

A solver for the stochastic master equation applied to gene regulatory networks

Markus Hegland^{a, b, *}, Conrad Burden^c, Lucia Santoso^{b, d}, Shev MacNamara^{a, b}, Hilary Booth[✉]

^aCentre for Mathematics and Its Application, Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, Australia

^bARC Centre in Bioinformatics, The University of Queensland, St Lucia, Qld 4072, Australia

^cCentre for Bioinformation Science, John Curtin School of Medical Research & Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, Australia

^dCentre for Bioinformation Science, Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, Australia

Received 1 August 2005

Abstract

An important driver of gene regulatory networks is noise arising from the stochastic nature of interactions of genes, their products and regulators. Thus, such systems are stochastic and can be modelled by the chemical master equations. A major challenge is the curse of dimensionality which occurs when one attempts to integrate these equations. While stochastic simulation techniques effectively address the curse, many repeated simulations are required to provide precise information about stationary points, bifurcation phenomena and other properties of the stochastic processes. An alternative way to address the curse of dimensionality is provided by sparse grid approximations. The sparse grid methodology is applied and the application demonstrated to work efficiently for up to 10 proteins. As sparse grid methods have been developed for the approximation of smooth functions, a variant for infinite sequences had to be developed together with a multiresolution analysis similar to Haar wavelets. Error bounds are provided which confirm the effectiveness of sparse grid approximations for smooth high-dimensional probability distributions.

© 2006 Elsevier B.V. All rights reserved.

MSC: 92C40; 62P10; 60J22

Keywords: Gene regulatory networks; Sparse grids; Master equations; Systems biology

1. Introduction

Within the nucleus of every cell of every organism, long coils of DNA (deoxyribonucleic acid) form the chromosomes that contain encoded information necessary for the organism to develop within a changing external environment. Each cell contains the same DNA, and yet there is a multitude of manifestations of genetic information in the various cell types, as not all genes are expressed at all times. In fact, whether a specific gene is expressed at a given time depends on

[✉]Deceased 26 May 2005

* Corresponding author. Australian National University, CMA, Mathematical Sciences Institute, Canberra ACT 0200, Australia.

Tel.: +61 06 6125 4501; fax: +61 06 6125 5549.

E-mail address: markus.hegland@anu.edu.au (M. Hegland).

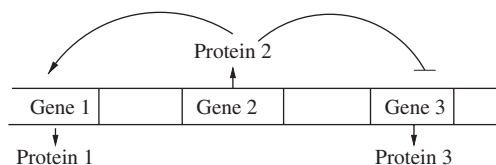


Fig. 1. Protein 2 enhances the expression of gene 1 and represses that of gene 3.

the need for its gene product within the cell. The mechanisms which have evolved to regulate the expression of genes are known as gene regulatory networks.

A gene is expressed when its genetic information is activated and transcribed into messenger RNA (ribonucleic acid), which is then translated into sequences of amino acids. These sequences fold into three-dimensional structures known as proteins or enzymes to perform various functions. One important function of proteins and enzymes is to regulate gene expression. Transcription is controlled by the binding of proteins to operator sites adjacent to the start (or ‘promoter’) of the gene, and this can either enhance or repress gene expression (Fig. 1). We will refer to the promoter–operator unit as the *gene switch*. Once an RNA polymerase (the enzyme which transcribes the gene) binds to the switch, gene transcription proceeds without stopping until completion. The translation process is similar to this, with the messenger RNA acting as the blueprint for amino acid sequences. For simplicity, in this paper we consider mainly prokaryotes, for which translation automatically follows transcription.

The processes described above which decode genomic information into their phenomic manifestation are a series of (controlled) chemical reactions. In essence, gene expression is controlled by gene products, which may originate from juxtaposed genes or more distant ones, in which case the controlling gene products would be considered simply as existing chemicals in the “pool”.

The number of molecules of a given chemical species in intra-cellular systems is typically in the order of hundreds (e.g. approximately 100 molecules in the bacterium *Escherichia coli* [6], corresponding to concentrations $> 0.2 \mu\text{M}$ [2]). At these numbers of molecules, relative fluctuations are of the order of 10^{-1} , and we cannot ignore the stochasticity of the system.

Such systems are often described by a joint probability distribution $p(x, t)$ over abundances x of various chemical species at time t . We will denote the (infinite) vector of probabilities $p(x, t)$ at time t by $p(\cdot, t)$. In cases where the time dependence is obvious, we will suppress the t and denote the values of such a distribution by $p(x)$ and the corresponding vector by $p(\cdot)$. The value of an operator A acting on $p(\cdot)$ will be denoted by $Ap(\cdot)$ or simply Ap and the abbreviated notation $Ap(x) = (Ap(\cdot))(x)$ will be used extensively.

We write the set of equations involving i reactions as

$$\frac{\partial p}{\partial t}(x, t) = - \sum_i a_i(x, t)p(x, t) + \sum_i a_i(x - z_i, t)p(x - z_i, t),$$

where $a_i(x)$ indicates the propensity of transition from state x , and z_i is a vector indicating the stoichiometry of reaction i . Propensities are measures of rates of transition, indicating the speeds of reactions, and are hence related to chemical reaction rate constants. We can rewrite the partial differential equation as an ordinary differential equation (ODE) for the vector $p(\cdot, t)$ as

$$\frac{dp(\cdot, t)}{dt} = Ap(\cdot, t),$$

where the transition rate matrix A contains the propensities a . The above equation is known as the stochastic master equation [4,21]. The matrix A consists of a sum

$$A = \sum_{i=1}^r A_i,$$

where each A_i describes one reaction and is of the form

$$A_i = (S_{z_i} - I)D_i,$$

where S_{z_i} is a shift matrix and D_i is a diagonal matrix containing the propensities. Thus, the matrix A is compactly represented by a pair $(z_i, \phi_i(x))$ where the function ϕ_i is such that $D_i p(\cdot, t) = \phi_i(\cdot) p(\cdot, t)$ and the multiplication is pointwise. In cases where x is large, these matrices (A, D , etc.) are thought of as linear operators, and the two terminologies are henceforth used interchangeably.

This framework expresses a stochastic system as a system of linear ODE, which can in principle be solved by numerical integration. However, for realistic biological networks, the system's dimension quickly grows to ostensibly unmanageable proportions. The perception of near intractability has caused many to shy away from the direct integration approach, and instead methods such as stochastic simulations are popularly used [1,20]. However, these methods can themselves be computationally expensive.

A gene regulatory system contains a large number of interacting proteins, DNA, multimers, RNA and auxiliary substances. In order to be computationally feasible, the complexity of these systems needs to be substantially reduced. A first principle to reduce effectively the state space is the quasi steady-state approximation [19]. It uses the fact that many reactions like dimerisation and gene switching are 100 to 1000 times faster than protein synthesis and degradation. A substantial simplification to the master equations can be achieved by assuming that, conditional on the protein levels, the probability distributions of chemical species driven by fast reactions are stationary [15]. In the following it will therefore be assumed that only protein levels determine the state of the system.

The probability distributions $p(\mathbf{x}, t)$ occurring in the master equations are defined over states \mathbf{x} which are vectors with potentially very many components (one component per protein species), where each component can take any integer value. In order to compute solutions to the master equations effectively one requires approximations which reduce biological problems to a feasible scale. First, one observes that for systems defined by their protein levels, the probability distributions are very small outside a bounded subdomain. It thus makes sense to approximate the infinite domain space \mathbb{N}^d by a set $[0, m_1 - 1] \times \dots \times [0, m_d - 1]$, or, in the simplest case, by $[0, m - 1]^d$ where $[0, m - 1] = \{0, \dots, m - 1\}$. Equivalently, one approximates the probability distributions by ones for which the probabilities are zero outside the supporting set. An algorithm for the approximation of the problem on the finite state space is the “finite state projection algorithm” [13] and methods for systematically increasing the space and error bounds have been provided.

While the truncation of the state space makes the problem feasible in principle, the state space can still be very large. For example, in the biological problems considered here one often has stationary mean values of numbers of proteins of a few hundreds, and m would have to be larger than this. Thus, a second method has to be used to reduce the state space. Here we propose to use aggregation which in practice amounts to approximating the probabilities by piecewise constant functions. The aggregation method can be slightly improved if one uses information about the shape of the probability functions within the aggregation domain, and this will be discussed elsewhere. For a discussion of aggregation in general, see [18,7,19,12]. The aggregation method requires an aggregation operator which is defined by a partitioning of the state space

$$X = \bigcup_y X_y,$$

where the index y is a state in the aggregated state space. The operator E acts on vectors $p(\cdot)$ and is then given by

$$Ep(y) = \sum_{x \in X_y} p(x).$$

One further requires a disaggregation operator F which is also positive and has to satisfy $EF = I_Y$ (the identity in Y). This aggregation/disaggregation pair defines a reduced operator $B = EAF$. This reduced operator also satisfies the zero column sum condition, and thus p_Y satisfying

$$\frac{dp_Y(\cdot, t)}{dt} = Bp_Y(\cdot, t)$$

and initial conditions $p_Y(\cdot, 0) = Ep_X(\cdot, 0)$ (where p_X is the original probability distribution) is also a probability distribution. Just as a footnote, we remark that one could now take B and try to reconstruct the original matrix A . The natural candidate FBE , however, is in general unacceptable as it typically has negative elements outside the main diagonal. This can be remedied by constructing from B, E and F a naturally disaggregated system

$$A' = FBE + \alpha(I - FE)$$

for an α which is large enough to remove all the negative elements. Note that the aggregated matrix does not change and one has

$$EA'F = B.$$

While a combination of truncation and aggregation does provide a substantial reduction of the state space, the dimensionality still leads to a state space of size which is exponential in the dimension. Consequently, even after these approximations one typically cannot solve the problems for systems with more than four or five proteins. An approximation which goes beyond state space reduction methods and allows approximation of high-dimensional problems uses sparse grids and is discussed in Section 3.

Once the state space is reduced to a feasible level one can deal with the solution of the initial value problem for the corresponding master equations, i.e., the linear differential equation

$$\frac{dp(\cdot, t)}{dt} = Ap(\cdot, t)$$

with the initial condition $p(\cdot, 0) = p_0(\cdot)$. The solution of this equation is of the form

$$p(\cdot, t) = e^{tA} p(\cdot, 0).$$

The exponential function of a (finite) square matrix A is well defined. However, the computation is nontrivial as A can typically be large and sparse. For the current investigation an adaptation of the Expokit software [17] has been used to compute $p(\cdot, t)$. The method uses matrix-free Krylov space projection methods which allow very large sparse matrices A to be dealt with, and which moreover define these matrices as procedures to compute matrix vector products and not as arrays of matrix elements.

In the remaining sections a hierarchical aggregation/disaggregation procedure will be developed, beginning with the case of a single protein in Section 2. Methods for dealing with more realistic multiple protein systems will be developed in Section 3 using sparse grids. Specific examples including a single protein with decay, a genetic toggle, a gene cascade, and the more biologically relevant examples of the bacteriophage- λ switch and the pleiotropic drug resistance network in yeast are given in Section 4. Conclusions are drawn in Section 5.

2. The one protein model

In one dimension the state is characterised by one integer $x \in \mathbb{N}$ which denotes the number of proteins of the single species considered in this case. The probability distribution is thus characterised by the sequence $p(x)$, $x=0, 1, 2, \dots$. A simple aggregation is based on blocking odd and even x together, i.e., by the partitioning $X = \bigcup_{y \in \mathbb{N}} X_y$ where $X_y = \{2y, 2y + 1\}$. In this case one gets the aggregation operator of the form

$$Ep(x) = p(2x) + p(2x + 1).$$

This particular aggregation operator shall be called the *pairwise aggregation operator* in the following. Consider now the delta function defined by $p_\delta(0) = 1$ and $p_\delta(x) = 0$ for $x \neq 0$. Applying the definition of E from above one gets $Ep_\delta = p_\delta$. In this case the distribution is not changed by aggregation. Consider now a shifted delta distribution with, say, $p(2^m) = 1$ and $p(x) = 0$ elsewhere. Here one gets $Ep(x) = p(2x)$ and so $Ep(x) = 1$ if $x = 2^{m-1}$ and zero elsewhere. Thus, the peak is maintained but shifted towards the origin. A peak at $x = 2^m + 1$ does move to the same place. Thus, aggregation “squeezes” the probability distribution. When the aggregation is applied multiple times (to an arbitrary p) the sequence of distributions obtained converges to p_δ :

$$E^n p \rightarrow p_\delta.$$

Moreover, if the support of p (the set of x where $p(x) \neq 0$) is contained in $[0, 2^m - 1]$ then one has

$$E^m p = p_\delta.$$

In this case it is safe to truncate the state space. Let the truncation operator Q_m be defined by $Q_m p(x) = p(x)$ if $x < 2^m$ and $Q_m p(x) = 0$ for $x \geq 2^m$. Note that $Q_m p$ is not a probability distribution as with the truncation the normalisation

vanishes. Now let (as usual) the supremum norm be defined by $\|f\|_\infty = \sup_x |f(x)|$. One obtains the following bounds on the error norm for aggregation:

Proposition 1. *Let Q_m be the truncation operator onto the interval $[0, 2^m - 1]$ and E be the aggregation operator. Then the truncation error is bounded by*

$$\|(I - Q_m)p\|_\infty \leq \|p_\delta - E^m p\|_\infty.$$

Proof. Using the definitions and bounding the supremum by the sum gives $\|(I - Q_m)p\|_\infty \leq \sum_{x=2^m}^\infty p(x)$ which is equal to $1 - \sum_{x=0}^{2^m-1} p(x) = p_\delta(0) - E^m p(0)$. The bound follows directly. \square

This provides a simple estimate on the size of the truncation error relating it to the size of aggregates of the probability distribution. Combined with the bound of the aggregation error (see proposition below) this provides a bound for the total error which originates from the two approximation procedures.

As mentioned in the previous section, when combined with disaggregation, the aggregation procedure defines an approximation scheme. In the following a simple disaggregation operator will be used which is defined by

$$Fp(2x) = Fp(2x + 1) = p(x)/2.$$

Note that, like in the case of aggregation, the disaggregation operator maps probability distributions onto probability distributions. One also sees that disaggregation followed by aggregation gives the identity, i.e., $EF = I$. Approximations T_n are obtained when the aggregation operator is applied n times followed by n applications of the disaggregation operator, i.e.,

$$T_n = F^n E^n$$

and set $T_0 = I$. Note that $F = \frac{1}{2}E^*$ where E^* is the adjoint of E . As p is a probability distribution it follows that $\|p\|_\infty \leq 1$ and, more generally, by the definition of T_n one has

$$\|T_n p\|_\infty \leq 2^{-n}.$$

The approximation $T_n p$ is a piecewise constant function where each piece contains 2^n contiguous points. As T_n is an averaging operator it also follows that

$$\|T_n p\|_\infty \leq \|p\|_\infty.$$

For the analysis of the error $(T_n - I)p$ one introduces the shift operator S which is defined by $Sp(x) = p(x + 1)$. Note that Sp is not necessarily a probability distribution. However, the difference $(S - I)p$ does provide a measure for the smoothness of the probability distribution. The central error bound is obtained by repeated use of “telescoping” and the triangle inequality.

Proposition 2. *Let $T_n = F^n E^n$ where E and F are the pairwise aggregation and disaggregation operators, respectively. Then the infinity norm of the approximation error is bounded by*

$$\|(T_n - I)p\|_\infty \leq \frac{2^n - 1}{2} \|(S - I)p\|_\infty.$$

Proof. The application of the definition of T_n and the triangular inequality gives

$$\|(T_n - I)p\|_\infty \leq \sum_{k=1}^n \|F^{k-1}(FE - I)E^{k-1}p\|_\infty.$$

Furthermore, F is an extension combined with scaling and so

$$\|F^{k-1}(FE - I)E^{k-1}p\|_\infty = \frac{1}{2^{k-1}} \|(FE - I)E^{k-1}p\|_\infty.$$

By substituting the definitions of F , E and S one gets

$$\|(FE - I)E^{k-1}p\|_\infty \leq \frac{1}{2}\|(S - I)E^{k-1}p\|_\infty.$$

Expanding $E^{k-1}p$ and telescoping gives

$$(S - I)E^{k-1}p(x) = \sum_{j=0}^{2^{k-1}-1} \sum_{s=0}^{2^{k-1}-1} [p(2^{k-1}x + s + 1 + j) - p(2^{k-1}x + s + j)]$$

and it follows that

$$\|(S - I)E^{k-1}p\|_\infty \leq (2^{k-1})^2\|(S - I)p\|_\infty.$$

Putting all together now leads to the bound to prove. \square

For very smooth functions, the term $\|(S - I)p\|_\infty$ is very small and the parameter n in T_n is chosen as large as possible such that the error is still acceptable. A different way to look at this is that the error is proportional to the size of the aggregation domain. This is not surprising given that this method is piecewise constant. The error is bounded by the variation of the density over the aggregation domain.

Consider now two simple examples. First, if $p = p_\delta$ then the smoothness parameter is $\|(S - I)p_\delta\|_\infty = 1$ and the approximation error is $\|(T_n - I)p_\delta\|_\infty = 1 - 2^{-n}$. Clearly, the bound of the previous proposition is useless in this case, in fact, any approximation error where the approximation is again a probability distribution is bounded by 1 from above. A more interesting example is the Poisson distribution where $p(x) = \exp(-\lambda)\lambda^x/x!$. Fig. 2 shows the case $\lambda = 200$ with an aggregation $T_n p$ for $n = 4$. In this case the smoothness parameter is $\|(I - S)p\|_\infty = 0.0012$ which leads to an error bound of $(2^n - 1)/2\|(S - I)p\|_\infty = 0.0093$. The actual error of the approximation is just slightly smaller, i.e., $\|(T_n - I)p\|_\infty = 0.0087$. Thus in this case the error bound is informative.

Hidden in the proof one also gets bounds for the components of a hierarchical decomposition of the probability distribution:

$$p(x) = T_n p(x) + \sum_{k=1}^n (T_{k-1} - T_k)p(x). \tag{1}$$

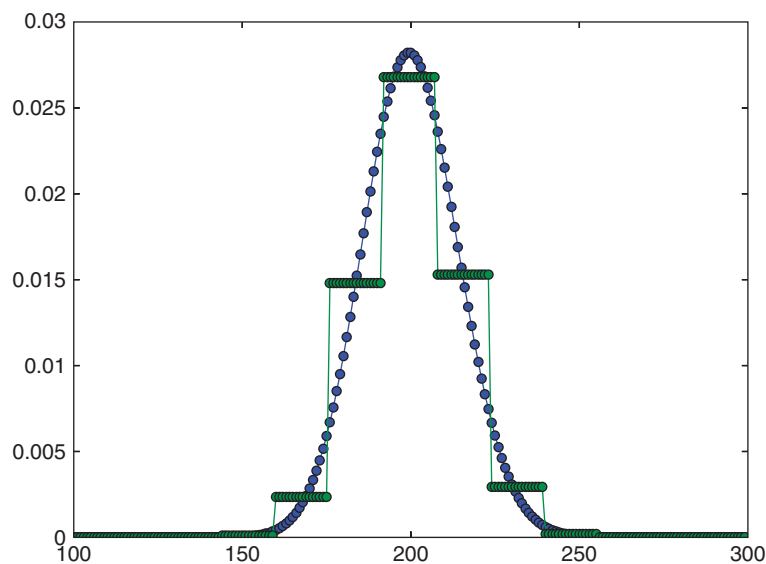


Fig. 2. Poisson distribution p and agglomerated approximation $T_n p$.

In particular, one has

$$\|(T_{k-1} - T_k)p\|_\infty \leq 2^{k-2} \|(S - I)p\|_\infty.$$

A specific approximation will truncate some of the early terms in the sum. However, these bounds are mostly important for the sparse grid approximations in Section 3.

One can now put this into a multiresolution framework. In particular, one introduces the operators \tilde{E}, \tilde{F} with $\tilde{E}p(x) = (p(2x) - p(2x + 1))/2$ and $\tilde{F}p(2x) = -\tilde{F}p(2x + 1) = p(x)$. Corresponding to Eq. (1) one gets the decomposition of the identity

$$I = F^n E^n + \sum_{k=1}^n F^{k-1} \tilde{F} \tilde{E} E^{k-1}.$$

(This follows, among other things, from the identity $FE + \tilde{F}\tilde{E} = I$.) This multiresolution defines an equivalent to Haar wavelets for the case of discrete sequence approximations.

In practice, one combines truncation with aggregation. Assuming that $m > n$ one has then

$$\|(T_n Q_m - I)p\|_\infty \leq \frac{2^n - 1}{2} \|(S - I)p\|_\infty + \|p_\delta - E^m p\|_\infty. \quad (2)$$

This follows because $\|(T_n Q_m - I)p\|_\infty \leq \|(T_n Q_m - Q_m)p\|_\infty + \|(Q_m - I)p\|_\infty$ and furthermore, as $T_n Q_m = Q_m T_n$ and because $\|Q_m f\|_\infty \leq \|f\|_\infty$ for any f . This bound provides a handle on how to choose both truncation and aggregation. In the following, the approximation space, i.e., the image of $Q_m T_n$ will be denoted by $V_{n,m}$. In the case of truncated spaces, it does not make sense to go beyond T_m when aggregating. This defines the coarsest level of approximation and the corresponding basis functions are constant over the full domain of interest $[0, 2^m - 1]$. If the finest level of detail is $n < m$ one gets the following representation for the projection $T_n p$:

$$T_n p = T_m p + \sum_{i=n}^{m-1} (T_k - T_{k+1})p. \quad (3)$$

Of course, this decomposition is just a telescoping expansion and holds trivially, it has been included here, however, as it forms the basis of nontrivial higher-dimensional approximations. As $T_k^2 = T_k$ and $T_k T_{k+1} = T_{k+1}$ the difference $R_k = T_k - T_{k+1}$ is also a projection and $R_{k_1} R_{k_2} = 0$ for $k_1 \neq k_2$ and $R_k T_m = 0$ for $k < m$. It follows that the decomposition provides a projection of p into a direct sum of subspaces.

The approximation of an evolving distribution $e^{tA} p_0$ has to deal with a changing smoothness $\|(S - I)e^{tA} p_0\|_\infty$. As the distribution converges to the stationary distribution and any other components decay exponentially in time, the smoothness also exponentially converges towards the smoothness of the stationary distribution. This is a consequence of the Perron theory for positive matrices [9]. The stationary distribution will asymptotically limit the effectiveness of the approximation. If the stationary distribution is known (at least conditional on the subsets used for the partitioning) this knowledge can be used in constructing the disaggregation operator to obtain approximations which converge to the stationary distribution.

One can now develop approximations for $e^{tA} p(\cdot)$. The theory developed so far can be directly applied to $T_n e^{tA} p(\cdot)$. Such an approximation, however, is computationally uninteresting as it still requires the determination of $e^{tA} p(\cdot)$. A more efficient technique would use an approximation of the form $F^n e^{A_n t} E^n p(\cdot)$. The operator A has the form

$$A = (S - I)D_1 + (S^* - I)D_2$$

where the D_i are diagonal and the S and S^* are shift operators. In this case the matrix A is tridiagonal. The two reactions taking place are protein production and decay which are birth and death processes, respectively. For a single species of protein produced with propensity $\alpha(x)$ and decaying with propensity $\delta(x)$, the master equations have the form

$$\begin{aligned} \frac{\partial p(x, t)}{\partial t} &= (Ap(\cdot, t))(x) \\ &= \alpha(x - 1)p(x - 1) + \delta(x + 1)p(x + 1) - (\alpha(x) + \delta(x))p(x). \end{aligned}$$

For this example the following approximation is used here:

$$A_n = E^n A F^n = E^n (S - I) D_1 F^n + E^n (S^* - I) D_2 F^n$$

which provides a good compromise between computational efficiency and approximation quality. The efficiency is based on the equality

$$E^n (S - I) F^n = \frac{1}{2^n} (S - I)$$

and a similar equality for S^* . With this in mind, one gets

$$A_n = \frac{1}{2^n} (S - I) D_n^{(1)} + \frac{1}{2^n} (S^* - I) D_n^{(2)}$$

where the diagonal matrices contain the even elements of D_1 (decay) and the odd elements for D_2 (production). It follows that the aggregated matrix A_n is tridiagonal again. As the “flow of the probability” between the aggregated regions is only influenced by the boundaries the diagonal matrices only contain propensities from states at the boundary of the aggregated domain. In order to derive error bounds for the approximation one needs to compare $e^{At} p$ with $F^n e^{A_n t} E^n p$. For this one introduces first the operator

$$B_n = T_n (S - I) D_1 T_n + T_n (S^* - I) D_2 T_n$$

and it follows that $F^n e^{A_n t} E^n = e^{B_n t} - I + T_n$. For the error one now has

$$(F^n e^{A_n t} E^n - e^{At}) p(\cdot) = (e^{B_n t} - e^{At}) p(\cdot) - (I - T_n) p(\cdot).$$

The second term of the error has been studied previously and for the first term one uses the identity

$$(e^{B_n t} - e^{At}) p(\cdot) = \int_0^t e^{B_n(t-s)} (B_n - A) e^{As} p(\cdot) ds.$$

This can be recast as a sum of several terms each containing factors $T_n - I$ which then leads to an error bound. The effect of the aggregation, the approximation error, can be interpreted as extra noise as the aggregation will smooth or “widen” the peaks of the probability distribution.

3. Tensor products and sparse grids

In the previous section, the one-dimensional space of truncated piecewise constant functions $V_{n,m} = \text{range}(Q_m T_n)$ was used to approximate the solution of the master equation involving one protein. The simplest generalisation to multiple dimensions (or multiple proteins) is based on tensor products of these spaces, i.e., $V_{n_1,m_1} \times \dots \times V_{n_d,m_d}$. In this case the approximating space is a linear vector space with a dimension (number of basis vectors, not the number of variables d) which grows exponentially with the number of variables (or proteins) d . This curse of dimensionality prohibits computations which go beyond, say $d = 4$.

An alternative to the simple tensor products are the sparse grids [22]. Sparse grids are able to deal with the curse of dimensionality and the spaces are generated by collections of tensor products of mostly very simple spaces $V_{n,m}$. Here we discuss sparse grids for $d = 2$, higher dimensions are derived similarly.

First, introduce the operators $R_{k,m_i} = T_k - T_{k+1}$ for $k = n_i, \dots, m_i - 1$ and $R_{m_i,m_i} = T_{m_i}$. In the following, the second index m_i in R_{k,m_i} will be omitted to simplify notation. The projection into piecewise constant functions in two variables is obtained by $T_{n_1} \otimes T_{n_2}$ and one has

$$T_{n_1} \otimes T_{n_2} = \sum_{k_1, k_2 = n_1, n_2}^{m_1, m_2} R_{k_1} \otimes R_{k_2}$$

by induction. One can distinguish four different cases for $R_{k_1} \otimes R_{k_2}$. In the first case, $k_1 < m_1$ and $k_2 < m_2$ and, as $R_{k_1} \otimes R_{k_2} = (R_{k_1} \otimes I)(I \otimes R_{k_2})$ one gets by repeated application of the bound from the previous section

$$\|R_{k_1} \otimes R_{k_2} p\|_\infty \leq 2^{k_1+k_2-2} \|(I - S) \otimes (I - S) p\|_\infty.$$

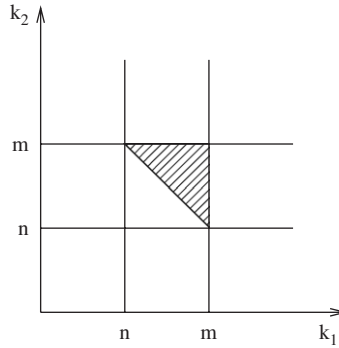


Fig. 3. Sparse grid index domain.

If $k_1 = m_1$ and $k_2 = m_2$ one can use the bound on the aggregated distribution to get

$$\begin{aligned} \|R_{m_1} \otimes R_{m_2} p\|_\infty &= \|T_{m_1} \otimes T_{m_2} p\|_\infty \\ &\leq 2^{-m_1 - m_2} \leq \|p\|_\infty. \end{aligned}$$

A combination of these bounds is used in the other two cases. For example, if $k_1 = m_1$ and $k_2 < m_2$ then

$$\|R_{m_1} \otimes R_{k_2} p\|_\infty \leq 2^{k_2 - 1} \|I \otimes (I - S)p\|_\infty.$$

Similar bounds are obtained for systems with more than two proteins.

The approximation error for the case of two proteins is a consequence of Proposition 2

$$\|(T_{n_1} \otimes T_{n_2} - I)p\|_\infty \leq \frac{2^{n_1} - 1}{2} \|(I - S) \otimes I p\|_\infty + \frac{2^{n_2} - 1}{2} \|I \otimes (I - S)p\|_\infty.$$

This follows by a triangle inequality argument:

$$\|(T_{n_1} \otimes T_{n_2} - I)p\|_\infty \leq \|(T_{n_1} \otimes I)(I \otimes T_{n_2} - I \otimes I)\|_\infty + \|(T_{n_1} \otimes I - I \otimes I)p\|_\infty$$

and remembering that $\|T_n p\|_\infty \leq \|p\|_\infty$. Again, the bounds for $d > 2$ are obtained in the same way.

Using the representation with R_k , it now turns out that $T_{n_1} \otimes T_{n_2}$ can be further approximated without much loss in performance. This uses the consequence of the bounds on $R_{k_1} \otimes R_{k_2} p(\cdot)$ obtained earlier that the terms $R_{k_1} \otimes R_{k_2} p(\cdot)$ are small if both k_1 and k_2 are small. An application of this observation leads to the sparse grid approximation

$$p_{\text{sg}}(\cdot) = \sum_{k_1 + k_2 \geq n + m} R_{k_1} \otimes R_{k_2} p(\cdot).$$

Here the summation indices k_i are bounded above by m , see Fig. 3 for the domain over which the summation is done.

The approximation error of this sparse grid approximation is then

$$q_e = \sum_{k_1 + k_2 < n + m} R_{k_1} \otimes R_{k_2} p.$$

The error norm of this approximation is bounded by

$$\begin{aligned} \|q_e\|_\infty &\leq \sum_{k_2 = n}^{m-1} \sum_{k_1 = n}^{m-1-k_2+n} 2^{k_1 + k_2 - 2} \|(I - S) \otimes (I - S)p\|_\infty \\ &\leq ((m - n - 1)2^{n+m-2} + 4^{(n-1)}) \|(I - S) \otimes (I - S)p\|_\infty. \end{aligned}$$

Note that only the first type of products $R_{k_1} \otimes R_{k_2}$ occurs in the error sum. Again this can be generalised to higher d . This error is substantially smaller than some of the terms which occur in the expansion which are of the order $4^m \|(I - S) \otimes (I - S)p\|_\infty$.

One reason why sparse grids are used is because they reduce the required storage space substantially. The size of the (truncated) product space is $2^{d(m-n)}$ but the size of the sparse grid space turns out to be $O((m - n)^{d-1} 2^{m-n})$ [22]. With this one is able to model networks with up to around 10 proteins. With adaptive sparse grids [8] one can expect, however, to model substantially larger problems.

For an efficient evaluation the sparse grid approximation can be reformulated in terms of the T_k only and one finds that

$$p_{\text{sg}}(\cdot) = \left(\sum_{k=n}^{m-1} T_k \otimes T_{m-1-k+n} - \sum_{k=n}^{m-2} T_k \otimes T_{m-2-k+n} \right) p(\cdot).$$

This formula leads to the *combination technique* [5] and there are corresponding formulae for the case of d proteins. The advantage of the combination formula is that it only requires approximations based on T_k and does not require evaluations of the R_k . The combination formula for the d protein case takes the form

$$T^{\text{sg}} = \sum_{\mathbf{k}} c_{\mathbf{k}} T_{\mathbf{k}},$$

where $\mathbf{k} = (k_1, \dots, k_d)$, and $T_{\mathbf{k}} = T_{k_1} \otimes \dots \otimes T_{k_d}$. The determination of the combination coefficients is based on the fact that the sparse grid space V^{sg} can be represented as the union of its maximal subspaces which are ranges of the $T_{\mathbf{k}}$, i.e., as

$$V^{\text{sg}} = \sum_{|\mathbf{k}|_1 = n+(d-1)m} V_{\mathbf{k},m}$$

where $V_{\mathbf{k},m} = \bigotimes_{i=1}^d V_{k_i,m}$ and $|\mathbf{k}|_1 = k_1 + \dots + k_d$. For more details on the determination of the coefficients, see [8].

The approximation of $e^{At} p$ suggested here is based on the combination formula and the ideas developed for the case $d = 1$ and is defined as

$$T^{\text{sg}} e^{At} p(\cdot) = \sum_{\mathbf{k}} c_{\mathbf{k}} T_{\mathbf{k}} e^{At} p(\cdot).$$

Thus with the combination formula the problem is reduced to finding approximations of the $T_{\mathbf{k}} e^{At} p$. These are determined as in the $d = 1$ case by approximations of the form $F_{\mathbf{k}} e^{E_{\mathbf{k}} A F_{\mathbf{k}}^t} E_{\mathbf{k}}$. The error bounds obtained here are a direct application of the bounds for the case of $d = 1$ and the sparse grid approximation error bounds developed above.

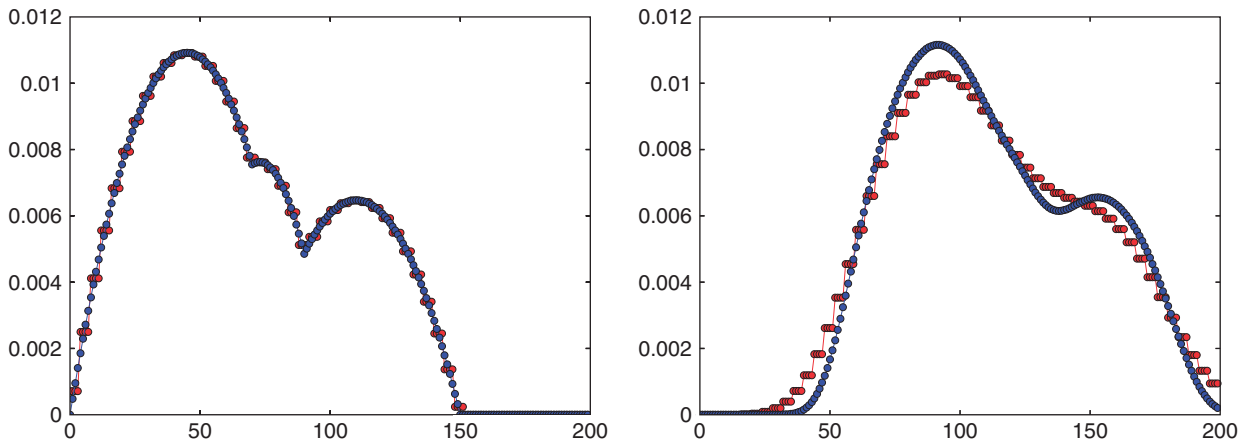


Fig. 4. Exact solution and (piecewise constant) solution of the aggregated equations (top: $t = 0$, bottom: $t = 5$); abscissa = number of protein molecules, ordinate = $p(x)$.

In summary, the computations consist of the following steps:

- (1) Given: initial conditions $p_0(\cdot)$
- (2) Determine the aggregated grids G_k and their aggregation operators E_k and disaggregation operators F_k
- (3) Compute aggregated initial conditions $p_0^{(k)}(\cdot) = E_k p_0(\cdot)$
- (4) Evolve these initial conditions over time using Expokit to get

$$p^{(k)}(\cdot, t) = e^{B_k t} p_0^{(k)}(\cdot),$$

where $B_k = E_k A F_k$

- (5) Evaluate the combination solution

$$p^{\text{sg}}(\cdot, t) = \sum_k c_k F_k p^{(k)}(\cdot, t)$$

at desired points of the state space and determine the displayed marginal distributions

4. Examples

In this section some typical examples will be considered using a protein production/decay (birth and death) model. Several examples model various control situations which are defined by the topology and the propensities $a_i(x)$.

4.1. Example 1: single protein produced at a constant rate and decaying with a rate proportional to the number of proteins

The operator A for this example is defined by

$$Ap(x) = \alpha p(x-1) + \delta \cdot (x+1)p(x+1) - (\alpha + \delta \cdot x)p(x).$$

Fig. 4 displays the initial conditions and results of an approximation using the aggregation technique describe in Section 2 where aggregation was done over $m=4$ consecutive points for the case for $\alpha=10$ and $\delta=0.01$. The top image shows the initial conditions and the bottom the computed probability distributions after five seconds with and without aggregation. It can be clearly seen that the aggregated equations introduce a stronger smoothing (= extra noise) into the results. Thus, peaked features get smoothed out much faster in the aggregated than in the original equations. However, the main features of the aggregated solution are still well maintained. To reduce aggregation noise, one could consider using adaptive aggregation which aggregates in smooth regions but not in the others. This needs to be further investigated.

4.2. Example 2: The toggle

A mutually repressing gene pair, or gene toggle (Fig. 5 (a)) can be found in systems such as the bacteriophage- λ [14,16], where there are two competing proteins. In this example we focus only on protein dynamics. The propensity

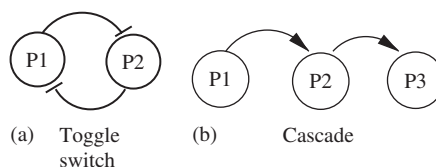


Fig. 5. Simple networks. (a) Toggle switch, (b) cascade.

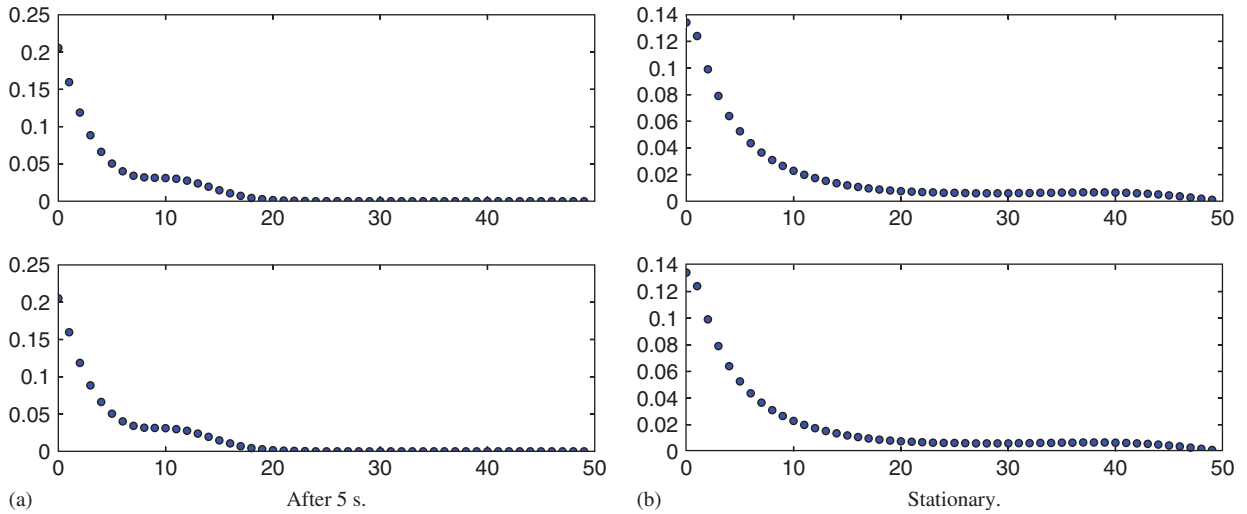


Fig. 6. Marginal distributions of the toggle model, abscissa = number of proteins, ordinate = $p(x)$. (a) After 5 s, (b) stationary.

of protein production is given in the quasi steady state approximation by the “soft switch” [15]

$$a_{\text{prod}}(x) = \frac{\alpha\beta}{\beta + \gamma x}.$$

Here α is the mass-action rate of protein production, β is the propensity of gene expression, and the propensity γx of gene repression is proportional to the abundance x of the repressing protein.

The toggle comprises two gene switches; each is a function of the other protein. The operator is therefore $A = A_1 + A_2$ the two components corresponding to production and decay of the two proteins.

In the toggle example the first protein stops the production of the second and vice versa. The first component is

$$A_1 p(x_1, x_2) = \frac{\alpha\beta}{\beta + \gamma x_2} p(x_1 - 1, x_2) + \delta(x_1 + 1) \cdot p(x_1 + 1, x_2) - \left(\frac{\alpha\beta}{\beta + \gamma x_2} + \delta \cdot x_1 \right) p(x_1, x_2)$$

and the second component is the same for the production and decay of x_2 controlled by x_1 .

It can be shown that the corresponding deterministic model in this case leads to a monostable point. A stochastic simulation does show a slightly different picture. Consider the case where $\delta = 0.05$, $\alpha = 1.0$, $\gamma = 1.0$, and $\beta = 0.4$. The simulations show that after $t = 5$ one has a case where both average values of both proteins and small levels of the one protein combined with higher level of the other protein are quite likely, and this remains the case for the stationary distribution as well (Figs. 6, 7).

4.3. Example 3: a cascade

A cascading process occurs when (usually) adjacent genes produce protein which enhances the expression of the succeeding gene. This is also a typical motif in genetic networks; one example can be found in the lytic phase of the λ -phage system [14]. We consider here a cascade of length 3 for illustration (Fig. 5(b)). The operator A of the cascade is of the form

$$A = A_1 + \dots + A_k,$$

where k is the length of the cascade. While the first component (A_1) is of the same form as the example 1, the components A_i for $i > 1$ satisfy

$$A_i p(x) = \frac{\beta x_{i-1}}{\beta x_{i-1} + \gamma} p(x - e_i) + \delta \cdot (x_i + 1) p(x + e_i) - \left(\frac{\beta x_{i-1}}{\beta x_{i-1} + \gamma} + \delta \cdot x_i \right) p(x)$$

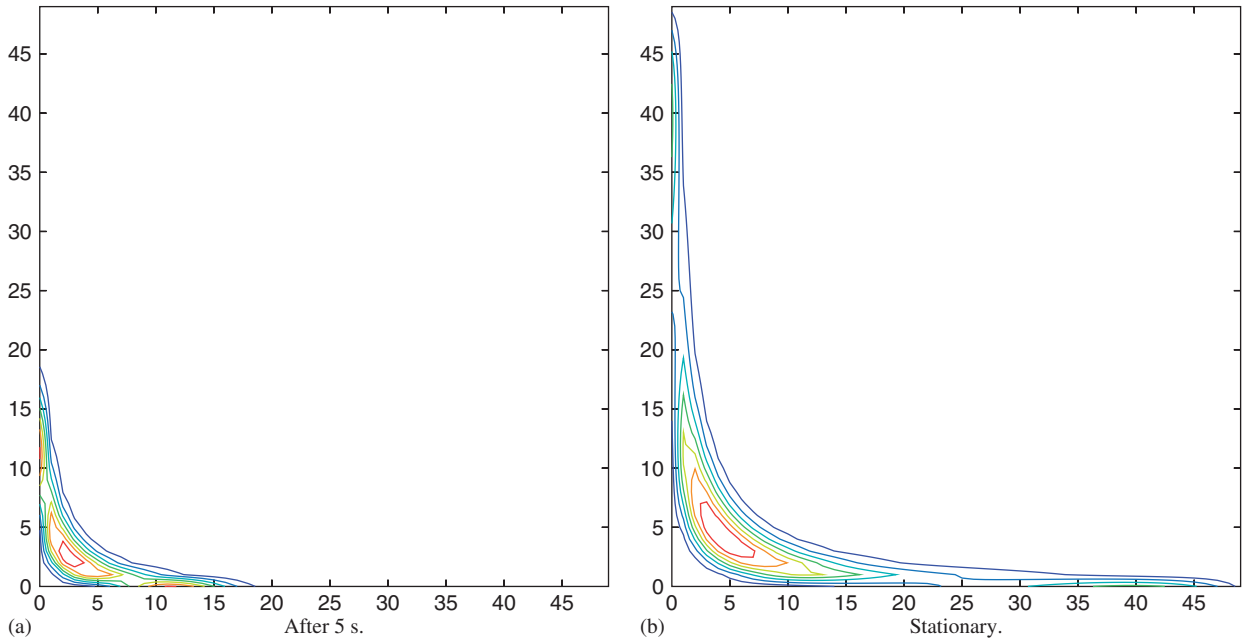


Fig. 7. Contour of the distributions for the toggle, axes denote the number of protein 1 (abscissa) and protein 2 (ordinate). (a) After 5 s, (b) stationary.

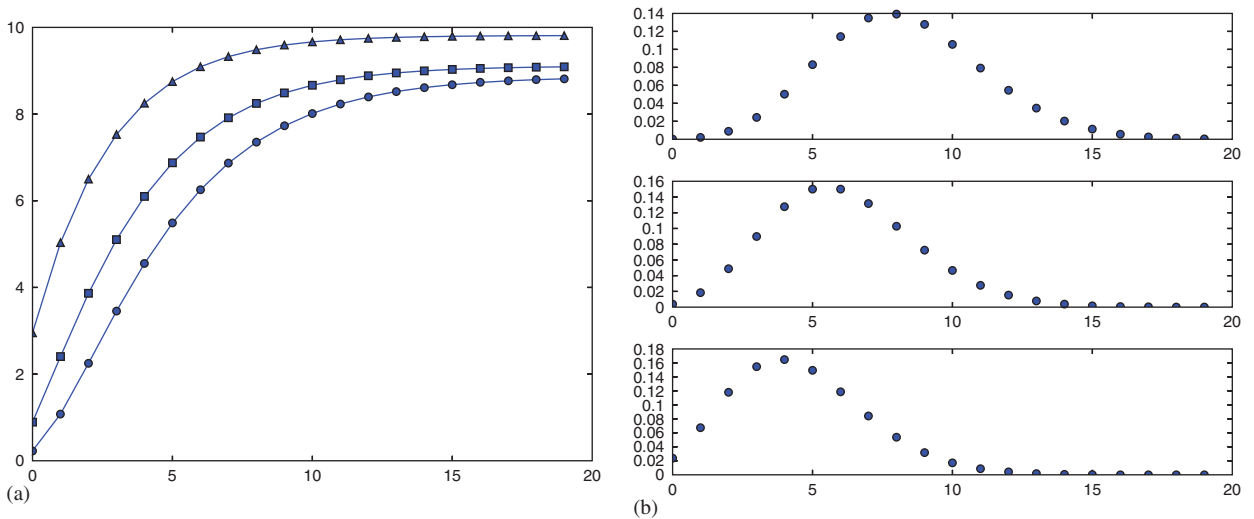


Fig. 8. Solution of the master equation pertaining to the cascade. (a) Expected value of protein levels; triangle, square, circle = proteins 1,2,3 as functions of time, (b) intermediate ($t = 80$ s) protein distributions as functions of protein abundance; top, middle, bottom = proteins 1,2,3.

where e_i is the i th standard basis vector of \mathbb{R}^k . For this cascade one can compute several approximations applying a combination technique.

In the cascade the expectations of the marginal distributions have a clear inbuilt delay. As an example, consider the case of three species with production of the first species $\alpha_0 = 0.7$, decay for all the species $\delta = 0.07$ and the production of species two and three equal to $x_{i-1}/(5.0 + x_{i-1})$ for $i = 2, 3$, respectively (Fig. 8).

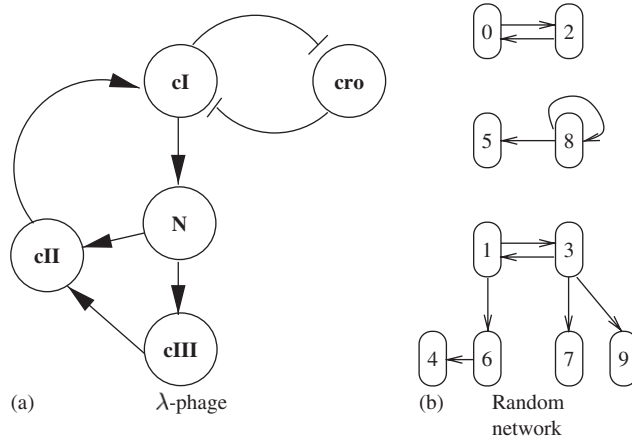


Fig. 9. Examples of larger networks. (a) λ-phage, (b) random network.

4.4. Example 4: lambda phage

The life cycle of bacteriophage-λ displays a naturally occurring toggle switch [14,16]. The λ-phage infects a bacterium *E. coli*, and depending on the environment, either stays dormant in the bacterial host or multiplies, reassembles itself and breaks out of the host. This switching behaviour is determined by two competing proteins *cI* (a high concentration of which establishes the dormant, or lysogenic phase) and *cro* (a high concentration of which establishes the active, or lytic, phase in which the phage breaks out). The initial establishment of either the lysogenic or lytic phase is determined by a series of proteins which sense the environmental conditions. Under favourable conditions, the establishment of *cII* initiates the production of *cI* and the system enters lysogeny. If an adequate quantity of *cII* cannot be established, the production of *cro* causes the system to enter lysis soon after invasion of the bacterium by the phage. Because the balance between *cI* and *cro* is controlled by a genetic toggle, it is also possible for a lysogenic bacterium to switch to the lytic state if the system is stressed, for example, by exposure to ultraviolet light.

A protein signalling network of the λ-phage switch is shown in Fig. 9. Let us denote the abundance of proteins *cro*, *cI*, *N*, *cIII*, *cII* as state variables x_1, x_2, \dots, x_5 . The core of this switch, *cro-cI* system is a toggle, thus A_1 is simply the toggle operator in Example 2. The production rate of x_2 is an increasing function of x_5 ,

$$A_2 p(x) = \frac{(\alpha_2 + kx_5)\beta_2}{\beta_2 + \gamma x_1} p(x - e_2) + \delta_2 \cdot (x_2 + 1)p(x + e_2) - \left(\frac{(\alpha_2 + kx_5)\beta_2}{\beta_2 + \gamma x_1} + \delta_2 x_2 \right) p(x)$$

and we choose here $k = 1$. The production of x_3 is enhanced by x_2 , giving us

$$A_3 p(x) = \frac{\alpha_3 \beta x_2}{\beta x_2 + \gamma} p(x - e_3) + \delta_3 \cdot (x_3 + 1)p(x + e_3) - \left(\frac{\alpha_3 \beta x_2}{\beta x_2 + \gamma} + \delta_3 x_3 \right) p(x).$$

The same form applies for A_4 , where production of x_4 is enhanced by x_3 . Production of x_5 is enhanced by x_3 and degradation is impeded by presence of x_4 ,

$$A_5 p(x) = \frac{\alpha_5 \beta x_3}{\beta x_3 + \gamma} p(x - e_5) + \frac{\delta_5}{1 + x_4} \cdot (x_5 + 1)p(x + e_5) - \left(\frac{\alpha_5 \beta x_3}{\beta x_3 + \gamma} + \frac{\delta_5}{1 + x_4} \cdot x_5 \right) p(x).$$

A simulation was run with parameters estimated from [1,16], $\alpha_1 = 0.5$, $\delta_1 = 0.0025$, $\alpha_2 = 1.0$, $\delta_2 = 0.0007$, $\alpha_3 = 0.15$, $\delta_3 = 0.0231$, $\alpha_4 = 0.3$, $\delta_4 = 0.01$, $\alpha_5 = 0.3$, $\delta_5 = 0.01$. When information is unavailable, switching parameters β and γ are assumed to be 1.0, except for $\beta_1 = 0.12$ and $\beta_2 = 0.6$. The results are shown in Fig. 10.

4.5. Example 5: the pleiotropic drug resistance network in yeast

The network consists of 10 proteins which are all positively regulated by two central proteins PDR1 and PDR3 [11]. This subnetwork has been identified in *Saccharomyces cerevisiae* and is thought to counter the action of toxic substances.

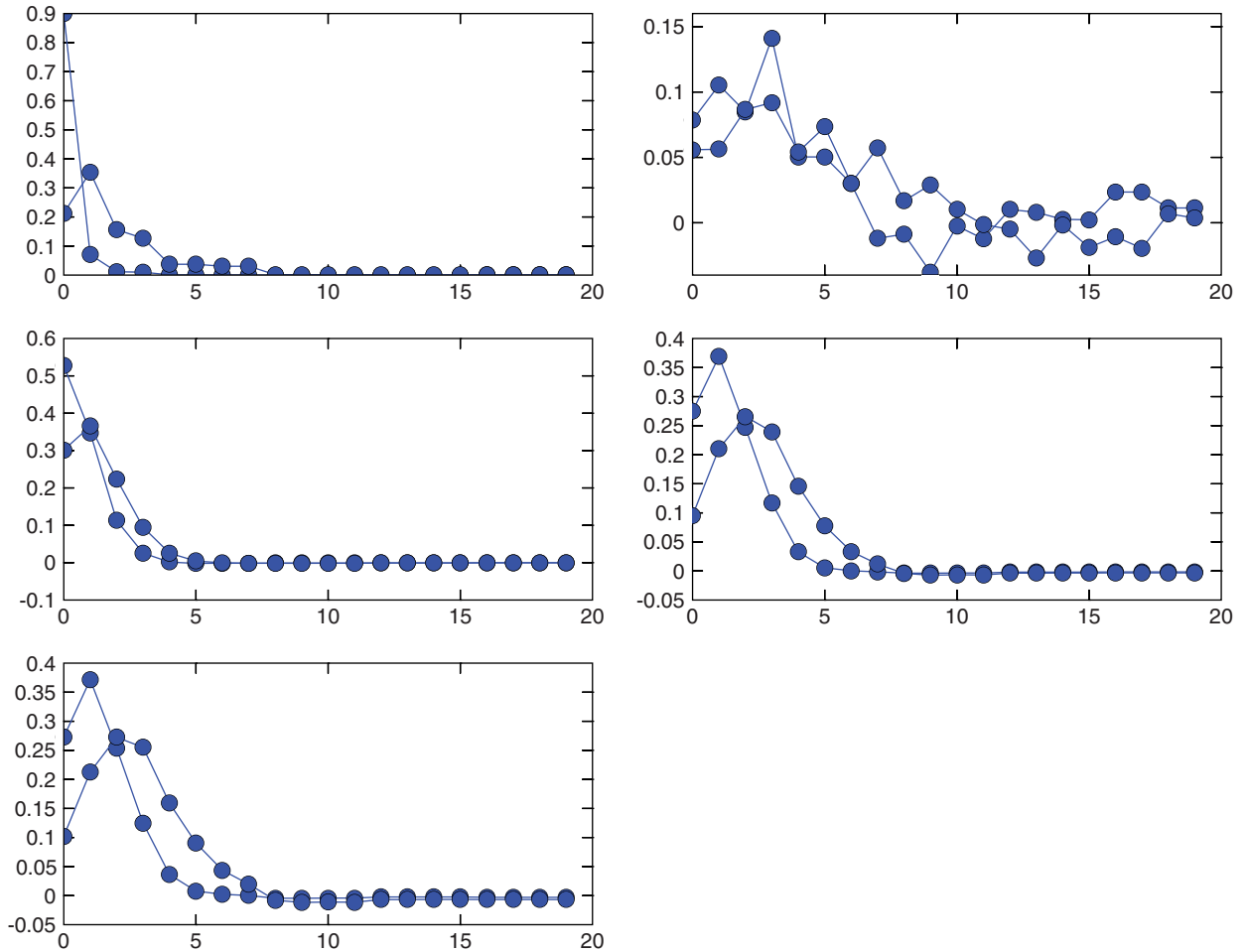


Fig. 10. Marginal protein probability distributions for the λ -phage switch at $t = 5, 10$ s. Top left = *cro*, top right = *cI*, mid left = *N*, mid right = *cII*, bottom left = *cIII*, abscissa = number of proteins and ordinate = $p(x)$.

Using the sparse grid technology discussed above the evolution of the marginal distributions has been determined and the results can be seen in Fig. 11. This example is used to illustrate the feasibility of sparse grid approximation for high-dimensional cases. Methods based on graphical models may provide promising alternatives but they will not be further considered here.

The sparse grid approximates a probability distribution over a state space with $16^{10} \approx 10^{13}$ nodes. The actual approximation required 1001 grids with 16 grid points each. On a 1.7 GHz Pentium 4 PC with 1 GByte memory the simulation required 40 min and on 2 GHz AMD 64 Bit PC with 1 GByte the same simulation required less than 15 min.

5. Conclusion

Probably the major computational challenge one faces when one attempts to model gene regulatory networks using the chemical master equation framework is the curse of dimensionality as these networks consist of large numbers of interacting substances. Until now, mostly stochastic approximation techniques have been used to address this curse [4,20]. These techniques, while being highly efficient, have the disadvantage that using them to estimate population characteristics does require a large number of simulations. Here, the curse of dimensionality is addressed using sparse grids. Networks containing up to around 10 proteins can be modelled with the classical sparse grids and larger networks become feasible using adaptive sparse grids.

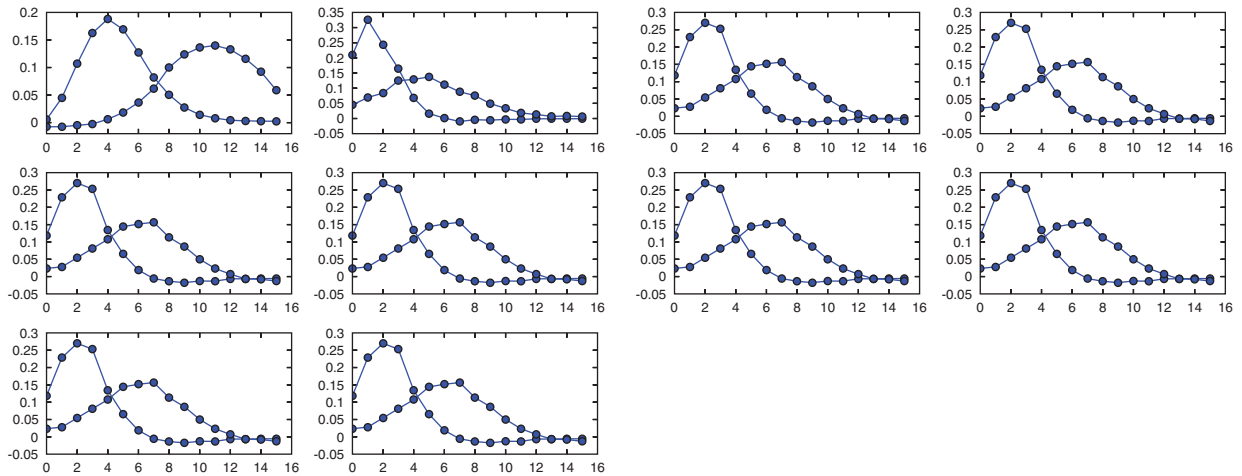


Fig. 11. Marginal distributions of PDR network.

The sparse grid approach has been implemented in Python (<http://www.python.org>) using an adaptation of the Expokit [17] code, SciPy (<http://www.scipy.org>), Numerical Python (sourceforge.net/projects/numpy) and Matplotlib (matplotlib.sourceforge.net). It features a basic class extending numerical arrays for the underlying numerical linear algebra, a class which allows the simple definition of gene regulatory networks and a sparse grid class. The main code (using sparse grids) for the toggle example is:

```
from SpaGri import SpaGri
production=(lambda x,: 1.0/(0.4 + x[1]),\
            lambda x : 1.0/(0.4 + x[0]))
decay=(lambda x : 0.05*x[0],\
       lambda x : 0.05*x[1])
cellx = SpaGri(dimension=2,\
               production=production, decay =decay )
dt =5 # time step
cellx.step(dt)
cellx.plot()
```

After importing the main class SpaGri an instance of a gene regulatory network is created (called cellx here). Then the production and decay propensities are specified as tuples of functions. The integrator is called using the “step” method and the marginal distributions are plotted by the “plot” method. Please contact the authors if you require further information.

Using this code it was possible to solve the equations for a network with 10 proteins on an AMD64 based PC with 2 GHz tact and 1GByte of memory in 15 min. The underlying state space for this problem has $16^{10} \approx 10^{12}$ states. The next step is the development of technology which allows the simulation of systems with 100 proteins again using the master equations. For this an adaptive sparse grid approximation will be implemented.

Earlier work on gene regulatory networks relevant to the methods here include the discussion of truncating state space [13], the linear noise approximation [3] using a combination of ODE- and PDE-based techniques, and multiplane modelling [10] which is closely related to graphical model approximations.

Acknowledgements

Initial parts of this research has been done while the first author visited the IWR of the University of Heidelberg. The research has been partially supported by the Australian Centre of Bioinformatics and by the Centre for Bioinformation Sciences. We would like to thank Professor Kevin Burrage, University of Queensland and Professor Sue Wilson, ANU

for their ongoing support and encouragement and Professors Allen Rodrigo and Bill Jordan for their invitations and feedback relating to the early stages of this work. While writing up this research, sadly, our coauthor Hilary Booth has died. We dedicate this paper to her memory.

References

- [1] A. Arkin, J. Ross, H.H. McAdams, Stochastic kinetic analysis of developmental pathway bifurcation in phage λ -infected *Escherichia coli*, *Genetics* 149 (1998) 1633–1648.
- [2] W.D. Donachie, A. Robinson, Cell division: parameter values and process, in: F. Neidhardt et al. (Eds.), *Escherichia coli and Salmonella typhimurium: Cellular and Molecular Biology*, ASM Press, Washington DC, USA, 1987, pp. 1347–1350.
- [3] J. Elf, M. Ehrenberg, Fast evaluation of fluctuations in biochemical networks with the linear noise approximation, *Genome Research* 13 (2003) 2475–2484.
- [4] D.T. Gillespie, *Markov Processes: An Introduction for Physical Scientists*, Academic Press, San Diego, USA, 1992.
- [5] M. Griebel, M. Schneider, C. Zenger, A combination technique for the solution of sparse grid problems, in: *Iterative Methods in Linear Algebra*, Brussels, 1991, North-Holland, Amsterdam, 1992, pp. 263–281.
- [6] P. Guptasarma, Does replication-induced transcription regulate the myriad of low copy number proteins of *Escherichia coli*?, *Bioessays* 17 (1995) 987–997.
- [7] M. Haviv, Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain, *SIAM J. Numer. Anal.* 24 (4) (1987) 952–966.
- [8] M. Hegland, Adaptive sparse grids, in: K. Burrage, R.B. Sidje (Eds.), *Proceedings of 10th Computational Techniques and Applications Conference CTAC-2001*, vol. 44, April 2003, pp. C335–C353.
- [9] R.A. Horn, C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990 Corrected reprint of the 1985 original.
- [10] A. Lipshtat, H.B. Perets, N.Q. Balaban, O. Biham, Modeling of negative autoregulated genetic networks in single cells, *Gene* 347 (2) (2005) 265–271.
- [11] A. Mazurie, S. Bottani, M. Vergassola, An evolutionary and functional assessment of regulatory network motifs, *Genome Biology* 6 (4) (2005) R35.1–R35.12.
- [12] C.D. Meyer, Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems, *SIAM Rev.* 31 (1989) 240–272.
- [13] B. Munsky, M. Khammash, The finite state projection algorithm for the solution of the chemical master equation, Technical Report CCEC-05-0505, Center for Control Engineering and Computation, University of California at Santa Barbara, 2005. (<http://ccec.mee.ucsb.edu/Year/2005.complete.html#CCEC-05-0505>).
- [14] M. Ptashne, *A genetic switch: phage- λ and Higher Organisms*, Cell, Cambridge, USA, 1992.
- [15] L. Santoso, M. Hegland, H. Booth, C. Burden, Approximation techniques for Markov models of genetic regulatory network, 2005, in preparation.
- [16] M.A. Shea, G.K. Ackers, The o_p control system of bacteriophage lambda, a physical-chemical model for gene regulation, *J. Mol. Biol.* 181 (1985) 211–230.
- [17] R.B. Sidje, EXPOKIT: Software package for computing matrix exponentials, *ACM Trans. Math. Software* 24 (1) (1998) 130–156.
- [18] H.A. Simon, A. Ando, Aggregation of variables in dynamic systems, *Econometrica* 29 (1961) 111–138.
- [19] W. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.
- [20] T. Tian, K. Burrage, Bistability and switching in the lysis/lysogeny genetic regulatory network of bacteriophage— λ , *J. Theoret. Biol.* 227 (2004) 229–237.
- [21] N.G. van Kampen, *Stochastic Processes in Physics and Chemistry*, North-Holland, Amsterdam, the Netherlands, 1981.
- [22] C. Zenger, Sparse grids, in: *Parallel Algorithms for Partial Differential Equations*, Kiel, 1990, Notes Numerical Fluid Mechanics, vol. 31, Vieweg, Braunschweig, 1991, pp. 241–251.