On the Approximation of Stochastic Concurrent Constraint Programming by Master Equation

Luca Bortolussi

Department of Mathematics and Computer Science,
University of Trieste, Italia
Center for Biomolecular Medicine, Area Science Park, Trieste
`luca@dmi.units.it`

Abstract

We explore the relation between the stochastic semantic associated to stochastic Concurrent Constrain Programming (sCCP) and its fluid-flow approximation. Writing the master equation for a sCCP model, we can show that the fluid flow equation is a first-order approximation of the true equation for the average. Moreover, we introduce a second-order correction and first-order equations for the variance and the covariance.
Keywords: Stochastic Concurrent Constraint Programming, ordinary differential equations, biological systems, fluid-flow approximation.

1 Introduction

Process algebras offer an elegant framework to describe a wide range of systems, from computer networks to biological systems. When additional information regarding (stochastic) speed of actions is added, we end up in the realm of stochastic process algebras (SPA), a widespread modeling technique in performance analysis [15] and in systems biology [20]. SPA are usually given a semantics in terms of continuous-time Markov Chains (CTMC [19]) - in fact, they can be seen as an high-level description language for CTMC. Therefore, all the analysis techniques for CTMC can be applied also to SPA, like steady-state or transient analysis. Moreover, many features of CTMC (like block decomposability) can be lifted to the level of SPA, greatly simplifying the analysis [15]. However, analyzing a stochastic system is a computationally expensive activity, and all previously mentioned techniques suffer severely from state space explosion. This phenomenon gets even worse when components are present in many copies, like molecules involved in chemical reactions. In order to handle more effectively these cases, in [16] a fluid-flow approximation method for PEPA [15] has been presented. The central idea in [16] was to approximate the number of each component's type with a continuous variable, describing its dynamical evolution with a set of ODEs derived analyzing statically the SPA model. The

same method has been developed also for other process algebras, like stochastic π-calculus [8] and stochastic Concurrent Constraint Programming [4]. The problem of continuously approximating SPA has, since then, received a lot of attention and it is considered an important topic for the applications in the biological field [7].

The main problem with these approximation methods is that their relation with the standard CTMC associated to a SPA model is not very clear. Practice showed that in many cases the approximation was satisfactory [7], although in other examples the method did not work [3]. Clearly, precise theoretical foundations are required, and research is moving quickly in this direction [14, 11, 12]. In [14, 11] the authors prove that for a particular class of PEPA models, namely those not cooperating,[1] the equations obtained —linear, in this case— describe exactly the evolution of the average of the system. Moreover, in [14] authors recognize that in the non-linear case the equations are only an approximation, as the differential equation for the average depends on all higher-order momenta, through a non-closed system of equations. Moreover, in [14] the authors suggest that, for some PEPA models, hybrid schemes of approximation should work better. This idea appeared also in [5], where authors use hybrid automata as a target formalism for approximating stochastic Concurrent Constraint Programming.

The question of relating stochastic and differential models is not a new one, and in fact it has been extensively studied in mathematical, physical, and chemical literature in order to compare stochastic and deterministic description of physical systems, from gases to chemical reactions [18]. Most of the methods used start from a (partial) differential equation for the time evolution of the probability mass distribution, known as Master Equation (ME). This equation is essentially equivalent to the Chapman-Kolmogorov forward equation [19], although it is more convenient to manipulate when the states of the system are vectors of (integer or real) variables whose evolution is determined by a fixed set of interactions. The ME is an exact equation, in the sense that it gives a precise picture of the evolution of the stochastic system. Unfortunately, it is almost impossible to solve, both analytically and numerically, hence approximate methods have been developed to deal with it.

In this paper we apply approximation methods for the ME to the fluid-flow approximation of SPA. The first step is, therefore, the definition of a master equation for stochastic Concurrent Constraint Programming (sCCP [1], a stochastic extension of CCP [21]). It turns out that defining the ME is rather simple, once some preliminary rewriting of sCCP programs in a more convenient graphical form is done. Besides, these first steps coincide with those needed to associate a set of ODEs to an sCCP program [4]. A related work in this direction is [9], where the author derives a master equation for a subset of stochastic π-calculus, in order to prove equivalence between the canonical stochastic model of a set chemical reactions and their description in stochastic π-calculus.

---

[1]More precisely, in [14] binary cooperation is possible only when on one side of the operator there is exactly one passive component.

$$Program = D.N; \quad D = \varepsilon \mid D.D \mid p : -A;$$
$$\pi = \text{tell}_\lambda(c) \mid \text{ask}_\lambda(c) \quad M = \pi.G \mid M + M;$$
$$G = \mathbf{0} \mid \text{tell}_\infty(c).G \mid p \mid M; \quad A = \mathbf{0} \mid M; \quad N = A \mid A \parallel N$$

Table 1: Syntax of restricted sCCP.

Instead, we will focus on the application of the master equation to justify the current derivations of differential equations from SPA model and to refine these methods. First of all, we will deduce the exact equation for the average of the stochastic process. Remarkably, when the rate functions describing interactions are linear, this equation coincide with the ODEs derived from fluid-flow approximation (thus giving a different proof of some results presented in [11, 14]). In general, we will show that these equations are just a first-order approximation of the exact equation for the average. Using a similar method, we will then deduce first-order equations for variance and covariance, introducing also a second-order correction for the average depending explicitly from variance and covariance.

The paper is organized as follows. Section 2 introduced briefly a subset of sCCP that will be used in the following. In particular, in Section 2.1 we overview the method to associate ODEs to an sCCP program. Section 3 contains the derivation of the master equation for an sCCP program. Section 4, instead, relates the ME with the average of the system, while in Section 5 we define the equations for the variance and the covariance. In Sections 5.1, 5.2, and 5.3 we give three simple examples, showing how the method works in practice. Finally, in Section 6 we draw conclusions and we present research directions to investigate further on.

2 Stochastic Concurrent Constraint Programming

Concurrent Constraint Programming (CCP [21]) is a process algebra in which agents interact exchanging information through a shared memory (the *constraint store*). Informational units are the *constraints*, i.e. interpreted first-order logical formulae stating relationships among variables. Agents interact asynchronously by adding new constraints (`tell`) and by checking if certain relations are entailed by the current configuration of the constraint store (`ask`).

The stochastic version of CCP (sCCP [1, 2, 6]) is obtained by adding an exponentially distributed stochastic duration to all instructions interacting with the constraint store. The rates are given by a *function* $\lambda : \mathcal{C} \to \mathbb{R}^+$ associating a positive real number to each configuration of the store. The underlying semantic model of the language (defined by structural operational semantics, cf. [1, 2]) is a CTMC.

In the following, we will use a restricted version of sCCP:

Definition 1 *A restricted sCCP-program (or* sCCP-network*) satisfies the following conditions:*

1. *its syntax is given by the grammar in Table 1;*

2. *the variables of the store are* $\mathbf{X} = (X_1, \ldots, X_n)$, *with domain* $\mathbb{Z}$; *they are all* stream *variables, i.e. growing lists representing time-varying quantities (cf. [1]). Moreover, their scope is always* global;

3. *constraints that can be added to the store are updates of stream variables of the form* $X' = X + k$, *with* $k \in \mathbb{Z}$ constant, *meaning that the value of* X *is replaced by* $X + k$;

4. *constraints that can be checked in* `ask` *instructions are positive boolean formulae containing* inequalities *involving stream variables only;*

5. *the* initial configuration *of the constraint store always consists in a conjunction of constraints of the form* $X = x_0$, *one for each variable in* $\mathbf{X}$. x_0 *is the* initial value *of variable* X.

The syntax defined in Table 1 imposes that all agents definable are *sequential*, i.e. they cannot contain any occurrence of the parallel operator, whose usage is restricted to the top level of the global network (denoted with N in Table 1). This implies that processes cannot be forked at run-time, hence their number remains constant during an execution. In addition, admitting only *global variables* we can avoid any parameter passing to called procedures.

In [2] and [6], we argued that sCCP can be conveniently used for modeling a wide range of biological systems, like biochemical reactions, genetic regulatory networks, the formation of protein complexes, and the process of folding of a protein. Actually, the restricted fragment of sCCP presented here suffices to deal with the first two classes of systems.

2.1 Reduced Transition Systems, Interaction Matrix and ODEs

Each sequential sCCP agent can be conveniently represented as a graph, with vertices corresponding to different stochastic choices and edges corresponding to transitions. The edges are labeled by the guard, the update, and the rate function of the corresponding transition.

Such graphs can be constructed from the syntactic tree of the sequential agent, simply merging all instantaneous transitions with the preceding stochastic one (updating consistently the label of the edges) and replacing a node corresponding to the call of a procedure p with the syntactic tree of p. This needs to be done at most once for each procedure p; all other nodes corresponding to calls to p are removed and their incoming edge redirected to the root of the unique copy of the syntactic tree of p.

The resulting graph for each component is called in [4] *reduced transition system (RTS)*. Given a sequential agent A, we denote its RTS by $RTS(A)$, the set of

its vertices (RTS-states) by $S(A) = \{s_1, \ldots, s_k\}$ and the set of its edges (RTS-transitions or simply transition) by $T(A) = \{t_1, \ldots, t_h\}$. Moreover, the guard of a transition t_j is denoted by $g(t_j) = g_j$ (with $g_j(\mathbf{X})$ we denote also its indicator function), its update by $u(t_j) = u_j$, and its rate function by $\lambda(t_j) = \lambda_j(\mathbf{X})$. As an example, below is a sCCP process composed by one single component A, together with its RTS ($*$ is shorthand for $true$):

A :- $\text{tell}_{\lambda_1}(X' = X + 1).\text{B}$
B :- $\text{ask}_{\lambda_2}(X > 0).\text{tell}_{\infty}(X' = X - 1).\text{A}$
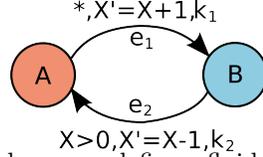
In order to simplify the analysis of an sCCP model, we can define a fluid-flow approximation of the system, by treating variables as continuous and describing their time-evolution by means of ODEs [4].

Starting from an sCCP network $\mathcal{N}$, with initial configuration $N = A_1 \parallel \ldots \parallel A_n$, we build the RTS for each sequential component A_i. Then, letting $S(N) = S(A_1) \cup \ldots \cup S(A_n)$ and $T(N) = T(A_1) \cup \ldots \cup T(A_n)$,[2] we associate a continuous variable to each RTS-state of $S(N)$. The variables $\mathbf{Y}$ of the differential equations will thus comprehend the stream variables $\mathbf{X}$ of the store, approximated as continuous, and all RTS-state variables.

Next, we build what we will call the *interaction matrix* ν of our sCCP-network. The interaction matrix has as many rows as system's variables $\mathbf{Y}$ and as many columns as the transitions in $T(N)$. In this matrix we store the updates (constant by Definition 1) that each transition induces on stream variables: for instance, if transition t_j increases variable X by 2, we put 2 in position $\nu_{X,j}$. Moreover, we put also a $^+\!/_-1$ in correspondence to the enter/exit RTS-state of the transition.

To write the ODEs, we simply need a final vector ϕ storing the (functional) rates of each transition in $T(N)$.[3] In addition, we multiply such rates by the indicator function of the guards of the RTS-edges (depending only on stream variables $\mathbf{X}$, cf. Definition 1) and by the variable corresponding to the exit RTS-state of the transition, denoted by $e(t_j) = e_j$. In summary,

$$\phi(t_j) = \phi_j(\mathbf{Y}) = \lambda_j(\mathbf{X})e_j g_j(\mathbf{X}). \tag{1}$$

Then, letting ν_j represent the j-th column vector of matrix ν, the differential equations are given by Φ^1:

$$\Phi^1 = \nu \cdot \phi = \sum_j \nu_j \phi_j. \tag{2}$$

The initial values are determined by the initial configuration of the store (cf. Definition 1) and by the initial state of each sequential agent. For the previous

---

[2] We consider states of different components as distinct.
[3] Here and in the following, we suppose to have fixed an ordering of RTS-states, stream variables and transitions.

example, we have ($\langle \cdot \rangle$ returns the logical value of a formula):

$$I = \begin{matrix} (X) \\ (A) \\ (B) \end{matrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix} \quad \phi = \begin{pmatrix} \lambda_1 \cdot A \\ \lambda_2 \cdot B \cdot \langle X > 0 \rangle \end{pmatrix} \quad \begin{cases} \dot{X} = \Phi_X^1 = \lambda_1 \cdot A - \lambda_2 \cdot B \cdot \langle X > 0 \rangle \\ \dot{A} = \Phi_A^1 = -\lambda_1 \cdot A + \lambda_2 \cdot B \cdot \langle X > 0 \rangle \\ \dot{B} = \Phi_B^1 = \lambda_1 \cdot A - \lambda_2 \cdot B \cdot \langle X > 0 \rangle \end{cases}$$

3 Master Equation for sCCP

The master equation for an sCCP network $N = A_1 \parallel \ldots \parallel A_n$ describes the time-variation of the probability mass function of the CTMC. First of all, we need to describe states of the CTMC by a set of variables. The choice is rather simple, given the method of the previous section. In fact, a state of the network is described by the values of the stream variables $\mathbf{X}$ and by the variables associated to RTS-states $S(N)$, i.e. by a valuation of the variables $\mathbf{Y}$. Note that in this case we are not assuming these variables to be continuous, like in the ODE construction; in fact, they will generally be integer-valued.

Consider now the probability of being in state $\mathbf{Y}$ at time t, denoted by $\mathcal{P}(\mathbf{Y}, t)$. Each transition entering in state $\mathbf{Y}$ will increase this probability, while all transitions leaving $\mathbf{Y}$ will decrease it. Consider now a transition $t_j \in T(N)$. Its effect on $\mathbf{Y}$ is described by the j-th column vector ν_j of the interaction matrix ν. In fact, the happening of t_j in $\mathbf{Y}$ will bring us in state $\mathbf{Y} + \nu_j$ (if $\mathbf{Y} + \nu_j$ does not belong to the domain of $\mathbf{Y}$, then transition t_j cannot happen — in this case, we assume its rate $\phi_j(\mathbf{Y})$ to be equal to zero). In addition, the probability that a transition t_j fires in the infinitesimal time dt, given that we are in state $\mathbf{Y}$, is $\phi_j(\mathbf{Y})dt$. The conditioning can be removed multiplying by $\mathcal{P}(\mathbf{Y}, t)$.

In summary, transition t_j increases $\mathcal{P}(\mathbf{Y}, t)$ in the infinitesimal time dt by

$$\mathcal{P}(\mathbf{Y} - \nu_j, t)\phi_j(\mathbf{Y} - \nu_j)dt,$$

and decrease it by

$$\mathcal{P}(\mathbf{Y}, t)\phi_j(\mathbf{Y})dt.$$

The first term describes the probability of t_j leading into state $\mathbf{Y}$, while the second term describes the probability of t_j leading out of state $\mathbf{Y}$.[4]

Summing over all transitions and dividing for dt, we get the *master equation for the sCCP network N*:

$$\frac{\partial \mathcal{P}(\mathbf{Y}, t)}{\partial t} = \sum_j \left(\phi_j(\mathbf{Y} - \nu_j)\mathcal{P}(\mathbf{Y} - \nu_j, t) - \phi_j(\mathbf{Y})\mathcal{P}(\mathbf{Y}, t) \right). \tag{3}$$

The initial conditions $\mathcal{P}(\mathbf{Y}, 0)$ for $\mathcal{P}$ are given by the initial configuration of the store (cf. Definition 1): at time 0 the system is in the state $\mathbf{Y_0}$ with probability 1, where the stream variables are determined by the constraints

---

[4]If $\mathbf{Y} - \nu_j$ does not belong to the domain of $\mathbf{Y}$, then we let $\phi_j(\mathbf{Y} - \nu_j) = 0$.

$X = x_0$, while the RTS-state variables are fixed by the initial state of each agent.

An important issue for the discussion to follow regards the continuity properties of rate functions ϕ_j. Recalling their definition given in equation 1, $\phi_j(\mathbf{Y}) = \lambda_j(\mathbf{X})e_j g_j(\mathbf{X})$, we note that ϕ_j is the product of three functions: e_j, λ_j, and g_j. e_j is a simple linear function (actually, a single variable), hence it is analytical. Generally, we can also expect λ_j to be an analytical function (this is the case, for instance, for biochemical reactions). On the contrary, g_j is an indicator function, hence discontinuous whenever the guard g_j is non trivial (i.e., different from *true*). Guards are generally used to synchronize agents; in this sense, *discontinuity seems the price to pay for synchronization*. In some cases, however, the discontinuous nature of g_j is absorbed by the function λ_j. For instance, suppose that $g_j(\mathbf{X}) = I(X_i > 0)$ and that λ_j vanishes for $X_i = 0$; for all non-negative reals it then holds $g_j \lambda_j = \lambda_j$, and so ϕ_j is continuous whenever λ_j is. This situation is not so uncommon, especially when guards are used to force upper and lower bounds on variables. An example are biochemical reactions, cf, below.

3.1 Relation with the Chemical Master Equation

Biochemical networks are generally expressed as a list of chemical reactions, encoding all possible actions the system can undergo. A general biochemical reaction has the form

$$R_1 + \ldots + R_n \to_{f(\mathbf{R};\mathbf{k})} P_1 + \ldots + P_m,$$
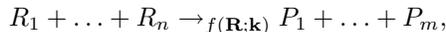
where $R_1, \ldots, R_n$ are the *reactants* and $P_1, \ldots, P_m$ are the *products*. The real-valued *kinetic function* of the reaction is $f(\mathbf{R}; \mathbf{k})$, depending on the reactants $\mathbf{R}$ and on some parameters $\mathbf{k}$. This function, generally analytical, can be one of the many used in biochemistry (cf. [10]), although usually it follows the mass action law (i.e., it is proportional to the concentration of the reactants). Moreover, it is required to satisfy the following *boundary condition*: it must be zero whenever one reactant is less than its amount consumed by the reaction. For instance, if the reaction consumes two molecules of R, then f must be zero for $R = 0, 1$[5]. In sCCP, each reaction is associated to the agent

$$f\text{-reaction}(\mathbf{R}, \mathbf{P}, \mathbf{k}) :-$$
$$\text{tell}_{f(\mathbf{R};\mathbf{k})}\big(\textstyle\bigwedge_{i=i}^{n}(R_i - 1) \wedge \bigwedge_{j=i}^{m}(P_j + 1)\big).$$
$$f\text{-reaction}(\mathbf{R}, \mathbf{P}, \mathbf{k}).$$

This agent is a simple recursive loop, modifying the value of reactants' and products' variables at a speed given by the kinetic law. Note that the boundary conditions for the rate function f imply that no stream variable will ever

---

[5]In case of mass action kinetics, this condition means that the rate for $R + R \to P$ must be $kR(R-1)$ and not kR^2. This is, however, consistent with the definition of the mass action principle in the stochastic setting.

become negative, as all reactions that may produce this effect have rate zero[6]. Consequently, we do not have to check domain constraints explicitly by guards, as their introduction would be redundant in the sense of the discussion at the end of previous section.

The RTS of this agent has just one state. In this case, the variable associated to it can be removed, as its equation would be $\dot{P} = 0$, $P(0) = 1$, and we can set $P \equiv 1$. Hence, the rate function ϕ associated to the agent f-reaction is $\phi = f(\mathbf{R}; \mathbf{k})$. Now, the *chemical master equation* is defined as equation 3 with f in place of ϕ, hence the master equation of the sCCP model of a set of biochemical reactions coincides with the chemical master equation. The same result has been proved in [9] for chemical π-calculus, for mass action reactions with at most 2 reactants.

4 First-order approximation

From the Master Equation 3 we can deduce an exact differential equation for the average of each system variable Y_i of $\mathbf{Y}$. At time t, the average value $\langle Y_i \rangle_t$ equals[7]

$$\langle Y_i \rangle_t = \sum_{\mathbf{Y}} Y_i \mathcal{P}(\mathbf{Y}, t),$$

where the sum ranges over all possible states of the system. As the only time-dependent quantity within the sum is $\mathcal{P}$, deriving both sides w.r.t. time t and substituting equation 3, we obtain

$$\frac{d\langle Y_i \rangle_t}{dt} = \sum_j \left[\sum_{\mathbf{Y}} Y_i \phi_j(\mathbf{Y} - \nu_j) \mathcal{P}(\mathbf{Y} - \nu_j, t) - \sum_{\mathbf{Y}} Y_i \phi_j(\mathbf{Y}) \mathcal{P}(\mathbf{Y}, t) \right]. \quad (4)$$

In the first summation in the right-hand side, we can substitute $\mathbf{Y}$ with $\mathbf{W_j} + \nu_j$, breaking the sum in two pieces (one piece for W_{ij} and one piece for ν_{ij}). The first part cancels out with the second summation in 4, hence we are left with

$$\frac{d\langle Y_i \rangle_t}{dt} = \sum_j \nu_{ij} \sum_{\mathbf{W_j}} \phi_j(\mathbf{W_j}) \mathcal{P}(\mathbf{W_j}, t).$$

Applying the definition of the average we obtain $\sum_{W_j} \phi_j(\mathbf{W_j}) \mathcal{P}(\mathbf{W_j}, t) = \langle \phi_j \rangle_t$; recalling equation 2, we get

$$\frac{d\langle Y_i \rangle_t}{dt} = \left\langle \Phi_i^1(\mathbf{Y}) \right\rangle_t \quad (5)$$

---

[6]Boundary conditions for f may be relaxed by checking explicitly with ask instructions that variables stay within their domain. For instance, for the reaction $R + R \rightarrow P$, we can precede `tell` by `ask(R > 1)` . This allows us to use the more common kR^2 as rate function, at the price of introducing discontinuity in the rates.

[7]We adopt the conventions used among physicists to denote average and covariance. Namely, $\langle X \rangle$ indicates the average of X, while $\langle\langle XY \rangle\rangle$ denotes the covariance of X and Y. $\langle\langle X^2 \rangle\rangle$ indicates the variance of X.

Thus, the average value of $\mathbf{Y}$ in the stochastic system evolves as the *average* of the function $\Phi^1(\mathbf{Y})$, which is the vector of functions of the ODE system associated to the sCCP-network N by the fluid-flow approximation reviewed in Section 2.1.

Remark 1 *If all rate functions ϕ_j are* linear, *then so is $\Phi_i^1(\mathbf{Y})$, and thus, thanks to the linearity of expectation $\langle \cdot \rangle_t$, the equation 5 reduces to*

$$\frac{d \langle Y_i \rangle_t}{dt} = \Phi_i^1(\langle \mathbf{Y} \rangle_t),$$

which is equation 2. Therefore, if all rate functions ϕ_j are linear, the method of Section 2.1 provides the exact equation for the average. This is essentially the same result proved in [11, 14].

Of course, if the linearity condition on ϕ_j does not hold, then the previous remark is no more valid. In this case, the equation 2 is no more the correct equation for the average of the system. However, it is a first order approximation, hence reasonable whenever fluctuations are small. To see this, consider the time-dependent Taylor expansion of function $\phi_j(\mathbf{Y})$ around the average value $\langle \mathbf{Y} \rangle_t$; for simplicity we truncate it at second order:

$$\phi_j(\mathbf{Y}) \approx \phi_j(\langle \mathbf{Y} \rangle_t) \quad + \sum_{k=1}^{|\mathbf{Y}|} \partial_k \phi_j(\langle \mathbf{Y} \rangle_t)(Y_k - \langle Y_k \rangle_t)$$
$$+ \frac{1}{2} \sum_{h,k=1}^{|\mathbf{Y}|} \partial_{hk}^2 \phi_j(\langle \mathbf{Y} \rangle_t)(Y_h - \langle Y_h \rangle_t)(Y_k - \langle Y_k \rangle_t),$$

where ∂_k denotes the partial derivative w.r.t. Y_k. Taking the average of both sides, the linear term cancels out, as $\langle (Y_k - \langle Y_k \rangle_t) \rangle_t = 0$; moreover the covariance of Y_h and Y_k at time t, $\langle \langle Y_h Y_k \rangle \rangle_t = \langle (Y_h - \langle Y_h \rangle_t)(Y_k - \langle Y_k \rangle_t) \rangle_t$, appears in the second-order term. Therefore, as $\langle \Phi^1(\mathbf{Y}) \rangle_t = \sum_j \nu_j \langle \phi_j(\mathbf{Y}) \rangle$, we obtain for $\langle \Phi^1(\mathbf{Y}) \rangle_t$ at second order:

$$\langle \Phi^1(\mathbf{Y}) \rangle_t \approx \Phi^1(\langle \mathbf{Y} \rangle_t) + \frac{1}{2} \sum_{h,k=1}^{|\mathbf{Y}|} \partial_{hk}^2 \Phi^1(\langle \mathbf{Y} \rangle_t) \langle \langle Y_h Y_k \rangle \rangle_t \tag{6}$$

If we truncate the previous approximation at first order, then we obtain $\langle \Phi^1(\mathbf{Y}) \rangle_t \approx \Phi^1(\langle \mathbf{Y} \rangle_t)$, and so

$$\frac{d \langle Y_i \rangle_t}{dt} = \Phi_i^1(\langle \mathbf{Y} \rangle_t).$$

The previous method works *only if* the functions ϕ_j can be expanded in Taylor series (at least up to second order, hence ϕ_j must have continuous second order derivatives). Preferably, ϕ_j should be analytic. In the presence of guards, this may not be true, even if all rates λ_j are analytic. In this case, a different treatment is needed, cf. Section 6 for further comments. However, for the rest of the paper, we suppose ϕ_j to be analytic.

5 Second-order approximation

Equation 6 shows that, at second order, the average depends also on the variance and covariance of system variables. By the way, by taking the expansion to higher orders, we end up having an equation potentially depending on all higher order momenta. We will comment more on this later. In any case, if we can get equations for the variance and covariance, we may use the equation 6 to improve the approximate equation 2 for the average, taking into account fluctuations. To get equations for the variance and covariance, we can proceed similarly to the average: we deduce an exact equation from the master equation and then we linearize it.

First of all, we derive the differential equation for $\langle Y_i Y_k \rangle_t$:

$$\frac{d \langle Y_i Y_k \rangle_t}{dt} = \sum_{\mathbf{Y}} Y_i Y_k \frac{\partial \mathcal{P}(\mathbf{Y}, t)}{\partial t}.$$

By substituting equation 3 to $\frac{\partial \mathcal{P}(\mathbf{Y}, t)}{\partial t}$, letting $\mathbf{W} = \mathbf{Y} - \nu_j$, as in Section 4, and setting

$$\Phi_{ik}^2(\mathbf{Y}) = \sum_j \nu_{ij} \nu_{kj} \phi_j(\mathbf{Y}),$$

we obtain the following

$$\frac{d \langle Y_i Y_k \rangle_t}{dt} = \left\langle \Phi_{ik}^2(\mathbf{Y}) \right\rangle_t + \left\langle Y_i \Phi_k^1(\mathbf{Y}) \right\rangle_t + \left\langle Y_k \Phi_i^1(\mathbf{Y}) \right\rangle_t.$$

The exact equation for the covariance $\langle\langle Y_i Y_k \rangle\rangle_t$ can now be obtained easily from:

$$\frac{d \langle\langle Y_i Y_k \rangle\rangle_t}{dt} = \frac{d}{dt} \left(\langle Y_i Y_k \rangle_t - \langle Y_i \rangle_t \langle Y_k \rangle_t \right).$$

After simple manipulations, we obtain

$$\frac{d \langle\langle Y_i Y_k \rangle\rangle_t}{dt} = \left\langle \Phi_{ik}^2(\mathbf{Y}) \right\rangle_t + \left\langle (Y_i - \langle Y_i \rangle_t) \Phi_k^1(\mathbf{Y}) \right\rangle_t + \left\langle (Y_k - \langle Y_k \rangle_t) \Phi_i^1(\mathbf{Y}) \right\rangle_t \quad (7)$$

This equation involves the average of functions over the state space, and the right hand side cannot in general even be computed. However, if we do a first-order expansion of functions Φ^1 and Φ^2, as in the previous section, we can obtain the following approximate equation:

$$\frac{d \langle\langle Y_i Y_k \rangle\rangle_t}{dt} = \Phi_{ik}^2(\langle \mathbf{Y} \rangle_t) \quad + \sum_{h=1}^{|\mathbf{Y}|} \partial_h \Phi_k^1(\langle \mathbf{Y} \rangle_t) \langle\langle Y_i Y_h \rangle\rangle_t$$

$$+ \sum_{h=1}^{|\mathbf{Y}|} \partial_h \Phi_i^1(\langle \mathbf{Y} \rangle_t) \langle\langle Y_k Y_h \rangle\rangle_t. \quad (8)$$

10

First of all, note that in case the functions ϕ_j are linear, then the previous equation is exact. If this is not the case, then expansion of Φ^1 up to first order is essential, otherwise higher order momenta appear in the equation. In fact, covariance appears in the RHS because the term $(Y_i - \langle Y_i \rangle_t)$ gets multiplied by the factor $(Y_h - \langle Y_h \rangle_t)$ of the first order term in the Taylor expansion of Φ^1. Therefore, the factor $(Y_h - \langle Y_l \rangle_t)(Y_h - \langle Y_l \rangle_t)$ of the second order term in this expansion would bring in the third order momenta. This remark shows that, when ϕ_j are not linear, then the exact system of ODEs for the average is not closed, in the sense that equation for momenta of order k involve momenta of higher order, leading to an infinite set of equations (this fact has been observed also in [14]). The only way to overcome this infinity is to linearize equations for the highest order momentum we are taking into account. However, even if in theory we can introduce momenta of order as high as desired, we are concretely limited by the number of variables needed: for k-th order momenta, we need $O(n^k)$ new variables, an effort that may be not justified.

Equation 8 is a first order ODE for the variance and covariance. Note that it can be derived from an sCCP network using the purely syntactic method presented in Section 2.1: the functions Φ^2_{ij} can be computed easily, given the knowledge of the rate vector ϕ_j and of the interaction matrix ν, while the derivatives of Φ^1 can be computed by symbolic derivation. Such equations can give a picture of the fluctuations of the system, even if in an approximate way. They are not exact, but they indeed are more informative than the equations for the average alone. We can see this in the following simple examples.

Once we have the equations for the variance and covariance of system variables, we can use equation 6 to add a second order correction also for the average, obtaining

$$\frac{d \langle Y_i \rangle_t}{dt} = \Phi^1(\langle \mathbf{Y} \rangle_t) + \frac{1}{2} \sum_{h,k=1}^{|T(N)|} \partial^2_{hk} \Phi^1(\langle \mathbf{Y} \rangle_t) \langle\langle Y_h Y_k \rangle\rangle_t. \tag{9}$$

5.1 Example: Random Walk

We consider a random walk in one variable X, expressed by the following sCCP process

$$\mathrm{RW}_X :\text{-} \mathrm{tell}_k(X = X + 1).\mathrm{RW}_X + \mathrm{tell}_k(X = X - 1).\mathrm{RW}_X,$$

where the rate of both tell actions is constant and equal to k. Since the RTS of this agent has a single state, with two edges t_1 and t_2 looping on it, we may forget about the state variable (it will have equation $\dot{A} = 0$, $A_0 = 1$, so it can be eliminated by setting $A \equiv 1$). The two rate functions will then be $\phi_1(X) = k$ and $\phi_2(X) = k$, while the interaction matrix ν will be equal to $\nu = (\nu_1, \nu_2) = (1, -1)$. Therefore, the Φ functions are

$$\Phi^1(X) = 0, \quad \Phi^2(X) = 2k,$$

11

giving the following two equations for the average $\langle X \rangle$ and the variance $\langle\langle X^2 \rangle\rangle$ of X:

$$\begin{cases} \langle \dot{X} \rangle = \Phi^1(\langle X \rangle) + \frac{1}{2}\langle\langle X^2 \rangle\rangle \partial^2_{XX}\Phi^1(\langle X \rangle) = 0 \\ \langle\langle \dot{X^2} \rangle\rangle = \Phi^2(\langle X \rangle) + 2\langle\langle X^2 \rangle\rangle \partial_X \Phi^1(\langle X \rangle) = 2k \end{cases}$$

These very simple equations can be readily solved, yielding $\langle X \rangle_t = X_0$ and $\langle\langle X^2 \rangle\rangle_t = 2kt + \langle\langle X_0^2 \rangle\rangle$. These equations are exact, as the rate functions are linear, giving the well known result that a random walk has constant average but linear growing variance. Note that the equation for the average alone, despite being exact, is not very informative, as it does not capture the fact that the variable X has unbounded variance (hence it will eventually reach any number in $\mathbb{Z}$ with probability 1).

5.2 Example: Dimerization

As another example, we consider the dimerization of a protein P. This is a biochemical system composed by two molecule's types, P and its dimer P_2, and two reactions, $P + P \rightarrow_{k_1} P_2$ and its inverse $P_2 \rightarrow_{k_2} P + P$. The rates associated to these to reactions are $\phi_1 = k_1 X(X-1)/2$ and $\phi_2 = k_2 Y$, with X giving the quantity of P molecules and Y giving the quantity of P_2. Due to the results in Section 3.1, these are also the rate functions of the sCCP network describing the dimerization system. Notice that this is a closed system, as the number of P molecules is *constant*; formally $X + 2Y = X_0 + 2Y_0 = 2C$. This observation allows to remove one variable from the system, setting $X = 2C - 2Y$. This one-dimensional system is a birth-death process, and so it can be solved exactly. For instance, setting $k_1 = 0.00166$, $k_2 = 0.2$, $C = 150$, and $Y_0 = 0$, we obtain that the average value of Y converges to the stationary value of $\langle Y \rangle_\infty = 80.3512$, while the variance converges to $\langle\langle Y^2 \rangle\rangle_\infty = 24.3005$.

Using the method just presented, we can write the first order and the second order approximation for $\langle Y \rangle$. For the first order, we have:

$$\langle \dot{Y} \rangle = k_1(C - \langle Y \rangle)(2C - 2\langle Y \rangle - 1) - k_2 \langle Y \rangle.$$

The second order correction, instead, gives rise to the following set of equations for $\langle Y \rangle$ and its variance $\langle\langle Y^2 \rangle\rangle$:

$$\begin{cases} \langle \dot{Y} \rangle = k_1(C - \langle Y \rangle)(2C - 2\langle Y \rangle - 1) - k_2 \langle Y \rangle + 2k_1 \langle\langle Y^2 \rangle\rangle \\ \langle\langle \dot{Y^2} \rangle\rangle = k_1(C - \langle Y \rangle)(2C - 2\langle Y \rangle - 1) + k_2 \langle Y \rangle + \langle\langle Y^2 \rangle\rangle (4k_1 \langle Y \rangle - 4k_1 C + k_1 - k_2) \end{cases}$$

These equations can be solved numerically, and they both predict for $\langle Y \rangle$ convergence to an asymptotic value. For first order approximation, this value equals $\langle Y \rangle_\infty = 80.2291$, while at second order we have $\langle Y \rangle_\infty = 80.3509$, closer to the true mean. Moreover, variance converges asymptotically to $\langle\langle Y^2 \rangle\rangle_\infty = 24.258$, also in good agreement with the true variance.

5.3 Example: Effects of Variance

Consider the following biochemical system, described by a set of mass action reactions:

$$\emptyset \to_k X; \quad \emptyset \to_k Y; \quad X \to_{\alpha_1} \emptyset$$
$$Y \to_{\alpha_2} \emptyset; \quad X + Y \to_{k_a} \emptyset$$
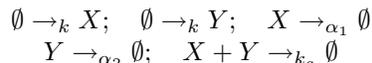
Its sCCP code can be obtained easily according to the prescriptions of Section 3.1. In Figure 1 we show the result of a stochastic simulation via Gillespie Algorithm [13]. As readily seen, the effect of fluctuations is predominant, and the standard deviation is of the same order of the average. Therefore, equations for the average, even if accurate, may fail to give a detailed picture of this system: the knowledge of the variance is essential.

Using the method of this paper, we can obtain approximate differential equations for the average and the variance. There is a total of 5 variables: two for the average ($\langle X \rangle$ and $\langle Y \rangle$), two for the variance ($\langle\langle X^2 \rangle\rangle$, and $\langle\langle Y^2 \rangle\rangle$) and one for the covariance ($\langle\langle XY \rangle\rangle$). The resulting equations are
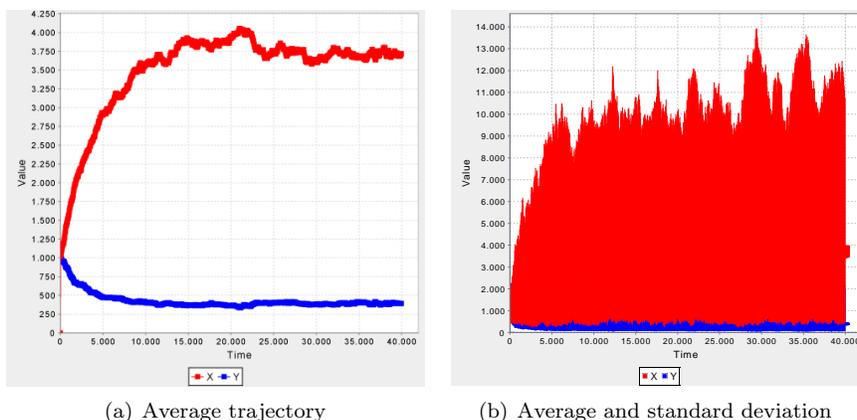

(a) Average trajectory   (b) Average and standard deviation

Figure 1: (**1(a)**) Average trajectory from 250 runs of the system of Section 5.3. Parameters were set equal to $k = 1000$, $\alpha_1 = 10^{-4}$, $\alpha_2 = k_a = 10^{-3}$. (**1(b)**) In this plot also the standard deviation is shown. As can be seen, the fluctuations are predominant: the sole average thus does not give an accurate picture. Note that standard deviation seems to be of the same order of the average.
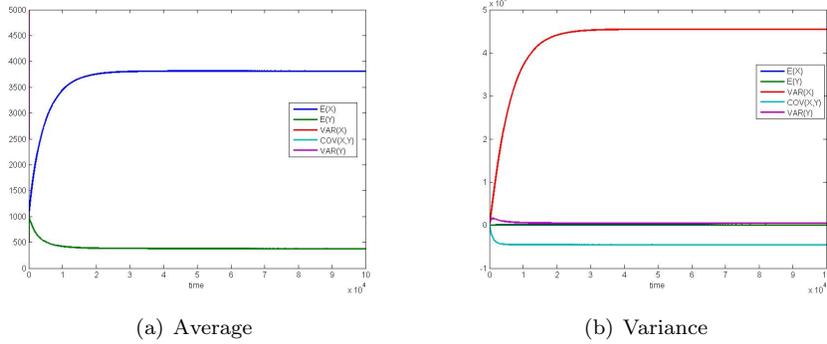
(a) Average

(b) Variance

Figure 2: (**2(a)**) Numerical solution for second-order equations for the average. Parameters are like those in caption of Figure 1. (**2(b)**) Numerical solution of equations for the variance and covariance.

$$
\begin{aligned}
\langle \dot{X} \rangle &= k - \alpha_1 \langle X \rangle - k_a \langle X \rangle \langle Y \rangle - k_a \langle\langle XY \rangle\rangle \\
\langle \dot{Y} \rangle &= k - \alpha_2 \langle Y \rangle - k_a \langle X \rangle \langle Y \rangle - k_a \langle\langle XY \rangle\rangle \\
\langle\langle \dot{X^2} \rangle\rangle &= k + \alpha_1 \langle X \rangle + k_a \langle X \rangle \langle Y \rangle - 2\alpha_1 \langle\langle X^2 \rangle\rangle \\
&\quad - 2k_a \langle Y \rangle \langle\langle X^2 \rangle\rangle - 2k_a \langle X \rangle \langle\langle XY \rangle\rangle \\
\langle\langle \dot{XY} \rangle\rangle &= k_a \langle X \rangle \langle Y \rangle - k_a \langle Y \rangle \langle\langle X^2 \rangle\rangle - \\
&\quad (\alpha_1 + \alpha_2 + k_a \langle X \rangle + k_a \langle Y \rangle) \langle\langle XY \rangle\rangle - k_a \langle X \rangle \langle\langle Y^2 \rangle\rangle \\
\langle\langle \dot{Y^2} \rangle\rangle &= k + \alpha_2 \langle Y \rangle + k_a \langle X \rangle \langle Y \rangle - 2\alpha_2 \langle\langle Y^2 \rangle\rangle \\
&\quad - 2k_a \langle X \rangle \langle\langle Y^2 \rangle\rangle - 2k_a \langle Y \rangle \langle\langle XY \rangle\rangle
\end{aligned}
$$

Their solution is shown in Figure 2. As we can see, the variance is of the order of 10^6 at equilibrium, suggesting a standard deviation of order 10^3, in agreement with the simulation results.

6 Conclusions and Future Work

In this paper we defined a differential equation for the time-evolution of the probability mass distribution, the so called master equation, for a model written in stochastic Concurrent Constraint Programming. From this equation, we derived exact differential equation for the average and the variance/covariance of the model. These equations contain averages on the right hand side, which can be eliminated by linearization or Taylor expansion up to second (or higher) order. As a collateral result, we obtained the "standard" fluid-flow equations as first-order approximations of the exact equation for the average. The most important fact is, however, the derivation of an approximate set of equations for the variance and covariance, which can be obtained by a simple modification of the method for building the fluid-flow approximation of a generic sCCP model.

These equations can be used to draw a more detailed picture of the system's evolution, giving information on the average magnitude of fluctuations and, consequently, on the goodness of the fluid-flow approximation.

Starting from the ME, there are other approximation methods that can be applied to sCCP. For instance, if one can introduce a meaningful notion of *system size* Ω for the sCCP model[8], then a Taylor expansion around Ω of the master equation can lead to the so-called *linear noise approximation* [18]: the equation for the average, following the first-order approximation, is coupled with a linear Fokker-Planck stochastic differential equation governing the offset from the solution of the average equation.

However, sCCP (and SPA in general) has characteristic features that present some challenges for the application of these approximation techniques. Specifically, in presence of synchronization, sCCP models have discontinuous rate functions, which cannot be expanded in Taylor series over their whole domain of definition. In this case, a different treatment is required. Usually, the functions ϕ_j will be piecewise analytic, and their analytic regions will be separated by discontinuity boundaries of measure zero. Consequently, we may use an hybrid approximation scheme, expanding in series functions ϕ_j only within their analytic regions, while moving discretely from one region to another when a boundary is hit. However, near discontinuity boundaries, approximations of stochastic processes with ODEs can lead to big errors: for instance, the continuous trajectory can converge asymptotically to the border, while the stochastic trajectory can cross it in finite time, entering in a different dynamical regime. In these case, we can keep the stochastic ingredient, using stochastic differential equations instead of ODEs and maintaining the transitions discrete near the boundaries, de facto approximating the sCCP model with a (stochastic) hybrid automaton [17].

As a matter of fact, the second order correction of fluid-flow approximation presented here is still too crude for sCCP models in which the stochasticity or the discreteness play a dominant role in the dynamical evolution (see [5] for an example). In these cases, we again expect that an approximation based on (stochastic) hybrid automata will outperform the one based solely on ODE's. We are currently investigating this direction [5].

References

[1] L. Bortolussi. Stochastic concurrent constraint programming. In *Proc. of QAPL 2006*, ENTCS 164, pages 65–80, 2006.

---

[8]For instance, in [11] the authors suggest the notion of density, i.e. the ratio between the current number of agents of a given type and the maximum number of agents of that type, which may work for closed systems.

[2] L. Bortolussi. *Constraint-based approaches to stochastic dynamics of biological systems*. PhD thesis, PhD in Computer Science, University of Udine, 2007. Available at `http://www.dmi.units.it/~bortolu/files/reps/Bortolussi-PhDThesis.pdf`.

[3] L. Bortolussi and A. Policriti. Dynamical systems and stochastic programming I - ordinary differential equations. Submitted to *Trans. Comp. Sys. Bio., available upon request*, 2007.

[4] L. Bortolussi and A. Policriti. Stochastic concurrent constraint programming and differential equations. In *Proc. of QAPL 2007*, ENTCS 16713, 2007.

[5] L. Bortolussi and A. Policriti. Hybrid approximation of stochastic process algebras for systems biology. *To be presented at IFAC 2008 world congress, Seoul, South Korea. Available upon request.*

[6] L. Bortolussi and A. Policriti. Modeling biological systems in concurrent constraint programming. *Constraints*, in press.

[7] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of rkip on the erk signalling pathway using the stochastic process algebra pepa. *Trans. on Comp. Sys. Bio.*, 4230:1–23, 2006.

[8] L. Cardelli. From processes to odes by chemistry. *downloadable from `http://lucacardelli.name/`*, 2006.

[9] L. Cardelli. A process algebra master equation. In *Proc. of QEST 2007*, 2007.

[10] A. Cornish-Bowden. *Fundamentals of Chemical Kinetics*. Portland Press, 3rd edition, 2004.

[11] J. Ding and J. Hillston. On odes from pepa models. In *Proc. of PASTA 2007*, 2007.

[12] N. Geisweiller, J. Hillston, and M. Stenico. Relating continuous and discrete pepa models of signalling pathways. *Theoretical Computer Science*, in print, 2008.

[13] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. of Physical Chemistry*, 81(25), 1977.

[14] R. A. Hayden and J. T. Bradley. Fluid-flow solutions in pepa to the state space explosion problem. In *Proc of PASTA 2007*, 2007.

[15] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.

[16] J. Hillston. Fluid flow approximation of pepa models. In *Proc. of QEST 2005*, 2005.

[17] J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In *Proc. of HSCC 2000*, LNCS 1790, 2000.

[18] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 1992.

[19] J. R. Norris. *Markov Chains*. Cambridge University Press, 1997.

[20] A. Regev and E. Shapiro. Cellular abstractions: Cells as computation. *Nature*, 419, 2002.

[21] V. A. Saraswat. *Concurrent Constraint Programming*. MIT press, 1993.

[22] J. M. G. Vilar, H. Yuan Kueh, N. Barkai, and S. Leibler. Mechanisms of noise resistance in genetic oscillators. *PNAS*, 99(9):5991, 2002.