Fluid and Mean Field Approximation: Hybrid fluid limits

Luca Bortolussi[1,2]

[1] Dipartimento di Matematica e Geoscienze
Università degli studi di Trieste
[2] CNR/ISTI, Via Moruzzi, Pisa, Italy

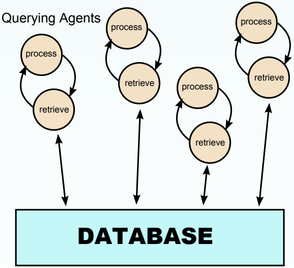
luca@dmi.units.it

IMT, Lucca, September 30 - October 4, 2013

Outline

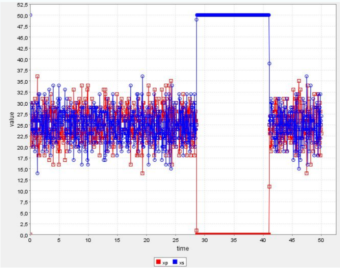
Outline

A SIMPLE EXAMPLE: SERVER WITH FAILURE



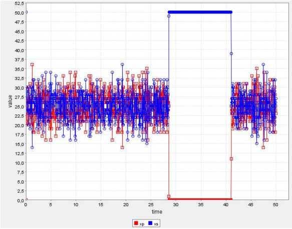
X_d, X_b: database server (working/broken)

X_s, X_p: clients (requesting/ processing)

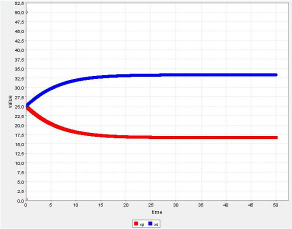


$$\text{request: } (\cdot, X_s' = X_s - 1 \wedge X_p' = X_p + 1, k_s X_s X_d)$$
$$\text{process: } (\cdot, X_p' = X_p - 1 \wedge X_s' = X_s + 1, k_p X_p)$$
$$\text{break: } (\cdot, X_d' = X_d - 1 \wedge X_b' = X_b + 1, k_b X_d)$$
$$\text{repair: } (\cdot, X_b' = X_b - 1 \wedge X_d' = X_d + 1, k_r X_b)$$

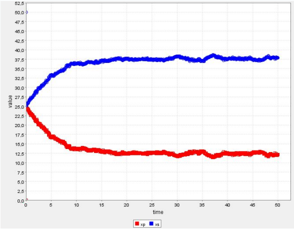
CTMC VS ODE



stochastic



ODE

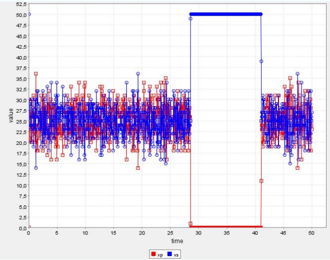


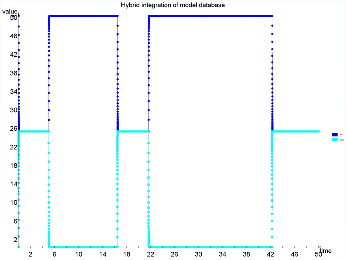
average

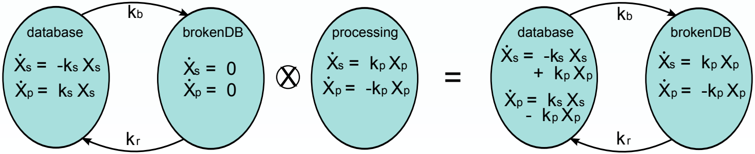
ISSUES

- Fluid semantics brings just (approximate) information about average
- The CTMC and ODE systems are not “behaviorally equivalent” (in terms of CSL/CTL formulae)

HYBRID SEMANTICS







HYBRID SYSTEMS

Many real systems have a double nature. They:

- evolve in a continuous way,
- are ruled by a discrete system,
- are subject to stochastic events.

MODELING?

(stochastic) hybrid systems/automata

HYBRID SYSTEMS

Many real systems have a double nature. They:

- evolve in a continuous way,
- are ruled by a discrete system,
- are subject to stochastic events.

MODELING?

(stochastic) hybrid systems/automata

HYBRID SYSTEMS

Many real systems have a double nature. They:

- evolve in a continuous way,
- are ruled by a discrete system,
- are subject to stochastic events.

MODELING?

(stochastic) hybrid systems/automata

HYBRID SYSTEMS

Many real systems have a double nature. They:

- evolve in a continuous way,
- are ruled by a discrete system,
- are subject to stochastic events.

MODELING?

(stochastic) hybrid systems/automata

HYBRID SYSTEMS

Many real systems have a double nature. They:

- evolve in a continuous way,
- are ruled by a discrete system,
- are subject to stochastic events.

MODELING?

(stochastic) hybrid systems/automata

Overview of the hybrid semantics

Population CTMC

↓

Transition Driven Stochastic Hybrid Automata

↓

Piecewise Deterministic Markov Processes

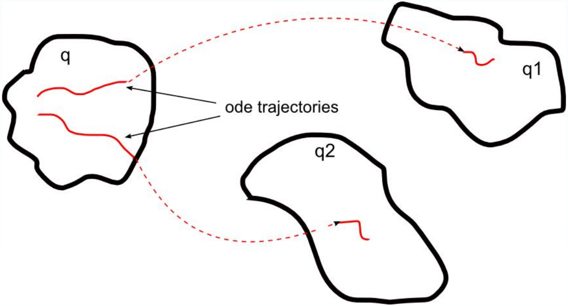
Piecewise Deterministic Markov Processes

A PDMP is a tuple $(Q, D, \mathcal{X}, \lambda, R)$, such that:

- Q is a finite set of *modes* or *discrete states*. $D_q \subseteq \mathbb{R}^n$ is an open set.
- The *hybrid state space* is $D = \bigcup_{q \in Q} \{q\} \times D_q$.
- $\mathcal{X}_q : D_q \to \mathbb{R}^n$ is a *locally Lipschitz continuous vector field*.
- $\lambda : D \to \mathbb{R}^+$ is the *jump rate*.
- $R : D \cup \partial D \times \mathcal{D} \to [0, 1]$ is the *transition measure* or *reset kernel*.

DYNAMICAL EVOLUTION

- Within each mode q, the process evolves according to the flow $\phi_q(t, \mathbf{x_0})$ of the vector field $\mathcal{X}$.
- While in a mode, the process can jump spontaneously with hazard given by the rate function λ.
- A jump is immediately performed whenever the boundary of the state space of the current mode is hit.



REGULARITY CONDITIONS

1. $\mathcal{X}$ locally Lipschitz continuous and explosion-free.
2. $\forall y_0 = (q, \mathbf{x_0}) \in D,\ \exists \varepsilon(y_0) > 0 : t \mapsto \lambda(q, \phi_q(t, \mathbf{x_0}))$ is integrable in $[0, \varepsilon(y_0)[$.
3. For each $A \in \mathcal{D}$, $y \mapsto R(y, A)$ is measurable.
4. $R(y, \{y\}) = 0$, for each $y \in D$.
5. $\forall t, \mathbb{E}N_t < \infty$, $N_t = \sum_k I_{t > T_k}$.

DYNAMICAL EVOLUTION – DETAILS

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

JUMP TIMES

- The times at which a PDMP jumps are random variables $\tau_1, \tau_2, \ldots$. As a PDMP has the strong Markov property, we can reason on τ_1. For τ_i, stop the process after the jump and restart it.
- Hitting time of the boundary ∂D_q:

$$t_*(y) = \begin{cases} \inf\{t > 0 \mid \phi_q(t, \mathbf{x}) \in \partial D_q\} \\ \infty, \text{ if no such time exists,} \end{cases}$$

DYNAMICAL EVOLUTION – DETAILS

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

JUMP TIMES

- The times at which a PDMP jumps are random variables $\tau_1, \tau_2, \ldots$. As a PDMP has the strong Markov property, we can reason on τ_1. For τ_i, stop the process after the jump and restart it.
- Hitting time of the boundary ∂D_q:

$$t_*(y) = \begin{cases} \inf\{t > 0 \mid \phi_q(t, \mathbf{x}) \in \partial D_q\} \\ \infty, \text{ if no such time exists,} \end{cases}$$

DYNAMICAL EVOLUTION – DETAILS

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0, 1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

JUMP TIMES

- The times at which a PDMP jumps are random variables $\tau_1, \tau_2, \ldots$. As a PDMP has the strong Markov property, we can reason on τ_1. For τ_i, stop the process after the jump and restart it.
- Hitting time of the boundary ∂D_q:

$$t_*(y) = \begin{cases} \inf\{t > 0 \mid \phi_q(t, \mathbf{x}) \in \partial D_q\} \\ \infty, \text{ if no such time exists,} \end{cases}$$

Dynamical evolution – details

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0, 1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

Jump times

- Survival function for τ_1:

$$F(t, y) = \mathbb{P}(T_1 \geq t) = I_{t<t_*(y)} \exp\left(-\int_0^t \lambda(q, \phi_q(s, \mathbf{x}))\mathrm{d}s\right).$$

- τ_1 can be determined by solving for t the equation $F(t, y) = U$, with U uniform r.v. in $[0, 1]$ (use one coordinate of $\mathcal{H}$).

Dynamical evolution – details

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

Jump times

- Survival function for τ_1:

$$F(t,y) = \mathbb{P}(T_1 \geq t) = I_{t<t_*(y)} \exp\left(-\int_0^t \lambda(q, \phi_q(s, \mathbf{x}))\mathrm{d}s\right).$$

- τ_1 can be determined by solving for t the equation $F(t,y) = U$, with U uniform r.v. in $[0,1]$ (use one coordinate of $\mathcal{H}$).

Dynamical evolution – details

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

Jump times

- Survival function for τ_1:

$$F(t,y) = \mathbb{P}(T_1 \geq t) = I_{t<t_*(y)} \exp\left(-\int_0^t \lambda(q, \phi_q(s,\mathbf{x}))\mathrm{d}s\right).$$

- τ_1 can be determined by solving for t the equation $F(t,y) = U$, with U uniform r.v. in $[0,1]$ (use one coordinate of $\mathcal{H}$).

Dynamical evolution – details

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

Target state of jumps

- At time τ_1^-, just before the first jump, the PDMP is in state $\mathbf{x}(\tau_1^-) = \phi_q(\tau_1, \mathbf{x}_0)$.
- The state after the jump has distribution $R(\mathbf{x}(\tau_1^-), \cdot)$.
- We can sample from this distribution using a uniform r.v. via the following inversion result: there is a function $\psi : [0,1] \times \mathcal{D} \to D$, $\psi_{\mathbf{x}}(u) = \psi(u, \mathbf{x})$, such that

$$R(\mathbf{x}, A) = \mu_\ell(\psi_{\mathbf{x}}^{-1}(A)).$$

DYNAMICAL EVOLUTION – DETAILS

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0, 1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

TARGET STATE OF JUMPS

- At time τ_1^-, just before the first jump, the PDMP is in state $\mathbf{x}(\tau_1^-) = \phi_q(\tau_1, \mathbf{x}_0)$.
- The state after the jump has distribution $R(\mathbf{x}(\tau_1^-), \cdot)$.
- We can sample from this distribution using a uniform r.v. via the following inversion result: there is a function $\psi : [0, 1] \times \mathcal{D} \to D$, $\psi_{\mathbf{x}}(u) = \psi(u, \mathbf{x})$, such that

$$R(\mathbf{x}, A) = \mu_\ell(\psi_{\mathbf{x}}^{-1}(A)).$$

Dynamical evolution – details

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

Target state of jumps

- At time τ_1^-, just before the first jump, the PDMP is in state $\mathbf{x}(\tau_1^-) = \phi_q(\tau_1, \mathbf{x}_0)$.
- The state after the jump has distribution $R(\mathbf{x}(\tau_1^-), \cdot)$.
- We can sample from this distribution using a uniform r.v. via the following inversion result: there is a function $\psi : [0,1] \times \mathcal{D} \to D$, $\psi_{\mathbf{x}}(u) = \psi(u, \mathbf{x})$, such that

$$R(\mathbf{x}, A) = \mu_\ell(\psi_{\mathbf{x}}^{-1}(A)).$$

Dynamical evolution – details

We construct a realization of a PDMP on the Hilbert cube $\mathcal{H} = [0,1]^\infty$ as a right continuous process. Each coordinate of the Hilbert cube, with Lebesgue measure, is a uniform random variable.

Target state of jumps

- At time τ_1^-, just before the first jump, the PDMP is in state $\mathbf{x}(\tau_1^-) = \phi_q(\tau_1, \mathbf{x}_0)$.
- The state after the jump has distribution $R(\mathbf{x}(\tau_1^-), \cdot)$.
- We can sample from this distribution using a uniform r.v. via the following inversion result: there is a function $\psi : [0,1] \times \mathcal{D} \to D$, $\psi_{\mathbf{x}}(u) = \psi(u, \mathbf{x})$, such that

$$R(\mathbf{x}, A) = \mu_\ell(\psi_{\mathbf{x}}^{-1}(A)).$$

Simulation of a PDMP

The previous definition of the dynamics of a PDMP provides also a way to simulate trajectories:

1. Find the next jump time τ.
2. Integrate the vector field up to τ.
3. Sample a new state from the reset kernel.

Finding the next jump time

- The time of the next jump is a time-inhomogeneous exponential, with rate $\lambda(q, \phi_q(t, \mathbf{x}))$.
- We can sample from it by solving for t the equation $\int_{t_0}^{t} \lambda(q, \phi_q(s, \mathbf{x}))ds = \Lambda(t) = -\log U$.
- We do this by introducing a new variable X_Λ satisfying the ODE $\frac{dX_\Lambda(t)}{dt} = \lambda(q, \phi_q(s, \mathbf{x}))$.
- We integrate the ODE and stop when $X_\Lambda(t) = -\log U$, using a root finding algorithm.

Transition Driven Stochastic Hybrid Automata

A Transition-Driven Stochastic Hybrid Automaton (TDSHA) is a tuple $\mathcal{T} = (Q, \mathbf{X}, \mathfrak{TC}, \mathfrak{TD}, \mathfrak{TS}, init)$, where:

- Q is a finite set of *control modes*.
- $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a set of real valued *variables*.
- $\mathfrak{TC}$ is the set of *continuous transitions or flows*, which are triples $\eta = (a, q, stoich, rate)$,
- $\mathfrak{TD}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$,
- $\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$,
- *init* is a point giving the initial state of the system.

CONTINUOUS TRANSITIONS

$\mathfrak{TC}$ is the set of *continuous transitions or flows*, which are tuples $\eta = (a, q, stoich, rate)$, where:

- a is the (optional) name of the transition.
- $q \in Q$ the mode in which the transition is active;
- *stoich* is the update or stoichiometric vector, of size $|\mathbf{X}|$
- $rate : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function (usually Lipschitz continuous).

CONTINUOUS TRANSITIONS

$\mathfrak{TC}$ is the set of *continuous transitions or flows*, which are tuples $\eta = (a, q, stoich, rate)$, where:

- a is the (optional) name of the transition.
- $q \in Q$ the mode in which the transition is active;
- *stoich* is the update or stoichiometric vector, of size $|\mathbf{X}|$
- $rate : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function (usually Lipschitz continuous).

Continuous Transitions

$\mathfrak{T}\mathfrak{C}$ is the set of *continuous transitions or flows*, which are tuples $\eta = (a, q, stoich, rate)$, where:

- a is the (optional) name of the transition.
- $q \in Q$ the mode in which the transition is active;
- $stoich$ is the update or stoichiometric vector, of size $|\mathbf{X}|$
- $rate : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function (usually Lipschitz continuous).

CONTINUOUS TRANSITIONS

$\mathfrak{TC}$ is the set of *continuous transitions or flows*, which are tuples $\eta = (a, q, stoich, rate)$, where:

- a is the (optional) name of the transition.
- $q \in Q$ the mode in which the transition is active;
- *stoich* is the update or stoichiometric vector, of size $|\mathbf{X}|$
- $rate : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function (usually Lipschitz continuous).

Continuous Transitions

$\mathfrak{TC}$ is the set of *continuous transitions or flows*, which are tuples $\eta = (a, q, stoich, rate)$, where:

- a is the (optional) name of the transition.
- $q \in Q$ the mode in which the transition is active;
- $stoich$ is the update or stoichiometric vector, of size $|\mathbf{X}|$
- $rate : \mathbb{R}^n \to \mathbb{R}$ is a sufficiently smooth function (usually Lipschitz continuous).

INSTANTANEOUS TRANSITIONS

$\mathfrak{T}\mathfrak{D}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- $priority : \mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

INSTANTANEOUS TRANSITIONS

$\mathfrak{T}\mathfrak{D}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- $priority : \mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

Instantaneous Transitions

$\mathfrak{TD}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- $priority : \mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

Instantaneous Transitions

$\mathfrak{T}\mathfrak{D}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- *priority* : $\mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

Instantaneous Transitions

$\mathfrak{T}\mathfrak{D}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- *priority* : $\mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

Instantaneous Transitions

$\mathfrak{T}\mathfrak{D}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- *priority* $: \mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

Instantaneous Transitions

$\mathfrak{ID}$ is the set of *instantaneous transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, priority)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the closed set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector.
- *priority* : $\mathbb{R}^n \to \mathbb{R}^+$ is a function giving a weight used to solve probabilistically non-determinism among two or more active transitions.

STOCHASTIC TRANSITIONS

$\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

STOCHASTIC TRANSITIONS

$\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

Stochastic Transitions

$\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

STOCHASTIC TRANSITIONS

$\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

STOCHASTIC TRANSITIONS

$\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

STOCHASTIC TRANSITIONS

$\mathfrak{TS}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

Stochastic Transitions

$\mathfrak{T}\mathfrak{S}$ is the set of *stochastic transitions*, which are tuples $\eta = (a, q_1, q_2, guard, reset, rate)$, where:

- a is the (optional) name of the transition.
- q_1 is the exit mode (where the transition is active);
- q_2 is the target mode
- *guard* is a quantifier-free positive first-order formula with free variables among $\mathbf{X}$, representing the set $G_\eta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$
- *reset* is an update of the form $\mathbf{X}' = r(\mathbf{X}, \mathbf{W})$, where $\mathbf{W}$ is a random vector (optional).
- $rate : \mathbb{R}^n \to \mathbb{R}^+$ is the rate function giving the hazard of taking transition η.

From TDSHA to PDMP

A TDSHA $\mathcal{T} = (Q, \mathbf{X}, \mathfrak{TC}, \mathfrak{TD}, \mathfrak{TS}, init)$ can be mapped quite straightforwardly into a PDMP $(Q, D, \mathcal{X}, \lambda, R)$:

- Q is the set of discrete modes and the domain within each mode is given by

$$D_q = \bigcap_{\delta \in \mathfrak{TD}} G_\delta^c,$$

  with $G_\delta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$.

- The vector field in mode q:

$$\mathcal{X}_q(\mathbf{x}) = \sum_{\tau \in \mathfrak{TC} \ \mid\ \mathbf{cmode}[\tau] = q} \mathbf{stoich}[\tau] \cdot \mathbf{rate}[\tau](\mathbf{x}).$$

- The rate function λ:

$$\lambda(q, \mathbf{x}) = \sum_{\eta \in \mathfrak{TS} \ \mid\ \mathbf{e}_1[\eta] = q} \lambda(\eta, q, \mathbf{x}),$$

  where $\lambda(\eta, q, \mathbf{x}) = I_{\mathbf{guard}[\eta](\mathbf{x})} \mathbf{rate}[\eta](\mathbf{x})$.

FROM TDSHA TO PDMP

A TDSHA $\mathcal{T} = (Q, \mathbf{X}, \mathfrak{TC}, \mathfrak{TD}, \mathfrak{TS}, init)$ can be mapped quite straightforwardly into a PDMP $(Q, D, \mathcal{X}, \lambda, R)$:

- Q is the set of discrete modes and the domain within each mode is given by

$$D_q = \bigcap_{\delta \in \mathfrak{TD}} G_\delta^c,$$

  with $G_\delta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$.

- The vector field in mode q:

$$\mathcal{X}_q(\mathbf{x}) = \sum_{\tau \in \mathfrak{TC} \mid \mathbf{cmode}[\tau]=q} \mathbf{stoich}[\tau] \cdot \mathbf{rate}[\tau](\mathbf{x}).$$

- The rate function λ:

$$\lambda(q, \mathbf{x}) = \sum_{\eta \in \mathfrak{TS} \mid \mathbf{e}_1[\eta]=q} \lambda(\eta, q, \mathbf{x}),$$

  where $\lambda(\eta, q, \mathbf{x}) = I_{\mathbf{guard}[\eta](\mathbf{x})}\mathbf{rate}[\eta](\mathbf{x})$.

From TDSHA to PDMP

A TDSHA $\mathcal{T} = (Q, \mathbf{X}, \mathfrak{TC}, \mathfrak{TD}, \mathfrak{TS}, init)$ can be mapped quite straightforwardly into a PDMP $(Q, D, \mathcal{X}, \lambda, R)$:

- Q is the set of discrete modes and the domain within each mode is given by
$$D_q = \bigcap_{\delta \in \mathfrak{TD}} G_\delta^c,$$
with $G_\delta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$.
- The vector field in mode q:
$$\mathcal{X}_q(\mathbf{x}) = \sum_{\tau \in \mathfrak{TC} \mid \mathbf{cmode}[\tau]=q} \mathbf{stoich}[\tau] \cdot \mathbf{rate}[\tau](\mathbf{x}).$$
- The rate function λ:
$$\lambda(q, \mathbf{x}) = \sum_{\eta \in \mathfrak{TS} \mid \mathbf{e}_1[\eta]=q} \lambda(\eta, q, \mathbf{x}),$$
where $\lambda(\eta, q, \mathbf{x}) = I_{\mathbf{guard}[\eta](\mathbf{x})} \mathbf{rate}[\eta](\mathbf{x})$.

FROM TDSHA TO PDMP

A TDSHA $\mathcal{T} = (Q, \mathbf{X}, \mathfrak{TC}, \mathfrak{TD}, \mathfrak{TS}, init)$ can be mapped quite straightforwardly into a PDMP $(Q, D, \mathcal{X}, \lambda, R)$:

- Q is the set of discrete modes and the domain within each mode is given by

$$D_q = \bigcap_{\delta \in \mathfrak{TD}} G_\delta^c,$$

with $G_\delta = \{\mathbf{x} \in \mathbb{R}^n \mid guard[\mathbf{x}]\}$.

- The vector field in mode q:

$$\mathcal{X}_q(\mathbf{x}) = \sum_{\tau \in \mathfrak{TC} \mid \mathbf{cmode}[\tau]=q} \mathbf{stoich}[\tau] \cdot \mathbf{rate}[\tau](\mathbf{x}).$$

- The rate function λ:

$$\lambda(q, \mathbf{x}) = \sum_{\eta \in \mathfrak{TS} \mid \mathbf{e_1}[\eta]=q} \lambda(\eta, q, \mathbf{x}),$$

where $\lambda(\eta, q, \mathbf{x}) = I_{\mathbf{guard}[\eta](\mathbf{x})} \mathbf{rate}[\eta](\mathbf{x})$.

From TDSHA to PDMP

The reset kernel R is defined by:

- $\mathbf{x} \in D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{T}\mathfrak{S}(q,\mathbf{x})} \frac{\lambda(\eta,q,\mathbf{x})}{\lambda(q,\mathbf{x})} \mathbb{P}\{(\mathbf{e}_2[\eta], r_\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathbf{x} \in \partial D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{T}\mathfrak{D}(q,\mathbf{x})} \frac{\mathbf{priority}[\delta]}{\mathbf{priority}[\mathfrak{T}\mathfrak{D}(q,\mathbf{x})]} \mathbb{P}\{(\mathbf{e}_2[\eta], r_\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathfrak{T}\mathfrak{D}(q,\mathbf{x})$ is the set of instantaneous transitions active in $(q,\mathbf{x})$.

Multiple instantaneous jumps

The previous definition works if the TDSHA can do at most one instantaneous jump in a row.
In general, this is not guaranteed, but a definition can be given if there are no infinite loops of instantaneous jumps.
Checking if there are no infinite loops is undecidable.

FROM TDSHA TO PDMP

The reset kernel R is defined by:

- $\mathbf{x} \in D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{TS}(q,\mathbf{x})} \frac{\lambda(\eta,q,\mathbf{x})}{\lambda(q,\mathbf{x})} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathbf{x} \in \partial D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{TD}(q,\mathbf{x})} \frac{\mathbf{priority}[\delta]}{\mathbf{priority}[\mathfrak{TD}(q,\mathbf{x})]} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathfrak{TD}(q,\mathbf{x})$ is the set of instantaneous transitions active in $(q,\mathbf{x})$.

MULTIPLE INSTANTANEOUS JUMPS

The previous definition works if the TDSHA can do at most one instantaneous jump in a row.
In general, this is not guaranteed, but a definition can be given if there are no infinite loops of instantaneous jumps.
Checking if there are no infinite loops is undecidable.

From TDSHA to PDMP

The reset kernel R is defined by:

- $\mathbf{x} \in D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{T}\mathfrak{S}(q,\mathbf{x})} \frac{\lambda(\eta,q,\mathbf{x})}{\lambda(q,\mathbf{x})} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathbf{x} \in \partial D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{T}\mathfrak{D}(q,\mathbf{x})} \frac{\mathbf{priority}[\delta]}{\mathbf{priority}[\mathfrak{T}\mathfrak{D}(q,\mathbf{x})]} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathfrak{T}\mathfrak{D}(q,\mathbf{x})$ is the set of instantaneous transitions active in $(q,\mathbf{x})$.

Multiple instantaneous jumps

The previous definition works if the TDSHA can do at most one instantaneous jump in a row.
In general, this is not guaranteed, but a definition can be given if there are no infinite loops of instantaneous jumps.
Checking if there are no infinite loops is undecidable.

FROM TDSHA TO PDMP

The reset kernel R is defined by:

- $\mathbf{x} \in D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{TS}(q,\mathbf{x})} \frac{\lambda(\eta,q,\mathbf{x})}{\lambda(q,\mathbf{x})} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathbf{x} \in \partial D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{TD}(q,\mathbf{x})} \frac{\textbf{priority}[\delta]}{\textbf{priority}[\mathfrak{TD}(q,\mathbf{x})]} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathfrak{TD}(q,\mathbf{x})$ is the set of instantaneous transitions active in $(q,\mathbf{x})$.

MULTIPLE INSTANTANEOUS JUMPS

The previous definition works if the TDSHA can do at most one instantaneous jump in a row.
In general, this is not guaranteed, but a definition can be given if there are no infinite loops of instantaneous jumps.
Checking if there are no infinite loops is undecidable.

FROM TDSHA TO PDMP

The reset kernel R is defined by:

- $\mathbf{x} \in D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{T}\mathfrak{S}(q,\mathbf{x})} \frac{\lambda(\eta,q,\mathbf{x})}{\lambda(q,\mathbf{x})} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathbf{x} \in \partial D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{T}\mathfrak{D}(q,\mathbf{x})} \frac{\textbf{priority}[\delta]}{\textbf{priority}[\mathfrak{T}\mathfrak{D}(q,\mathbf{x})]} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathfrak{T}\mathfrak{D}(q,\mathbf{x})$ is the set of instantaneous transitions active in $(q,\mathbf{x})$.

MULTIPLE INSTANTANEOUS JUMPS

The previous definition works if the TDSHA can do at most one instantaneous jump in a row.
In general, this is not guaranteed, but a definition can be given if there are no infinite loops of instantaneous jumps.
Checking if there are no infinite loops is undecidable.

From TDSHA to PDMP

The reset kernel R is defined by:

- $\mathbf{x} \in D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{TS}(q,\mathbf{x})} \frac{\lambda(\eta,q,\mathbf{x})}{\lambda(q,\mathbf{x})} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathbf{x} \in \partial D_q$ and $A \in \mathcal{D}$:

$$R((q,\mathbf{x}),A) = \sum_{\eta \in \mathfrak{TD}(q,\mathbf{x})} \frac{\textbf{priority}[\delta]}{\textbf{priority}[\mathfrak{TD}(q,\mathbf{x})]} \mathbb{P}\{(\mathbf{e_2}[\eta], r\eta(\mathbf{x},\mathbf{W})) \in A\}.$$

- $\mathfrak{TD}(q,\mathbf{x})$ is the set of instantaneous transitions active in $(q,\mathbf{x})$.

Multiple instantaneous jumps

The previous definition works if the TDSHA can do at most one instantaneous jump in a row.
In general, this is not guaranteed, but a definition can be given if there are no infinite loops of instantaneous jumps.
Checking if there are no infinite loops is undecidable.

Poisson representation

Consider a TDSHA $\mathcal{T} = (Q, \mathbf{X}, \mathfrak{T}\mathfrak{C}, \mathfrak{T}\mathfrak{D}, \mathfrak{T}\mathfrak{S}, init)$ with no instantaneous transitions ($\mathfrak{T}\mathfrak{D} = \emptyset$), and all resets of stochastic transitions of the constant increment type ($\mathbf{X}' = \mathbf{X} + \mathbf{k}_\eta$). Then, the associated PDMP admits a Poisson representation (using additional variables to encode modes, with null vector field)

$$\mathbf{X}(t) = \mathbf{X}(0) + \int_0^t \mathcal{X}(\mathbf{X}(s))ds + \sum_{\eta \in \mathfrak{T}\mathfrak{S}} \mathbf{k}_\eta \mathcal{Y}_\eta \left(\lambda_\eta(\mathbf{X}(s))ds \right).$$

HYBRID LIMIT

GIVEN A POPULATION CTMC

- Assume it has no guards.
- Partition variables into discrete and continuous, and transitions into continuous and stochastic.
- Construct a TDSHA according to the partition (it admits a Poisson representation).

Under a suitable scaling of rates and updates of transitions, we can prove that the sequence of CTMC models associated with a population CTMC for increasing population size N only of continuous variables, converges to the PDMP associated with it via the TDSHA.

COSTRUCTION OF THE HYBRID LIMIT

Partition variables (discrete/ continuous) and transitions (continuous/ stochastic)

MODES AND VARIABLES

- Discrete state space: all possible values of discrete variables;
- Continuous variables become the continuous variables of SHA.

FLUID TRANSITIONS

They define a set of ODE, as usual.

STOCHASTIC TRANSITIONS WITH VANISHING JUMPS

They become stochastic edges of the SHA, connecting modes so to reflect updates on discrete variables, with the same rate function as the stochastic model, and identity reset.

STOCHASTIC TRANSITIONS WITH DISCONTINUOUS JUMPS

Like the previous edges, but with an arbitrary reset.

COSTRUCTION OF THE HYBRID LIMIT

Partition variables (discrete/ continuous) and transitions (continuous/ stochastic)

MODES AND VARIABLES

- Discrete state space: all possible values of discrete variables;
- Continuous variables become the continuous variables of SHA.

FLUID TRANSITIONS

They define a set of ODE, as usual.

STOCHASTIC TRANSITIONS WITH VANISHING JUMPS

They become stochastic edges of the SHA, connecting modes so to reflect updates on discrete variables, with the same rate function as the stochastic model, and identity reset.

STOCHASTIC TRANSITIONS WITH DISCONTINUOUS JUMPS

Like the previous edges, but with an arbitrary reset.

COSTRUCTION OF THE HYBRID LIMIT

Partition variables (discrete/ continuous) and transitions (continuous/ stochastic)

MODES AND VARIABLES

- Discrete state space: all possible values of discrete variables;
- Continuous variables become the continuous variables of SHA.

FLUID TRANSITIONS

They define a set of ODE, as usual.

STOCHASTIC TRANSITIONS WITH VANISHING JUMPS

They become stochastic edges of the SHA, connecting modes so to reflect updates on discrete variables, with the same rate function as the stochastic model, and identity reset.

STOCHASTIC TRANSITIONS WITH DISCONTINUOUS JUMPS

Like the previous edges, but with an arbitrary reset.

COSTRUCTION OF THE HYBRID LIMIT

Partition variables (discrete/ continuous) and transitions (continuous/ stochastic)

MODES AND VARIABLES

- Discrete state space: all possible values of discrete variables;
- Continuous variables become the continuous variables of SHA.

FLUID TRANSITIONS

They define a set of ODE, as usual.

STOCHASTIC TRANSITIONS WITH VANISHING JUMPS

They become stochastic edges of the SHA, connecting modes so to reflect updates on discrete variables, with the same rate function as the stochastic model, and identity reset.

STOCHASTIC TRANSITIONS WITH DISCONTINUOUS JUMPS

Like the previous edges, but with an arbitrary reset.

SCALING

VARIABLES

Variables are partitioned into two sets:

- Continuous variables, increasing with N;
- Discrete variables, independent of N;

TRANSITIONS CAN OBEY DIFFERENT SCALING LAWS W.R.T. SYSTEM SIZE N:

- Continuous/Fluid Transitions: rate is $\Theta(N)$, increment is $O(N^{-1})$. They act only on continuous variables.
- Discrete/Stochastic Transitions with vanishing jumps on continuous variables: rate is $\Theta(1)$, increment is $O(N^{-1})$ on continuous variables.
- Discrete/Stochastic Transitions with discontinuous jumps on continuous variables: rate is $\Theta(1)$, increment is $\Theta(1)$ on continuous variables.

SCALING

VARIABLES

Variables are partitioned into two sets:

- Continuous variables, increasing with N;
- Discrete variables, independent of N;

TRANSITIONS CAN OBEY DIFFERENT SCALING LAWS W.R.T. SYSTEM SIZE N:

- Continuous/Fluid Transitions: rate is $\Theta(N)$, increment is $O(N^{-1})$. They act only on continuous variables.
- Discrete/Stochastic Transitions with vanishing jumps on continuous variables: rate is $\Theta(1)$, increment is $O(N^{-1})$ on continuous variables.
- Discrete/Stochastic Transitions with discontinuous jumps on continuous variables: rate is $\Theta(1)$, increment is $\Theta(1)$ on continuous variables.

SCALING

VARIABLES

Variables are partitioned into two sets:

- Continuous variables, increasing with N;
- Discrete variables, independent of N;

TRANSITIONS CAN OBEY DIFFERENT SCALING LAWS W.R.T. SYSTEM SIZE N:

- Continuous/Fluid Transitions: rate is $\Theta(N)$, increment is $O(N^{-1})$. They act only on continuous variables.
- Discrete/Stochastic Transitions with vanishing jumps on continuous variables: rate is $\Theta(1)$, increment is $O(N^{-1})$ on continuous variables.
- Discrete/Stochastic Transitions with discontinuous jumps on continuous variables: rate is $\Theta(1)$, increment is $\Theta(1)$ on continuous variables.

SCALING

VARIABLES

Variables are partitioned into two sets:

- Continuous variables, increasing with N;
- Discrete variables, independent of N;

TRANSITIONS CAN OBEY DIFFERENT SCALING LAWS W.R.T. SYSTEM SIZE N:

- Continuous/Fluid Transitions: rate is $\Theta(N)$, increment is $O(N^{-1})$. They act only on continuous variables.
- Discrete/Stochastic Transitions with vanishing jumps on continuous variables: rate is $\Theta(1)$, increment is $O(N^{-1})$ on continuous variables.
- Discrete/Stochastic Transitions with discontinuous jumps on continuous variables: rate is $\Theta(1)$, increment is $\Theta(1)$ on continuous variables.

LIMIT THEOREM

- Let $\mathbf{x}(t)$ be the state of the PDMP at time t, and $\hat{\mathbf{X}}^{(N)}(t)$ be the state of the (normalized) CTMC at time t.
- Assume that $\hat{\mathbf{X}}^{(N)}(0) \to \mathbf{x}(0)$ in probability (a.s.).
- Assume the vector fields of the PDMP to be (locally) Lipschitz continuous.

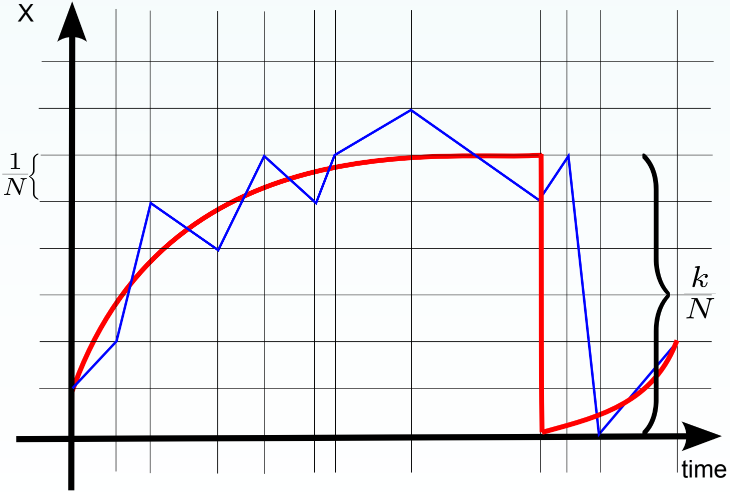
THEOREM (HYBRID LIMIT OF CTMC)

For each $T < \infty$ (in the absence of guards),

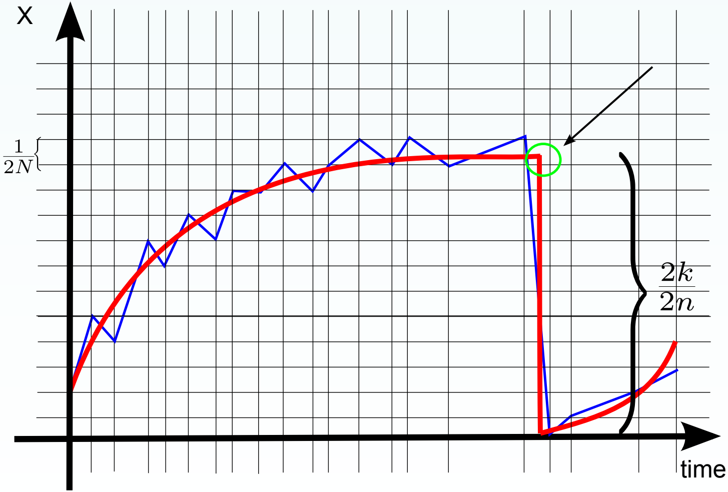
$$\lim_{N\to\infty} \|\hat{\mathbf{X}}^{(N)}(T) - \mathbf{x}(T)\| = 0 \quad \textit{in probability (a.s.).}$$

Then $\hat{\mathbf{X}}^{(N)} \Rightarrow \mathbf{x}$ (weak convergence) in the space of cadlag functions equipped with the Skorokhod metrics.

THE HYBRID CASE — CTMC TO PDMP — INTUITION



THE HYBRID CASE — CTMC TO PDMP — INTUITION



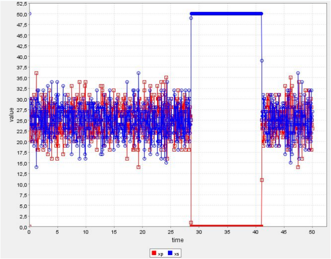
THE HYBRID CASE — CTMC TO PDMP

THEOREM (HYBRID LIMIT OF CTMC)

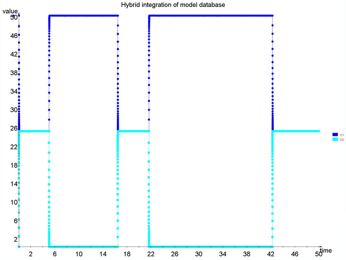
For each $T < \infty$ (in the absence of guards),

$$\lim_{N\to\infty} \|\hat{\mathbf{X}}^{(N)}(T) - \mathbf{x}(T)\| = 0 \text{ in probability (a.s.).}$$

Then $\hat{\mathbf{X}}^{(N)} \Rightarrow \mathbf{x}$ (weak convergence) in the space of cadlag functions equipped with the Skorokhod metrics.



$n = 1$



Fluid approximation

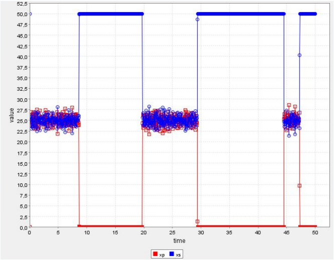
THE HYBRID CASE — CTMC TO PDMP

THEOREM (HYBRID LIMIT OF CTMC)

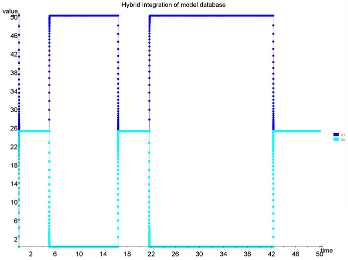
For each $T < \infty$ (in the absence of guards),

$$\lim_{N\to\infty} \|\hat{\mathbf{X}}^{(N)}(T) - \mathbf{x}(T)\| = 0 \quad \textit{in probability (a.s.).}$$

Then $\hat{\mathbf{X}}^{(N)} \Rightarrow \mathbf{x}$ (weak convergence) in the space of cadlag functions equipped with the Skorokhod metrics.



$n = 10$



Fluid approximation

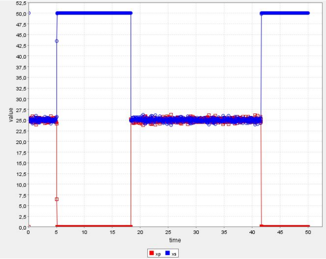
THE HYBRID CASE — CTMC TO PDMP

THEOREM (HYBRID LIMIT OF CTMC)

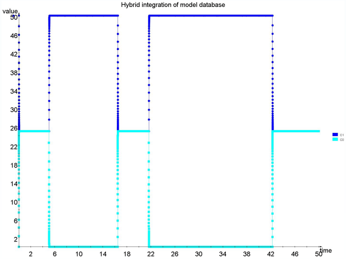
For each $T < \infty$ (in the absence of guards),

$$\lim_{N\to\infty} \|\hat{\mathbf{X}}^{(N)}(T) - \mathbf{x}(T)\| = 0 \quad \textit{in probability (a.s.).}$$

Then $\hat{\mathbf{X}}^{(N)} \Rightarrow \mathbf{x}$ (weak convergence) in the space of cadlag functions equipped with the Skorokhod metrics.



$n = 10$



Fluid approximation

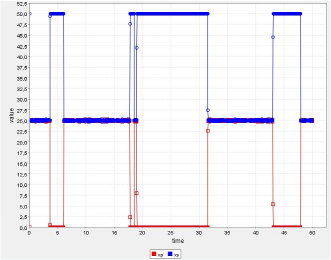
THE HYBRID CASE — CTMC TO PDMP

THEOREM (HYBRID LIMIT OF CTMC)

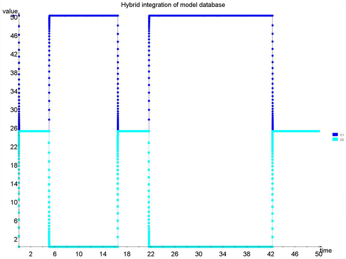
For each $T < \infty$ (in the absence of guards),

$$\lim_{N\to\infty} \|\hat{\mathbf{X}}^{(N)}(T) - \mathbf{x}(T)\| = 0 \quad \text{in probability (a.s.).}$$

Then $\hat{\mathbf{X}}^{(N)} \Rightarrow \mathbf{x}$ (weak convergence) in the space of cadlag functions equipped with the Skorokhod metrics.



$n = 1000$



Fluid approximation

EXTENDING THE FRAMEWORK

INSTANTANEOUS TRANSITIONS

- We can consider instantaneous transitions in **sCCP**, becoming forced transitions in the limit PDMP.
- We need to enforce restrictions to avoid infinite loops: we can allow only a finite number of finite sequences of instantaneous transitions.
- One needs some care on the interaction between the flow of the vector field and the boundary: we want to (almost surely) avoid tangential activation of guards.
- Weak convergence still holds, if we assume that the PDMP is non-Zeno.

Extending the framework

Random Resets

- We can consider random reset functions for instantaneous transitions and for transitions being discrete and stochastic in the limit PDMP.
- We have to properly interface it with the state space for finite N, for instance requiring that $r^{(N)}(\mathbf{X}, \mathbf{W}) \rightarrow r(\mathbf{X}, \mathbf{W})$ in probability, uniformly on $\mathbf{X}$.
- We can also allow random resets for continuous transitions, with a proper scaling of variance (conditions on increments of the normalized CTMC sequence restricted to continuous transitions), e.g. finite first and second order moments.
- Weak convergence still holds, if we assume that the PDMP is non-Zeno.

Dealing with guards

Guards on continuous transitions

- Guards on fluid transitions cause discontinuities in the vector fields (typically in computer science models).
- In this case, we obtain a piecewise-smooth dynamical system (in each mode), which is still the limit of the sequence of CTMC (direct proof working with piecewise systems or via differential inclusions).

Guards on stochastic transitions

We can allow also guarded stochastic transitions, but some care has to be taken in how the vector field of the limit PDMP interacts with them (no sliding on guard surfaces with probability one).

FROM sCCP TO PDMP TO ODE

SIZE-DEPENDENT SEQUENCE OF PDMP

Consider a sequence of PDMP with discrete variables increasing with N, and discrete transitions satisfying the continuous scaling. Call $\mathbf{x}^{(N)}(t)$ the sequence of PDMP and $\mathbf{x}(t)$ the solution of the fluid approximation of the original **sCCP**program.

THEOREM (CONVERGENCE OF PDMP TO FLUID ODE)

If stochastic rates are locally Lipschitz, then for each $T > 0$,

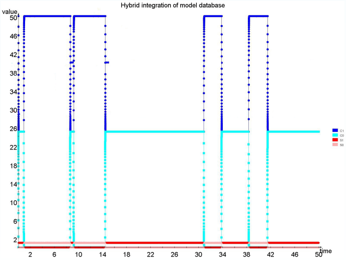
$$\sup_{t \leq T} \|\mathbf{x}^{(N)}(t) - \mathbf{x}(t)\| \rightarrow 0 \quad \textit{in probability}$$

The hybrid case — PDMP to ODE

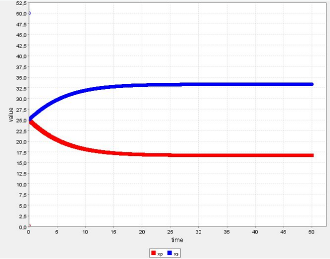
Theorem (Convergence of PDMP to fluid ODE)

If stochastic rates are locally Lipschitz, then for each $T > 0$,

$$\sup_{t \le T} \|\mathbf{x}^{(N)}(t) - \mathbf{x}(t)\| \to 0 \quad \textit{in probability}$$



$n = 1$



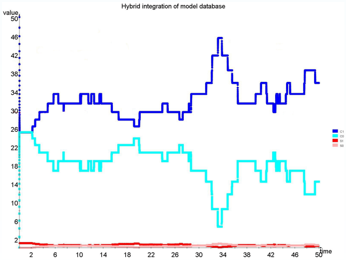
Fluid approximation

The hybrid case — PDMP to ODE

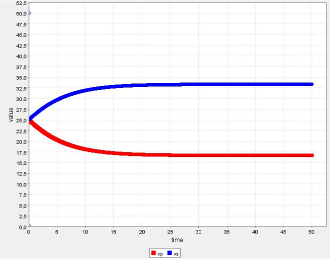
Theorem (Convergence of PDMP to fluid ODE)

If stochastic rates are locally Lipschitz, then for each $T > 0$,

$$\sup_{t \leq T} \|\mathbf{x}^{(N)}(t) - \mathbf{x}(t)\| \to 0 \quad \textit{in probability}$$



$n = 10$



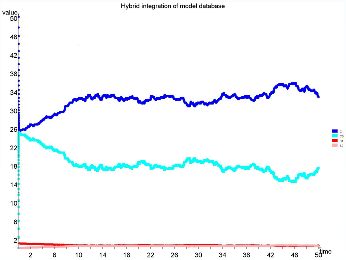
Fluid approximation

THE HYBRID CASE — PDMP TO ODE

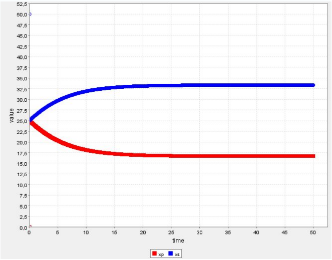
THEOREM (CONVERGENCE OF PDMP TO FLUID ODE)

If stochastic rates are locally Lipschitz, then for each $T > 0$,

$$\sup_{t \leq T} \|\mathbf{x}^{(N)}(t) - \mathbf{x}(t)\| \to 0 \quad \textit{in probability}$$



$n = 10$



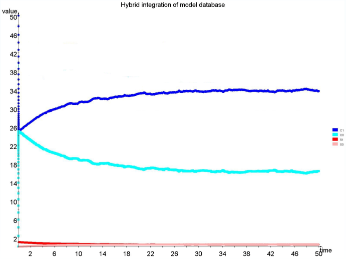
Fluid approximation

The hybrid case — PDMP to ODE

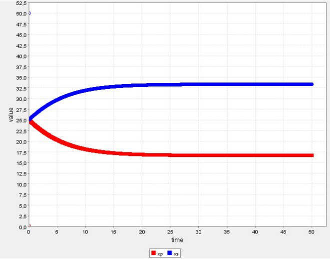
Theorem (Convergence of PDMP to fluid ODE)

If stochastic rates are locally Lipschitz, then for each $T > 0$,

$$\sup_{t \le T} \|\mathbf{x}^{(N)}(t) - \mathbf{x}(t)\| \to 0 \quad \textit{in probability}$$

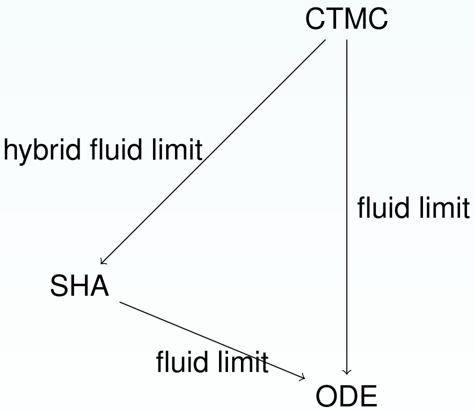


$n = 1000$



Fluid approximation

SUMMARIZING: RELATION BETWEEN sCCP SEMANTICS



Outline

1. Hybrid Mean Field
2. Mean Field for Piecewise Smooth Systems

EXAMPLE: QUEUE NETWORK

THE SCENARIO

- Network of N servers, with buffer of size 1.
- Client arrival rate: $N\lambda$.
- Client servicing rate: μ.
- Servicing policy: clients are redirected to the server with the shortest queue.

Example: queue network

Variables

- X_0: idle server, empty buffer
- X_1: busy server, empty buffer
- X_2: busy server, full buffer

Transitions

- Arrival to idle server $(X_0 > 0, (-1, 1, 0), N\lambda)$
- Arrival to busy server $(X_0 = 0 \wedge X_1 > 0, (0, -1, 1), N\lambda)$
- Servicing by server with empty buffer $(\top, (1, -1, 0), \mu X_1)$
- Servicing by server with full buffer $(\top, (0, 1, -1), \mu X_2)$

EXAMPLE: QUEUE NETWORK

FLUID ODE

The sequence of model (for increasing N) satisfies scaling assumptions. Normalized models live in $E = \{\mathbf{x} \in [0,1]^3 \mid x_0 + x_1 + x_2 = 1\}$.

$$\begin{cases} \dot{x}_0 = \mu x_1 - \lambda I_{x_0>0} \\ \dot{x}_1 = \mu x_2 - \mu_x 1 + \lambda I_{x_0>0} - \lambda I_{x_0=0 \wedge x_1>0} \\ \dot{x}_2 = \lambda I_{x_0=0 \wedge x_1>0} - \mu x_2 \end{cases}$$

EXAMPLE: QUEUE NETWORK

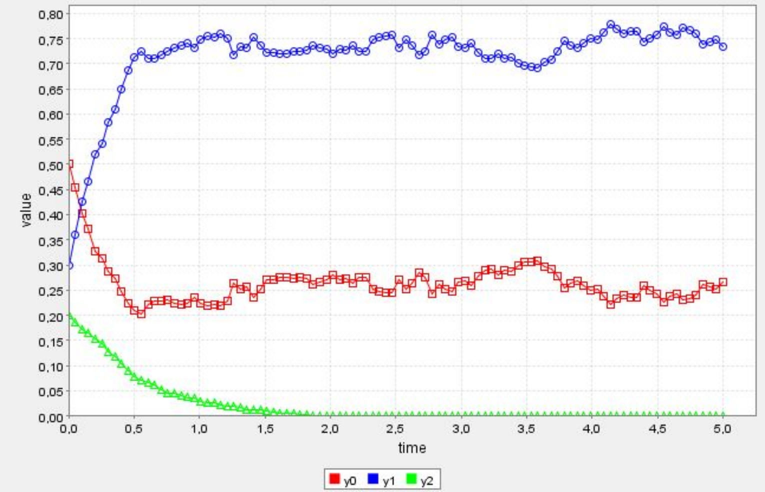
WARNING

The fluid vector field is discontinuous (on the plane $x_0 = 0$), due to the presence of guards!
The theorem cannot be applied in E!
But it can be applied in a proper subset of E, not containing the plane $x_0 = 0$.

QUEUE NETWORK: $\lambda = 1.5$, $\mu = 2$, $\mathbf{x_0} = (0.5, 0.3, 0.2)$

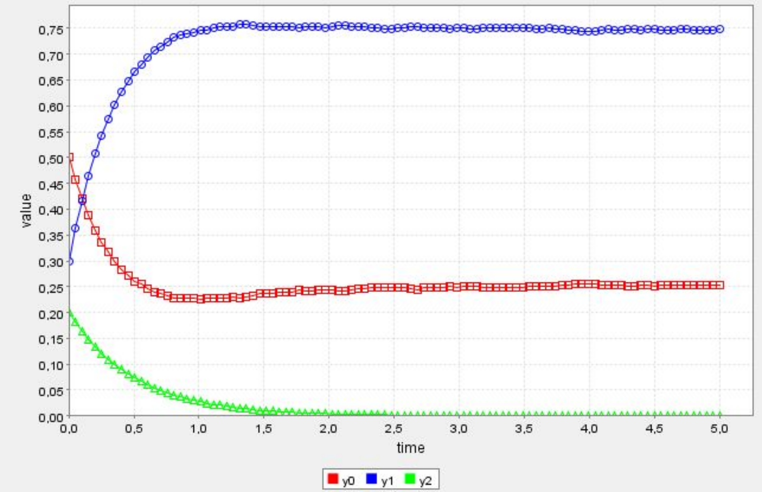
CTMC $N = 100$

QUEUE NETWORK: $\lambda = 1.5$, $\mu = 2$, $\mathbf{x_0} = (0.5, 0.3, 0.2)$



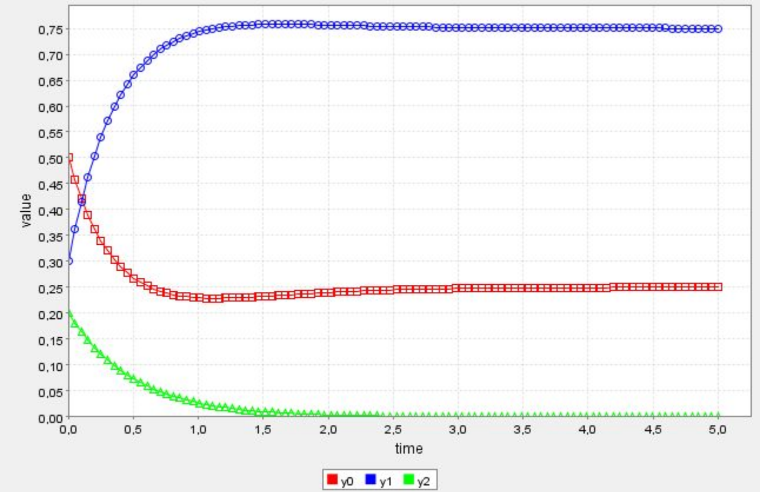
CTMC $N = 1000$

QUEUE NETWORK: $\lambda = 1.5, \mu = 2, \mathbf{x_0} = (0.5, 0.3, 0.2)$



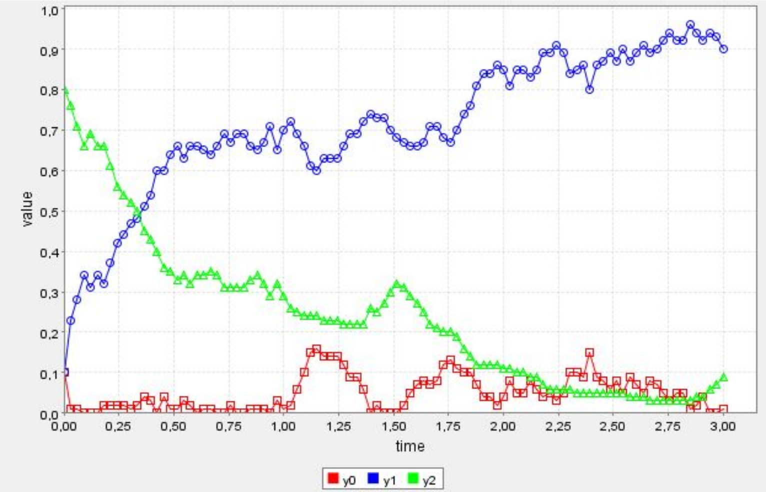
CTMC $N = 100000$

QUEUE NETWORK: $\lambda = 1.5$, $\mu = 2$, $\mathbf{x_0} = (0.5, 0.3, 0.2)$



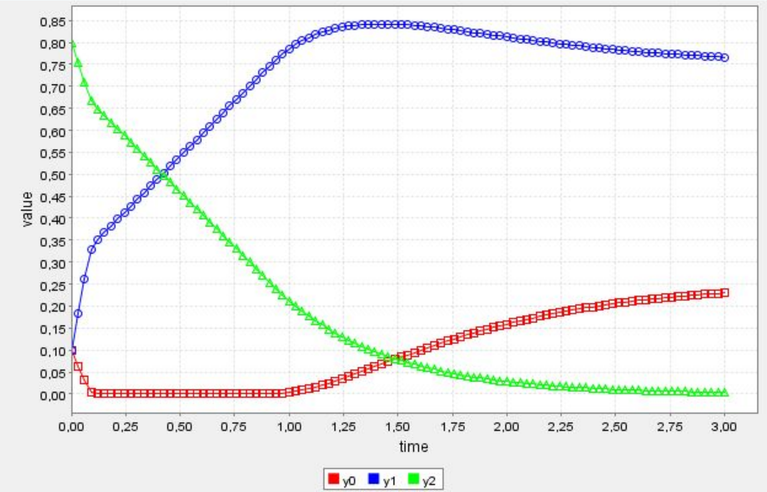
Limit ODE: it works!

QUEUE NETWORK: $\lambda = 1.5, \mu = 2, \mathbf{x_0} = (0.1, 0.1, 0.8)$



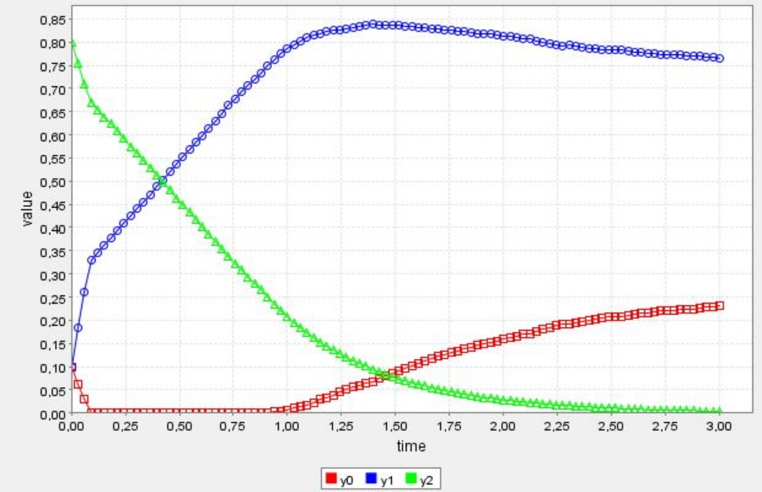
CTMC $N = 100$

QUEUE NETWORK: $\lambda = 1.5, \mu = 2, \mathbf{x_0} = (0.1, 0.1, 0.8)$



Limit ODE: THEOREM CANNOT BE APPLIED!

QUEUE NETWORK: $\lambda = 1.5, \mu = 2, \mathbf{x_0} = (0.1, 0.1, 0.8)$



CTMC $N = 100000$

Piecewise smooth dynamical systems

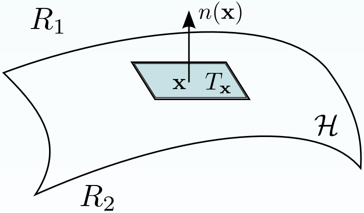
What is the solution of an ODE with discontinuous vector field?

- Within regions of continuity: behaves like standard ODE.
- What happens if a trajectory hits a discontinuity boundary?
- Concept of Filippov solution.

Hypothesis

- Surface of discontinuity between two continuous regions is defined as the zero set of a smooth function $h : E \to \mathbb{R}$: $\mathcal{H} = \{\mathbf{x} \mid h(\mathbf{x}) = 0\}$.
- Continuity regions: $\mathcal{R}_1 = \{\mathbf{x} \mid h(\mathbf{x}) > 0\}$, with vector field f_1, $\mathcal{R}_2 = \{\mathbf{x} \mid h(\mathbf{x}) < 0\}$, with vector field f_2.
- In our setting: surfaces of discontinuity are determined by non-trivial guards.

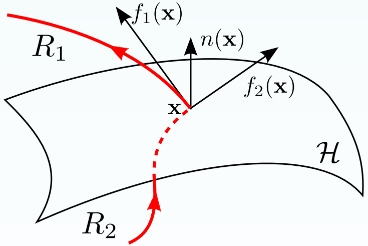
PIECEWISE SMOOTH DYNAMICAL SYSTEMS



NORMAL VECTOR

Normal vector to $\mathcal{H}$: $n(\mathbf{x}) = \nabla h(\mathbf{x}) / \|\nabla h(\mathbf{x})\|$

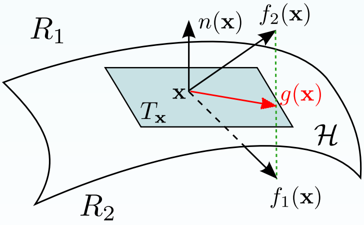
PIECEWISE SMOOTH DYNAMICAL SYSTEMS



TRANSVERSAL CROSSING

Both $n(\mathbf{x})^T f_1(\mathbf{x})$ and $n(\mathbf{x})^T f_2(\mathbf{x})$ are non-zero and have the same sign (If only one between $n(\mathbf{x})^T f_1(\mathbf{x})$ and $n(\mathbf{x})^T f_2(\mathbf{x})$ is zero, we have tangential crossing).

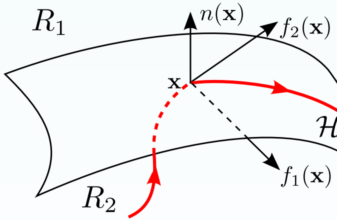
PIECEWISE SMOOTH DYNAMICAL SYSTEMS



SLIDING MOTION

If $n(\mathbf{x})^T f_1(\mathbf{x}) < 0$ and $n(\mathbf{x})^T f_2(\mathbf{x}) > 0$, the system cannot escape $\mathcal{H}$. The solution follows a vector field obtained as convex combination of f_1 and f_2.

PIECEWISE SMOOTH DYNAMICAL SYSTEMS



SLIDING MOTION

If $n(\mathbf{x})^T f_1(\mathbf{x}) > 0$ and $n(\mathbf{x})^T f_2(\mathbf{x}) < 0$, the sliding motion is stable. If signs are inverted, the sliding motion in unstable.

Piecewise smooth dynamical systems

Existence and uniqueness

Solution of piecewise smooth systems exists and is unique in a point $\mathbf{x} \in \mathcal{H}$ iff:

- At least one between $n(\mathbf{x})^T f_1(\mathbf{x})$ and $n(\mathbf{x})^T f_2(\mathbf{x})$ is non-zero.
- Either $n(\mathbf{x})^T f_1(\mathbf{x})$ and $n(\mathbf{x})^T f_2(\mathbf{x})$ have the same sign, or $n(\mathbf{x})^T f_1(\mathbf{x}) \leq 0$ and $n(\mathbf{x})^T f_2(\mathbf{x}) \geq 0$.

CONVERGENCE OF CTMC TO PWS SOLUTIONS

THEOREM

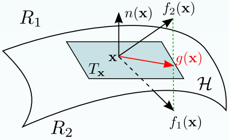
The fluid approximation theorem extends to PWS if the PWS solution $\mathbf{x}(t)$ starting in $\mathbf{x_0}$, for $t \leq T < \infty$, satisfies the following conditions:

1. $\mathbf{x}(t)$ is the *unique* solution in $[0, T]$ starting from $\mathbf{x_0}$;
2. if $\mathbf{x}(t)$, for $t \in [T_1, T_2]$, undergoes a sliding motion, then this motion happens on the discontinuity surface of a single guard.
3. the number of times $\mathbf{x}(t)$ undergoes transversal crossing in $[0, T]$ and the number of traits of sliding motion of $\mathbf{x}(t)$ in $[0, T]$ are *finite*, i.e. we exclude Zeno trajectories.

PROOF SKETCH

- Split $\mathbf{x}(t)$ in the intervals $[0, T_1], [T_1, T_2], \ldots, [T_n, T]$ in which either $\mathbf{x}(t)$ is within a continuity region or it undergoes transversal motion.
- In continuity regions, fluid limit theorem holds.
- Prove a convergence theorem for sliding motion.
- Glue convergence proofs in $[T_i, T_{i+1}]$ using convergence of exit times of CTMC to exit times of ODE.

PROOF SKETCH

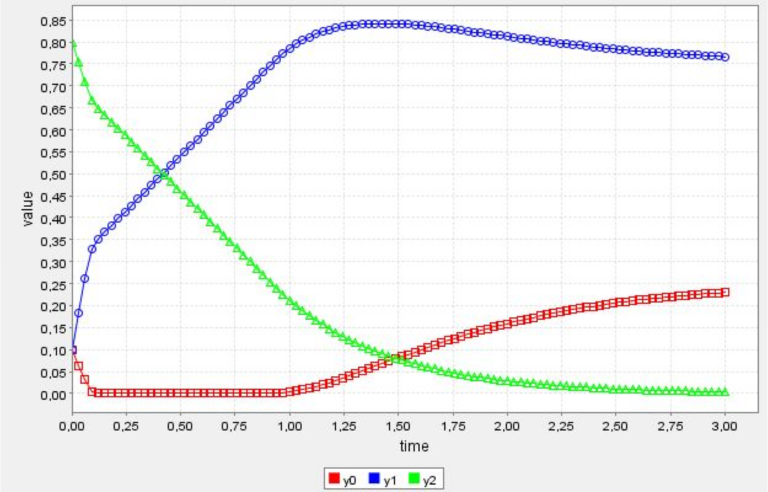


SLIDING MOTION

Proving convergence for sliding motion is the difficult part.
Intuition:

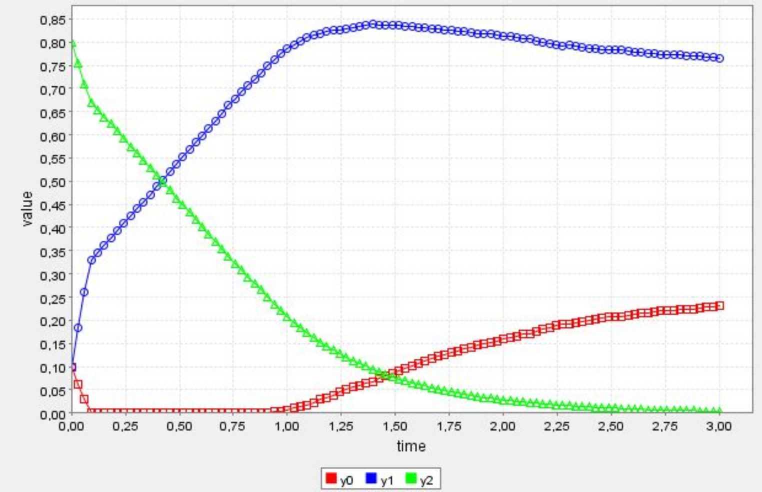
- when $\mathbf{X}^{(N)}(t)$ is in $\mathcal{R}_1$ near $\mathcal{H}$, it is pushed in $\mathcal{R}_2$, and viceversa: $\mathbf{X}^{(N)}(t)$ cannot escape from $\mathcal{H}$.
- the probability of $\mathbf{X}^{(N)}(t)$ being in $\mathcal{R}_1$ converges to the weight of f_1 in the convex combination defining the Filippov vector field in $\mathcal{H}$.

QUEUE NETWORK: SLIDING MOTION



Limit ODE: we have a (good) sliding solution.

QUEUE NETWORK: SLIDING MOTION



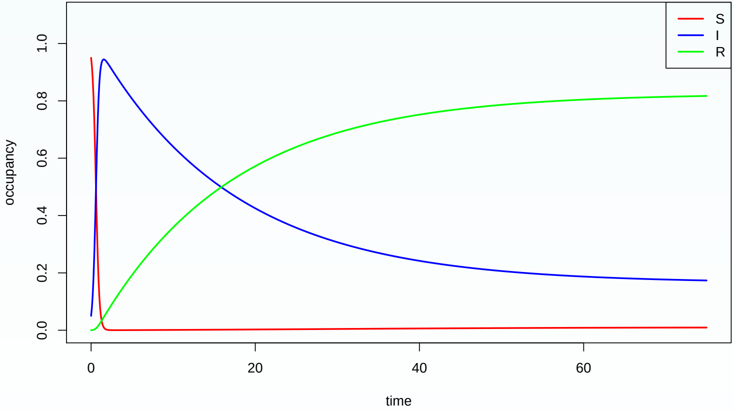
CTMC $N = 100000$

Example: controlling epidemics outspread

Controlling policy

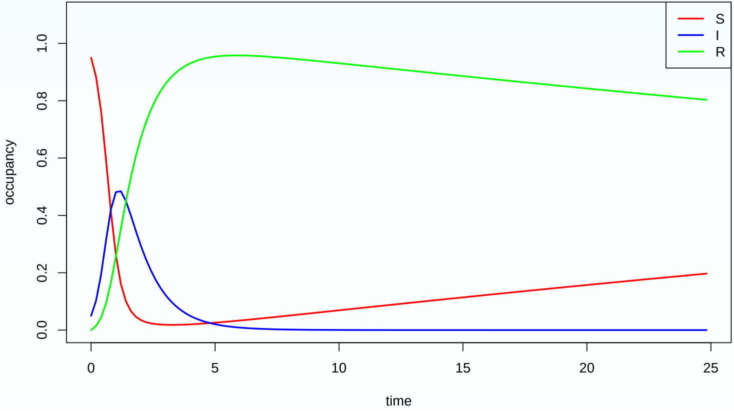
When the fraction of infected nodes is bigger than a given threshold, the patching rate (recovery rate) is increased.

EXAMPLE: CONTROLLING EPIDEMICS OUTSPREAD



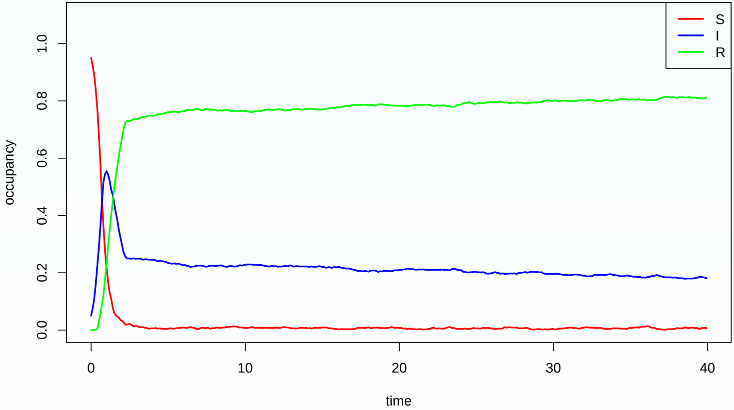
ODE, low recovery rate.

EXAMPLE: CONTROLLING EPIDEMICS OUTSPREAD



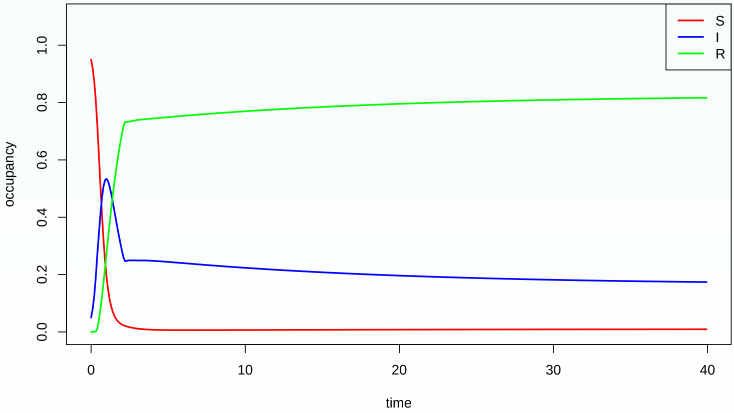
ODE, high recovery rate.

EXAMPLE: CONTROLLING EPIDEMICS OUTSPREAD



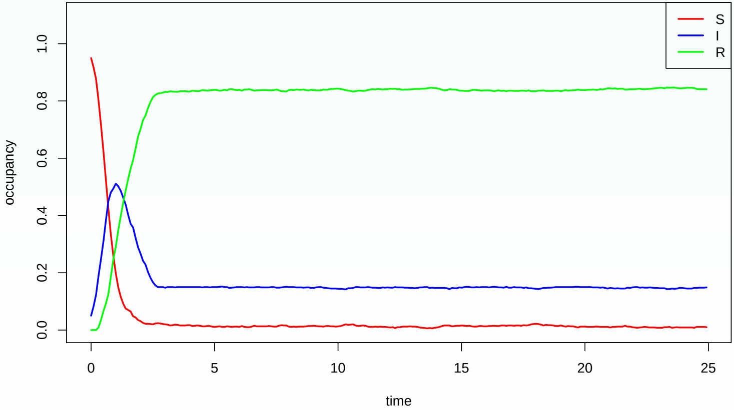
CTMC $N = 1000$, switch threshold 0.25

EXAMPLE: CONTROLLING EPIDEMICS OUTSPREAD



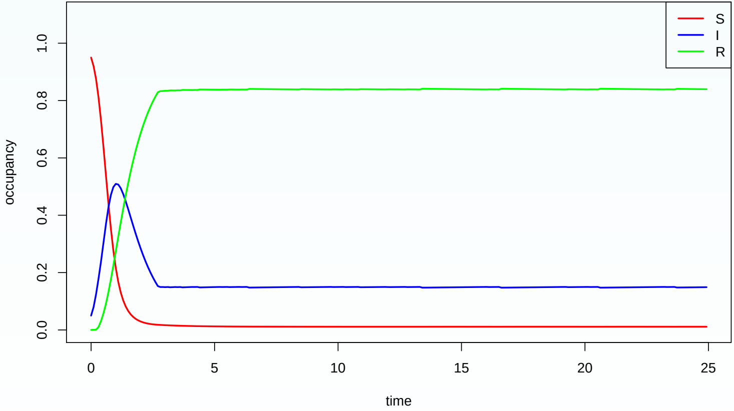
Limit ODE: switch threshold 0.25.

EXAMPLE: CONTROLLING EPIDEMICS OUTSPREAD



CTMC $N = 1000$, switch threshold 0.15

EXAMPLE: CONTROLLING EPIDEMICS OUTSPREAD



Limit ODE: switch threshold 0.15.

REFERENCES

- L. Bortolussi. Hybrid behaviour of Markov population models. Submitted to Information and Computation.
- L. Bortolussi, Hybrid limits of continuous time markov chains. in Proc. of 8th Int. Conf. on the Quantitative Evaluation of Systems, QEST 2011.
- N. Gast and B. Gaujal, Mean field limit of non-smooth systems and differential inclusions, SIGMETRICS Perform. Eval. Rev., vol. 38, pp. 30-32, October 2010.
- L. Bortolussi, J. Hillston, D. Latella, M. Massink. Continuous Approximation of Collective Systems Behaviour: a Tutorial. Submitted to Performance Evaluation.
- L. Bortolussi and A. Policriti. Programmable models of growth and mutation of cancer-cell populations, CompMod 2011.