PDMPs with ODE Dynamics

Sam Power
(*joint with Sergio Bacallado*)

Cambridge Centre for Analysis
Cantab Capital Institute for the Mathematics of Information

sp825@cam.ac.uk

November 15, 2018

Overview

PDMPs

- Informally: Deterministic dynamics + Jump Process

PDMPs

- Informally: Deterministic dynamics + Jump Process
- Stochastic process Z_t which

PDMPs

- Informally: Deterministic dynamics + Jump Process
- Stochastic process Z_t which
  1. Follows a *deterministic* path, until

PDMPs

- Informally: Deterministic dynamics + Jump Process
- Stochastic process Z_t which
  1. Follows a *deterministic* path, until
  2. An *event* occurs, at a certain rate, upon which

PDMPs

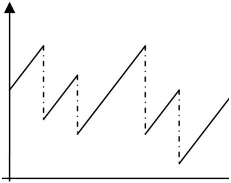
- Informally: Deterministic dynamics + Jump Process
- Stochastic process Z_t which
  1. Follows a *deterministic* path, until
  2. An *event* occurs, at a certain rate, upon which
  3. The position *jumps*, and then

PDMPs

- Informally: Deterministic dynamics + Jump Process
- Stochastic process Z_t which
  1. Follows a *deterministic* path, until
  2. An *event* occurs, at a certain rate, upon which
  3. The position *jumps*, and then
  4. Resumes following the deterministic path

PDMPs

- Informally: Deterministic dynamics + Jump Process
- Stochastic process Z_t which
  1. Follows a *deterministic* path, until
  2. An *event* occurs, at a certain rate, upon which
  3. The position *jumps*, and then
  4. Resumes following the deterministic path



Specifying a PDMP

- Today: PDMPs from ODEs

Specifying a PDMP

- Today: PDMPs from ODEs
  - Vector field $\phi(z)$
    - Use dynamics $\frac{dz}{dt} = \phi(z)$

Specifying a PDMP

- Today: PDMPs from ODEs
  - Vector field $\phi(z)$
    - Use dynamics $\frac{dz}{dt} = \phi(z)$
  - Event rate $\lambda(z) \geqslant 0$
    - Dictates how often events happen (inhomogeneous Poisson process)

Specifying a PDMP

- Today: PDMPs from ODEs
  - Vector field $\phi(z)$
    - Use dynamics $\frac{dz}{dt} = \phi(z)$
  - Event rate $\lambda(z) \geqslant 0$
    - Dictates how often events happen (inhomogeneous Poisson process)
  - Transition dynamics $Q(z \to dz')$
    - Dictates what happens at events (Markov jump kernel)

PDMPs for MCMC

- Want $\pi(dx)$, but work on extended target:

PDMPs for MCMC

- Want $\pi(dx)$, but work on extended target:
  - Set $z = (x, v)$.

PDMPs for MCMC

- Want $\pi(dx)$, but work on extended target:
  - Set $z = (x, v)$.
  - Choose your own $\psi(dv)$.

PDMPs for MCMC

- Want $\pi(dx)$, but work on extended target:
  - Set $z = (x, v)$.
  - Choose your own $\psi(dv)$.
  - Target is then $\mu(dz) = \pi(dx)\psi(dv)$.

PDMPs for MCMC

- Want $\pi(dx)$, but work on extended target:
  - Set $z = (x, v)$.
  - Choose your own $\psi(dv)$.
  - Target is then $\mu(dz) = \pi(dx)\psi(dv)$.
- Typically, jumps fix $x \rightsquigarrow X_t$ has continuous sample paths.

PDMPs for MCMC

- Want $\pi(dx)$, but work on extended target:
  - Set $z = (x, v)$.
  - Choose your own $\psi(dv)$.
  - Target is then $\mu(dz) = \pi(dx)\psi(dv)$.
- Typically, jumps fix $x \rightsquigarrow X_t$ has continuous sample paths.
- **Question**:

$$\text{Given target measure } \mu, \text{ vector field } \phi, \tag{1}$$
$$\text{how can I build } (\lambda, Q) \text{ to sample } \mu\,? \tag{2}$$

Aside on Reversibility, Symmetry

- Reversibility
  - Much MCMC work built on reversible methods
  - PDMPs are generally non-reversible
  - To design algorithms, *locality* is the important part

Aside on Reversibility, Symmetry

- Reversibility
  - Much MCMC work built on reversible methods
  - PDMPs are generally non-reversible
  - To design algorithms, *locality* is the important part
- Symmetry
  - Existing PDMPs are highly symmetric (BPS, ZZ)
  - A priori, not necessary to have symmetry
  - Want to be able to use *all* ODEs!

Time-Augmented PDMPs

- Idea:

Time-Augmented PDMPs

- Idea:
  1. Introduce ‘direction of time’ variable $\tau \in \{\pm 1\}$

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.
- Write $\phi(z, \tau) = \tau \cdot \phi(z)$; use dynamics $\frac{dz}{dt} = \phi(z, \tau)$

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.
- Write $\phi(z, \tau) = \tau \cdot \phi(z)$; use dynamics $\frac{dz}{dt} = \phi(z, \tau)$
  - Solve system forwards **and** backwards in time

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.
- Write $\phi(z, \tau) = \tau \cdot \phi(z)$; use dynamics $\frac{dz}{dt} = \phi(z, \tau)$
  - Solve system forwards **and** backwards in time
- Let $\lambda = \lambda(z, \tau)$

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.
- Write $\phi(z, \tau) = \tau \cdot \phi(z)$; use dynamics $\frac{dz}{dt} = \phi(z, \tau)$
  - Solve system forwards **and** backwards in time
- Let $\lambda = \lambda(z, \tau)$
- Stipulate that, at events, $\tau \mapsto -\tau$, i.e.

$$Q((z, \tau) \to (dz', d\tau')) = Q^{\tau}(z \to dz') \cdot \delta(-\tau, d\tau') \tag{3}$$

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.
- Write $\phi(z, \tau) = \tau \cdot \phi(z)$; use dynamics $\frac{dz}{dt} = \phi(z, \tau)$
  - Solve system forwards **and** backwards in time
- Let $\lambda = \lambda(z, \tau)$
- Stipulate that, at events, $\tau \mapsto -\tau$, i.e.

$$Q((z, \tau) \to (dz', d\tau')) = Q^{\tau}(z \to dz') \cdot \delta(-\tau, d\tau') \tag{3}$$

- 'Trajectorial Reversibility' $\rightsquigarrow$ checking exactness becomes *local*!

Time-Augmented PDMPs

- Idea:
  1. Introduce 'direction of time' variable $\tau \in \{\pm 1\}$
  2. Target $\tilde{\mu}(dz, d\tau) = \mu(dz) R(d\tau)$.
- Write $\phi(z, \tau) = \tau \cdot \phi(z)$; use dynamics $\frac{dz}{dt} = \phi(z, \tau)$
  - Solve system forwards **and** backwards in time
- Let $\lambda = \lambda(z, \tau)$
- Stipulate that, at events, $\tau \mapsto -\tau$, i.e.

$$Q((z, \tau) \to (dz', d\tau')) = Q^{\tau}(z \to dz') \cdot \delta(-\tau, d\tau') \tag{3}$$

- 'Trajectorial Reversibility' $\rightsquigarrow$ checking exactness becomes *local*!
  - 'in at z forwards in time = out at z backwards in time'

Choice of Event Rate (1)

- Consider 'probability current'

$$r(z, \tau) \triangleq \underbrace{\langle \nabla H(z), \phi(z, \tau) \rangle}_{\text{Energy Gain}} - \underbrace{\text{div}_z \phi(z, \tau)}_{\text{Compressibility Penalty}} \tag{4}$$

Choice of Event Rate (1)

- Consider 'probability current'

$$r(z,\tau) \triangleq \underbrace{\langle \nabla H(z), \phi(z,\tau) \rangle}_{\text{Energy Gain}} \quad - \quad \underbrace{\mathrm{div}_z \phi(z,\tau)}_{\text{Compressibility Penalty}} \tag{4}$$

- Define 'natural' event rate as

$$\lambda^0(z,\tau) = (r(z,\tau))_+ \tag{5}$$

where $(u)_+ = \max(0,u)$

Choice of Event Rate (1)

- Consider 'probability current'

$$r(z,\tau) \triangleq \underbrace{\langle \nabla H(z), \phi(z,\tau) \rangle}_{\text{Energy Gain}} \quad - \quad \underbrace{\text{div}_z \phi(z,\tau)}_{\text{Compressibility Penalty}} \tag{4}$$

- Define 'natural' event rate as

$$\lambda^0(z,\tau) = (r(z,\tau))_+ \tag{5}$$

where $(u)_+ = \max(0,u)$

- Let $\gamma(z) \geqslant 0$ be some 'refreshment rate'.

Choice of Event Rate (1)

- Consider ‘probability current’

$$r(z,\tau) \triangleq \underbrace{\langle \nabla H(z), \phi(z,\tau)\rangle}_{\text{Energy Gain}} \quad - \quad \underbrace{\mathrm{div}_z \phi(z,\tau)}_{\text{Compressibility Penalty}} \tag{4}$$

- Define ‘natural’ event rate as

$$\lambda^0(z,\tau) = (r(z,\tau))_+ \tag{5}$$

where $(u)_+ = \max(0,u)$

- Let $\gamma(z) \geqslant 0$ be some ‘refreshment rate’.
- We will take $\lambda(z,\tau) = \lambda^0(z,\tau) + \gamma(z)$

Choice of Transition Dynamics

- Define ‘jump measure’:

$$J^{\tau}(dz) \propto \mu(dz)\lambda(z, \tau) \tag{6}$$

Choice of Transition Dynamics

- Define 'jump measure':

$$J^{\tau}(dz) \propto \mu(dz)\lambda(z,\tau) \tag{6}$$

- Want trajectorial reversibility

Choice of Transition Dynamics

- Define 'jump measure':

$$J^{\tau}(dz) \propto \mu(dz)\lambda(z, \tau) \tag{6}$$

- Want trajectorial reversibility
  - $\Longrightarrow$ Need jump chain reversible w.r.t. jump measure

Choice of Transition Dynamics

- Define 'jump measure':

$$J^{\tau}(dz) \propto \mu(dz)\lambda(z,\tau) \tag{6}$$

- Want trajectorial reversibility
  - $\implies$ Need jump chain reversible w.r.t. jump measure
  - $\rightsquigarrow$ Choose $q^{\tau}(z \to dz')$ to be J^{τ}-reversible

Putting together the ingredients

Theorem

If (ϕ, λ, Q) are chosen in this way, then the resulting PDMP is trajectorially reversible, and admits $\tilde{\mu}$ as a stationary measure.

Putting together the ingredients

Theorem

If (ϕ, λ, Q) *are chosen in this way, then the resulting PDMP is trajectorially reversible, and admits* $\tilde{\mu}$ *as a stationary measure.*

Theorem

If (ϕ, λ, Q) *is a trajectorially-reversible,* $\tilde{\mu}$*-stationary TA-PDMP, then* $\exists\, \gamma \geqslant 0$ *such that*

$$\lambda(z, \tau) = \lambda^0(z, \tau) + \gamma(z) \tag{7}$$

and for $\tau \in \{\pm 1\}$*,* Q^τ *is* J^τ*-reversible*

Split PDMPs (1)

- Many PDMPs in use have different types of event

Split PDMPs (1)

- Many PDMPs in use have different types of event
  - Refreshment
  - Zig-Zag
  - Local BPS (Factor Graph)
  - Subsampling
  - $\cdots$

Split PDMPs (1)

- Many PDMPs in use have different types of event
  - Refreshment
  - Zig-Zag
  - Local BPS (Factor Graph)
  - Subsampling
  - $\cdots$
- Each event type affects different parts of the system

Split PDMPs (1)

- Many PDMPs in use have different types of event
  - Refreshment
  - Zig-Zag
  - Local BPS (Factor Graph)
  - Subsampling
  - $\cdots$
- Each event type affects different parts of the system
- Key point: Different event types correspond to *decompositions* of r

Split PDMPs (2)

- $z = (z_1, \cdots, z_D)$, $\tau = (\tau_1, \cdots, \tau_D) \in \{\pm 1\}^D$
- $\phi(z, \tau) = \tau \odot \phi(z) = (\tau_1 \phi_1(z), \cdots, \tau_D \phi_D(z))$

Split PDMPs (2)

- $z = (z_1, \cdots, z_D)$, $\tau = (\tau_1, \cdots, \tau_D) \in \{\pm 1\}^D$
- $\phi(z, \tau) = \tau \odot \phi(z) = (\tau_1 \phi_1(z), \cdots, \tau_D \phi_D(z))$
- Assume decomposition

$$r(z, \tau) = \sum_{j=1}^{M} r_j(z, \tau) \tag{8}$$

and existence of involutions $\mathcal{F}_j : \{\pm 1\}^D \to \{\pm 1\}^D$ such that

$$r_j(z, \mathcal{F}_j(\tau)) = -r_j(z, \tau) \tag{9}$$

Split PDMPs (2)

- $z = (z_1, \cdots, z_D)$, $\tau = (\tau_1, \cdots, \tau_D) \in \{\pm 1\}^D$
- $\phi(z, \tau) = \tau \odot \phi(z) = (\tau_1 \phi_1(z), \cdots, \tau_D \phi_D(z))$
- Assume decomposition

$$r(z, \tau) = \sum_{j=1}^{M} r_j(z, \tau) \tag{8}$$

and existence of involutions $\mathcal{F}_j : \{\pm 1\}^D \to \{\pm 1\}^D$ such that

$$r_j(z, \mathcal{F}_j(\tau)) = -r_j(z, \tau) \tag{9}$$

- Events of type j happen at rate $\lambda_j(z, \tau)$
  - and then jump according to $Q_j^{\tau}(z \to dz') \cdot \delta(\mathcal{F}_j(\tau), d\tau')$

Making Split-PDMPs work (1)

- Define

$$\lambda_j^0(z,\tau) = (r_j(z,\tau))_+ \quad (10)$$

$$\lambda_j(z,\tau) = \lambda_j^0(z,\tau) + \gamma_j(z,\tau) \quad (11)$$

Making Split-PDMPs work (1)

- Define

$$\lambda_j^0(z,\tau) = (r_j(z,\tau))_+ \tag{10}$$

$$\lambda_j(z,\tau) = \lambda_j^0(z,\tau) + \gamma_j(z,\tau) \tag{11}$$

- Define

$$J_j^\tau(dz) \propto \mu(dz)\lambda_j(z,\tau) \tag{12}$$

and for each $\tau \in \{\pm 1\}^D$, take Q_j^τ to be J_j^τ-reversible.

Making Split-PDMPs work (1)

- Define

$$\lambda_j^0(z,\tau) = (r_j(z,\tau))_+ \tag{10}$$

$$\lambda_j(z,\tau) = \lambda_j^0(z,\tau) + \gamma_j(z,\tau) \tag{11}$$

- Define

$$J_j^\tau(dz) \propto \mu(dz)\lambda_j(z,\tau) \tag{12}$$

and for each $\tau \in \{\pm 1\}^D$, take Q_j^τ to be J_j^τ-reversible.

Theorem

This leads to trajectorially-reversible, $\tilde{\mu}$-stationary Split PDMPs.

Making Split-PDMPs work (1)

- Define

$$\lambda_j^0(z,\tau) = (r_j(z,\tau))_+ \tag{10}$$

$$\lambda_j(z,\tau) = \lambda_j^0(z,\tau) + \gamma_j(z,\tau) \tag{11}$$

- Define

$$J_j^\tau(dz) \propto \mu(dz)\lambda_j(z,\tau) \tag{12}$$

and for each $\tau \in \{\pm 1\}^D$, take Q_j^τ to be J_j^τ-reversible.

Theorem

This leads to trajectorially-reversible, $\tilde{\mu}$-stationary Split PDMPs.

Theorem

Given a fixed splitting, all trajectorially-reversible, $\tilde{\mu}$-stationary Split PDMPs take this form.

Algorithm Design Pipeline (1)

- Non-negotiable: we want samples from $\pi(dx)$.

Algorithm Design Pipeline (1)

- Non-negotiable: we want samples from $\pi(dx)$.
  1. Decide on v.
  2. Decide on ϕ.
  3. Decide on $\psi(dv)$ (and hence μ).

Algorithm Design Pipeline (1)

- Non-negotiable: we want samples from $\pi(dx)$.
  1. Decide on v.
  2. Decide on ϕ.
  3. Decide on $\psi(dv)$ (and hence μ).
  4. Write down r, decide on a splitting.

Algorithm Design Pipeline (1)

- Non-negotiable: we want samples from $\pi(dx)$.
  1. Decide on v.
  2. Decide on ϕ.
  3. Decide on $\psi(dv)$ (and hence μ).
  4. Write down r, decide on a splitting.
  5. Write down λ^0, decide on γ (and hence λ).

Algorithm Design Pipeline (1)

- Non-negotiable: we want samples from $\pi(dx)$.
  1. Decide on v.
  2. Decide on ϕ.
  3. Decide on $\psi(dv)$ (and hence μ).
  4. Write down r, decide on a splitting.
  5. Write down λ^0, decide on γ (and hence λ).
  6. Decide on Q.

Algorithm Design Pipeline (2)

- Choosing Q is often least obvious; order of preference:

Algorithm Design Pipeline (2)

- Choosing Q is often least obvious; order of preference:
  1. Sample from J^τ directly.

Algorithm Design Pipeline (2)

- Choosing Q is often least obvious; order of preference:
    1. Sample from J^{τ} directly.
    2. Sample from its restriction to a finite set. (e.g. BPS)

Algorithm Design Pipeline (2)

- Choosing Q is often least obvious; order of preference:
  1. Sample from J^{τ} directly.
  2. Sample from its restriction to a finite set. (e.g. BPS)
  3. (Use a Metropolis-Hastings step).

Algorithm Design Pipeline (2)

- Choosing Q is often least obvious; order of preference:
  1. Sample from J^{τ} directly.
  2. Sample from its restriction to a finite set. (e.g. BPS)
  3. (Use a Metropolis-Hastings step).
- Choosing ψ could make a big difference; dictates μ.
  - Can have $\psi(dv|x)$ (relatively unexplored)

Algorithm Design Pipeline (2)

- Choosing Q is often least obvious; order of preference:
  1. Sample from J^τ directly.
  2. Sample from its restriction to a finite set. (e.g. BPS)
  3. (Use a Metropolis-Hastings step).
- Choosing ψ could make a big difference; dictates μ.
  - Can have $\psi(dv|x)$ (relatively unexplored)
- Choosing ϕ: some room for creativity here.

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs
  - **Conjecture**: Split as little as possible

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs
  - **Conjecture**: Split as little as possible
  - **Conjecture**: Refresh as little as possible

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs
  - **Conjecture**: Split as little as possible
  - **Conjecture**: Refresh as little as possible
  - Pinch of salt / 'Pre-Asymptopia': Maire, Vialaret (2018)

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs
  - **Conjecture**: Split as little as possible
  - **Conjecture**: Refresh as little as possible
  - Pinch of salt / ‘Pre-Asymptopia’: Maire, Vialaret (2018)
- Implementation remains challenging
  - Splittings may help

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs
  - **Conjecture**: Split as little as possible
  - **Conjecture**: Refresh as little as possible
  - Pinch of salt / 'Pre-Asymptopia': Maire, Vialaret (2018)
- Implementation remains challenging
  - Splittings may help
- *Speculation*: Better dynamics $\phi \rightsquigarrow$ opportunities

Remarks, Open Questions, Takeaways

- Andrieu, Livingstone (2018): Peskun-type ordering for (some) PDMPs
  - **Conjecture**: Split as little as possible
  - **Conjecture**: Refresh as little as possible
  - Pinch of salt / 'Pre-Asymptopia': Maire, Vialaret (2018)
- Implementation remains challenging
  - Splittings may help
- *Speculation*: Better dynamics $\phi \rightsquigarrow$ opportunities
- *Curiosity*: Tempering?

Thank you!