Zig Zag in high dimensions for simulation of Bridges

Frank, Joris, Moritz, Sebastiano

November 2018

1 Preliminary Notes

Disclaimer: these notes are informal and may contain typos. The intention is just to keep track of derivations/ideas and share them.

1.1 Example

Let's derive some results from a simple example. Suppose we do not know how to simulate the Ornstein Uhlembeck process conditioned on the final point given by the following sde:

$$dX_t = (a + b\,X_t)dt + dW_t, \qquad X_T = v.$$

Define with $\mathbb{P}^*$ the conditioned measure induced by this sde and with $\mathbb{Q}^*$, the measure induced by the relative local martingale conditioned at the same final point

$$dM_t = dW_t, \qquad M_T = v.$$

The two measures are absolute continuous and the ratio of this two measures is proportional to the Girsanov Formula:

$$\frac{d\mathbb{P}^*}{d\mathbb{Q}^*}(X) \propto \frac{d\mathbb{P}}{d\mathbb{Q}}(X) = \exp\left\{\int_0^T (a + b\,X_s)dX_s - \frac{1}{2}\int_0^T (a + b\,X_s)^2 ds\right\}. \tag{1}$$

Furthermore, we can represent any Brownian Bridge from u at time 0 to v at time 1 with the Ciesielski construction

$$X(t) = \phi_1(t) + \phi_2(t) + \sum_{i=0}^{\infty}\sum_{j=0}^{2^i-1} \phi_{ij}(t)\xi_{ij}, \tag{2}$$

where ξ_{iJ} are standard Gaussian i.i.d. variables, $\phi_1(t) = u * (1-t)$ is the left-triangle, $\phi_2(t) = v * t$ is the right-triangle , $\phi_{ij}(t) = 2^{-i/j}\phi_{00}(2^i t - j)$ and $\phi_{00}(t) = t\mathbb{1}\{0 \le t < 0.5\} + (1-t)\mathbb{1}\{0.5 \le t \le 1\}$ are the Schauder functions (Wavelet). If we plug (2) in (1), writing momentarily the Ciesielsky expansion with a unique summation (for easing the notation), we get:

$$\exp\left\{\int_0^T a + b\,(\sum_i \phi_i\xi_i)d(\sum_i \phi_i\xi_i) - \frac{1}{2}\int_0^T (a + b\sum_i \phi_i\xi_i)^2 ds\right\}$$

If we truncate the sum, we can interchange summation and integration and we get

$$\exp\Big\{a\sum_i \xi_i \int_0^T d(\phi_i(t)) + b\sum_{ij}\xi_i\xi_j\int_0^T \phi_i(t)d(\phi_j(t)) - \frac{1}{2}\big(a^2T + 2ab\sum_i \xi_i\int_0^T \phi_i(t)dt + b^2\sum_{ij}\xi_i\xi_j\int_0^T \phi_i(t)\phi_j(t)dt\big)\Big\} \tag{3}$$

Finally if we want to write the likelihood ratio with respect to the Lebesgue measure and we want to write it in matrix notation we get:

$$\log(\frac{d\mathbb{P}}{d\mathbb{L}}) = \log(\frac{d\mathbb{P}}{d\mathbb{Q}}\frac{d\mathbb{Q}}{d\mathbb{L}}) = \log(\frac{d\mathbb{P}}{d\mathbb{Q}}e^{\sum_i \frac{-\xi_i^2}{2}}) = a\xi^\top(A - bC) + \xi^\top(bB - \frac{b^2}{2}D - \frac{1}{2}I)\xi + c, \tag{4}$$

with

$$A_i = \int_0^T d(\phi_i(t)), \quad B_{ij} = \int_0^T \phi_i(t)d(\phi_j(t)), \quad C_i = \int_0^T \phi_i(t)dt, \quad D_{ij} = \int_0^T \phi_i(t)\phi_j(t)dt, \quad c = \frac{a^2}{2}T.$$

Note that the matrices B and D share the same 0 entries and are symmetric. The elements of the vector A are all equal to 0 except from the triangles ϕ_1 and ϕ_2. A working-in-progress implementation can be found at `https://github.com/mschauer/Ciesielski.jl`

1.2 The ZigZag for one dimensional Gaussian

The paper I refer to is: `https://arxiv.org/pdf/1607.03188.pdf`. I am trying to keep the same notation of the paper. For a standard Gaussian variable The *potential* is defined as

$$\psi(\xi) = -ln(C) + \frac{(\xi-\mu)^2}{2\sigma^2}.$$

The ZigZag rate is therefore defined as

$$\lambda(\xi,\theta) = (\theta\partial\psi(\xi))^+ = \left(\frac{\theta(\xi-\mu)}{\sigma^2}\right)^+.$$

Finally, given the current position ξ and momentum θ, the inhomogeneous Poisson process is defined as

$$m(s) = \lambda(\xi+\theta s,\theta),$$
$$P(\tau > t) = \int_0^t m(s)ds.$$

Since $m(s)$ is piece-wise linear and positive (*), its integral is quadratic and non-decrasing and admits a unique inverse. Once we find the inverse, we can apply the CDF inversion technique for simulating the inhomogeneous Poisson Process by throwing a uniform random variable from 0 to 1.

$$u = P(\tau \geq t) = \exp\{-\int_0^t m(s)ds)\};$$
$$t = F_x^{-1}(u) = \sqrt{-2\sigma^2\ln(u) + \max(\theta(\xi-\mu),0)^2} - \theta(\xi-\mu).$$

Note: in case the function $m(s)$ is not of the form (*), we have to find a dominating function $M(s)$ for each s satisfying condition (*) and then applying the thinning scheme. Given $m(s)$, I have to figure out how to find $M(s)$ such that $M(s) \geq m(s)$. The paper mentions function $M(s) = (a+bs)^+$ or constant functions $M(s) = c$. I think a valid alternative could be proposing a function that is piecewise constant. The only requirement is that it has to be injective.

1.3 ZigZag d-dimensional Gaussian

Do the same, but in the multi-dimensional caseit gets more complicated and we need to invert the variance covariance matrix Σ. Computational costs of inverting a d-dimensional matrix?

$$\psi(\xi) = -\ln(C) + \frac{1}{2}(\xi-\mu)^\top\Sigma^{-1}(\xi-\mu)$$

$$\lambda(\xi,\theta) = (\theta\partial\psi(\xi))^+ = \theta.^*\left(\Sigma^{-1}(\xi-\mu)\right)^+$$

$$-\ln(P(\tau>t)) = \int_0^t(\theta.^*\Sigma^{-1}(\xi+\theta.^*s))^+ds = \begin{bmatrix} \int_0^t\left(\theta_1(\sum_j v_{1j}(\xi_j+s\theta_j))\right)^+ds \\ \vdots \\ \int_0^t\left(\theta_n(\sum_j v_{nj}(\xi_j+s\theta_j))\right)^+ds \end{bmatrix} = \begin{bmatrix} \int_0^t\left(a_1+b_1s\right)^+ds \\ \vdots \\ \int_0^t\left(a_n+b_ns\right)^+ds \end{bmatrix}$$

where v_{ij} are the entries of Σ^{-1}. Note that $a_i, b_i \in \mathbb{R}$. In the multi-dimensional case there is the ulterior passage of choosing which coordinate to switch for each Poisson event i.e after trowing d uniform rvs and inverting them, we switch the coordinates corresponding to the minimum of the d waiting times just derived and we take $\tau_0 = \min(\tau_i)$.

A raw implementation of the algorithms can be found at: `https://github.com/SebaGraz/ZigZagprocess`

1.4 Independent ZigZag sampler (gain in efficiency)

1.5 Bouncy Particle and Coordinate sampler with multi-dimensional Gaussian target

The Bouncy Algorithm is described in the paper: `https://arxiv.org/abs/1510.02451`. We describe briefly how to generate the Inhomogeneous Poisson Process (IPP) when the target distribution is Gaussian. Given the energy function $\psi(\xi)$ and the velocity θ, the rate of the IPP is

$$\lambda(\xi, \theta) = (\langle \nabla\psi(\xi), \theta \rangle)^+$$

therefore as before we want to invert:

$$\Lambda(t) = \int_0^t (\langle \nabla\psi(\xi + s * \theta), \theta \rangle)^+ ds$$

that, given the Gaussian energy function, boils down to inverting

$$\Lambda(t) = \int_0^t (a + bs)^+ ds$$

for certain a and b.

The Coordinate Sampler Algorithm is described in the paper: `https://arxiv.org/abs/1809.03388`.

The d-dimensional vector space of velocity is chosen to be $\mathbb{V} = \{\pm e_i, i = 1, 2, ..., d\}$ where e_i is a d-dimensional vector with 1 at the ith entry and 0 elsewhere.

The waiting time is the same as the bouncy particle (althugh it is simplified cause the velocity are restricted to be 0 everywhere except one dimension) while the transition dynamic is given by

$$Q(dx', dv'|dx, dv) = \sum_{v^* \in \mathbb{V}} \frac{\lambda(-v^*, x)}{\lambda(x)} \delta_{v^*}(v') \delta_x(x')$$

and $\lambda(x) = \sum_{i=1}^d |\frac{\partial \nabla U(x)}{\partial x_i}|$. Check that $\int_{x\in\Omega} \int_{v\in\mathcal{V}} f(x, v) dx dv = 1$ for any initial condition v', x'.

1.6 Algorithms comparisons

Algorithm 1 ZigZag

```
procedure EUCLID(a, b)                 ▷ The g.c.d. of a and b
    r ← a mod b
    while r ≠ 0 do                     ▷ We have the answer if r is 0
        a ← b
        b ← r
        r ← a mod b
    return b                           ▷ The gcd is b
```

Algorithm 2 Bouncy

```
procedure EUCLID(a, b)                 ▷ The g.c.d. of a and b
    r ← a mod b
    while r ≠ 0 do                     ▷ We have the answer if r is 0
        a ← b
        b ← r
        r ← a mod b
    return b                           ▷ The gcd is b
```

Algorithm 3 Coordinate sampler

```
procedure EUCLID(a, b)                 ▷ The g.c.d. of a and b
    r ← a mod b
    while r ≠ 0 do                     ▷ We have the answer if r is 0
        a ← b
        b ← r
        r ← a mod b
    return b                           ▷ The gcd is b
```

2 Merging section 1.1 with the zigzag Algorithm

As you can see in (4), we have a quadratic expression on ξ for the density of the bridge. In the d-dimensional Gaussian ZigZag experiment we have a quadratic form for the density as well. Therefore the implementation of the ZigZag for simulating the Ornstein Uhlembeck process should not be so different of the one for the d-dimensional Gaussian. Remember that in section 1 we already get the precision matrix while in the d-dimenional Gaussian we had to derive it taking the inverse of the covariance matrix. As already mentioned both the matrices B and D are sparse (see appendix of Moritz thesis for the shape of the matrix).

2.1 Conjectures and Observations:

- As L $\to \infty$, the Standard ZigZag process is going to change the direction always i.e. $\tau \to 0$. waiting time equal to 0. Easy to prove: the components at every level are independent between each other.

- if we keep the same velocity for all the components, every component change direction (I guess that would mean that we explored enough the space).

- Understand how the velocity affects the waiting time; Answer: time re-scale in one dimension.The question is now what if the velocities are different? What's the role of γ and how to use it.

- the Slower the velocity, the larger the waiting time
- The velocity at every level should be different. We could find a relation between velocity and level.
- The truncation can be done when we do not expect that the component in that level are not going to trigger the waiting time.

3 What happens if things get non-linear

$$dX_t = \sin(X_t)dt + dW_t$$

We begin writing down the likelihood using Girsanov as before:

$$\frac{d\mathbb{P}^*}{d\mathbb{Q}^*}(X) \propto \frac{d\mathbb{P}}{d\mathbb{Q}}(X) = \exp\left\{\int_0^T \sin(X_s)dX_s - \frac{1}{2}\int_0^T \sin^2(X_s)ds\right\}. \tag{5}$$

We are using the same notation as at the begginnig of these notes. We would like to implement a ZigZag on this target distribution expanding the continuous Brownian path with the Ciesielski construction of Brownian Bridge. It can be useful to get rid of the stochastic integral found in the Girsanov formula with the following trasformation valid whenever the drift is everywhere differentiable. set $A(x) = \int_0^X \sin(s)ds$, by ito:

$$dA(X) = \partial_X A(X)d(X) + \frac{1}{2}\partial_{X^2}A(X)dt$$

Therefore equation 5 can be written as

$$\frac{d\mathbb{P}}{d\mathbb{Q}}(X) = \exp\left\{1 - \cos(X_T) - \frac{1}{2}\int_0^T \left(\sin^2(X_t) + \cos(X_t)\right)dt\right\}.$$

Now let us try to expand the continuous path X with the Faber-Shauder basis and then take the derivative as required by the three algorithms. We hope to be able to bound the target function in order to be able to implement it. Suppose we want to simulate a bridge 0,0; 0,1. Then

$$\ln(\frac{d\mathbb{P}}{d\mathbb{Q}})(X) \propto -\frac{1}{2}\int_0^T \left(\sin^2(X_t) + \cos(X_t)\right)dt.$$

We know an upper bound and lower bound for the integrand and therefore also for the whole integral (Maybe this is the solution).

3.1 TODO:

- Comparison with Euler discretization: see - Exact simulation of diffusion processes
- Understand how to control the approximation when truncating the summation (can we give a prior distribution also at the truncation error?). Answer: An Introduction to Infinite Dimensional Analysis
- Davis for Piecewise deterministic Markov Processes
- How many events occurs in a small time space ϵ, given d dimensions for each algorithm and what is its complexity. Motivate the answer reading the small paragraph of coordinate sampler.