PERFECT Simulation

Lecture 3

Perfect simulation III

Dominated Processes and Uniform Coupler

Mark Huber
Claremont McKenna College
July, 2018

Supported by NSF grant DMS 1418495

The story so far

Coupling from the past: set up

Ingredients

- Need update function $\phi(x, r)$
- Need set A such that

$$(\forall r \in A)(\forall x \in \Omega)(\phi(x, R) = \{a\})$$

So no matter what random choices I pick in A, all the states in the state space get updated to move to the same state a

Coupling from the past: execution

`CFTP`

1. Draw R
2. If $R \in A$, $X \leftarrow$ only element of $\phi(\Omega, R)$
3. Else $Y \leftarrow$ `CFTP`, $X \leftarrow \phi(Y, R)$
4. Output X

Big question: how do we find such an A?

Coupling from the past: monotonicity

- Suppose we have a partial order on the state space [So for some states $a \preceq b$]
- There is a maximum state $x_{\max}$ where $(\forall x)(x \preceq x_{\max})$
- There is a minimum state $x_{\min}$ where $(\forall x)(x_{\min} \preceq x)$
- For all random choices r

$$x \preceq y \Rightarrow \phi(x, r) \preceq \phi(y, r)$$

Coupling from the past: utilizing monotonicity

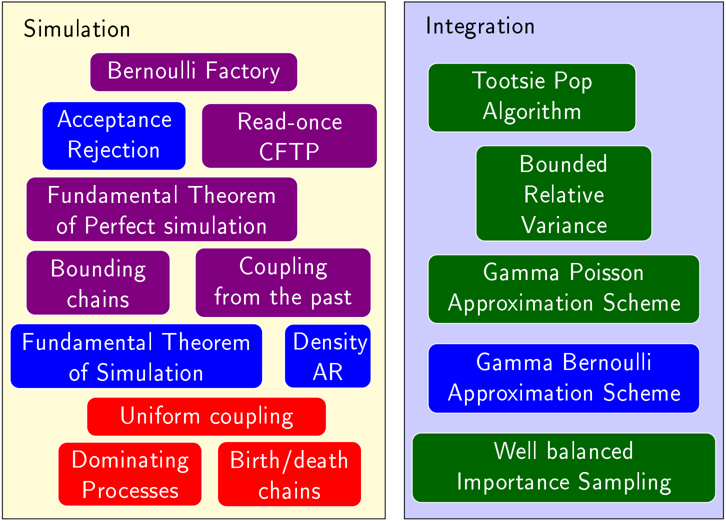
With a monotonic update function

$$A = \{r : \phi(x_{\min}, r) = \phi(x_{\max}, r)\}$$

Everything else is trapped between the update for the minimum and maximum states!

Two obstacles to using this

1. What if there is no $x_{\max}$ state?
2. What if the state space is continuous, and $\phi(x_{\min}, r)$ never quite reaches $\phi(x_{\max}, r)$?


Simulation
Bernoulli Factory
Acceptance Rejection
Read-once CFTP
Fundamental Theorem of Perfect simulation
Bounding chains
Coupling from the past
Fundamental Theorem of Simulation
Density AR
Uniform coupling
Dominating Processes
Birth/death chains
Integration
Tootsie Pop Algorithm
Bounded Relative Variance
Gamma Poisson Approximation Scheme
Gamma Bernoulli Approximation Scheme
Well balanced Importance Sampling

What if there is no $x_{\max}$ state?

Unfortunately, this happens a lot

- Perpetuities with state space $[0, \infty)$
- Point processes with unlimited numbers of points
- Queuing networks

Perpetuities

Perpetuities

Model things that grow or shrink randomly with random addition

$$X_{t+1} = A_t X_t + B_t,$$

where

$$A_1, A_2, \ldots \sim A$$
$$B_1, B_2, \ldots \sim B$$

When $B_t = 1$, $A_t = U_t^{1/\beta}$, where $\{U_t\}$ are Unif$([0,1])$ call this a **Vervaat Perpetutity** [1]

---

[1] W. Vervaat, On a Stochastic Difference Equation and a Representation of Non-negative Infinite Divisible Random Variables, *Adv. in Appl. Probab.*, 11(4):750–783, 1979

Another view

Can also view Vervaat Perpetuities as infinite sums of products of iid $U_1, U_2, \ldots \sim \text{Unif}([0,1])$

$$Y = U_1^{1/\beta} + [U_1 U_2]^{1/\beta} + [U_1 U_2 U_3]^{1/\beta} + \cdots$$

- Lower state is 0
- No upper bound on the state of this chain

Simplify things

- For simplicity of exposition, let $\beta = 1$
- Can extend techniques to general beta

Naive update function is monotonic

- For any $U \in [0,1]$,

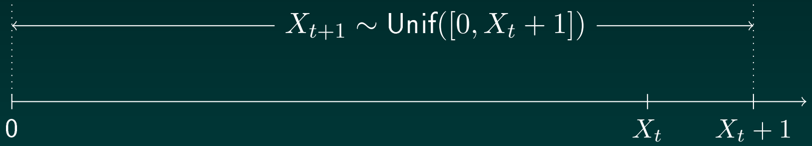
$$x \leq y \Rightarrow U(x+1) \leq U(y+1)$$

- Lower bound is 0
- No upper bound!
- Also strictly monotonic

$$x < y \Rightarrow U(x+1) < U(y+1)$$

so upper and lower processes will never meet

In pictures



Dominating process

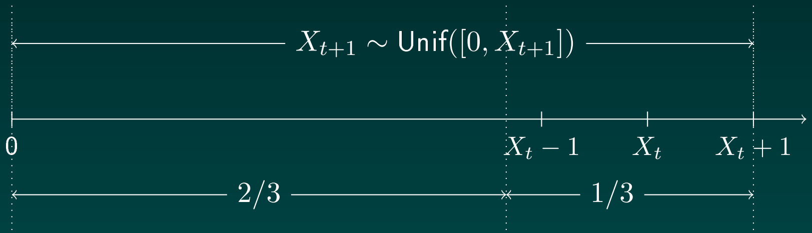
Definition

Say that Y_t **dominates** X_t if $X_t \leq Y_t$ for all t.

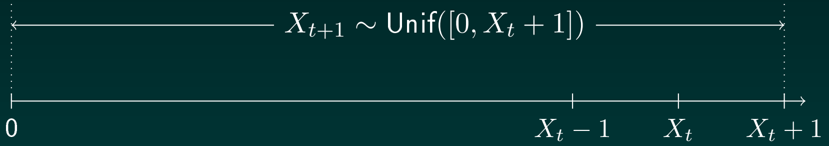
Getting a dominating process

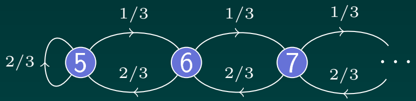
Suppose that $X_t \geq 5$. Then

$$\mathbb{P}((X_t + 1)U_t \leq X_t - 1) = \mathbb{P}(U \leq 2/3) = 2/3$$



The Dominating Process for $\beta = 1$





Why is it dominating?

$$X_{t+1} = (X_t + 1)U_t,$$
$$Y_{t+1} = Y_t - \mathbb{1}(U_t \leq 2/3, Y_t \geq 5) + \mathbb{1}(U_t > 2/3)$$

Fact

If $X_0 \leq Y_0$, $X_{t+1} = (X_t + 1)U_t$, *and* $Y_{t+1} = Y_t - \mathbb{1}(U_t \leq 2/3, Y_t \geq 5) + \mathbb{1}(U_t > 2/3)$ *then*

$$(\forall t)(X_t \leq Y_t)$$

Why is this useful?

Can calculate the stationary distribution for Y_t

$$(\forall i \in \{1,2,3\ldots\})(\mathbb{P}(Y_\infty = 4+i) = (1/2)^i)$$

Or another way to say it

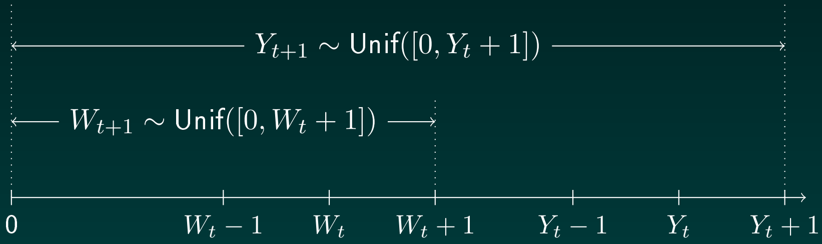
$$Y_\infty \sim 4 + G,\ G \sim \text{Geo}(1/2)$$

So

1. Draw $Y_0 \leftarrow Y_\infty$, and then set $W_0 \leftarrow 0$
2. Draw $U_0, \ldots, U_{k-1}$ iid Unif$([0,1])$
3. For $i = 1$ to k, $Y_i \leftarrow (Y_{i-1}+1)U_{i-1}$, $W_i \leftarrow (W_{i-1}+1)U_{i-1}$

Then $W_i \leq X_i \leq Y_i$

Bringing the processes together

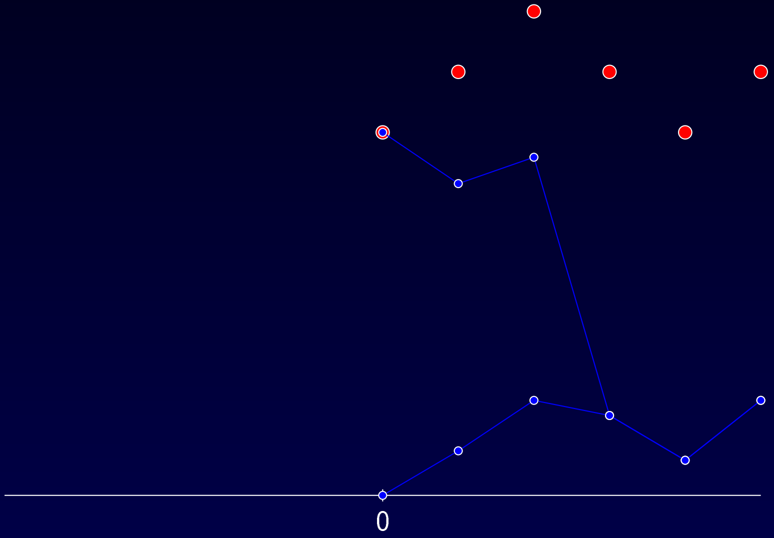


- If we draw $X \sim \text{Unif}([0, Y_t + 1])$ and it falls in $[0, W_t + 1]$, then it is also uniform over $[0, W_t + 1]$
- So probability that they come together is $(W_t + 1)/(Y_t + 1)$

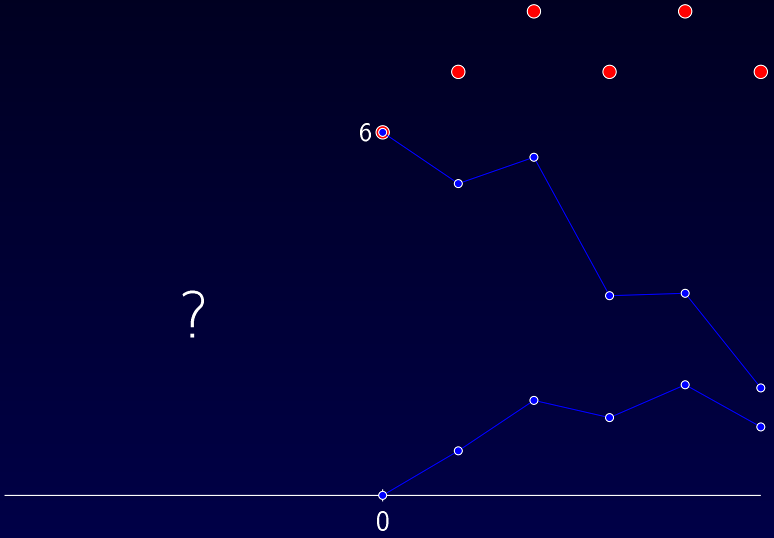
Putting this together

1. Draw $Y_0 \leftarrow Y_\infty$, and then set $W_0 \leftarrow 0$
2. Draw $U_0, \ldots, U_{k-1}, U_k$ iid Unif$([0,1])$
3. For $i = 1$ to k, $Y_i \leftarrow (Y_{i-1} + 1)U_{i-1}$, $W_i \leftarrow (W_{i-1} + 1)U_{i-1}$
4. $Y_{k+1} \leftarrow U_k(1 + Y_k)$
5. If $Y_{k+1} \leq W_k + 1$ then $W_{k+1} \leftarrow Y_{k+1}$
6. Else $W_{k+1} \leftarrow (W_k + 1)[Y_{k+1} - (W_k + 1)]/[Y_k - W_k]$

Great if it converges



What if it doesn't converge?



Change from regular CFTP

If it does not converge

- Need a coupled draw (X_0, Y_0) conditioned on $Y_0 = 6$
- To get it, use reversibility
- Recall that if

$$\pi(dx)p(x, dy) = \pi(dy)p(y, dx)$$

  then the chain is **reversible**, and π is a stationary distribution of the chain
- For reversible chains in a stationary state, the path looks the same run forward and backward

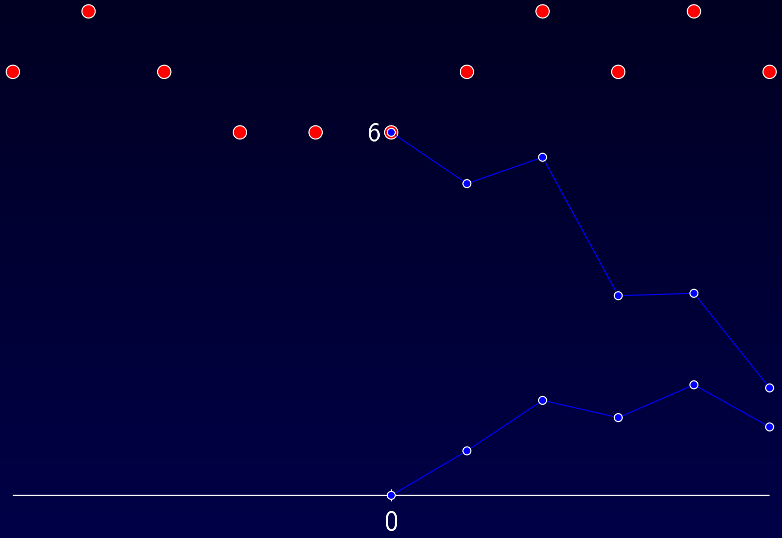
Building the dominated process that ends at the right spot

3 steps

1. Run the dominated chain backwards in time to beginning of the block
2. Impute the forward uniforms from the backward run
3. Use the forward uniforms to update the underlying chain

The result is a run of the underlying chain whose dominating chain ends at the proper spot

Run dominating process back in time



Impute forward uniforms

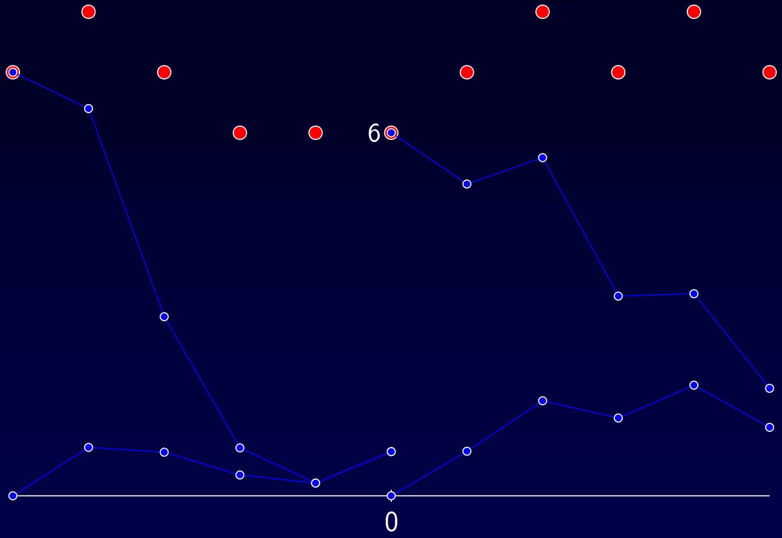
Example If $Y_{-3} = 7$ and $Y_{-2} = 6$, then

$$U_{-2} \sim \text{Unif}[0, 2/3]$$

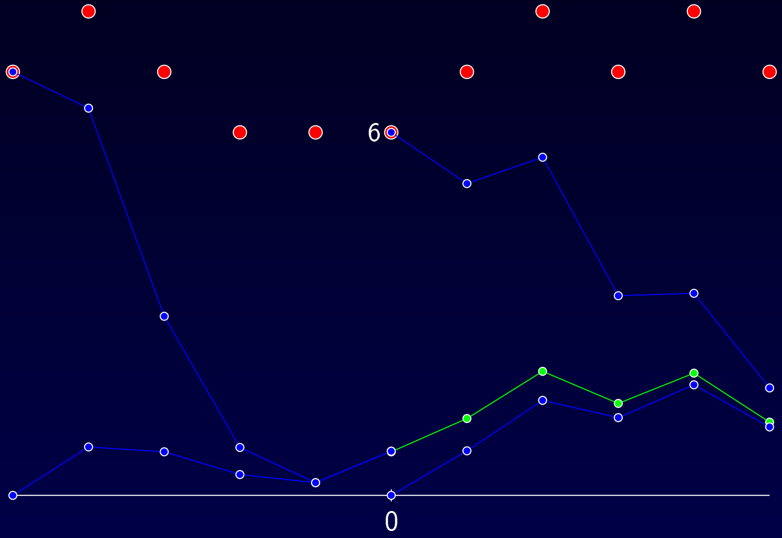
If $Y_{-5} = 7$ and $Y_{-4} = 8$ then

$$U_{-4} \sim \text{Unif}[2/3, 1]$$

Use uniforms to drive upper and lower processes forward



Now drive (X_0, Y_0) *forward to get* X_t



Spatial processes

Poisson point process

Rate λ over a window A

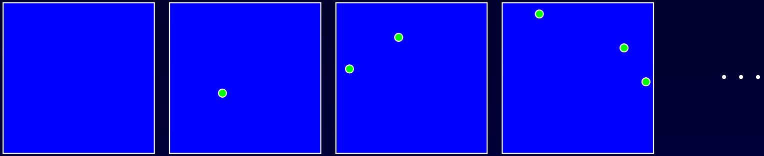
1. Draw $N \leftarrow \mathsf{Pois}(\lambda \cdot \mathsf{area}(A))$
2. Draw $P_1, \ldots, P_N$ uniform over A

Remarks

- The number of possible points is unbounded
- Resulting point process is in the **exponential space**

Picture of PPP

Region with $\lambda \cdot \text{area} = \mu$



μ^0 $\quad$ $\mu^1/1!$ $\quad$ $\mu^2/2!$ $\quad$ $\mu^3/3!$ $\quad \cdots$

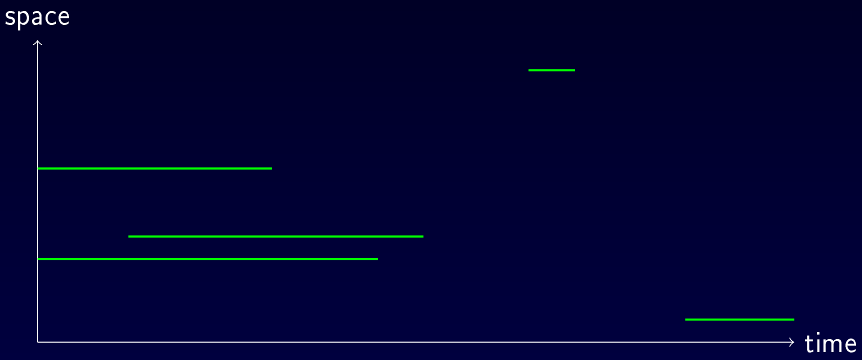
PPP in time

Jump process

- Continuous time Markov chain
- This is a birth-death chain
- Points are born into the process
- Time between births exponential distribution of rate $\lambda \cdot \text{area}(A)$
- Points live for a time, then die (and are removed)
- Stationary distribution is PPP

PPP in time

Suppose space is 1D, two points alive at time 0



- Lifetime of point is Exponential rate 1
- Time between births is Exponential rate μ

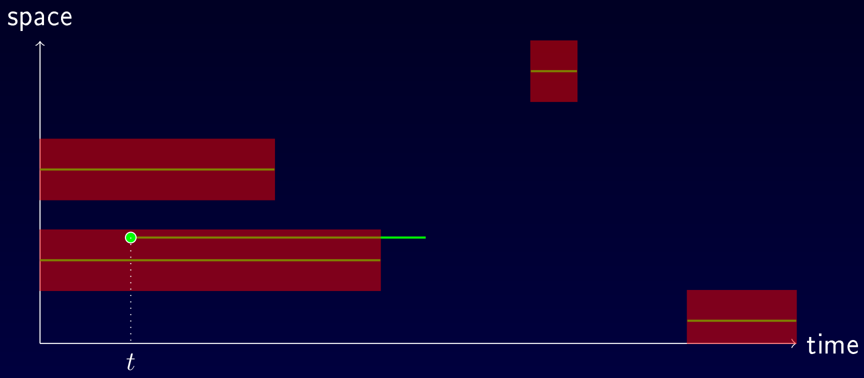
Modifying for spatial point processes

Want to draw from bounded density with respect to PPP (for $\gamma \in [0, 1]$)

$$f_{\text{Strauss}}(P) = \gamma^{t(P)}, \quad t(P) = \#\{\{x_i, x_j\} \in P : \text{dist}(x_i, x_j) < R\}$$

- Deaths work as in the original PPP–remove the point
- When point born, roll $U \sim \text{Unif}([0, 1])$ to see if point added
- For instance, if new point would increase $t(P)$ by 2, only accept the birth with probability γ^2
- This makes the chain reversible with the correct stationary distribution

Picture



- Point born at t too close to existing node
- Birth only occurs with probability γ

Pseudocode for birth-death step

`Birth-Death Strauss step`

Input: current state $P_1, \ldots, P_n$, intensity function λ

1. Draw $T_B \leftarrow \text{Exp}(\mu)$, $T_D \leftarrow \text{Exp}(n)$
2. If $T_D < T_B$ (death) then let $I \leftarrow \text{Unif}(\{1, \ldots, n\})$, remove point P_I from the set
3. Else (possible birth) draw P from λ normalized over the region
   - *3.1* Let $b \leftarrow \{i : \text{dist}(P_i, P)$, draw $U \leftarrow \text{Unif}([0, 1])$
   - *3.2* If $U \leq \gamma^b$, then let $P_{N+1} \leftarrow P$

Using with dominated CFTP

The PPP continuous time Markov chain is the dominating process

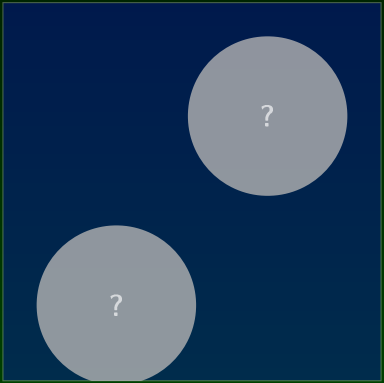
- The Strauss ctmc is the underlying process
- DCFTP works as with the perpetutities

One step in the process

- Generate a birth or a death
- Always accept deaths and remove the point
- Births are either added for certain, not added for certain, or might be added (?)

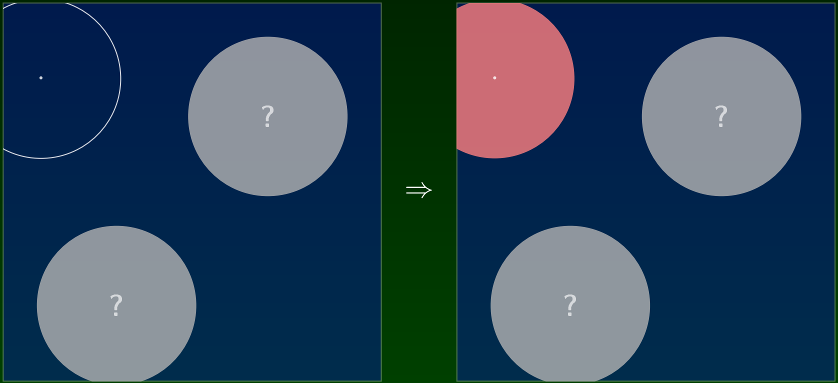
When the ? points are gone, the process has coupled

At time 0, all points uncertain



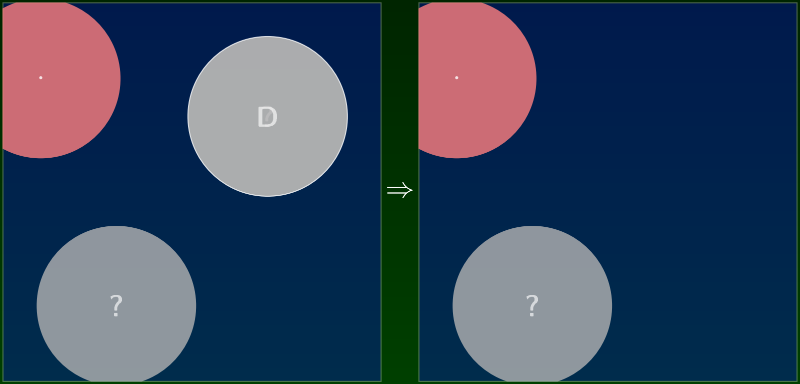
Radius of disk is $R/2$

Unblocked birth



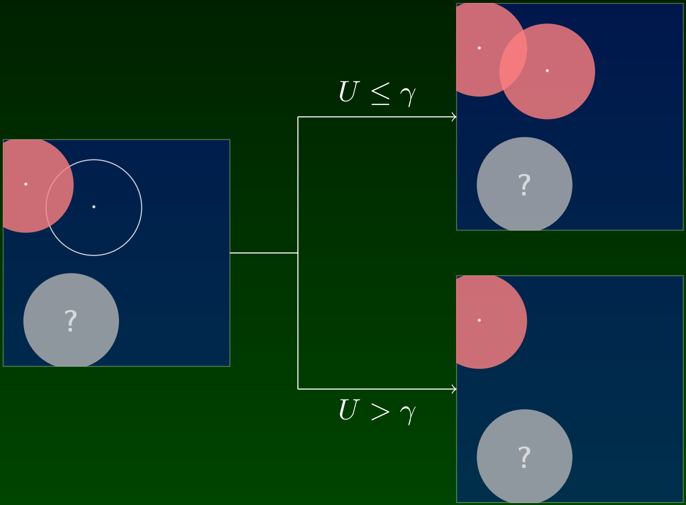
If point is born that does not conflict with any existing disks, it is certainly added

Death



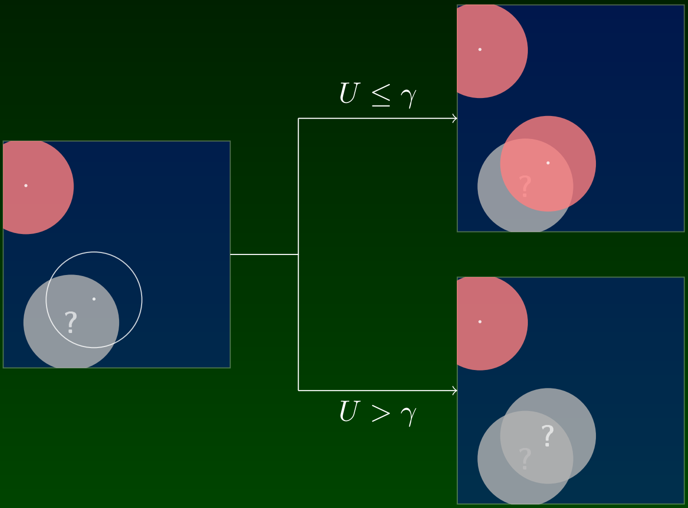
If point dies–whether certain or uncertain–it is always removed

Birth near known point



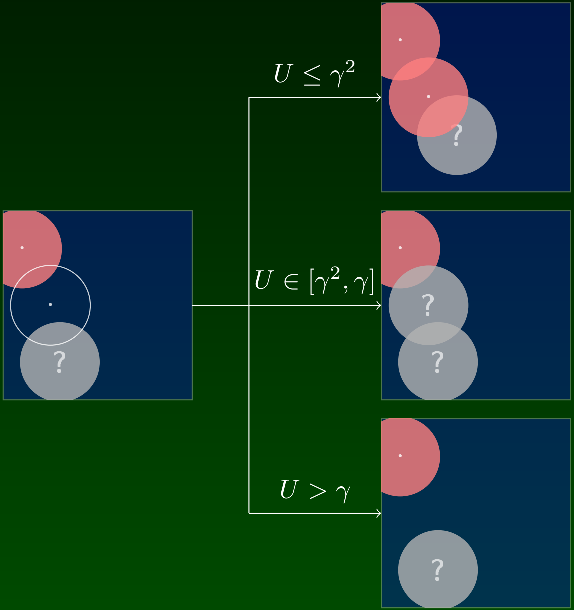
If point dies–whether certain or uncertain–it is always removed

Birth near unknown point

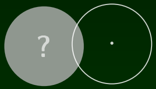


Point might or might not be born

Birth near both



New ? points must be next to existing points



- Let B_R be the ball of radius R around the point
- Average # of ? children a ? point has before dying is at most

$$\alpha < \lambda(B_R)(1 - \gamma)$$

- So if $\alpha < 1$, then ? points die away exponentially fast

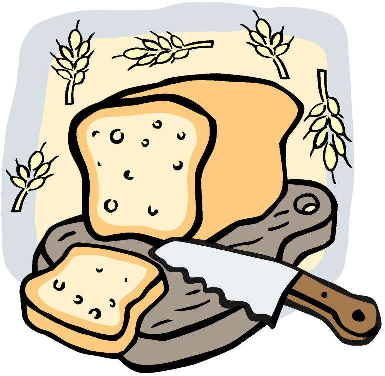
Birth-death-swap chains

- Birth-death-swap chains improve this framework [2]
- Death same as before
- Births can be "blocked" by one or more points
- If birth blocked by exactly one point, can swap with proability $1/4$: take the point's place (so removed blocking point and add the birth)
- This small change guarantees efficiency when

$$\lambda \cdot \text{area}(B_R)(1 - \gamma) < 2$$

[2] M. Huber, Spatial birth-death swap chains, *Bernoulli*, arXiv:1006.5934, 18(3):1031–1041, 2012

Slice sampling



Drawing uniforms

- Recall that if we draw X uniformly from B where $A \subseteq B$, and $X \in A$, then $X \sim \text{Unif}(A)$
- Mira, Møller, and Roberts[3] used this fact to create a **perfect slice sampler**

[3] A. Mira, J. Møller, and G. O. Roberts, Perfect slice samplers, *J. R. Statist. Soc. B*, 63(3):593–606, 2001

Slice sampler

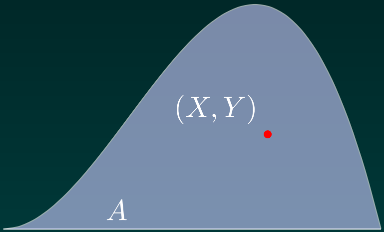
Remember Fundamental Theorem of Simulation, to draw

$$X \sim f_X$$

instead draw

$$(X, Y) \sim \mathsf{Unif}(\{(x, y) : 0 \leq y \leq f_X(x)\})$$

Example: drawing from beta distribution



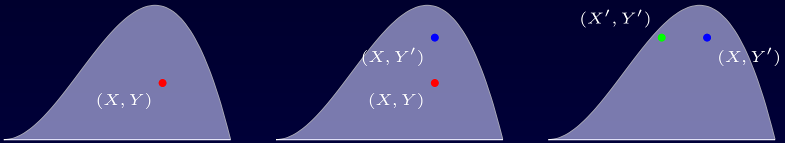
$\mathsf{Beta}(3,2)$

$X \sim \mathsf{Beta}(3,2)$

$(X, Y) \sim \mathsf{Unif}(A)$

Gibbs sampler

1. Given X, draw $Y \leftarrow \text{Unif}([0, f_X(X)])$
2. Given Y, draw $X \leftarrow \text{Unif}(\{x : f_X(x) \geq Y\})$



This is the **slice sampler** since X is drawn from the slice of the volume under the density of height Y

Using monotonicity with perfect slice sampler

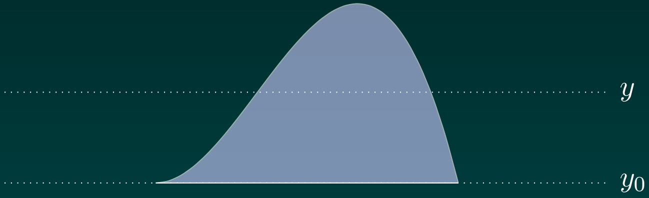
Use the following partial order on states

$$(X, Y) \preceq (W, Z) \Leftrightarrow Y \leq Z$$

Is there a monotonic update function?

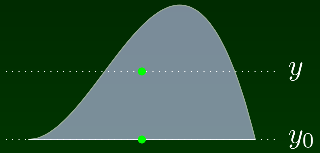
Here's the idea

- The hard part is drawing uniformly from $\{x : f_X(x) \geq y\}$
- Want to draw simulataneously for all y
- Use a nested approach

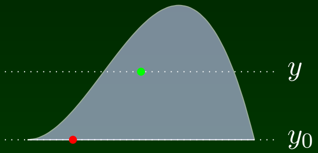


- Draw X uniformly from $\{x : f_X(x) \geq y_0\}$
- If $X \in \{x : f_X(x) \geq y\}$, also accept as uniform over this set

Illustration of monotonic slice update

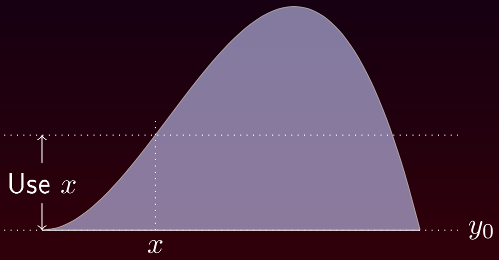


Same value for y_0 and y



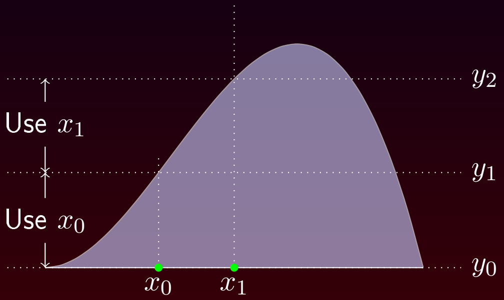
Different value for y_0 and y

Range of y for which choice of x works

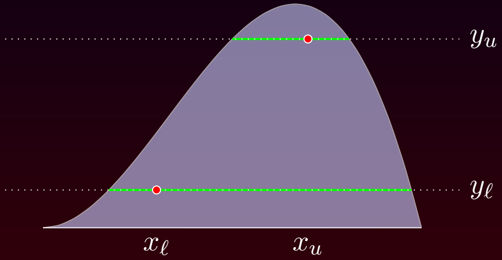


All y in $[0, f_X(x)]$ will have x as their value

Can repeat to get values for all y's needed



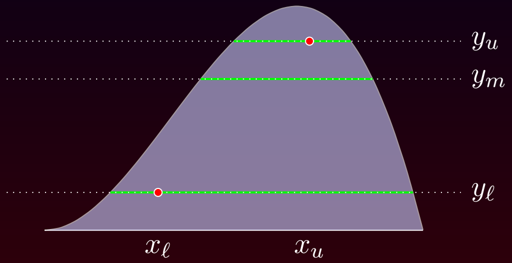
For CFTP, inital run only needs upper and lower process



Given lower step

1. If $f_X(x_\ell) \geq y_u$ then $x_u \leftarrow x_u$
2. Else draw x_u uniformly from $\{x : f_X(x) \geq y_u\}$

Recursive step: upper, lower, and middle process



Given upper and lower step

1. Draw x_m uniformly from $\{x : f_X(x) \geq y_m\}$
2. If $f_X(x_m) \geq y_u$ then $x_m \leftarrow x_u$

Summary

Domination

- Create a chain/process which upper bounds chain
- Naturally works with birth-death chains for spatial point processes
- Solution to chains with no $x_{\max}$

Uniform coupling

- For $A \subseteq B$, if $X \sim \text{Unif}(B)$ has $X \in A$, then $X \sim \text{Unif}(A)$
- Draws from our earlier work on AR
- Solution to moving continuous chains together