Time-dependent modified next reaction method

October 9, 2016

Core algorithm

Number of reactions: M
$rnd()$ is function that return a random number in $[0, 1]$ (uniform)

- Initialization:
  1. for k from 1 to M, $\Delta T_k = \log\left(\frac{1}{rnd()}\right)$
     This picks numbers from an exponential distribution $p(x) = \exp(-x)$
  2. set $t = 0$
- Loop:
  1. for k from 1 to M, calculate Δt_k so that $\Delta T_k = \int_t^{t+\Delta t_k} a_k(X(t), s)ds$
     This translates the time left from the exponential distribution (the Poisson process) into a physical time that should pass until the event fires.
  2. call μ the index for which $\Delta t_\mu = \min(\Delta t_1, ..., \Delta t_M)$
     This is the event that shall fire first.
  3. for k from 1 to M except μ, change ΔT_k to $\Delta T_k - \int_t^{t+\Delta t_\mu} a_k(X(t), s)ds$
     Note that because Δt_μ is the smallest of them all, the integral will be smaller than the one in 1, en ΔT_k will stay positive (unless there's some really strange propensity function, which of course should not happen)
  4. add Δt_μ to t
  5. fire event μ (change state vector)
  6. set $\Delta T_\mu = \log\left(\frac{1}{rnd()}\right)$
     For this particular event, no next one had been calculated, so we need to pick a new internal time from an exponential distribution (again for the poisson process)

An optimization

It may not be necessary to do the $\int_t^{t+\Delta t_\mu} a_k(X(t), s)ds$ calculation every time. If the propensity function changes due to each event, then we really do need to calculate every

$$\Delta T_{k,1} = \int_{t_0}^{t_1} a_k(X(t), s)ds,\ \Delta T_{k,2} = \int_{t_1}^{t_2} a_k(X(t), s)ds,\ \Delta T_{k,3} = \int_{t_2}^{t_3} a_k(X(t), s)ds, \text{ etc.}$$

However, if the propensity does not change for a particular event k, then instead of calculating each $\Delta T_{k,i}$ above, we can save some unnecessary recalculations by just calculating an integral

$$\Delta T_{k,sum} = \int_{t_0}^{t_{end}} a_k(X(t), s)ds$$

when really needed.

To make this work, some additional bookkeeping is needed to be able to determine when the events would fire in real time.

- Initialization:
    1. for k from 1 to M, $\Delta T_k = \log\left(\frac{1}{rnd()}\right)$
       This picks numbers from an exponential distribution $p(x) = \exp(-x)$
    2. set $t = 0$
    3. for each k, we must also know the time at which this calculation of ΔT took place. For now this is just $t = 0$, so we set $t_k^c = 0$ for all k.
    4. for each k, map these internal Poisson intervals ΔT_k to event fire times t_k^f using the propensities:

       $$\Delta T_k = \int_{t_k^c}^{t_k^f} a_k(X(t_k^c), s)ds$$

- Loop:
    1. for k from 1 to M, calculate the minimum real time that would elapse until an event: $\Delta t_\mu = \min(t_1^f - t, ..., t_M^f - t)$. Here μ is the index of the event that corresponds to this minimal value.
    2. add Δt_μ to t
    3. only for the events k for which the propensities will be affected by μ, we need to do the following:
        - Diminish the internal time ΔT_k with the internal time that has passed:

          $$\Delta T_k := \Delta T_k - \int_{t_k^c}^{t} a_k(X(t_k^c), s)ds$$

          Here t is the new time, and the propensities are still the *old* propensities!
        - Set $t_k^c = t$
    4. fire event μ (change state vector), generate a new random number and ΔT value, set $t_\mu^c = t$ and calculate t_μ^f accordingly.
    5. only for the events k for which the propensities were affected by μ, we need to recalculate the real fire times of these events: calculate t_k^f so that this holds:

       $$\Delta T_k = \int_{t_k^c}^{t_k^f} a_k(X(t_k^c), s)ds$$

       Note that here we're working with the *new* propensities.

If one keeps track of which event affects which, this can really save some calculation time. Furthermore, if certain events are stored in a list sorted on real event fire times, the minimum may very easily be calculated: if these times increase, one will only need to look at the first event instead of them all.

One might argue that keeping such a list ordered may require some computation as well, but if the list (or a part of it) does not need to be updated due to a certain event, no computation is needed for that part.

A slightly re-ordered version (for positive times only since we use a negative one as a marker):

- Initialization:
  1. set $t = 0$
  2. for k from 1 to M, let $\Delta T_k = \log\left(\frac{1}{rnd()}\right)$ (this picks numbers from an exponential distribution $p(x) = \exp(-x)$). Set $t_k^c = 0$ and set $t_k^f = -1$ to indicate that this event time still needs to be calculated from the ΔT_k version.
- Loop:
  1. for k from 1 to M, if $t_k^f < 0$ then calculate t_k^f from the stored ΔT_k value so that:
  $$\Delta T_k = \int_{t_k^c}^{t_k^f} a_k(X(t_k^c), s)ds$$
  2. for k from 1 to M, calculate the minimum real time that would elapse until an event takes place: $\Delta t_\mu = \min(t_1^f - t, ..., t_M^f - t)$. Here μ is the index of the event that corresponds to this minimal value.
  3. add Δt_μ to t
  4. only for the events k for which the propensities will be affected by μ, we need to do the following:
     - Diminish the internal time ΔT_k with the internal time that has passed:
     $$\Delta T_k := \Delta T_k - \int_{t_k^c}^{t} a_k(X(t_k^c), s)ds$$
     Here t is the new time, and the propensities are still the *old* propensities!
     - Set $t_k^c = t$ and set $t_k^f = -1$ to indicate that it still needs to be calculated from the remaining ΔT_k.
  5. fire event μ (change state vector), generate a new random number and ΔT value, set $t_\mu^c = t$ and set $t_\mu^f = -1$ to indicate that t_c^f should be calculated from ΔT_μ.