(Pseudo) Randomness

When would we want randomness?

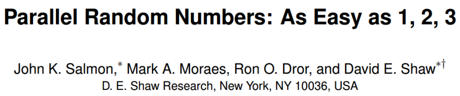
- Stochastic input (e..g spikes arriving via a Poisson process).
- Stochastic channel gating.
- Random channel parameters.
- Random morphology.
- Random network connections.

Guiding principles

- Randomness should be **reproducible**.
  - e.g. use random seeds.
- Randomness should be **independent of the number of processors**.

Random123

Random123 is a **cryptographic quality** generator, suitable for managing separate **independent**, **reproducible**, and **restartable** streams; the algorithm used is called Philox in the paper:



Parallel Random Numbers: As Easy as 1, 2, 3

John K. Salmon,[*] Mark A. Moraes, Ron O. Dror, and David E. Shaw[*†]
D. E. Shaw Research, New York, NY 10036, USA

```
from neuron import h
r = h.Random()
r.Random123(seed, id1, id2)
print(r.uniform(0,1)) # or binomial, or geometric, or lognormal, or negexp, or ...
```

A reasonable choice for `id1` is the gid of the cell; a reasonable choice for `id2` is the section index in `cell.all` or the synapse index.

Example: random synaptic inputs ("noise")

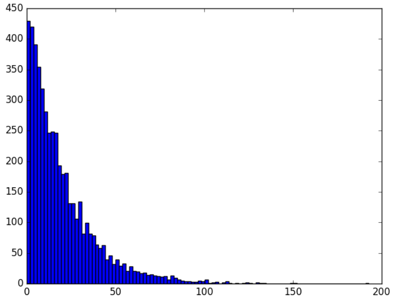
```
from neuron import h, gui
import numpy
from matplotlib import pyplot

r = h.Random()
r.negexp(1)
r.Random123(1, 2, 3)
ns = h.NetStim()
ns.noiseFromRandom(r)
ns.number = 1e20
ns.start = 5
ns.interval = 20
ns.noise = 1

spike_times = h.Vector()
nc = h.NetCon(ns, None)
nc.record(spike_times)

pc = h.ParallelContext()
pc.set_maxstep(10)
h.stdinit()
pc.psolve(100000)

spike_times = spike_times.as_numpy()
isis = spike_times[1:] - spike_times[:-1]
print numpy.mean(isis), numpy.std(isis)
pyplot.hist(isis, 100)
pyplot.show()
```



Example: random distribution of leak conductance

```
import sys
seed = float(sys.argv[1])

class Pyramidal:
    def __init__(self, gid):
        self._gid = gid
        self._setup_morphology()
        self._discretize()
        self.random_stream = [self.new_random_stream(i) for i in range(len(self.all))]
        self._add_channels()
        self._register_netcon()
    def new_random_stream(self, id2):
        r = h.Random()
        r.Random123(seed, self._gid, id2)
        return r
    def _add_channels(self):
        for sec in self.soma:
            sec.insert('hh')
        for sec, stream in zip(self.all, self.random_stream):
            sec.insert('pas')
            for seg in sec:
                seg.pas.g = max(0, stream.normal(1e-3, 8e-6))
    # the rest stays as before
```

If you try this with seed 1, you'll notice the raster looks significantly different. Why?