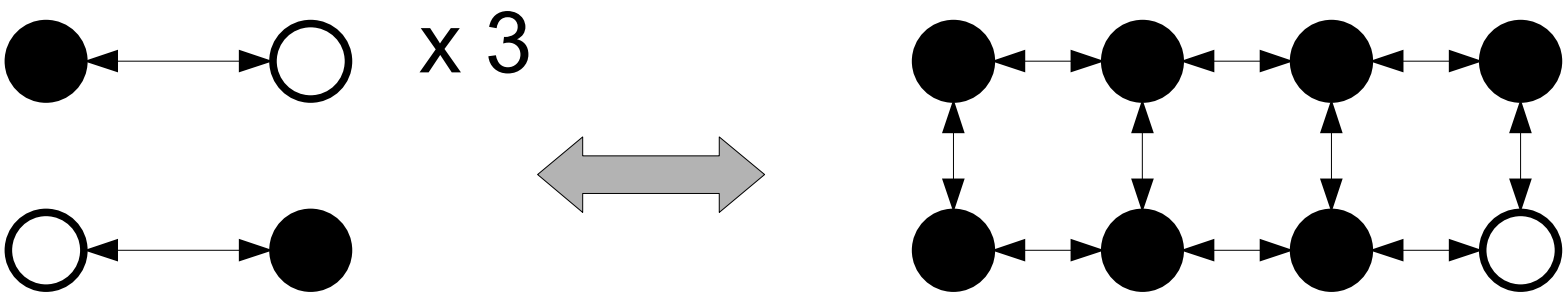
PSICS (parallel stochastic ion channel simulator)

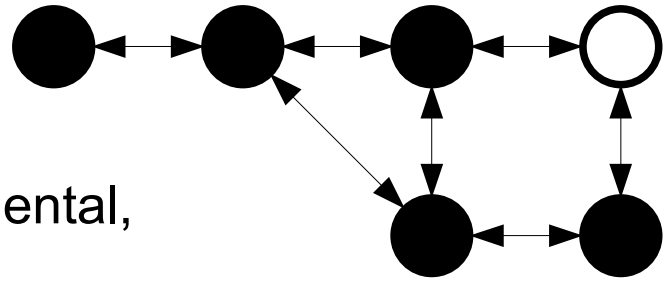
- Problem domain
- History and development process
- Algorithms
- Model specification and usage
- Comparison with other systems
- Icing – user interface for channel distributions

Ion channels are discrete entities, either open or closed
What if you model them this way?

HH



General
Kinetic schemes
More realistic, less temperamental,
different properties?



...spontaneous activity, stochastic resonance, propagation failures, “symmetry breaking”,...

Matt Nolan + Textensor Ltd. - discussions and estimates

BBSRC Tools and Resource application

Awarded

Work starts

No delays hiring people, no relocation or startup costs

Formal contract drawn up with Edinburgh University

Payments dependent on milestones being met

Calculations

Documentation

Examples

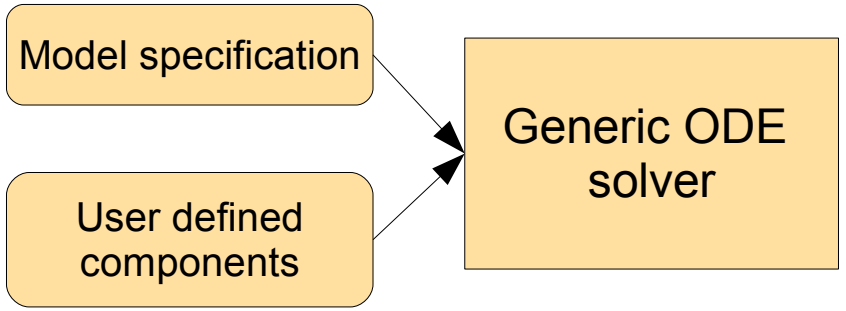
Even the boring bits get done because the contractor gets paid to do them.

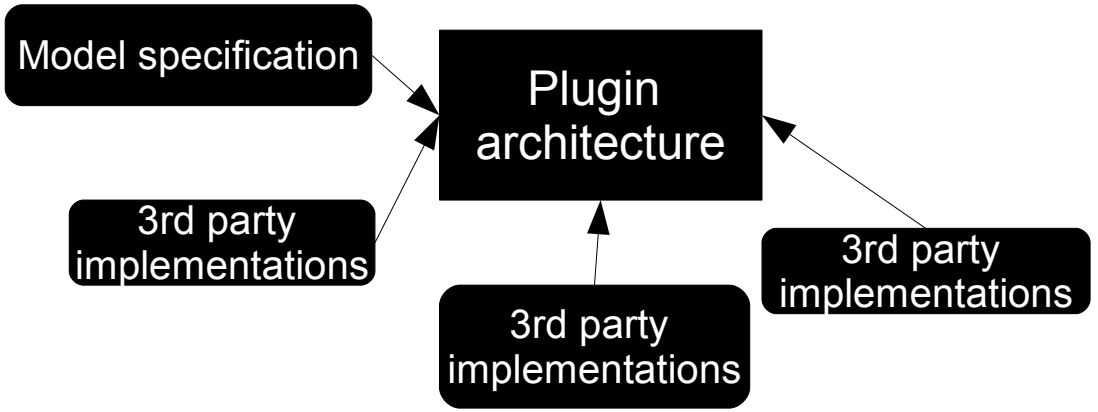
Validation, etc

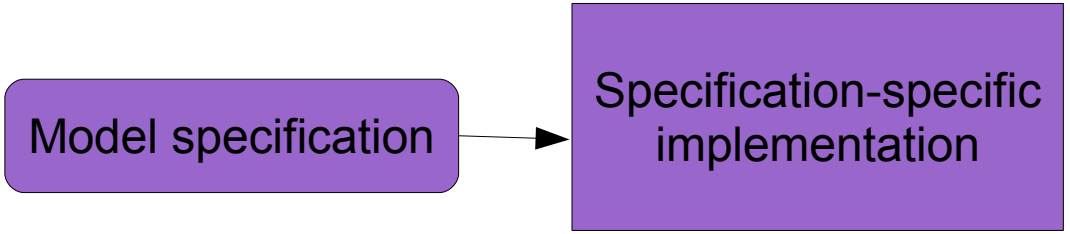
.....

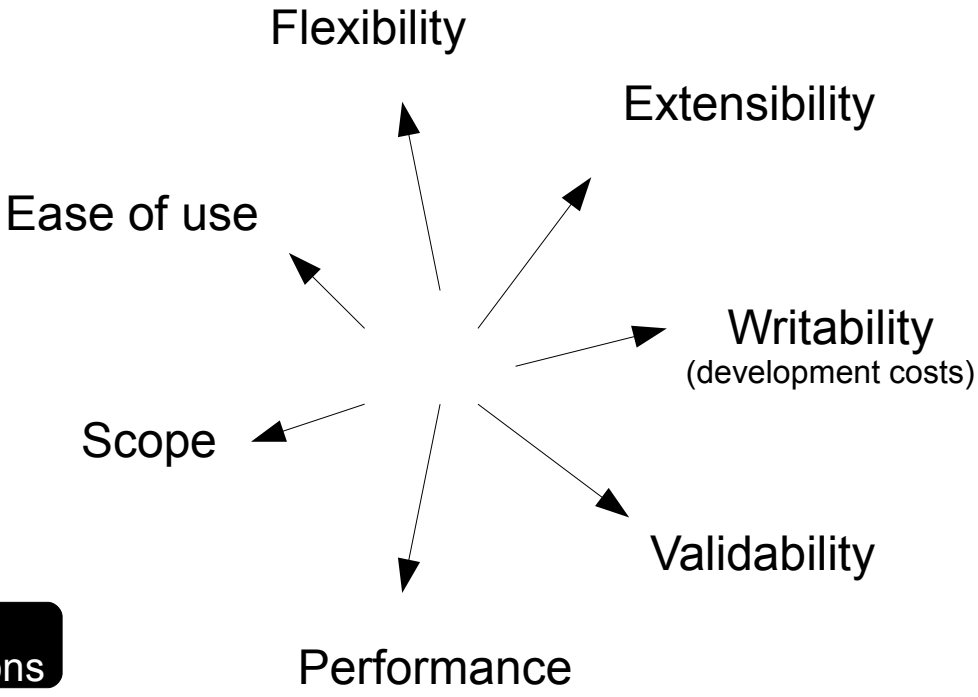
Completion

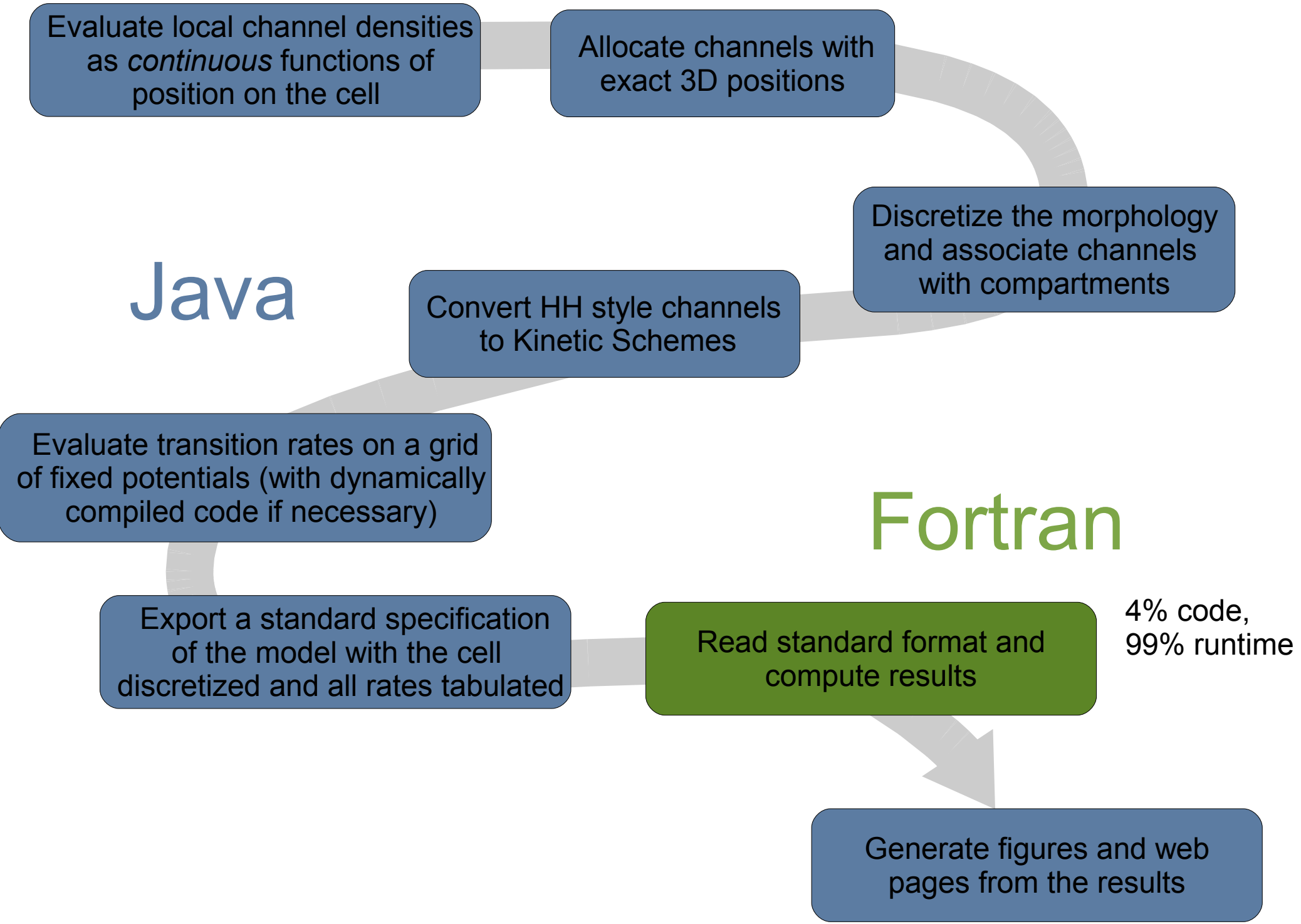
Ways to write a simulator







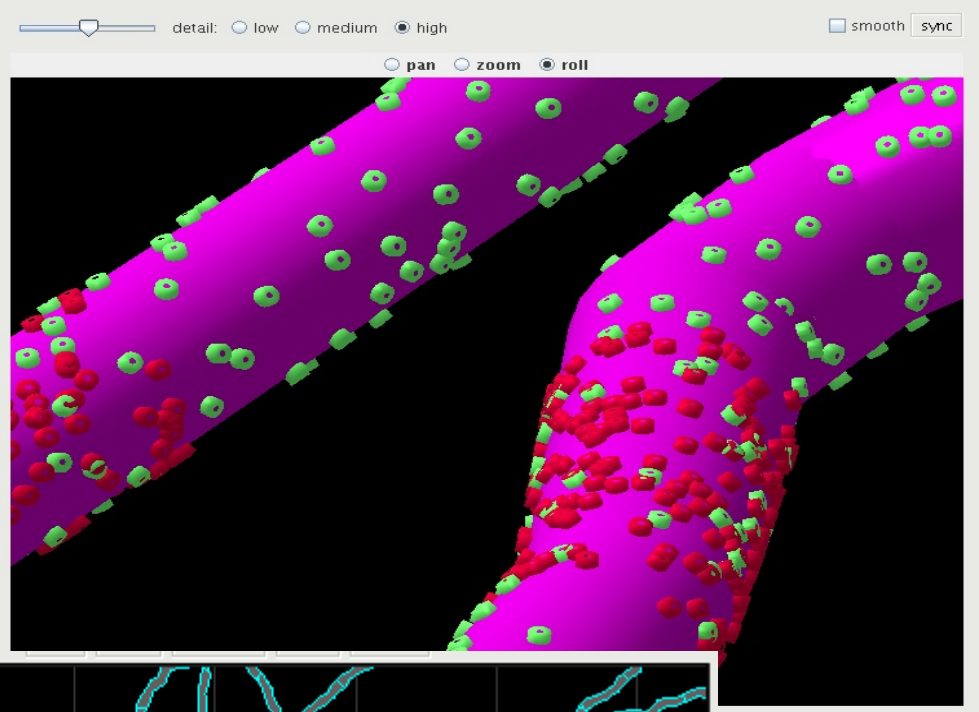


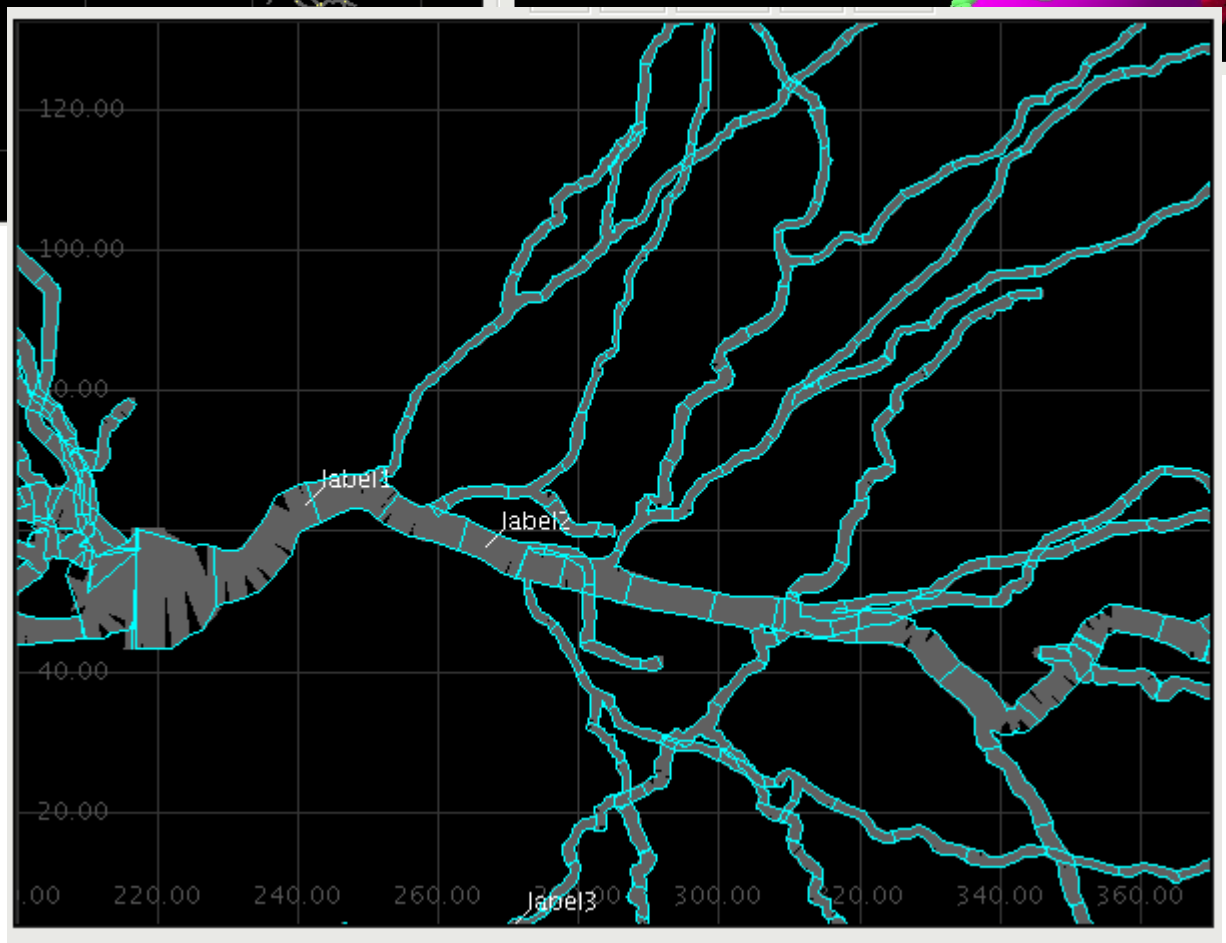

Evaluate local channel densities as *continuous* functions of position on the cell
Allocate channels with exact 3D positions
Discretize the morphology and associate channels with compartments
Java
Convert HH style channels to Kinetic Schemes
Evaluate transition rates on a grid of fixed potentials (with dynamically compiled code if necessary)
Fortran
Export a standard specification of the model with the cell discretized and all rates tabulated
Read standard format and compute results
4% code,
99% runtime
Generate figures and web pages from the results

Choice of algorithms - Objectives

- Model (channel numbers and positions, capacitance, distances etc) should be independent of spatial discretization and choice between stochastic and continuous calculation.
- Accuracy/efficiency trade-off
  - Channels on isopotential compartments are genuinely interchangeable
  - Is the RNG the dominant cost (it should be)
  - How few random numbers can you get away with?


200.00
100.00
0.0
-100.00
-200.00
100.00
200.00
300.00
400.00
label1
label2
label3


detail: low medium high
smooth sync
pan zoom roll


120.00
100.00
40.00
20.00
220.00
240.00
260.00
300.00
320.00
340.00
360.00
label1
label2
label3

Channel updates

At a given potential, the probability of occupying a given state, p_i evolves as

$$dp/dt = M\ p$$

so for a fixed interval δt,

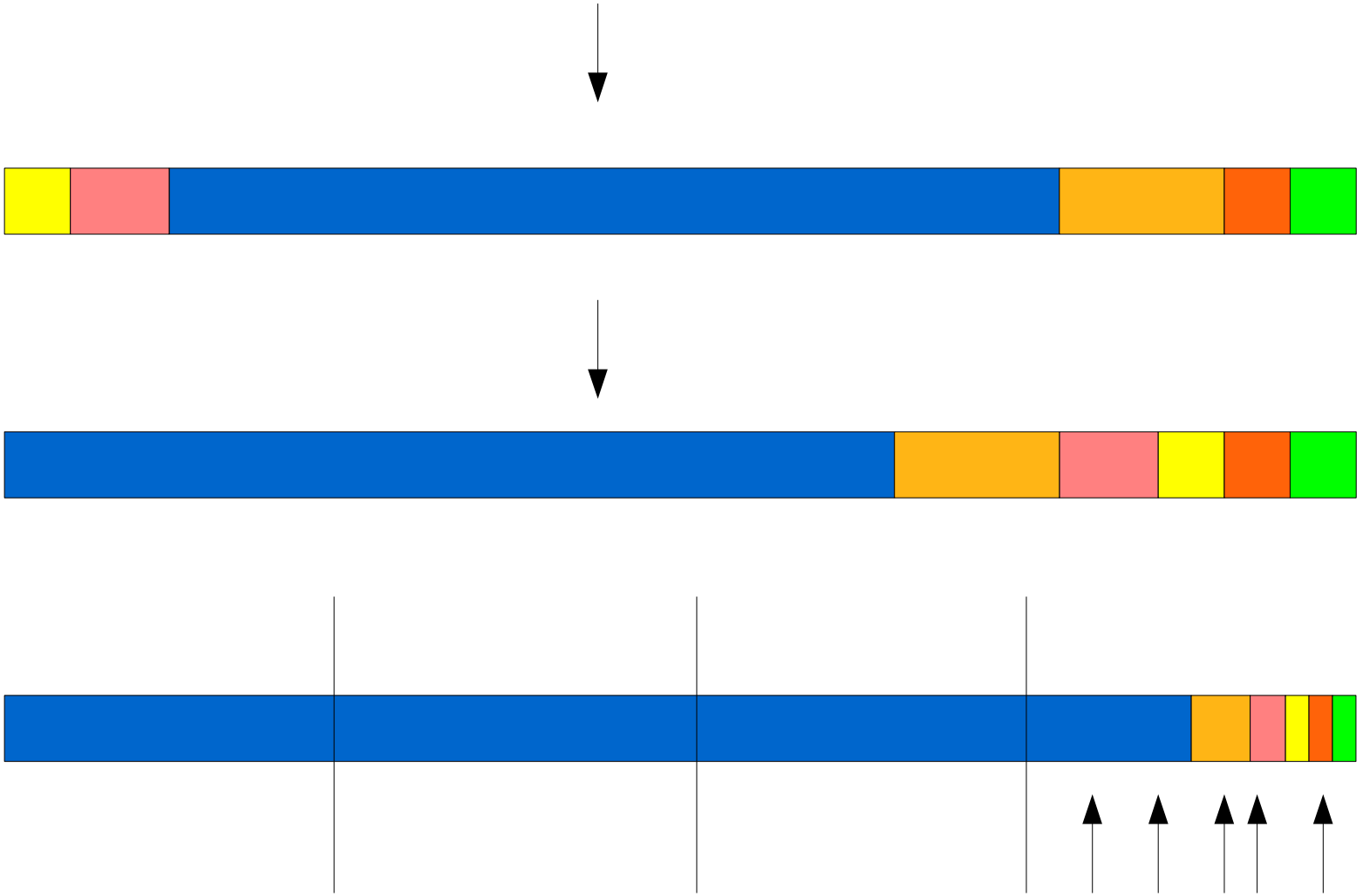
$$p_{t+\delta t} = e^{M\delta t}\ p$$

Given n channels in state p_i at time t, where are they at time $t + \delta t$?

Slow method: generate n uniform rvs, r_k, and sample row i of $e^{M\delta t}$ to generate n new states. Subtract elements from r_k until you reach zero – thats the selected state.

Better: sort row i of $e^{M\delta t}$ to put the largest element first (so most loops terminate at the first comparison)

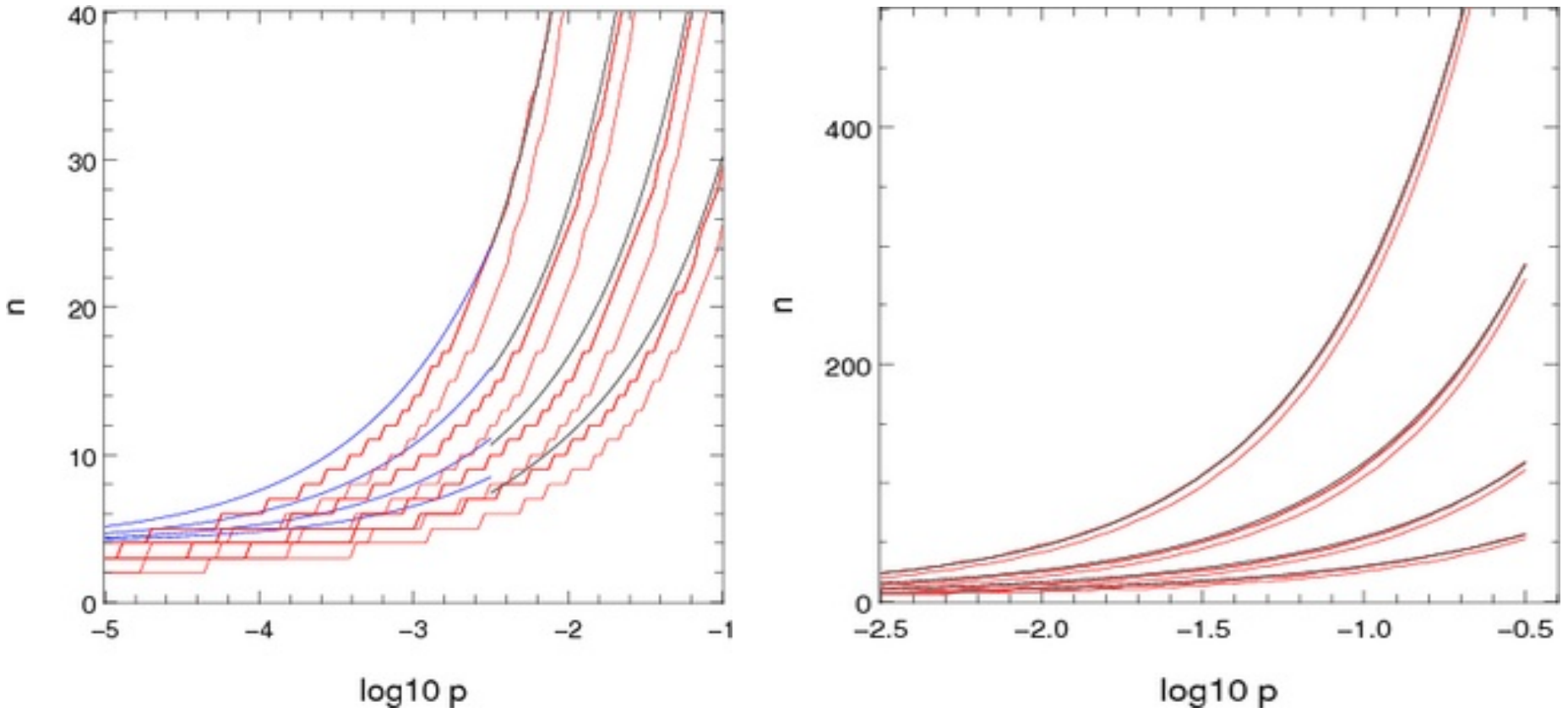
Better still: sort row i and then only sample m out of n channels $(m << n)$ restricted to the range $(1 - m/n, 1)$. ie, only consider those channels that have a non-negligible chance of changing state this step. *Under normal conditions, almost all channels are in the same state at the end of a step as they are at the beginning.*



Assume the last section gets its exact share of points and only generate these. Sometimes will underestimate the number falling in the colored area. Pick bounds to keep this rare enough to ignore.

$nc = 4 + 8 * \sqrt{n\ p}$ for $p < 0.003$

$nc = 6 + 5 * \log n + n\ p0.5 + 0.12 * \log n + 4 \log p$ for $0.003 < p < 0.15$



Approximations used for computing the number of channels to consider stochastically when there is one dominant destination. The x-axis shows log10 p where p is the probability of not ending up in the dominant location. The pairs or red lines show the points where the chance of n channels not ending up in the dominant destination are 10^{-5} and 10^{-7}. The blue and black lines show the two approximations presented above. The different sets of lines correspond to different population sizes: 2000 (top set), 700, 250 and 100 (bottom set).

Design issues

- Separate one-off preprocessing from core calculations
- Get as much as possible in the former: run user-defined code there and tabulate the results
- Core calculations work on standardized tables
- Fortran offers lots of advantages here:
  - Multi-dimensional arrays
  - Modules and data structures rather than objects and methods
  - Range of sizes for reals (useful to keep within the cache)
  - Compiler directives (non addressable arrays, fixed allocation)
  - Data-parallel statements (nice to write but slower on a single core)
  - Format based IO

Model Specification

- Collection of XML files – 100% declarative
  - Channel kinetics
  - Membrane properties (channel distribution)
  - Environment
  - Stimulation and recording
  - Morphology
  - Master file combining references to the others, control parameters, and output visualization

rallpack3/cell.xml

```
<CellMorphology id="cell">
        <Point id="p0" x="0" y="0" z="0" r="0.5" />
        <Point parent="p0" id="p1" x="1000" y="0" z="0" r="0.5" />
</CellMorphology>
```

rallpack3/environment.xml

```
<CellEnvironment id="environment">
        <Ion id="LEAK" name="Non-specific leak" reversalPotential="-65mV" />
        <Ion id="K" name="Potassium" reversalPotential="-77mV" />
        <Ion id="Na" name="Sodium" reversalPotential="50mV" />
</CellEnvironment>
```

rallpack3/membrane.xml

```
<CellProperties id="membrane" cytoplasmResistivity="100ohm_cm" membraneCapacitance="1uF_per_cm2">
        <ChannelPopulation channel="leak" density="25per_um2" />
        <ChannelPopulation channel="HH_Na" density="60per_um2" />
    <ChannelPopulation channel="HH_K" density="18per_um2" />
</CellProperties>
```

rallpack3/HH_K.xml

```
<KSChannel id="HH_K" permeantIon="K" gSingle="20pS">
        <KSComplex id="n" instances="4">
                <ClosedState id="c" />
                <OpenState id="o" />
                <ExpLinearTransition from="c" to="o" rate="0.1per_ms" midpoint="-55.mV" scale="10mV" />
                <ExpTransition from="o" to="c" rate="0.125per_ms" midpoint="-65.mV" scale="-80mV" />
        </KSComplex>
</KSChannel>
```

rallpack3/recording.xml

```
<Access id="recording">
        <CurrentClamp at="p0" lineColor="red" hold="0.1nA">
        </CurrentClamp>
    <VoltageRecorder at="p1" lineColor="blue" />
</Access>
```

rallpack3/HH_Na.xml

```
<KSChannel id="HH_Na" permeantIon="Na" gSingle="20pS">
        <KSComplex id="m" instances="3">
                <ClosedState id="c" />
                <OpenState id="o" />
                <ExpLinearTransition from="c" to="o" rate="1.per_ms" midpoint="-40.mV" scale="10mV" />
                <ExpTransition from="o" to="c" rate="4.per_ms" midpoint="-65.mV" scale="-18mV" />
        </KSComplex>
        <KSComplex id="h">
                <ClosedState id="c" />
                <OpenState id="o" />
                <ExpTransition from="c" to="o" rate="0.07per_ms" midpoint="-65.mV" scale="-20.mV" />
                <SigmoidTransition from="o" to="c" rate="1per_ms" midpoint="-35mV" scale="10mV" />
        </KSComplex>
</KSChannel>
```

rallpack3/run.xml

```
<PSICSRun timeStep="0.1ms" runTime="250ms" startPotential="-65mV" morphology="cell" environment="environment"
          properties="membrane" access="recording" stochThreshold="0">
        <StructureDiscretization baseElementSize="1um" />
        <info>Cable with HH sodium and potassium channels, different timesteps</info>
        <RunSet vary="timeStep" values="[10, 20, 50, 100]us" filepattern="ts-$" />
        <ViewConfig>
                <LineGraph width="500" height="400">
                <XAxis min="0" max="250" label="time / ms" />
                <YAxis min="-80" max="60" label="potential / mV" />
                <Line file="ref_axon_0_neuron.txt" color="white" width="1" rescale="[1000, 1000.]" />
                <Line file="ref_axon_x_neuron.txt" color="white" width="1" rescale="[1000, 1000.]" />
                <LineSet file="ts-10.txt" color="red" />
                <LineSet file="ts-20.txt" color="blue" />
                <LineSet file="ts-50.txt" color="cyan" />
            <LineSet file="ts-100.txt" color="green" />
                <View id="whole" xmin="-10." xmax="260." ymin="-100." ymax="80." />
                <View id="start" xmin="0." xmax="30." ymin="-100." ymax="80." />
                 <View id="end" xmin="210." xmax="255." ymin="-100." ymax="80." />
                </LineGraph>
        </ViewConfig>
</PSICSRun>
```

Units

- Almost all dimensional quantities require units
- Most commonly used unit choices are accepted

per_ms_per_M	per second per mole	L^3T^{-1}	$0.16605387280149467 * 10^{-29}$
per_ms_per_mM	per second per mole	L^3T^{-1}	$0.16605387280149467 * 10^{-32}$
l per s per mol	per second per mole	L^3T^{-1}	$0.16605387280149467 * 10^{-26}$
velSI	velocity	LT^{-1}	
accSI	acceleration	LT^{-2}	
N	newton	MLT^{-2}	
J	joule	ML^2T^{-2}	
A	ampere	A	
mA	milliampere	A	10^{-3}
uA	microampere	A	10^{-6}
nA	nanoampere	A	10^{-9}
pA	picoampere	A	10^{-12}
C	coulomb	TA	
V	volt	$ML^2T^{-3}A^{-1}$	
mV	millivolt	$ML^2T^{-3}A^{-1}$	10^{-3}
KV	Kilovolt	$ML^2T^{-3}A^{-1}$	10^3
MV	Megavolt	$ML^2T^{-3}A^{-1}$	10^6
per_mV	per millivolt	$M^{-1}L^{-2}T^3A$	10^3
ohm	ohm	$ML^2T^{-3}A^{-2}$	

(Embarrassing) Parallelization

- Most model specifications involve multiple runs
  - multiple realizations with different seed
  - Parameter sweeps
- Each run is defined in a single file for the Fortran core calculations
- In a parallel environment (GridEngine) these are automatically submitted as separate jobs
- Otherwise, they are run sequentially
- Final plots and documentation are submitted as a job that depends on completion of the others

Comparisons

PSICS	Others
No interpreter – declarative only	Interpreters – Hoc, Python etc
Efficient for stochastic channels	Neuron has Gillespie (exact but slow) method
Units supplied with quantities	Implicit unit sets or user's responsibility
Irregular shaped compartments	Cell gets straightened
User extensions only at preprocessing stage – important?	Coded user extensions
Command line only	GUIs
Partial input of NeuroML	Minimal if any?
No synapses, calcium dynamics or networks yet	

ICING (interactive channel insertion gui)

- Facilitate creation of membrane properties XML files
- 1-1 correspondence with model specification schema
- Generates XML that is still hand-editable
- Visualization of the effects of particular parameter choices

Model components

- /tmp/icing30674.tmp
  - KSChannel
  - CellProperties
    - membrane
  - SWC Morphology
    - C050800E2.CNG
    - NM1
  - CellMorphology
    - cell3d
    - cell

Ready

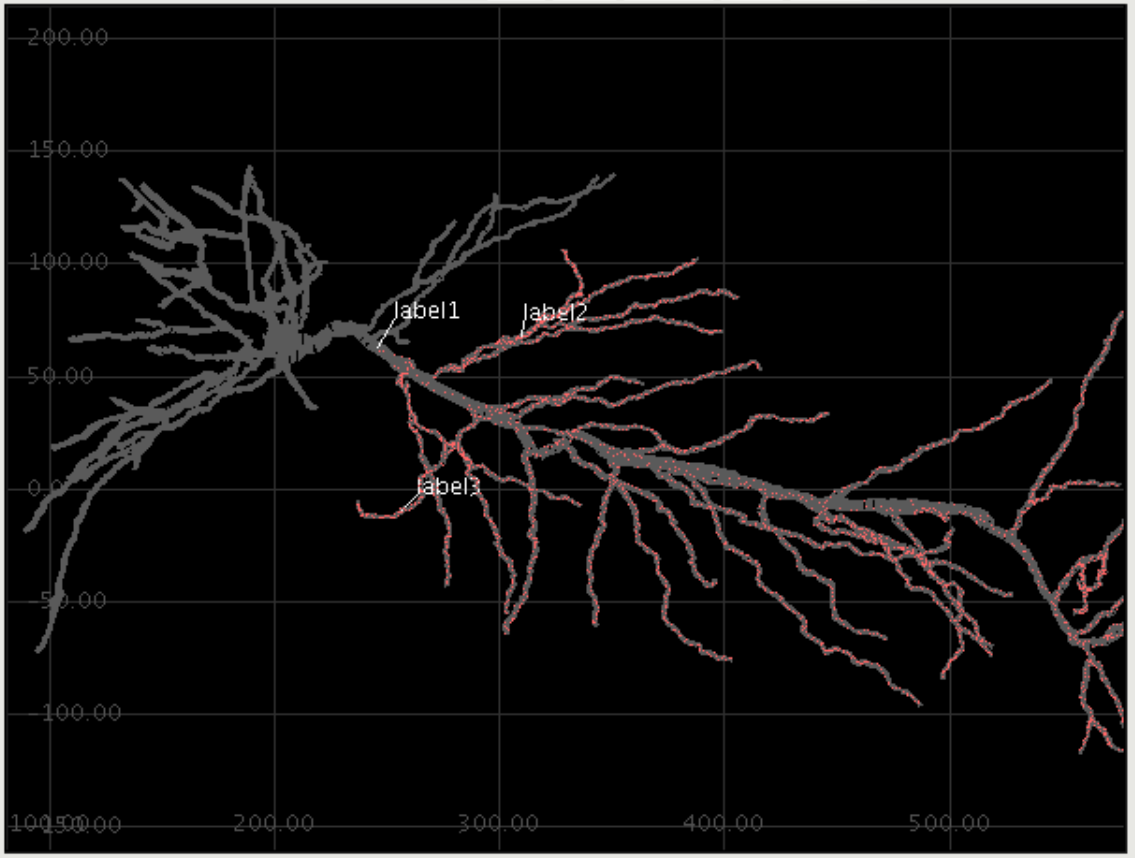
mem: 63 / 905 Mb

Clean up

Morphology

Display style: Filled | discretization at 10 um^(3/2) Change | sections: tapered

Pan | Zoom | Roll | R- | R+ | Reset | 3D | sync | Smooth



Point labels

Show all | Hide | New

label1
label2
label3

label: label2 | Delete

Labelled structures

show colors on cell

1
3
4

3

no active point

738 compartments

97066 channels

Channel populations

- population_1
- population_2
- population_3

Up | Down | Add | Delete

Population properties

population_1 | Rename

Channel type: none

Channel density | Relative

3 per um2

x none

Distribution: Regular | Reseed

Fix total: 0 channels

Cap density: 0.000 per um2

Update channel positions

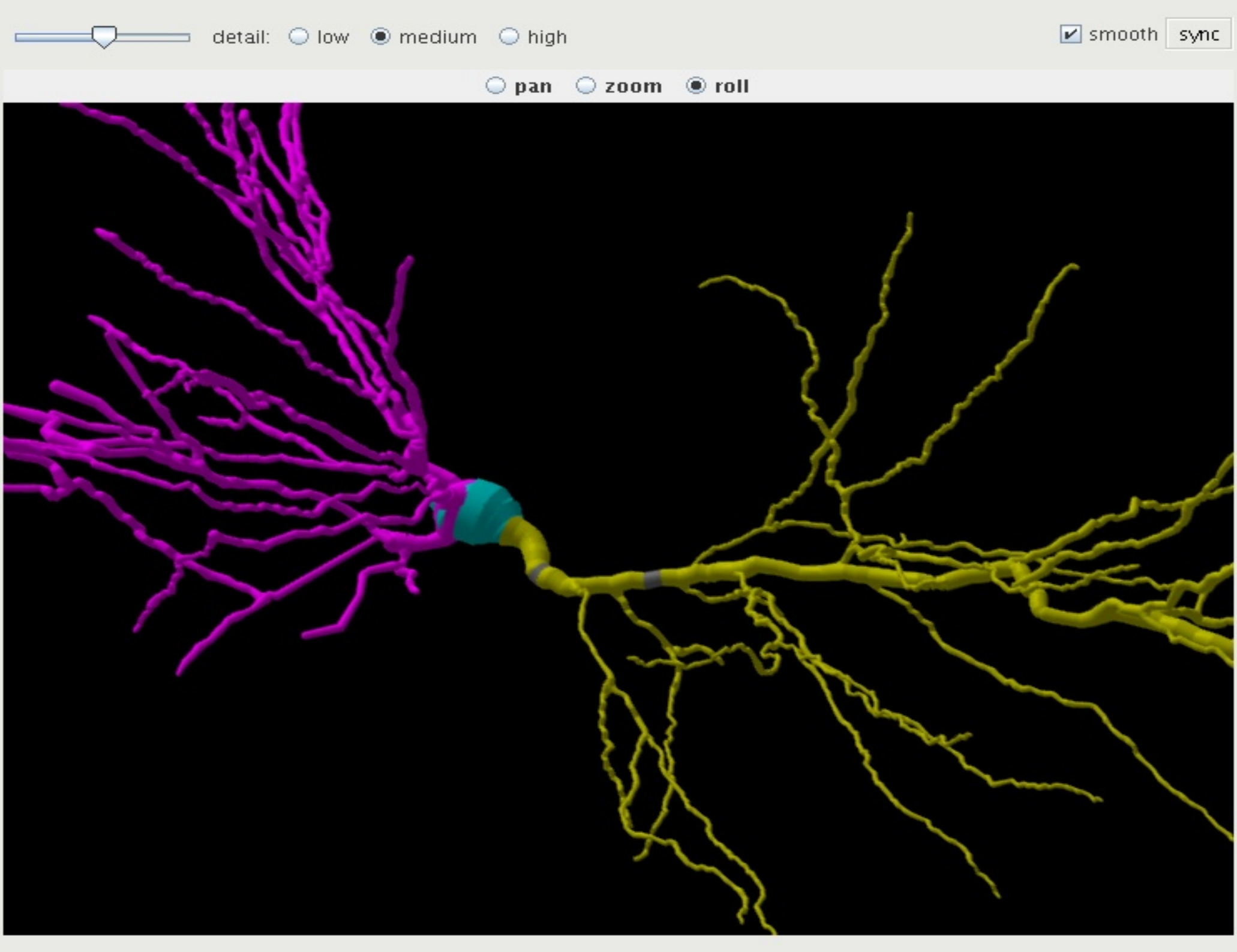
Region constraints

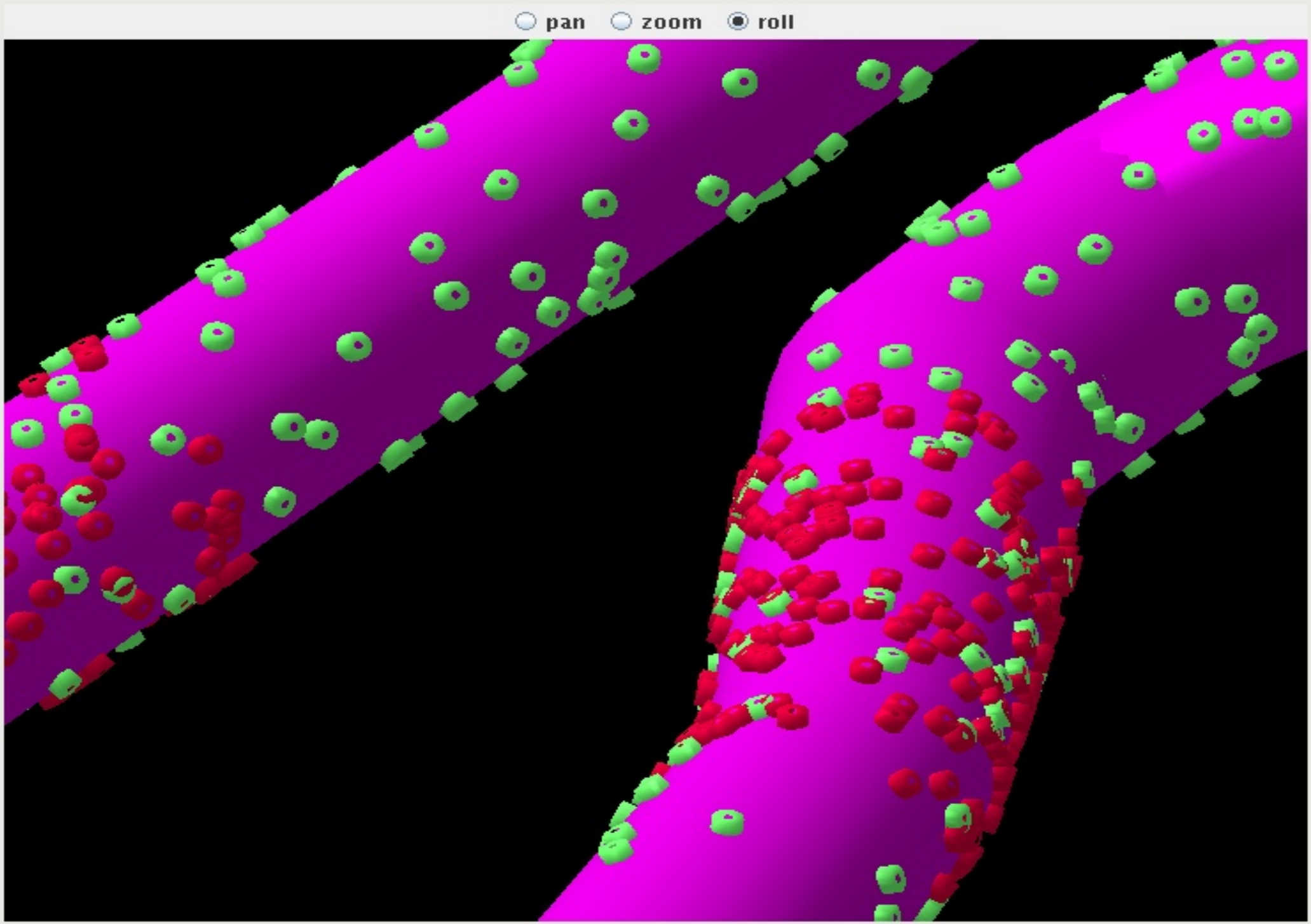
Conditions are applied in order defaulting to the who...

restrict to points where region>label1 | X

Restrict to | region distal to | label1

Clear | Apply


detail: low medium high
smooth
sync
pan zoom roll


detail: low medium high
smooth sync
pan zoom roll