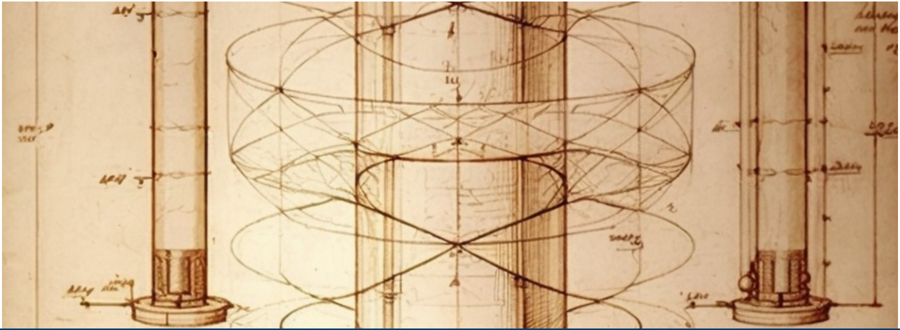


ARBOR, NEUROML2,
AND THE QUEST FOR WARP SPEED

March 2023 | T. Hater, L. Landsmeer, and the Arborists | Forschungszentrum Jülich

JÜLICH Forschungszentrum
JÜLICH SUPERCOMPUTING CENTRE

What is NeuroML2?

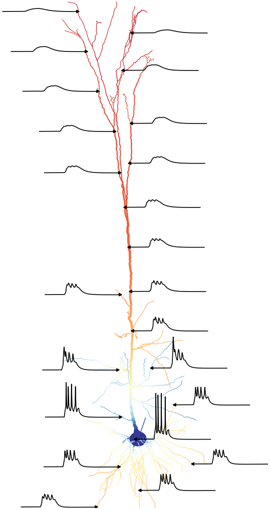
```xml
<ComponentType name="gate" />

<ComponentType name="hh_exp_gate" extends="gate" />
<ComponentType name="hh_lin_gate" extends="gate" />

<ComponentType name="ion_channel">
  <Children name="gates" type="gate"/>
</ComponentType>
```

- Declarative specification of a full simulation.
- A library of components written in LEMS, elaborating dynamics in a OO-like style.

- Morphology
- Ion channels
- Cell Parameters
- Connectivity
- Network Parameters

Objectives

- Unlock NML models for Arbor users.
- Keep the core tenets of NML
  - Extensibility: Can add components and types at any time.
  - Flexility: Components can be combined freely.
  - Encapsulation: No need to unterstand lower levels details.
- Get the most performance we feasibly can.
- Enter stage left: `nmlcc`



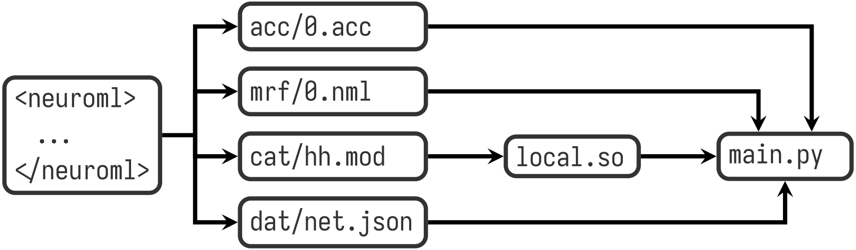
```xml
<cell id="L5PC">
  <biophysicalProperties id="biophys">
    <membraneProperties>
      <channelDensity id="basal" ionChannel="Ih" />
    <membraneProperties>
  </biophysicalProperties>
</cell>
```

Dynamic Compilation

Issue

New components – especially ion channels – may appear at ay time. Ion channels may be composed dynamically from gates.

- Thus, we cannot leverage statically built components.
- Our solution must be to translate NML on demand.
- We then add a simple driver script to read these and run a simulation.



Optimisation

The boring bits

- Ion channels are – usually! – the main performance bottleneck.
- NMODL is hard to get fast and harder to get correct *and* fast.
- NeuroML2 on the other hand is not concerned with performance.
- We take the lessons learned from optimising NMODL by hand and encode them into our compiler.
  - Removing `RANGE` and `ASSIGNED`.
  - Preferring computation over data movement.
  - Inlining `FUNCTION` and `PROCEDURE`.
  - Many 'standard' optimisations: folding and inlining constants, eliminating dead code, propagating constants …

Boring, yet…

This gives roughly $2\times$ the performance of the NML2 reference implementation.

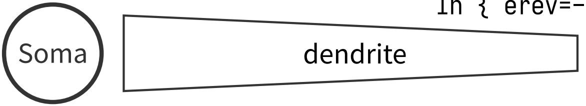
JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

Optimisation

Leveraging NeuroML2

```
Ih { erev=-65 g=0.3 }
```

```
Ih { erev=-45 g=0.2 }
```




```
<channelDensity segmentGroup="soma" condDensity="0.3" ionChannel="Ih" erev="-65">
<channelDensity segmentGroup="dend" condDensity="0.2" ionChannel="Ih" erev="-45">
```

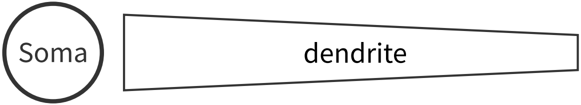
- We can generate one NMODL *per region* and *channel*.
- Then, we can make all parameters into constants…
- …and rerun our optimisation machinery.
- Saves memory traffic and uncovers more optimisations.

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

Optimisation

Leveraging NeuroML2

```
Ih { erev=-35 g=0.2 }
```




```
pas { erev=-90 g=0.4 }
```

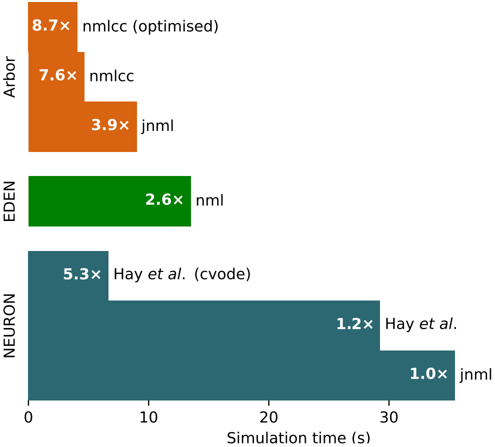
```
<channelDensity condDensity="0.2" ionChannel="Ih" segmentGroup="soma" erev="-45"/>
<channelDensity condDensity="0.4" ionChannel="pas" segmentGroup="soma" erev="-90"/>
```

- We can generate one NMODL *per region*.
- Saves calling into different channels and enables sharing.

Super-Mechanisms

These two optimisations combined yield a single bespoke NMODL file per region without any parameters. Performance gain is typically $20 - 30\%$

Results



- Layer 5 pyramidal cell by Hay et al.
- NML2 by P. Gleeson et al.
- 15 channels including non-uniform parameters.
- Translated via `nmlcc bundle` and run unaltered.
- 'optimised' uses super-mechanisms in addtion.
- Measured the simulation time excluding setup.

Support

- We strive to support *all* of NML2 that is feasible in Arbor.
- Currently the follwing features are enabled:
  - Morphologies and (non-uniform) parametrisations
  - Ion channels, concentration models and synapses.
  - Networks and their parameters.
- Support is missing For
  - Spike source cells (easy, but time constraints).
  - Artificial cells (hard, but doable via voltage processes).
- Expanding on super-mechanisms
  - Re-write regions s.t. each segment maps to exactly one region.
  - Apply similar transformations to synapses.

Conclusions

- Scientists: Using NeuroML2 becomes two calls on the shell.
  - Performance is effortless and competitive with best-in-class tools.
  - Unlocks validating simulations across simulators.
  - Makes your models portable, without expertise in Arbor.
  - Enables use of the NeuroML2 database.
- Developers: NeuroML2 drives fatures and incubates optimisations.
- Community: Lower barrier to entry, more users.
- NeuroML2: More implementations drive utilisation, feature development, and correctness.

F·I·N