Introducing a New Method for Simulating Kinetic Models

- Faster & More Accurate than NEURON's sparse solver
- For models which are Linear & Time-Invariant
  - Markov Models of Ion Channels
- Proof of concept implemented for NMODL files

Presented by David McDougall, 2022

What is a Linear & Time-Invariant Differential Equation?

Problem: $dx/dt = A * x$

- Where X is a variable
- Where A is a constant

Solution: $x_{t+dt} = x_t * \exp(A * dt)$

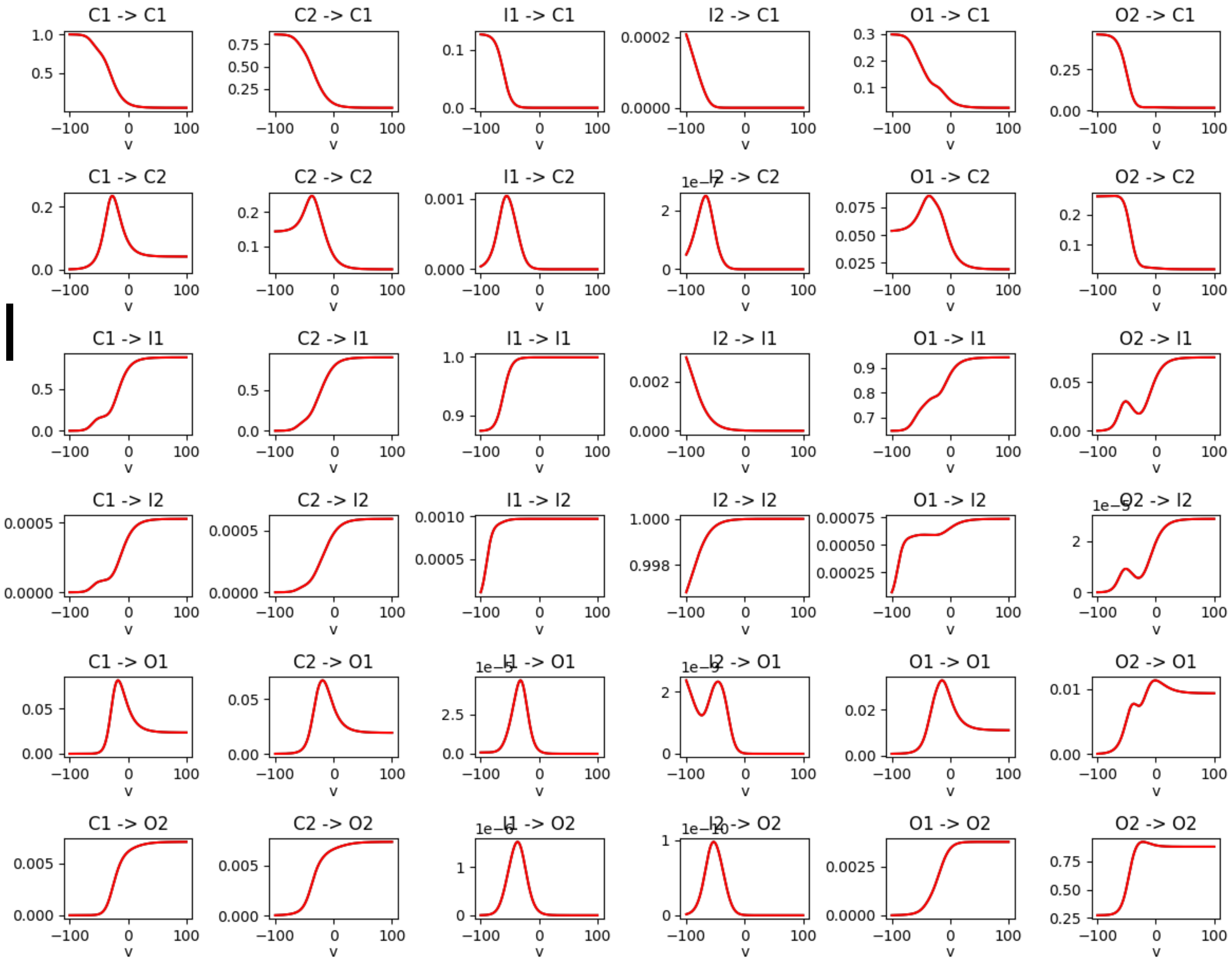
- But what if X is a vector and A is a matrix?

  The exponential function is defined for matrices

“A” is a function of the input

- Assume the input is constant for the duration of each time step

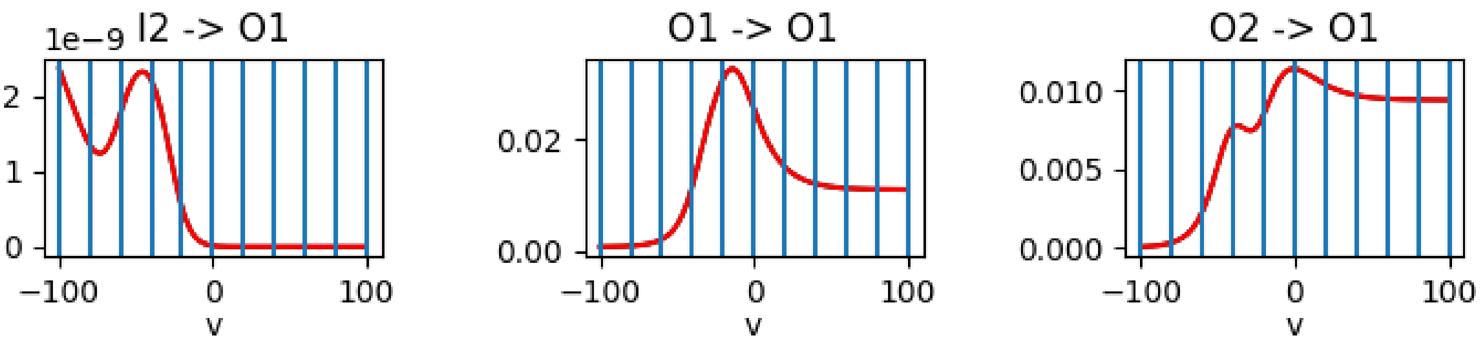
Example: Kinetic Model of Nav1.1 with 6 states



These plots shows the fraction of each state that transitions to each other state, as a function of voltage, over the course of one time step.

How to approximate the matrix?

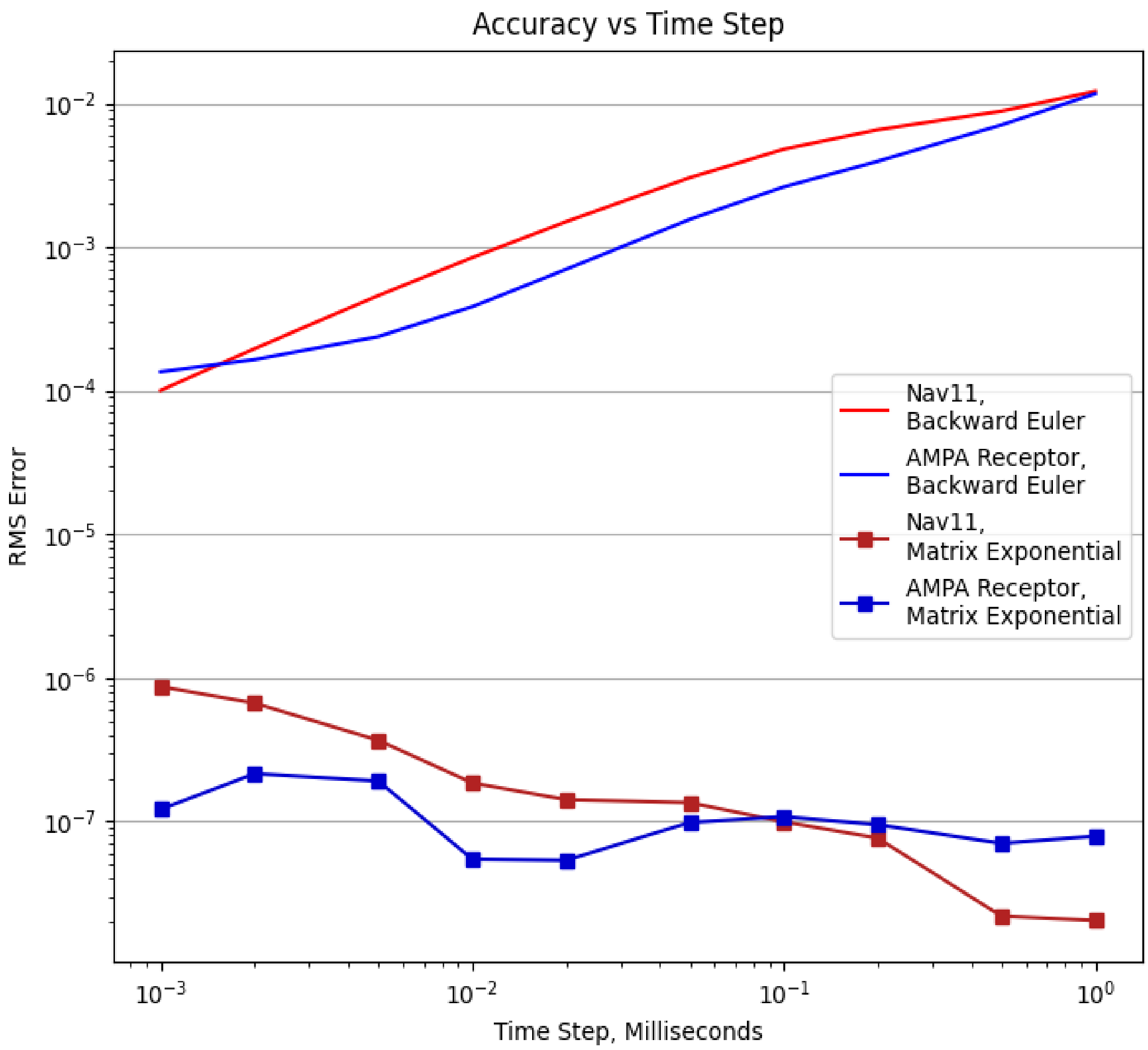
- Divide the input space into small pieces
- Fit a polynomial approximation to each piece



Accurate

- The matrix exponential is an exact solution
- Its approxiation is the only source of error
- Automatically measure and control the error by:
  - Increasing the number of input bins
  - Increasing the order of the polynomials

For this figure, the maximum approximation error was 10^-6

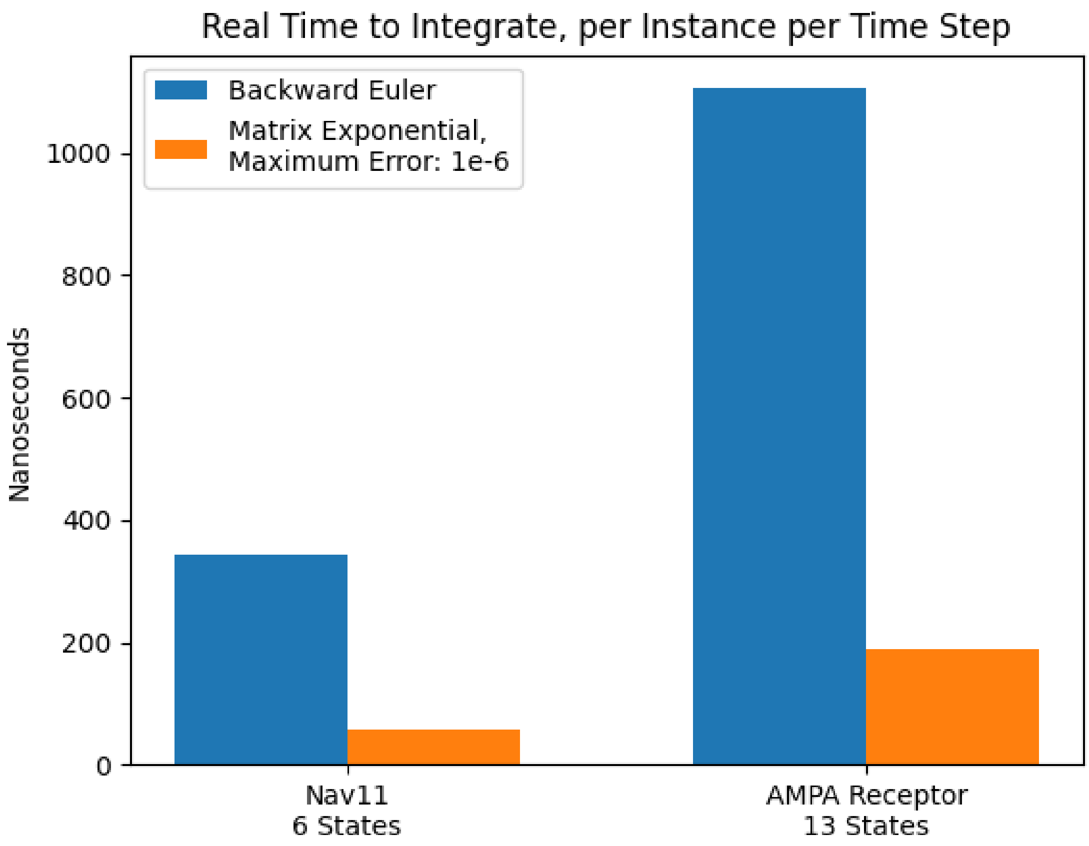


Fast

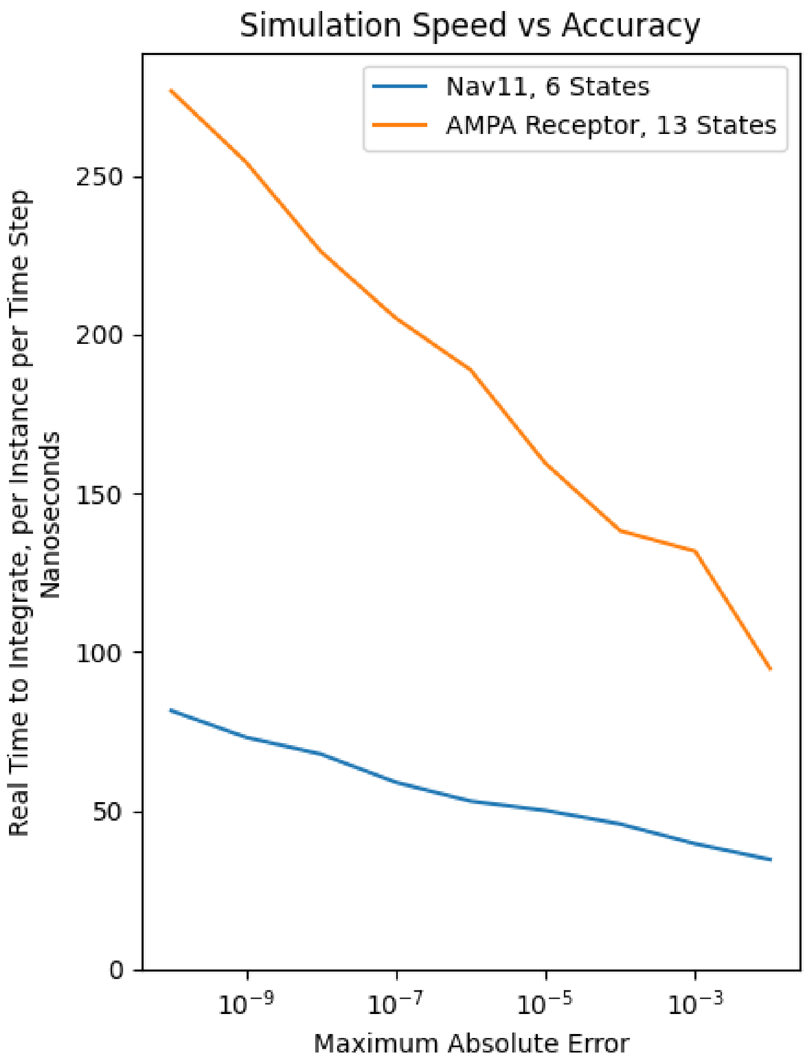
Loading the approximation from RAM is slow

Use CPU cache and process large batch

- Approx must fit in cache
- This figure measured 10,000 instances



Speed vs Accuracy



Pros & Cons

Pros

- Fast
- Accurate

Cons

- Only 1 or maybe 2 inputs allowed
- Variables become constant
  - Temperature
  - Time Step
  - Parameters
  - ASSIGNED block
- Complicated to implement
- Slow startup

- Reference:

  Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. Rotter S, Diesmann M (1999). https://doi.org/10.1007/s004220050570

- Source code:

  https://github.com/ctrl-z-9000-times/lti_sim

  MIT License