Parallel Computing of Action Potentials in the Hodgkin-Huxley Model via the Parareal Algorithm

Eric Boerman, Khanh Pham, Katie Peltier
Virtual Research and Creative Activity Day
West Chester University
April 29th, 2021
The Action Potential

Above: Numerical approximation of an action potential in the Hodgkin-Huxley model

Below: Numerical solution as reported by Hodgkin and Huxley in 1952

Source: A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve, Hodgkin & Huxley, 1952
The Cell Membrane

Cell Exterior
Higher steady-state concentration of Na⁺

Cell Interior
Higher steady-state concentration of K⁺

The Cell Membrane

The Hodgkin-Huxley Model

Ohm’s Law – current equals voltage times conductance

Total current is the sum of all component currents:

\[ I = I_1 + I_2 + \cdots + I_n \]

For each ionic current, \( I_{\text{ion}} = \text{conductance} \times (V_{\text{equilibrium}}) \times \text{distance from voltage equilibrium} \):

\[ I = g_K(V - V_K) + g_Na(V - V_{Na}) + g_l(V - V_l) + C_m \frac{dv}{dt} \]

where \( C_m \frac{dv}{dt} \) is the current from the membrane’s function as a capacitor
The Hodgkin-Huxley Model

Conductance for Na\(^+\) and K\(^+\) (\(g_{\text{Na}}\) and \(g_{\text{K}}\)) are gated by voltage

n, m, and h are proportions (0 ≤ n, m, h ≤ 1) that vary with voltage and define gate activation or inactivation

\(\bar{g}_{\text{Na}}\) and \(\bar{g}_{\text{K}}\) are the maximum possible conductances for a given set of parameters

\[ g_{\text{K}} = \bar{g}_{\text{Na}} n^4 \]
\[ g_{\text{Na}} = \bar{g}_{\text{K}} m^3 h \]

\(g_i\) does not meaningfully vary with voltage, and is treated as constant
The Hodgkin-Huxley Model

\[
\frac{dv}{dt} = \frac{(I - \bar{g}_Kn^4(V - V_k) - \bar{g}_Na m^3 h(V - V_{Na}) - g_l(V - V_l))}{Cm}
\]

\[
\frac{dm}{dt} = \alpha_m (1 - m) - \beta_m m
\]

\[
\frac{dn}{dt} = \alpha_n (1 - n) - \beta_n n
\]

\[
\frac{dh}{dt} = \alpha_h (1 - h) - \beta_h h
\]

<table>
<thead>
<tr>
<th>i</th>
<th>$\alpha_i$</th>
<th>$\beta_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>$\frac{2.5 + 0.1V}{e^{2.5+0.1V} - 1}$</td>
<td>$\frac{V}{4e^{110}}$</td>
</tr>
<tr>
<td>n</td>
<td>$\frac{0.01V + 0.1}{e^{0.1V + 1} - 1}$</td>
<td>$0.125e^{\frac{V}{410}}$</td>
</tr>
<tr>
<td>h</td>
<td>$0.07e^{\frac{V}{25}}$</td>
<td>$\frac{1}{1 + e^{3+0.1V}}$</td>
</tr>
</tbody>
</table>
Numerical Methods

Forward Euler Method
- Fastest numerical method
- Relatively inaccurate: 1st-order accuracy

4th-Order Runge-Kutta Method
- Increased accuracy given same parameters
- Computationally more expensive

For both methods, V, n, m, and h are solved for simultaneously within each step.
Experimental Results

Forward Euler Positive Threshold

- $V(0) = 6.60$
- $V(0) = 6.61$
Experimental Results

Anode Break Inhibition

![Graph showing voltage (mV) over time (ms)]
Experimental Results

Constant Applied Current, $3\mu A$

![Graph showing voltage over time with repetitive peaks at 20, 40, 60, and 80 ms.]
Parareal

- A unique parallel-in-time algorithm, developed by Lions, Meday, and Turinici in 2001
- Utilizes two temporal discretizations – one coarse, one fine – and solves them numerically
- Predicts reasonable starting values, then calculates fine mesh values in parallel
- Converges to a solution over multiple iterations
- Does not increase accuracy over sequential method, but can offer significant time savings
Preliminary estimations for parallelization in a 48-CPU system suggest a significant possible decrease in computational time. At 47 iterations, time savings is negative compared to sequential calculations, but the Parareal algorithm finishes well before then, even for tolerances within $1/100,000,000^{th}$ of a millivolt.

At increased CPU counts (100, 200, etc.), iteration count seems to fall around ~5% of maximum at this tolerance level.

While computational overhead limits maximum possible time savings, preliminary results suggest that for most real-world scenarios increasing the CPU count will increase efficiency.

<table>
<thead>
<tr>
<th>Tolerance (mV)</th>
<th>Iterations (47 max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-5}$</td>
<td>3</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>4</td>
</tr>
<tr>
<td>$10^{-7}$</td>
<td>4</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>5</td>
</tr>
</tbody>
</table>
References


https://www.youtube.com/watch?v=oa6rvUJlg7o

Questions?