Numerical Approximation of Neuronal Action Potentials

MATH 478 - Computational Methods in Applied Mathematics

Theo Fabi

Abstract

This paper outlines a numerical method to solve the Hodgkin-Huxley partial differential equations which model the propagation of action potentials in neural tissue. A stable, implicit finite difference method is utilized in combination with an adaptive time stepping algorithm that both enforces the convergence of the iterative non-linear solution procedure and increases computation speed when appropriate. The method is implemented in C++ and exposes its functionality to Python with the Python C API.

1 Introduction

The most widely used system of equations in modelling the dynamics of neuron electrophysiology is undoubtedly the *Hodgkin-Huxley* system of equations [1]. This mathematical model, for which its architects received the *Nobel Prize in Physiology or Medicine* in 1963, describes the propagation of electrical impulses in neurons. Such impulses have a crucial role in the flow of information throughout the nervous system[6], and as such are relevant in many areas of neuroscientific research, such as in computational neuroscience models and software [18, 12]. Their use is, however, not limited to the study of the nervous system, and can be extended to accurately model different kinds of biological excitable tissues[2].

The Hodgkin-Huxley equations yield a highly nonlinear dynamical system, and are often treated as a set of *ordinary* differential equations. However, this treatment requires either the assumption of voltage uniformity throughout the spatial domain[1], resulting in a solely time-dependent phenomenon, or the restriction to *steady propagation* solutions [1]. Without such assumptions, the system is dependent on both space and time and yields a set of *partial* differential equations.

Given its nature, solutions to the system are found numerically in the general case. A variety of schemes have been used to these ends. In this paper, only finite difference schemes or slight modifications thereof shall be considered, but it is worth noting that methods of higher sophistication have been employed with success. A common discretization scheme used in the field is that of Crank-Nicolson given its numerical accuracy and stability. Notably, the NEURON simulation environment, a set of software tools used in neuroscientific simulations and research to model neurons and neuronal networks[12], defaults to a Crank-Nicolson discretization scheme to numerically integrate action potentials [10]. Numerous other papers present the results of this method applied to the Hodgkin-Huxley partial differential equations [3, 4, 7, 16, 17]. The backward Euler method has also shown to be favourable in modelling these equations[5, 7, 8], whereas classical explicit time stepping is usually considered too unstable for most applications[17, 15]. As such, implicit methods are most commonly used[15], necessitating the solution of algebraic equations. For nonlinear systems such as those of the Hodgkin-Huxley model, solving for the implicit terms increases in complexity and usually requires iterative means[3, 5, 16, 8] when no explicit terms are available.

In this paper, a fully implicit[2] backward Euler scheme similar to that developed by Mascagani[5] is described. Its C++ implementation, along with a Python API wrapper, is available at this github repository. An adaptive time stepping

[1] Spatial uniformity is obtained either by simply modelling the voltage dynamics over time at a specific point along the spatial domain, or by applying a *voltage clamp* to enforce $\frac{\partial V}{\partial x} = 0$. The latter technique was used in early investigations of action potentials by Hodgkin and Huxley[1].

[2] The distinction between a *fully* implicit and a semi-implicit scheme will be made clear further in this document.

procedure is introduced so as to both ensure the convergence of the iterative nonlinear solver and to speed up computations when suitable. This method of time discretization was chosen over the Crank-Nicolson method because only the former has been formally proven to converge[3][5, 7]. Further, the argument of stability proposed in the literature for the latter scheme requires a specific set of boundary conditions that permit Von-Neumann stability analysis[10, 17] for the nonlinear system. The scheme employed herein uses boundary conditions that have a stronger biological basis but its stability cannot be analyzed with Fourier methods.

2 Theory

2.1 Biological preliminaries [4]

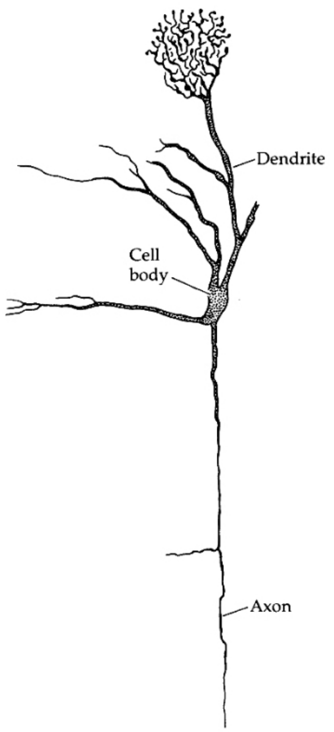

Figure 1: Representation of the structure of a neuron [13]

Neurons are cells that transmit information via electrical signals called *action potentials.* It receives input at the ends of *dendrites* which carry signals to the cell body, the *soma*, which contains the nucleus, and propagates down the *axon*, a long, cable-like branch that eventually branches out to communicate with other cells. The nature of the transmitted signal is a voltage difference between the inside and the outside of the cell, which is referred to as the *membrane potential* in that the cell's *membrane* is what separates its interior and exterior. The *resting potential* of a cell is the voltage difference across its membrane as a result of the concentration of charged ions on each side. For example, the resting potential of a neuron is of -70 mV, which means that there are more positively charged ions on the outside of the cell at resting conditions. When the membrane's potential is *depolarized* beyond a certain threshold, the potential spikes in a disproportionate manner, forming the impulse that we call the action potential.

Ions can travel in and out of the cell via *ion channels*, of which there are many kinds. These

[3] As noted by Mascagani[7], it is believed that a proof can be obtained for the convergence of the Crank-Nicolson scheme. Also note that recently, a similar proof was published for an explicit forward Euler scheme[11].

[4] Citations are omitted in this section given that it consists only of basic, introductory biology. All of this information was taken from two textbooks [13, 14].

channels are proteins that exist within the membrane and allow the transport of specific substances such as ions to and from the cell's environment, and may or may not be *gated*. Of particular importance with respect to action potentials are *voltage gated* ion channels, which, as suggested by their name, are ion channels that open and close in response to the membrane potential. Thus, they *influence* the membrane voltage by letting charged ions travel across the membrane, and *are influenced by* the membrane potential because the extent to which they allow the transportation of ions is controlled by the voltage difference. Therein lies the mechanism of the all-or-nothing nature of action potentials: depolarization beyond the threshold potential gives rise to a *positive feedback loop*.

In this paper, the phenomenon of action potential *propagation* is studied. The Hodgkin-Huxley partial differential equations model the transmission of these impulses throughout a single neuron, and for this reason only intra-cellular transmission will be considered. When a neuron receives an electrical input sufficient to initiate an action potential, it is generally true that the resulting spike travels along the length of the neuron in a "travelling-wave" fashion.

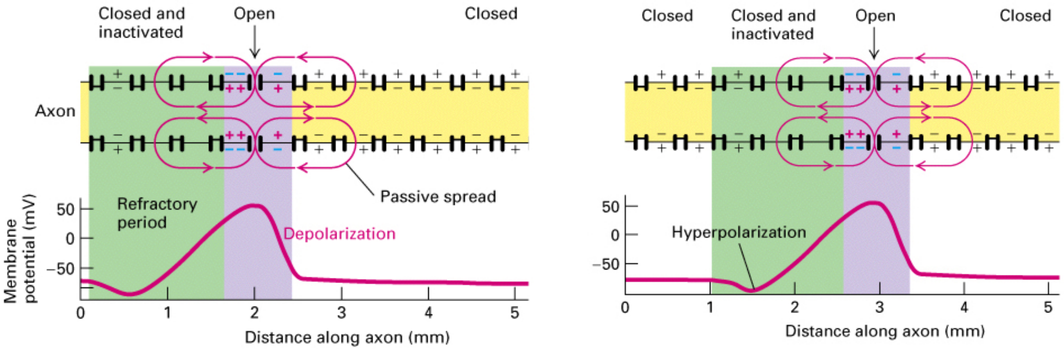

Figure 2: Propagation of action potential with respect to Na$^+$ voltage gated ion channels[9]

The travelling of these electrical impulses is reliant on the dynamics of the voltage-gated ion channels, as depicted in fig. 2. Although there are multiple ion channels at work in action potential dynamics, the aforementioned figure solely represents the mechanism of channels selective to sodium ions (Na$^+$). Such channels allow the passage of Na$^+$ when depolarized, and are otherwise closed. By virtue of concentration differences[5], a net inward flow of positive Na$^+$ ions occurs when a sodium channel opens. Depolarization of a region of the cell membrane spreads passively to neighbouring regions, which, in turn, activates proximate Na$^+$ channels and subsequently gives rise to more depolarization through an increased inward Na$^+$ current. However, such ion channels become temporarily

[5] The flow of ionic substances is dependent on both *concentration* gradients and *electrical* gradients. The inside of the cell has a lower sodium ion concentration, so Na$^+$ ions tend to diffuse into the cell when sodium channels open.

inactivated after having been opened, and will remain closed during this short period regardless of the membrane potential. As such, sufficient depolarization at one end of, say, a neuron's axon, will cause a *unidirectional* movement of an action potential: the initial stimulus can only spread forward and away from its origin, deactivating previous regions as it moves along the axon while simultaneously activating downstream sodium channels.

2.2 Derivation of the Hodgkin-Huxley equations

The *cable equation* bridges the gap between the biological mechanism of neural voltage dynamics and the Hodgkin-Huxley model. Cable theory offers a mathematical description of the electrical properties of the neuron's membrane by analyzing its *equivalent circuit*, a simplified electrical model of the tissue composed of fundamental electrical components such as resistors and capacitors.

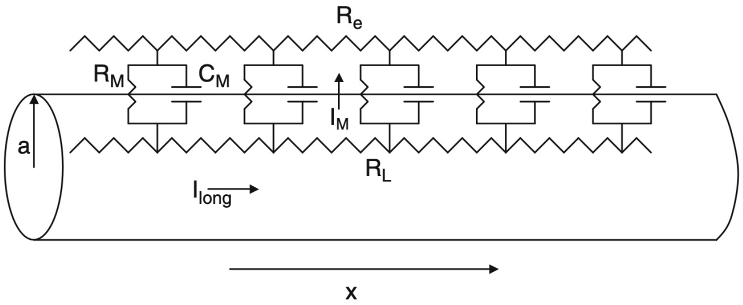

Figure 3: The equivalent circuit for a uniform passive cable representing the neuronal membrane[14]

Ohm's law states that the change in potential through a change Δx is equal to the current times the resistance encountered through Δx. Taking the limit $\Delta x \to 0$ and expressing the equation with the current isolated yields

$$I(x,\, t) = \lim_{\Delta x \to 0} \Big(\underbrace{\frac{\pi a^2}{\Delta x\, r_{\mathrm{L}}}}_{R} \,\Delta V \Big) = -\frac{\pi a^2}{r_{\mathrm{L}}} \frac{\partial V}{\partial x}(x,\, t) \tag{1}$$

which represents the current along the inside of the cable (I_{long} in fig. 3). Now, Kirchhoff's current summation law states that the total current entering a junction is equal to the current leaving it. Thus, at any point on the membrane, the sum of incoming axial current, outgoing axial current and trans-membrane current is zero:

$$I_{\mathrm{long}}^{\mathrm{in}} + I_{\mathrm{long}}^{\mathrm{out}} + I_{\mathrm{M}} = 0. \tag{2}$$

As described in 2.1, ions flow through the cell membrane. The total ionic current over Δx that flows across the membrane is

$$I_{\mathrm{ion}} = (2\pi a \Delta x) i_{\mathrm{ion}} \tag{3}$$

where i_{ion} is the current flow per unit area. In the equivalent circuit, ionic conductance is represented by R_{M}.

The cell membrane is mainly composed of a *phospholipid bilayer* which has *dielectric properties* in that it can store and release charge in the form of currents, not unlike a capacitor. The amount of current required to change the potential difference across the capacitive membrane over a length Δx at a rate $\frac{\partial V}{\partial t}$ is

$$I_{\text{cap}} = (2\pi a \Delta x) c_{\text{M}} \frac{\partial V}{\partial t} \tag{4}$$

where $(2\pi a \Delta x) c_{\text{M}} = C_{\text{M}}$ is the total cable capacitance over Δx.

The total trans-membrane current I_{M} is the sum of the ionic and capacitive currents. Substituting the sum of 3 and 4 for I_{M} in 2, and using 1 for the inward and outward ionic currents, one obtains the following formulation:

$$-\frac{\pi a^2}{r_{\text{L}}} \frac{\partial V}{\partial x}(x,\, t) + \frac{\pi a^2}{r_{\text{L}}} \frac{\partial V}{\partial x}(x + \Delta x,\, t) + (2\pi a \Delta x) i_{\text{ion}} + (2\pi a \Delta x) c_{\text{M}} \frac{\partial V}{\partial t} = 0. \tag{5}$$

Simplifying and taking the limit $\Delta x \to 0$ yields the *cable equation*:

$$c_{\text{M}} \frac{\partial V}{\partial t} = \frac{a}{2 r_{\text{L}}} \frac{\partial^2 V}{\partial x^2} - i_{\text{ion}} \tag{6}$$

Now, it has been shown in 2.1 that voltage-gated ion channels respond to voltage differences over time. These channels control the ionic flow, which is mathematically represented by i_{ion} in 6. The three major ion channels involved in the mechanisms of action potentials are voltage gated sodium channels, voltage gated potassium (K^+) channels, and *leak channels.* Leak channels are always open and allow the passage of a variety of ions.

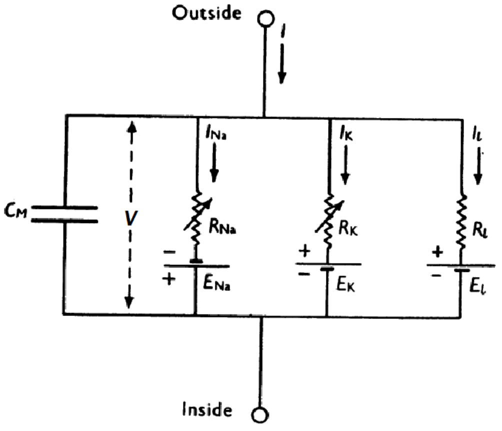

Figure 4: Equivalent circuit of voltage gated Na^+ channels, voltage gated K^+ channels, and leak channels [1]

In fig. 4, the resistors R_{Na} and R_{K}, which model the rate of Na^+ and K^+ flow respectively, are drawn with a superimposed diagonal arrow. This symbolizes *variable* resistance, which reflects the fact that the associated ion channels are voltage controlled. It can also be observed that R_{L} is not represented in this manner because it corresponds to the leak channels which are *not* voltage gated.

From 4 and Ohm's law, an equation can be derived for the ionic current:

$$\begin{aligned} i_{\text{ion}} &= i_{\text{Na}} + i_{\text{K}} + i_{\text{leak}} \\ &= \frac{1}{r_{\text{Na}}}(V - E_{\text{Na}}) + \frac{1}{r_{\text{K}}}(V - E_{\text{K}}) + \frac{1}{r_{\text{leak}}}(V - E_{\text{leak}}) \\ &\equiv g_{\text{Na}}(V - E_{\text{Na}}) + g_{\text{K}}(V - E_{\text{K}}) + g_{\text{leak}}(V - E_{\text{leak}}) \end{aligned} \tag{7}$$

where $E_\circ$ is the equilibrium potential[6] for $\circ =$ Na, K, leak.

Hodgkin and Huxley derived expressions[1] for the ion conductances g_{Na} and g_{K} from experimental data. They expressed these in terms of three *gating* variables m, n, h:

$$g_{\text{Na}} = \bar{g}_{\text{Na}}\, m^3 h \tag{8}$$

$$g_{\text{K}} = \bar{g}_{\text{K}}\, n^4. \tag{9}$$

The constants $\bar{g}_{\text{Na}}$, $\bar{g}_{\text{K}}$ are the maximum conductances, and the gating variables represent the probability that a particular *subunit* of an ion channel is open or closed. For example, K^+ voltage gated channels are composed of four identical units, all of which need to be simultaneously activated to fully open the channel and to permit the passage of potassium ions. If n represents the probability that a subunit is activated, it follows that the probability of activation of a channel is n^4 by basic probability theory. In the case of Na^+, there are two kinds of subunits: one that governs its activation (represented by m, of which there are three, hence the exponent m^3), and the other, h, that controls its inactivation [7].

To model the dynamics of the ionic currents, Hodgkin and Huxley derived the following differential equations for the change of the gating variables over time:

$$\frac{dw}{dt} = \alpha_w(V)(1 - w) - \beta_w(V)w \tag{10}$$

where $w = m$, n or h symbolizes a gating variable, and α_w, β_w are experimentally determined[8] *rate functions* for each $w = m$, n, h that solely depend on the voltage. In brief, these equations describe the rate of transition from one subunit's state to another. The function α_w corresponds to the rate at which closed subunits transition to the open state (hence the factor $(1 - w)$, the probability of subunit *inactivation*), whereas β_w governs the opposite transition.

[6] Also known as the *Nernst* or *reversal* potential, the equilibrium potential represents the membrane potential at which there is no *net* movement of the ion across the membrane.

[7] As seen in 2.1, sodium channels can also become temporarily inactived post-depolarization.

[8] The expressions underlying the rate equations do not have a particular physical intuition, they were constructed so as to match experimental data [1, 14].

2.3 The model

The Hodgkin-Huxley equations consist of 4 differential equations. The first is obtained by substituting 7 into the cable equation (6), and the other three are the gating variable equations 10.

Thus, the system is defined as follows:

$$c_{\mathrm{M}} \frac{\partial V}{\partial t} = \frac{a}{2r_{\mathrm{L}}} \frac{\partial^2 V}{\partial x^2} - g_{\mathrm{Na}}(V - E_{\mathrm{Na}}) + g_{\mathrm{K}}(V - E_{\mathrm{K}}) + g_{\mathrm{leak}}(V - E_{\mathrm{leak}}) \tag{11}$$

$$\frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m \tag{12}$$

$$\frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h \tag{13}$$

$$\frac{dn}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n \tag{14}$$

where $g_{\mathrm{Na}} = \bar{g}_{\mathrm{Na}}\, m^3 h$ and $g_{\mathrm{K}} = \bar{g}_{\mathrm{K}}\, n^4$.

In this paper, a scheme that numerically approximates the propagation of a neuronal action potential along a one dimensional axon of constant radius is described and analyzed. The equations are computed over the computational domain $\Omega = [0,\, L] \times [0,\, T]$ where the spatial domain represents the span of the axon. The following boundary conditions are imposed:

$$\frac{\partial V(0,\, t)}{\partial x} = \frac{R\, I(t)}{\pi a^2} \tag{15}$$

$$\frac{\partial V(L,\, t)}{\partial x} = 0\,. \tag{16}$$

The former, a Neumann boundary condition, corresponds to an externally applied current at the $x = 0$ end of the axon, related to the potential difference by Ohm's law, $\Delta V = RI$. The latter condition, often referred to as a *sealed end* boundary condition, requires that no current exits the $x = L$ end of the cable.

For this problem, the canonical initial conditions were utilized:

$$V(x,\, 0) = V_0(x) \tag{17}$$

$$\frac{dm}{dt} = \frac{dh}{dt} = \frac{dn}{dt} = 0 \tag{18}$$

where $V_0(x)$ is generally chosen to be close to 0, and the initial values for the gating variables depend entirely on V_0 after fixing the derivatives to 0.

Various current injection functions $I(t)$ are modelled in the scheme. Usually, a step, sine or cosine function is chosen, since such functions are most akin to biologically-founded neural inputs.

3 Numerical method

3.1 Discretization

The only spatial derivative that is present in the Hodgkin-Huxley system occurs in 11 and is of second order. It is discretized by means of a centered difference approximation *inside the domain*:

$$\left.\frac{\partial^2 V}{\partial x^2}\right|_i \approx \frac{V_{i+1} - 2V_i + V_{i-1}}{(\Delta x)^2} \tag{19}$$

Direct application of this discretization to the boundary points $i = 0,\ N$ yields:

$$\left.\frac{\partial^2 V}{\partial x^2}\right|_0 \approx \frac{V_1 - 2V_0 + V_{-1}}{(\Delta x)^2} \tag{20}$$

$$\left.\frac{\partial^2 V}{\partial x^2}\right|_N \approx \frac{V_{N+1} - 2V_N + V_{N-1}}{(\Delta x)^2} \tag{21}$$

where V_{-1} and V_{N+1} are outside of the spatial domain. To handle these cases, *imaginary* grid points are introduced. The imposed boundary conditions 15 and 16 can be used to obtain explicit formulations for these points by means of central differences:

$$\left.\frac{\partial V}{\partial x}\right|_0 \approx \frac{V_1 - V_{-1}}{2\Delta x} \implies V_{-1} \approx V_1 - 2\Delta x \frac{R\, I(t)}{\pi a^2} \tag{22}$$

$$\left.\frac{\partial V}{\partial x}\right|_N \approx \frac{V_{N+1} - V_{N-1}}{2\Delta x} \implies V_{N+1} \approx V_{N-1} \tag{23}$$

A fully implicit time stepping scheme is employed to promote stability. As outlined in 1, time is discretized with the backward Euler method, and is chosen over the generally more accurate Crank-Nicolson scheme because the latter has not yet been formally proven to converge when applied to the Hodgkin-Huxley model [7]. Further, it has been shown that larger Δt values can be used with backward Euler without compromising stability [7].

Inside the domain, applying the implicit time discretization to 11 in combination with the spatial discretization provided above yields the following:

$$\begin{aligned} C\frac{V_i^{n+1} - V_i^n}{\Delta t} = \frac{a}{2R}&\frac{V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}}{(\Delta x)^2} \\ &+ \bar{g}_{\mathrm{Na}} \left(m_i^{n+1}\right)^3 h_i^{n+1} \left(V_i^{n+1} - E_{\mathrm{Na}}\right) \\ &+ \bar{g}_{\mathrm{K}} \left(n_i^{n+1}\right)^4 (V_i^{n+1} - E_{\mathrm{K}}) \\ &+ \bar{g}_{\mathrm{L}}(V_i^{n+1} - E_{\mathrm{L}}) \end{aligned} \tag{24}$$

The discretization can be applied everywhere for the gating variables as these are ordinary differential equations without boundary conditions. The equations 12,13 and 14 admit the following discrete forms:

$$\begin{aligned} \frac{w_i^{n+1} - w_i^n}{\Delta t} &= (1 - w_i^{n+1})\alpha_w(V_i^{n+1}) - w_i^{n+1}\beta_w(V_i^{n+1}) \\ \implies w_i^{n+1} &= \frac{w_i^n + \alpha_w(V_i^{n+1})\Delta t}{1 + \left(\alpha_w(V_i^{n+1}) + \beta_w(V_i^{n+1})\right)\Delta t} \end{aligned} \tag{25}$$

where $w = m,\, h,\, n$.

For the sake of readability, the following notation is introduced:

$$\begin{aligned} \sigma &= \frac{a\,\Delta t}{2RC(\Delta x)^2} \\ \gamma_i^{n+1} &= \frac{\Delta t}{C}\left(\bar{g}_{\text{Na}}(m_i^{n+1})^3 h_i^{n+1} - \bar{g}_{\text{K}}(n_i^{n+1})^4 - \bar{g}_{\text{L}}\right) \\ D_i^{n+1} &= 1 + 2\sigma + \gamma_i^{n+1} \\ \Gamma_i^{n+1} &= \frac{\Delta t}{C}\left(\bar{g}_{\text{Na}}(m_i^{n+1})^3 h_i^{n+1} E_{\text{Na}} - \bar{g}_{\text{K}}(n_i^{n+1})^4 E_{\text{K}} - \bar{g}_{\text{L}} E_{\text{L}}\right) \end{aligned} \tag{26}$$

with which the discretization 24 can be expressed more concisely as:

$$-\sigma V_{i-1}^{n+1} + (1 + 2\sigma + \gamma_i^{n+1})V_i^{n+1} - \sigma V_{i+1}^{n+1} = V_i^n + \Gamma_i. \tag{27}$$

At the $x = 0$ boundary, the following is obtained:

$$2\Delta x\,\sigma\left(\frac{RI(t)}{\pi a^2}\right) + (1 + 2\sigma + \gamma_0^{n+1})V_0^{n+1} - 2\sigma V_1^{n+1} = V_0^n + \Gamma_0^{n+1} \tag{28}$$

and the case $x = L$ is similar.

It is convenient to express 11 in matrix form so as to succinctly encompass the entire spatial domain:

$$\begin{bmatrix} D_0^{n+1} & -2\sigma & & & \\ -\sigma & D_1^{n+1} & -\sigma & & \\ & \ddots & \ddots & \ddots & \\ & & -\sigma & D_{N-1}^{n+1} & -\sigma \\ & & & -2\sigma & D_N^{n+1} \end{bmatrix} \vec{V}^{n+1} = \vec{V}^n + \vec{\Gamma}^{n+1} + \begin{bmatrix} \frac{I(t)\,\Delta t}{\pi a C \Delta x} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{29}$$

$$\equiv \quad A^{n+1}\vec{V}^{n+1} = \vec{V}^n + \vec{\Gamma}^{n+1} + \delta_{i0}\frac{I(t)\,\Delta t}{\pi a C \Delta x}$$

This results in a *fully* implicit scheme (in contrast with a *semi*-implicit scheme) in that all the equations in the system (11, 12, 14, 13) are discretized backwards in time. Thus, we need to implicitly solve for V^{n+1}, m^{n+1}, n^{n+1} and h^{n+1} *simultaneously*. Such a situation calls for iterative methods.

3.2 Solving the nonlinear equations

One may choose to apply Newton iteration to solve this system. However, this approach requires an $N^4 \times N^4$-sized matrix, which is not ideal in terms of memory complexity. Of course, there exist sparse methods and computational libraries that provide the abstraction of such a matrix whilst keeping memory usage at a minimum. This method is used by Cox and Griffith [8].

The method described in this paper uses a different approach. It consists of first *approximating* the voltage at the next time step V^{n+1} by solving the system 29 using the previous gating variables M^n, N^n and H^n, which yields $\tilde{V}^{n+1}$. The implicit problem now becomes a *linear* one, which can be solved by simple matrix inversion. $\tilde{V}^{n+1}$ is subsequently used to approximate the gating variables, yielding $\tilde{M}^{n+1}$, $\tilde{N}^{n+1}$ and $\tilde{H}^{n+1}$. The process is then an *explicit* one by using the voltage guess. This entire procedure is repeated until the values converge within a prescribed tolerance criterion.

Algorithm 1 Iterative procedure to solve the implicit nonlinear equations at each time step

```
**prescribe** $\epsilon$
$\xi \leftarrow \infty$
$V^*$, $M^*$, $H^*$, $N^* \leftarrow V^n$, $M^n$, $H^n$, $N^n$
**while** $\xi < \epsilon$ **do**
    $\tilde{A} \leftarrow$ approximation of $A^{n+1}$ with $M^*$, $N^*$, $H^*$
    $\tilde{\Gamma} \leftarrow$ approximation of $\vec{\Gamma}^{n+1}$ with $M^*$, $N^*$, $H^*$
    $\tilde{V}^{n+1} \leftarrow \tilde{A}^{-1}\left(\vec{V^*} + \tilde{\Gamma} + \delta_{i0}\frac{I(t)\Delta t}{\pi a C \Delta x}\right)$
    $\tilde{M}^{n+1}$, $\tilde{H}^{n+1}$, $\tilde{N}^{n+1} \leftarrow$ approximation of $M^{n+1}$, $H^{n+1}$, $N^{n+1}$ using $\tilde{V}^{n+1}$
    $\xi \leftarrow \max\Big( ||\tilde{V}^{n+1} - V^*||_2, ||\tilde{M}^{n+1} - M^*||_2,$
                $||\tilde{H}^{n+1} - H^*||_2, ||\tilde{N}^{n+1} - N^*||_2 \Big)$
    $V^*$, $M^*$, $H^*$, $N^* \leftarrow \tilde{V}^{n+1}$, $\tilde{M}^{n+1}$, $\tilde{H}^{n+1}$, $\tilde{N}^{n+1}$
**end while**
$V^{n+1}$, $M^{n+1}$, $H^{n+1}$, $N^{n+1} \leftarrow V^*$, $M^*$, $H^*$, $N^*$
```

3.3 A brief note on convergence

Mascagani [5] formally proved the convergence of the backward Euler method for the Hodgkin-Huxley partial differential equations. The scheme outlined in their paper utilizes a similar iterative procedure to solve the implicit nonlinear system, and this process was also shown to be convergent. In brief, Mascagani argues that a sufficient condition for the iterates to converge is that the mapping

$V^{(k+1)} \leftarrow V^{(k)}$ defined by the iterative updates be *contractive.* One method by which this criterion can be verified is by evaluating the L^{∞} norm of the Jacobian matrix of the mapping:

$$\Big\|\frac{\partial V^{(k+1)}}{\partial V^{(k)}}\Big\|_{\infty} \equiv \max_i \sum_j \Big|\frac{\partial V_i^{(k+1)}}{\partial V_j^{(k)}}\Big| \tag{30}$$

If it is < 1, the iterative method is convergent. They have shown that the norm of the Jacobian is $O\big((\Delta t)^2\big)$ for a given fixed Δx. It then follows that, for a fixed spatial discretization level, a *sufficiently small* Δt can be chosen so as to ensure convergence. Note that this *grossly simplified* explanation relies on other claims made in the paper that are not covered herein.

3.4 Adaptive time stepping

In section 3.3, it was established that the convergence of the iterative procedure requires

$$O\big((\Delta t)^2\big) < 1 \implies r\,(\Delta t)^2 < 1 \ : \exists\, r$$

where r is not known, but depends on Δx. Although it is true that choosing a sufficiently small Δt will generally ensure the convergence of the iterative scheme, one may wish to use larger time steps without compromising the convergence.

To achieve this, an adaptive time stepping method was implemented in the scheme. The idea is to simply decrease the size of Δt by a prescribed penalty factor α if the error ξ [9] at the current iteration step is greater or equal to that of the previous time step. That is, the magnitude of the time step is decreased whenever divergence may occur. Further, if the error is smaller that the tolerance ϵ by some prescribed factor β, Δt may also be *increased* without compromising the convergence of the nonlinear solution procedure so as to increase computation speed. The adaptive time stepping mechanism requires only slight modifications to the scheme's implementation but offers significant improvements.

3.5 Implementation details

The method described in this paper has been implemented in C++, and exposes its functionality to the Python programming language via the Python C API. The C++ implementation uses the Eigen linear algebra library for most operations. This highly-optimized library uses vectorized operations and concurrent algorithms to significantly speed up computations. The entirety of the code is available at this github repository [10].

[9] See algorithm 1 for the expression for ξ.

[10] The file *neuro.cpp* is likely to be of higher relevance to the reader. The bulk of the numerical method is contained therein. The *.py* files mostly contain plotting functions and build tools.

4 Results

The results listed in this paper are not exhaustive. This section serves as a brief overview of what was achieved in the project, but most of the results are qualitative in nature and are best shown by video. Please refer to the github repository for videos by accessing the link and navigating to the `videos/` directory. *For some eye candy, try the* `videos/3d/` *directory.*

4.1 Similarity to biological phenomena

4.1.1 Refractory period

In 2.1, the idea of Na^+ channel *inactivation* was mentioned. In addition to contributing to the unidirectional flow of nerve impulses, the inactivation phenomenon causes what is known as the refractory period. This is a period during which otherwise sufficient electrical inputs cannot initiate action potentials. It can be seen in `videos/2d/current_wave_refractory_period.mp4` that the numerical method accurately models this phenomenon. In this video, the current injection function is

$$I(t) = \max\Big(1.2\cos(2t),\, 0\Big) \tag{31}$$

The first current rise causes an action potential, the second ($\sim$ 20 s) does *not*, whereas the third one ($\sim$ 40 s) does.

4.1.2 All-or-nothing & failed initiations

Integral to the nature of action potentials is the fact they only arise when a depolarization beyond a certain threshold is applied. For this reason it is said that it is an *all-or-nothing phenomenon*, and depolarizations that do not successfully create action potentials are referred to as *failed initiations*. In `videos/2d/failed_current_injection.mp4`, a current of $0.2\mu A$ is applied for 0.1 ms and causes a visible depolarization without initiating an action potential. In `videos/2d/failed_initiation_v0.mp4` another subthreshold depolarization can be observed, where the voltage comes directly from the initial conditions.

4.2 Iterative procedure & adaptive time stepping

It was found that the iterative nonlinear solution procedure outlined in algorithm 1 converges surprisingly quickly, provided a small enough Δt is used to ensure convergence (if adaptive time stepping is not used, otherwise this is handled automatically). In fact, when the applied current $I(t)$ is zero (either no current is ever injected, or current was previously injected but is not *currently*

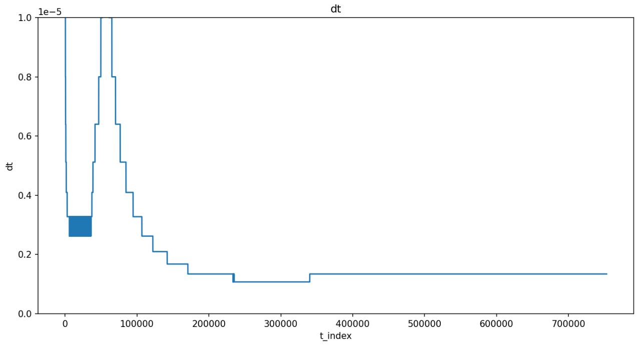

Figure 5: Change in Δt over time

being injected), the number of iterations required for convergence with a prescribed tolerance $\epsilon = 5 \cdot 10^{-4}$ generally never exceeds 3.

In fig. 5, the changing time step Δt is plotted over time in accordance with the procedure outlined in section 3.4. The dynamics associated with this plot can be viewed in the file `videos/3d/dt_change.mp4` (notice that this one is in the `videos/3d/` directory), which corresponds to a short current injection, causing a depolarization, which is then stopped. At this point, the voltage slowly drops, but the gating feedback loops kick in and eventually yields the action potential in an exponential manner. The change in Δt in fig. 5 reflects this activity: the current injection and subsequent depolarization calls for smaller time steps. Stopping the current injection yields a less active system, and so larger time steps can be temporarily afforded while the membrane potential slowly drops. As the action potential begins to manifest, it is clear the Δt adapts to the increasing level of activity.

5 Conclusion

In sum, this paper has described an implicit numerical method to solve the Hodgkin-Huxley partial differential equations. It was shown that the method yields a model that exhibits biologically accurate behaviour. The adaptive time stepping approach allows for a higher time stepping flexibility and it was demonstrated that it accommodates different levels of dynamical activity, adjusting for lower Δt when high speed activity is present, and taking larger steps when the inverse is true.

References

[1] Alan Hodgkin and Andrew Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve". In: *The Journal of Physiology* 117.4 (1952), pp. 500–544. DOI: 10.1113/jphysiol.1952.sp004764.

[2] D. Noble. "Cardiac Action and Pacemaker Potentials based on the Hodgkin-Huxley Equations". In: *Nature* 188.4749 (Nov. 1960), pp. 495–497. DOI: 10.1038/188495b0.

[3] Cooley JW and Jr. Dodge FA. "Digital computer solutions for excitation and propagation of the nerve impulse". In: Biophysical Journal 6.5 (1966), pp. 583–99. DOI: 10.1016/S0006-3495(66)86679-1. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1368016/.

[4] Stockbridge N. "Solution of the Hodgkin-Huxley and cable equations on an array processor". In: Ann Biomed Eng. 17.3 (1989), pp. 253–68. DOI: 10.1007/BF02368045. URL: https://pubmed.ncbi.nlm.nih.gov/2735583/.

[5] Michael Mascagni. "The Backward Euler Method for Numerical Solution of the Hodgkin-Huxley Equations of Nerve Conduction". In: *SIAM Journal on Numerical Analysis* 27.4 (1990), pp. 941–962. ISSN: 00361429. URL: http://www.jstor.org/stable/2157690.

[6] Arnold Burgen. "Information flow in the nervous system". In: *European Review* 1.1 (1993), pp. 31–39. DOI: 10.1017/S1062798700000363.

[7] Michael Mascagni and Arthur Sherman. "Numerical Methods for Neuronal Modeling". In: *Methods in Neuronal Modeling: From Ions to Networks* (Apr. 2000). URL: https://www.cs.fsu.edu/~mascagni/papers/RCEV1996_1.pdf.

[8] Boyce E. Griffith Steven J. Cox. *A Fast, Fully Implicit Backward Euler Solver for Dendritic Neurons*. Tech. rep. Rice University, 2000. URL: https://hdl.handle.net/1911/101958.

[9] Harvey F. Lodish and James E. et al. Darnell. *Molecular Cell Biology*. 5th ed. W. H. Freeman & Co, 2003. ISBN: 9780716743668.

[10] Nicholas T. Carnevale and Michael L. Hines. *The NEURON Book*. 1st ed. Cambridge University Press, 2004.

[11] Kenneth H. Karlsen Monica Hanslien and Aslak Tveito. "A Maximum Principle for an Explicit Finite Difference Scheme Approximating the Hodgkin-Huxley Model". In: BIT Numerical Mathematics 45.4 (2005), pp. 725–741. DOI: 10.1007/s10543-005-0023-2. URL: https://link.springer.com/article/10.1007/s10543-005-0023-2.

[12] T. Carnevale. “Neuron simulation environment”. In: *Scholarpedia* 2.6 (2007). revision #154131, p. 1378. DOI: 10.4249/scholarpedia.1378. URL: http://www.scholarpedia.org/article/Neuron_simulation_environment.

[13] James Sneyd James Keener. *Mathematical Physiology*. 2nd ed. Interdisciplinary Applied Mathematics. Springer New York, NY, 2008. DOI: https://doi.org/10.1007/978-0-387-75847-3.

[14] David H. Terman G. Bard Ermentrout. *Mathematical Foundations of Neuroscience*. 1st ed. Interdisciplinary Applied Mathematics. Springer New York, NY, 2010. DOI: https://doi.org/10.1007/978-0-387-87708-2.

[15] David L. Chopp Richard A. Kublik. “A locally adaptive time stepping algorithm for the solution to reaction diffusion equations on branched structures”. In: *Advances in Computational Mathematics* 42.3 (2016), pp. 621–649. DOI: 10.1007/s10444-015-9437-9. URL: https://link.springer.com/article/10.1007/s10444-015-9437-9.

[16] Juergen Geiser and Dennis Ogiermann. *Iterative Implicit Methods for Solving Hodgkin-Huxley Type Systems*. 2019. arXiv: 1905.00697 [math.NA]. URL: https://arxiv.org/abs/1905.00697.

[17] R. Park. *A comparison of six numerical methods for integrating a compartmental Hodgkin-Huxley type model*. 2021. arXiv: 2105.08159 [math.NA]. URL: https://arxiv.org/abs/2105.08159.

[18] Zurich MedTech. *Neuronal Tissue Models: T-NEURO*. URL: https://zmt.swiss/sim4life/tissue-models/t-neuro/.