Exam in FY8904

Computational Physics

Candidate: 10019

April, 2024

Abstract

In this study, the processes behind the functioning of a brain are discussed from a physical perspective and modeled with computational routines. A model to describe the exchange of information among nodes in a network is initially presented, followed by a focused study on how the impulse is transmitted inside a neuron and through its axion.

Contents

1. Introduction

Understanding the working mechanism in biological systems is usually a very complex task, as many components are interacting with each other resulting in non-linear effects that also happen at different length scales. It is quite common that some minor concentration changes can lead to a cascade of effects eventually causing large variations in the properties and functions of a system. When the system in question is the brain, this task is particularly challenging, requiring knowledge from many different fields such as biology, chemistry physics and computer science.
A first necessary step, which is the topic of this work, is the understanding of the mechanisms that regulate the transmission of signals among the complex network of neurons. Section 2 presents a computational model to describe the transmission among a fully connected network of cells, eventually generalizing to two disjoint sub-networks. In section 3, the length scale is reduced and the transmission of impulses inside a neuron is discussed and modeled, gradually introducing effects to improve the agreement with experimental data.

2. Network model of the brain

In this section, the transmission of information among a closed system of neurons is modeled, assuming that the information is represented as positive charges and every fixed amount of time each node transfers this charge equally to the neighbouring nodes.

2.1. Single neural network

We consider a simple neural network, consisting of $N = 21$ synchronous nodes, each directly connected with 4 neighbours. The state at each node is represented by a vector $V(t)$ where each component contain the charge at the corresponding node. The time evolution is described by

$$V(t + \mathrm{d}t) = T \cdot V(t) \tag{1}$$

where T is a transformation matrix, defined by the network topology. The matrix used in the implementation in a Julia code is defined in Appendix A.1. A first test of time evolution was started from an initial vector $V(t = 0) = (1, 0, \ldots, 0)$ and the results are presented in Fig. 1 and in Animation 1. Looking at the figures is possible to see that the localized charge is spread after each timestep, initially fully transmitting the information to the 4 neighbours and continuing expanding to their neighbouring nodes and so on. At the end, the charges are equally spread among all nodes. It is

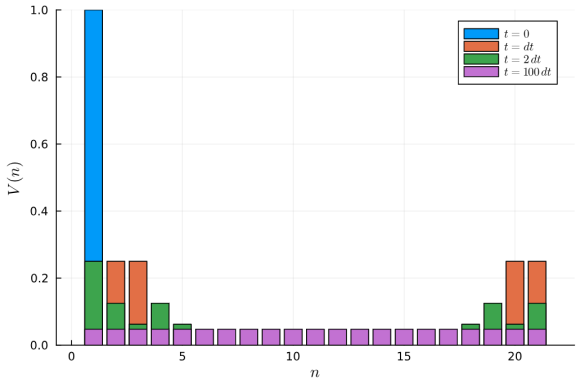


Figure 1: Results of the time evolution of a starting vector $V(t = 0) = \delta_{i1}$. The charges for each element of the vectors are presented at $t = 0, dt, 2dt, 100dt$.

interesting to look at eigenvalues and eigenvectors of the transformation matrix. The smallest and biggest eigenvalues are found using Lanczos algorithm in the `Arpack.jl` package, giving the same results (up to order 10^{-15}) of the complete diagonalization with `LinearAlgebra.jl`. There is a single eigenvalue 1.0, and all the others are in pairs and decreasing. The complete list can be found in Appendix A.3. The eigenvector corresponding to eigenvalue of 1, is $\mathbf{v}_N = \frac{1}{\sqrt{N}}(1, 1, \ldots, 1)$.
The time evolution is repeated 2 more times up to $t = 100dt$, starting with

$$V(t = 0)_i = \begin{cases} \frac{1}{5} & \text{if } i = 2, 7, 8, 15, 18, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

and with a randomly generated vector, initialized with random values in [0,1] for each entry and then divided by the sum of the elements to make it a stochastic vector. Again, also in these 2 cases the simulation ended with the charges uniformly distributed among all nodes. The results are presented in animations 2 and 3. This specific behaviour at longer times is strongly connected to the properties of the transformation matrix, as longer times means that we are applying a large number of times the matrix to the initial vector. Eventually, we will always obtain a vector parallel to the eigenvector corresponding to eigenvalue of 1, and thus equally divided among nodes. A more precise derivation is presented in Appendix A.4.

2.2. Disjoint neural network

We now move to consider a neural network made of two disjoint sub-networks with 11 and 10 connected nodes each. The two sub-networks are not connected to each other but inside each, every two nodes are connected by a continuous sequence of links. The transformation matrix is block-diagonal,

with zeros on the off-diagonal blocks, as defined in Appendix A.2. Eigenvalues and eigenvectors are also defined in blocks, with the eigenvalues of the 11-block that are a single 1.0 value and 5 pairs and the eigenvalues of the 10-block that are 4 pairs, a 1.0 and a zero (10^{-16}). This means that we have two eigenvectors corresponding to eigenvalues of 1, $\mathbf{v}_{N-1} = \frac{1}{\sqrt{10}}(0,\dots,0,1,\dots,1,)$ and $\mathbf{v}_N = \frac{1}{\sqrt{11}}(1,\dots,1,0,\dots,0)$. The final distribution of charges will be determined by the scalar product of the initial vector and these two eigenvectors, as described in Appendix A.4. This was also tested and confirmed computationally in the Appendix (see animations 4 - 7).

2.3. Origin of a distribution

From a given state vector, the original distribution can be obtained tracing back the time evolution. This can be done solving the linear system in Eq. 1, where now the unknown vector is $V(t)$. This was implemented in Julia, with a direct inversion of the T matrix, with LU factorization, with Conjugate Gradient (CG) and with Generalized minimum residual method (GMRES). The considered state vector was $V(t)_i = 1/i$, solving the linear system for $t' = t - dt$ and $t' = t - 5dt$. All three methods were compared and the results can be found in A.1. The resulting vector is oscillating, with alternating positive and negative values. The LU factorization and Conjugate Gradient performed similarly in time, but with errors larger for the CG method, which was the most efficient in terms of allocations. The GMRES method was slower than the other 2 methods and also required more allocations. The direct inversion was by far the most expensive in both time and allocations. Jacobi, Gauss-Seidel and Successive Over Relaxation (SOR) methods could not be used since the matrix has zeros on all the diagonal.

3. Transmission of electric impulses

In this section the focus is shifted to the description of the transmission of electric impulses inside neurons, from the main cell body where the impulses are collected and through the axion (see Fig. 2). Along the axions, there are many ion channels, in particular voltage-gated Na^+, leaky K^+ and Cl^-, that are such to maintain a balanced flow of ions that originates a non zero membrane potential. When an impulse is capable to overcome a threshold potential value, the ion channels open and the potential is rapidly increased and transmitted along the axion [1]. Models to describe this phenomenon are presented in this section.

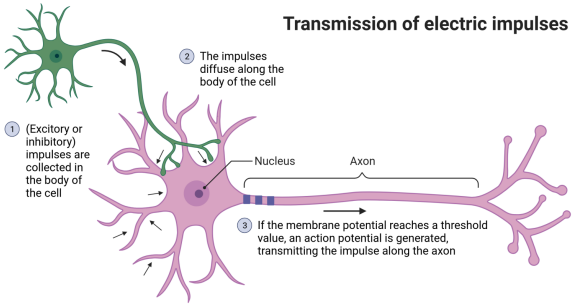


Figure 2: Schematic representation of the transmission of electric impulses along a neuron. From Exam documentation.

3.1. Linear cable equation

One possibility to describe the current through an axion is to use the so-called linear cable equation

$$\lambda^2 \frac{\partial^2 V(x,t)}{\partial x^2} - \tau \frac{\partial V(x,t)}{\partial t} = V(x,t) \qquad (3)$$

where $\lambda = \sqrt{a\kappa/(2g_{\text{tot}})}$ and $\tau = C/g_{\text{tot}}$. Here a is the radius of the cable, κ is the fluid's electrical conductivity, C is the capacitance per area and g_{tot} is the total conductance per area. The problem in Eq. 3 is solved in a Julia code, using Euler explicit, Euler implicit and Crank-Nicolson schemes to numerically solve the equation. The system is bounded in $[0, L]$ and discretized in N points $x_i = i\,dx$ with $i = 0, 1, \dots, N-1$, such that $dx = L/(N-1)$. For simplicity, $\lambda = L = 1.0$ m and $\tau = 1.0$ s were chosen, with $N = 51$. For the boundaries, Neumann boundary conditions were used and a narrow gaussian for the initial condition. Details on the implementations of the three schemes can be found in Appendix B, together with the expression of an analytical solution for the unbound problem. The analytical solution is valid only at small times, in particular for $t << 0.0625$, since for longer times the voltage at the boundaries is not small enough to be a good approximation.

The resulting time evolutions for $N = 51$ and $dt = 1.5\,10^{-4}$ for 500 steps can be found in animations 8-11 and Fig. 3 (first row). The initially localized gaussian is gradually spreading on the entire interval. The potential values for the Crank-Nicolson scheme are in fact in between those of the explicit and implicit Euler methods and they better relate to the exact unbound solution. After 100 steps, we can start to see first deviations at the extremes, as we are at about one quarter of the validity time of 0.0625. After 500 steps the profile given by the analytical solution is very different, giving lower values at the extremes than the numerical results.

With the chosen parameters, $\alpha = (\lambda^2 dt)/(\tau dx^2) = 0.375$, which is crucial to have lower than 0.5 to

have a stable Euler explicit scheme. This was also true in the implementation of the diffusion equation of assignment 1. To show this, dt was increased to $2.004\,10^{-4}$, giving $\alpha = 0.501$ and from the results in Fig. 3 (second row) and animations 12-15 we can see that in fact the Euler explicit scheme is unstable, with very large oscillations. The other methods are instead stable. Increasing a lot α, to 2.5, creates some spurious small oscillations at the peak of the gaussian for the Crank-Nicolson scheme.

3.2. Introducing ion channels

The effect of the ion channels needs to be introduced in Eq. 3, to obtain a form of activation and a traveling wave. When the membrane permeability to Na^+ and K^+ is included, the expanded equation reads

$$\lambda^2 \frac{\partial^2 V(x,t)}{\partial x^2} - \tau \frac{\partial V(x,t)}{\partial t} = \frac{g_{\mathrm{Na}}(V(x,t))}{g_{\mathrm{K}}} \left[V(x,t) - \mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}} \right] + \left[V(x,t) - \mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}} \right]. \tag{4}$$

where $\mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}} = 56$ mV and $\mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}} = -76$ mV are the Nernst potential of sodium and potassium, used to represent the equilibrium ionic distributions inside and outside the cell. K^+ channels are contributing with a constant effect with a permeability $g_{\mathrm{K}} = 5.0\,\Omega^{-1}\ \mathrm{m}^{-2}$ whereas the Na^+ contribution is dependent on the value of the potential, as these are voltage-gated channels. Its permeability is defined as

$$g_{\mathrm{Na}}(V) = \frac{100}{1 + e^{\gamma(V^* - V)}} + 0.2\,, \tag{5}$$

expressed in $[\Omega^{-1}\ \mathrm{m}^{-2}]$. $V^* = -40$ mV is a threshold potential and $\gamma = 0.5\ \mathrm{mV}^{-1}$, which means that for very low V the permeability is $0.2\ \Omega^{-1}\ \mathrm{m}^{-2}$ and then increases very rapidly for potentials between -50 mV and -30 mV and eventually stabilizes at $100\ \Omega^{-1}\ \mathrm{m}^{-2}$.

The problem in Eq. 4 is solved using Crank-Nicolson scheme in a julia code. The system is bounded in $[0, L]$ and discretized in N points $x_i = i\,dx$ with $i = 0, 1, \ldots, N-1$, such that $dx = L/(N-1)$. For simplicity, $L = 1.0$ mm is chosen and $N = 51$, with $\lambda = 0.18$ mm and $\tau = 2.0$ ms to be biologically relevant. For the boundaries, Neumann boundary conditions were used and the initial impulse is defined as

$$V(x,0) = (V_{\mathrm{appl}} - V_{\mathrm{mem}}) \exp\left(-\frac{(x - x_0)^2}{2\lambda^2} \right) + V_{\mathrm{mem}} \tag{6}$$

with $\lambda = 0.18$ mm and V_{mem} =-70 mV. This is such that at x_0 the potential is V_{appl}. More details can be found in Appendix C. A $dt = 0.002$ ms was chosen, that gives $\alpha = 0.081$, running the simulation for 500 steps up to 1.0 ms. The results of the time evolutions, choosing $x_0 = 0.5$ mm and $V_{\mathrm{appl}} = -60, -50, -47, -46, -43, -40$ mV are presented in Fig. 4, looking at how the potential changes with time at a position 0.26 mm to the right of the center of the initial impulse.

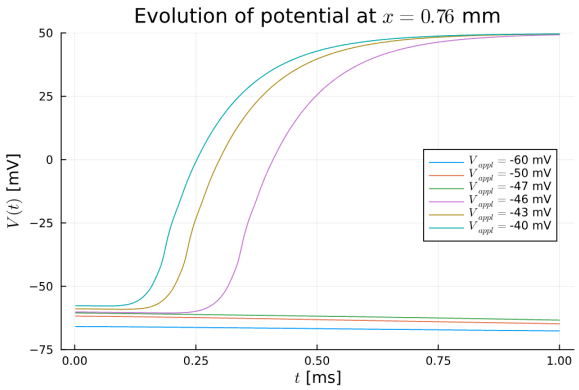
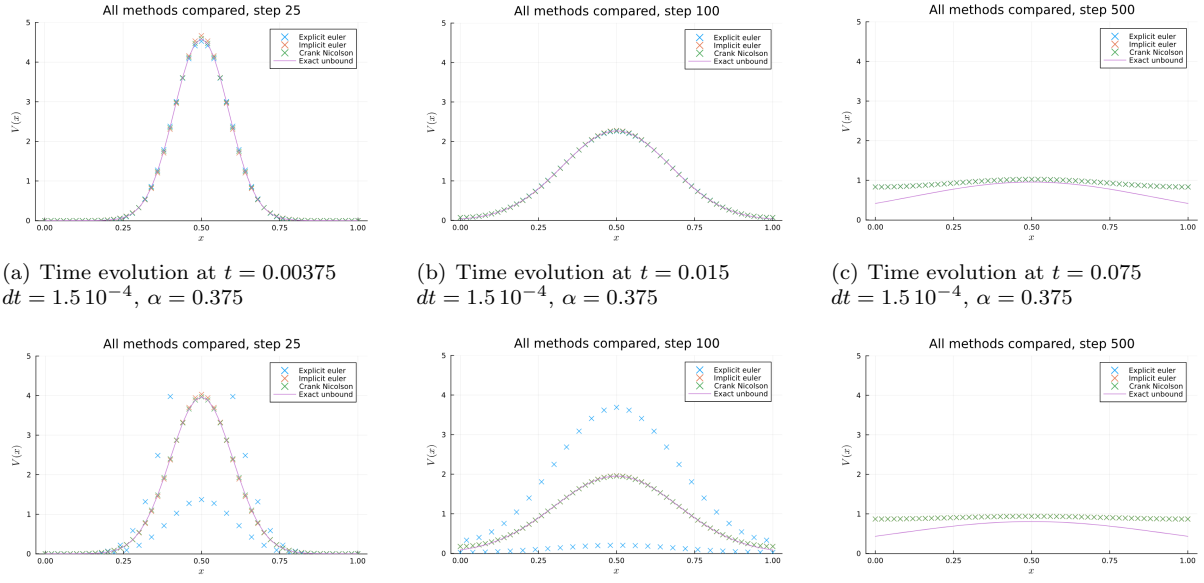


Figure 4: Electric potential profile at different timesteps of a point 0.26 mm to the right of the initial impulse, choosing $V_{\mathrm{appl}} = -60, -50, -47, -46, -43, -40$ mV.

Looking at Fig. 4 and animation 16 for $V_{\mathrm{appl}} = -40$ mV and animation 17 for $V_{\mathrm{appl}} = -50$ mV, it is possible to see there are two possible behaviours. A $V_{\mathrm{appl}} \leq -47$ mV is not capable to activate the ion channels enough to spread the impulse to the membrane and slowly disappears thanks to the constant effect of the K^+ channels. When $V_{\mathrm{appl}} \geq -46$ mV, the impulse is strong enough that is quickly spread through the membrane, raising the voltage to 50 mV on the entire system. This can effectively represent the effect of a voltage-gated activation and spreading.

3.3. Shifted impulse with localized channels

In the previous model, the channels are distributed on the entire surface of the membrane but in reality, impulses are collected on the body of the cell and only then possibly transmitted through the axion. To model this effect, we consider the case of a larger system, with $L = 2.5$ mm, an impulse with the same λ but centered at $x_0 = 0.75$ mm and the Na^+ channels that are only effective 0.25 mm to the right. This is enforced by setting the permeability to zero for $x < 1.0$ mm. The simulation is run with $dt = 0.01$ ms, giving $\alpha = 0.0648$, for 500 steps up to 5.0 ms. All other parameters are kept the same as the previous simulation. The results of the time evolutions are presented



(a) Time evolution at $t = 0.00375$ $dt = 1.5\,10^{-4}$, $\alpha = 0.375$

(b) Time evolution at $t = 0.015$ $dt = 1.5\,10^{-4}$, $\alpha = 0.375$

(c) Time evolution at $t = 0.075$ $dt = 1.5\,10^{-4}$, $\alpha = 0.375$

(d) Time evolution at $t = 0.00375$ $dt = 2.004\,10^{-4}$, $\alpha = 0.501$

(e) Time evolution at $t = 0.015$ $dt = 2.004\,10^{-4}$, $\alpha = 0.501$

(f) Time evolution at $t = 0.075$ $dt = 2.004\,10^{-4}$, $\alpha = 0.501$

Figure 3: Electric potential distribution at different times, with $N = 51$. In the first row $\alpha = 0.375$ and all methods are stable. In the second row $\alpha = 0.501$ and Euler explicit is clearly unstable, with oscillations that are out of scale in the bottom right plot.

in Fig. 5, for $V_{\text{appl}} = -40, -20, -14, -13, -10, 10$ mV. The entire simulation can be found in animation 18 for $V_{\text{appl}} = -13$ mV and in animation 19 for $V_{\text{appl}} = -14$ mV.

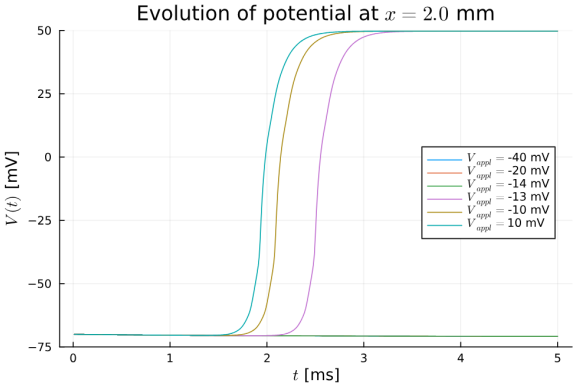


Figure 5: Electric potential profile at different time-steps of a point at x=2.0 mm, choosing $V_{\text{appl}} = -40, -20, -14, -13, -10, 10$ mV. Ion channels are active only for $x > 1.0$ mm and the initial impulse is centered at $x_0 = 0.75$ mm.

Again, two different behaviours can be observed, but with a required potential significantly higher: for $V_{\text{appl}} \leq -14$ mV the signal is dissipated as the potential at the start of the channels is not strong enough to activate the effect but for $V_{\text{appl}} \geq -13$ mV instead, the activation if effective. The applied potential is much higher since only a fraction of it is felt 0.25mm to the right of the impulse. Also, in both cases eventually the signal is dissipated in the initial area, free of ion channels and if the potential is effective, the signal is spread to the right for $x \geq 1.0$ mm, reaching a voltage of 50 mV.

3.4. Interpretation and comparison with experimental data

This model is thus capable of better represent the effect of the induced activation, which require an applied voltage higher than the one necessary to simply activate the effect since comes from the body of the cell. Nevertheless, only the activation is modeled, the deactivation of the Na^+ channels needs to be introduced to fully restore the initial conditions. In Fig. 6 we see that in fact the potential is in fact rapidly increasing but is also suddenly decreasing below the equilibrium point and eventually reaching equilibrium again. In cells this is due to the fact that about half a millisecond after peaking, the conductance of Na^+ drops to zero. The one of K^+ instead increases slowly in time, causing the potential to get even lower than the initial situation, in a phenomenon called afterhyperpolarization. Eventually, the K^+ conductance resets to its original value, with a potential that goes back to V_{mem} [1]. Another possible effect is related to the presence of Cl^- ions and channels, that have a Nernst potential of $\mathcal{V}_{\text{Cl}}^{\text{Nernst}} = -62$ mV [1]. This means that when the potential is activated and far

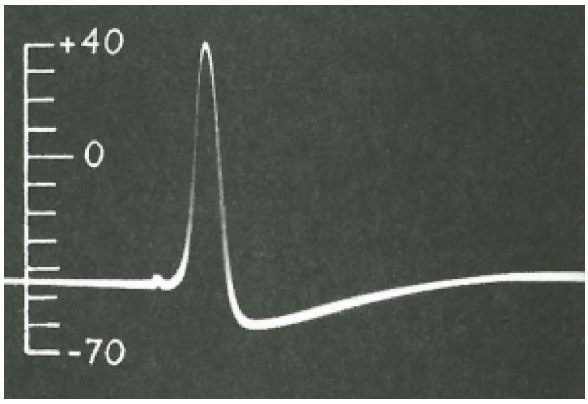


Figure 6: Electric potential between the inside and outside of an axon at different time-steps. From [2].

above this value, we have a movement of negative ions towards the inside of the cell, causing an increment of hyperpolarization and thus inhibition. The introduction of their effect could be another cause of termination of the action potential.

A possible change in the model, to try to include both the termination of activation and afterhyperpolarization is proposed in Appendix C.2, ignoring the possible effect of Cl^-. Running the simulation for 1000 time-steps with $V_{\text{appl}} = -10$ mV and keeping every other parameters as Sec. 3.3, gives the results in Fig. 7 and animation 20.

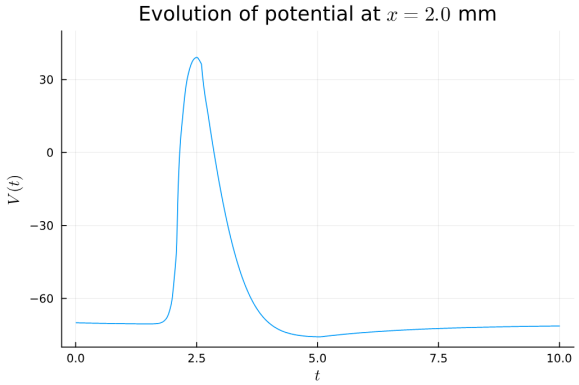


Figure 7: Electric potential profile at different time-steps of a point at x=2.0 mm, choosing $V_{\text{appl}} = -10$ mV. Ion channels are active only for $x > 1.0$ mm and the initial impulse is centered at $x_0 = 0.75$ mm. Termination of activation and afterhyperpolarization are introduced as explained in Appendix C.2.

The time evolution of the potential is quite similar to the experimental one in Fig. 6, with a clear termination of the activation and a phase of afterhyperpolarization that in the animation can be seen moving to the right and restoring the resting potential. The time-scale in the experimental results is not available, but it would be interesting to compare them as the termination is modeled from biological relevant data. The resting V_{mem} was kept to -70 mV, to be consistent with previous simulations whereas the experimental one is of about -45 mV.

4. Conclusion

The model used to describe the transmission of signal in a neural network shows the capability of spread the charge on the entire network. If sub-networks exists and are disjoint, only the charge that is initially present in the sub-network is spread. The model is also capable of reconstruct the signal that originates a certain distribution at a given time.
The transmission of impulses in a nerve cell is a more complex phenomenon, that cannot be described as a linear cable as activation effects after a certain threshold are not included. The addition of this effect with the introduction of terms related to the ion channels is capable to describe the activation and transmission of the impulse along the cell but it is missing the deactivation of the signal. With a proposed modified version of the model, this effect is introduced, together with the afterhyperpolarization generated after the termination and potential profiles similar to experimental results are obtained.

5. Animations on GitHub

All animations are in the public GitHub repository linked.

1. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_100000.gif
2. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_somepoints.gif
3. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_random.gif
4. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_disj_100000.gif
5. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_disj_000100.gif
6. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_disj_100100.gif
7. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob2/time_evol_disj_300100.gif
8. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Euler_explicit.gif
9. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Euler_implicit.gif
10. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson.gif
11. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Exact.gif
12. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Euler_explicit_badalpha.gif
13. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Euler_implicit_badalpha.gif
14. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson_badalpha.gif
15. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Exact_badalpha.gif
16. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson_channel_v40.gif
17. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson_channel_v50.gif
18. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson_shifted_channel_13.gif
19. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson_shifted_channel_14.gif
20. https://github.com/FedeRoxx/ComputationalPhysics/blob/main/Exam/Prob3/Crank_Nicolson_shifted_modified.gif

References

[1]P. Nelson, *Biological physics. energy, information, life* (Chiliagon Science, 2020) Chap. 12.

[2]R. Phillips, J. Kondev, J. Theriot, and H. Garcia, *Physical biology of the cell* (Garland Science, 2013).

Appendix A. Transformation matrix

The transformation matrix defined in Sec. 2, is determined in this Appendix, together with its eigenvalues, eigenvectors and properties.

Appendix A.1. Single network

First of all, the matrix needs to be symmetric since if node A is connected to node B, also the opposite is true. No node is directly connected to itself, which means that the diagonal is zero. Then the connection to the 4 neighbours of node i are represented with ones, to the positions $i-2, i-1, i+1, i+2$. To enforce periodic boundary condition, the first two and last two nodes need to be adjusted.
For $N = 8$, the matrix is

$$T_8 = \frac{1}{4}\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{A.1}$$

which can be easily generalized to $N = 21$. In this work, unless differently specified, the transformation matrix used was $T = T_{21}$. The pre-factor is such that the sum of the elements in each column and row is one. This ensures that T is a stochastic matrix, in such a way that returns a stochastic vector when applied on a stochastic vector.

Appendix A.2. Disjoint sub-networks

The transformation matrix in this case is composed of 4 blocks, where the off-diagonal blocks are all zeros to ensure no connections among node of different sub-networks. The diagonal blocks are instead matrices T_n of the same form of Eq. A.1. For $N = 21$ the matrix is then the following

$$T_{\text{disj}} = \begin{pmatrix} T_{11} & 0 \\ 0 & T_{10} \end{pmatrix} \tag{A.2}$$

Appendix A.3. Eigenvalues and eigenvectors

Smaller and bigger eigenvalues, with Lanczos algorithm: (-0.5617449009293667, 0.9999999999999997)
Full list of eigenvalues:

(-0.5617449009293668,	-0.5617449009293664,	-0.5,	-0.4999999999999998,
-0.45705036631935214,	-0.4570503663193519,	-0.32916088912170105,	-0.329160889121701,
-0.18385542373171573,	-0.18385542373171568,	-0.1387395330218426,	-0.13873953302184253,
-0.016629010219494294,	-0.01662901021949407,	0.20048443395120946,	0.20048443395120952,
0.5957898993411948,	0.5957898993411952,	0.8909057900510672,	0.8909057900510674,
1.0000000000000007)			

For the disjoint network, the two smaller and biggest eigenvalues are:
(-0.5590169943749478, -0.5590169943749473, 0.9999999999999993, 1.0000000000000002)

Eigenvectors can be easily printed from the code.

Appendix A.4. Properties of the matrix

Any diagonalizable matrix, can be expressed using the Spectral Theorem as

$$T = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^T \tag{A.3}$$

where λ_i and $\mathbf{v}_i$ are the eigenvalues and orthonormalized eigenvectors of the matrix.
Thanks to the orthonormality among the eigenvectors,

$$T^n = \sum_i \lambda_i^n \mathbf{v}_i \mathbf{v}_i^T. \tag{A.4}$$

For the transformation matrix T used in this work, we observed that we get one single eigenvalue $\lambda_N = 1$ and all the other eigenvalues have module lower than 1, which means that all the contributions with $i \neq N$ are eventually zero for big enough n, giving $T^n \approx \mathbf{v}_N \mathbf{v}_N^T$.
From this is clear that evolving the neural network for longer times, which correspond to applying a big n of times the operator T to the initial vector $V(t=0) = V_0$, always gives a vector that is proportional to the eigenvector corresponding to $\lambda = 1$ and thus equally spread on the entire network.

For the disjoint transformation matrix, a similar procedure gives $T^n_{\text{disj}} \approx \mathbf{v}_{N-1}\mathbf{v}_{N-1}^T + \mathbf{v}_N \mathbf{v}_N^T$, with the difference that now we have two eigenvectors with eigenvalue one, each localized in one of the 2 blocks. This means that if we start from an initial vector purely localized in one block, it will only spread on that single block. If the initial vector is delocalized in both blocks, at the end it will disperse on each one according to the initial scalar products on one and the other blocks. To test this, 4 different time-evolutions were started from

1. $V(t=0) = \delta_{i,1}$ (Anim 4)
2. $V(t=0) = \delta_{i,12}$ (Anim 5)
3. $V(t=0) = \frac{1}{2}(\delta_{i,1} + \delta_{i,12})$ (Anim 6)
4. $V(t=0) = \frac{1}{2}(0.75\delta_{i,1} + 0.25\delta_{i,12})$ (Anim 7)

and in fact, in the first two cases the charges were localized for the entire time on a single block and were eventually equally subdivided only on that block. In the third case the final distribution was of 0.5 equally distributed in the first 11 nodes and another 0.5 among the last 10 nodes. In the fourth and last case instead, the final distribution was of 0.75 equally distributed among the first 11 nodes and 0.25 among the last 10 nodes.

Appendix A.5. Solving linear system

Results from Sec. 2.3, with different algorithms. Direct inversion is performed with `LinearAlgebra.jl`, LU factorization and GMRES with `LinearSolve.jl` and Conjugate gradient (CG) with `IterativeSolvers.jl`.

	Inversion	LU	CG	GMRES
Time on $t - dt$	0.007	0.004	0.003	0.005
Error on $t - dt$	0.0	$5.48\ 10^{-15}$	$1.11\ 10^{-11}$	$1.48\ 10^{-13}$
Allocations on $t - dt$	14.7	8.0	1.5	9.8
Time on $t - 5dt$	0.044	0.015	0.012	0.025
Error on $t - 5dt$	0.0	$1.05\ 10^{-7}$	$9.80\ 10^{-7}$	$4.59\ 10^{-7}$
Allocations on $t - 5dt$	74.0	40.7	8.3	49.5

Table A.1: Comparison of 3 methods to solve a linear system. Time is expressed in ms, allocations in kB and the error is evaluated as the 2-norm of the difference between the solution vectors and the one obtained with a direct inversion.

The solution from direct inversion at $t - dt$ is:

(-4.392413086569437 4.898762292918643 -2.6185858903738186 3.8251939881583876
-2.9982098611742583 1.7180334586294346 -2.124641556414003 1.9643240960965425
-0.11271912212314628 1.0193272199077166 -0.41456531514581024 -1.0370396588275856
0.4940679246793808 -0.7654964961079515 2.5247937777736564 -1.6961077579111645
2.234202996006405 -3.74350027767211 3.1501083754566768 -3.465981391329691
5.185804988784868)

The solution from direct inversion at $t - 5dt$ is:

(-5.022436010100753e7 4.935916265733784e7 -4.7391080373523004e7 4.436495642693291e7
-4.034719399795014e7 3.5427645382880546e7 -2.971839723839771e7 2.3344596429316275e7
-1.6448584053954722e7 9.187362847310062e6 -1.7199777008958207e6 -5.7869307672044085e6
1.3162477552019583e7 -2.0245132275456358e7 2.68768340739996e7 -3.2906502684426703e7
3.820201960447851e7 -4.2645129983768314e7 4.613444937933757e7 -4.859346969504266e7
4.99672581633732e7)

Appendix B. Implementation of computational schemes

In this Appendix, the computational schemes to solve Eq. 3 for Euler explicit, Euler implicit and Crank-Nicolson are presented. In all methods we are assuming to work in $[0, L]$ with vectors discretized in N points $x_i = idx$ with $i = 0, 1, \ldots, N-1$, such that $dx = L/(N-1)$. Neumann boundary conditions are enforced. Also, the notation used is

$$V_i^n = V(t_n, x_i).$$

Some intermediate factors α and β are defined as

$$\alpha = \frac{\lambda^2 dt}{\tau dx^2} \qquad \beta = \frac{dt}{\tau}. \tag{B.1}$$

The initial impulse was generated as a narrow gaussian centered in $x_0 = L/2$ and with $\sigma = 0.01$. The discretization was done evaluating the gaussian at each point, normalizing afterwards to have the gaussian integrate to 1 on $[0, L]$.

Appendix B.1. Euler explicit

We start expanding the second space derivative using central differences and the time derivative using forwards Euler. In this method the points at time t_{n+1} are defined from the ones at the previous time-step.

$$\frac{\lambda^2}{dx^2}(V_{i+1}^n - 2V_i^n + V_{i-1}^n) - \frac{\tau}{dt}(V_i^{n+1} - V_i^n) = V_i^n \tag{B.2}$$

$$\alpha(V_{i+1}^n - 2V_i^n + V_{i-1}^n) - V_i^{n+1} + V_i^n = \beta V_i^n \tag{B.3}$$

$$V_i^{n+1} = (1-\beta)V_i^n + \alpha(V_{i+1}^n - 2V_i^n + V_{i-1}^n) \tag{B.4}$$

For the extreme points, enforcing the boundary condition gives

$$V_0^{n+1} = (1-\beta-2\alpha)V_0^n + 2\alpha V_1^n \tag{B.5}$$

$$V_N^{n+1} = (1-\beta-2\alpha)V_N^n + 2\alpha V_{N-1}^n \tag{B.6}$$

Appendix B.2. Euler implicit

We start expanding the second space derivative using central differences at time t_{n+1} and the time derivative using backwards Euler. In this method the points at time t_n are implicitly defined from the ones at the next time-step.

$$\frac{\lambda^2}{dx^2}(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}) - \frac{\tau}{dt}(V_i^{n+1} - V_i^n) = V_i^{n+1} \tag{B.7}$$

$$\alpha(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}) - V_i^{n+1} + V_i^n = \beta V_i^{n+1} \tag{B.8}$$

$$(1+\beta)V_i^{n+1} - \alpha(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}) = V_i^n \tag{B.9}$$

$$\mathbf{A}\mathbf{V}^{n+1} = \mathbf{V}^n \tag{B.10}$$

Where $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{pmatrix} (1+\beta+2\alpha) & -2\alpha & 0 & \ldots & 0 \\ -\alpha & (1+\beta+2\alpha) & -\alpha & \ldots & 0 \\ 0 & -\alpha & (1+\beta+2\alpha) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\alpha \\ 0 & 0 & \ldots & -2\alpha & (1+\beta+2\alpha) \end{pmatrix} \tag{B.11}$$

The boundary conditions are enforced with the double α contribution on the first term of the upper diagonal and the last term of the lower diagonal.

Appendix B.3. Crank-Nicolson

We start expanding the second space derivative and the time derivative using central differences and taking the average of the results at times t_n and t_{n+1}.

$$\frac{\lambda^2}{2dx^2}(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1} + V_{i+1}^n - 2V_i^n + V_{i-1}^n) - \frac{\tau}{dt}(V_i^{n+1} - V_i^n) = \frac{1}{2}(V_i^{n+1} + V_i^n) \tag{B.12}$$

$$\frac{1}{2}\alpha(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1} + V_{i+1}^n - 2V_i^n + V_{i-1}^n) - V_i^{n+1} + V_i^n = \frac{1}{2}\beta(V_i^{n+1} + V_i^n) \tag{B.13}$$

$$(1 + \frac{1}{2}\beta)V_i^{n+1} - \frac{1}{2}\alpha(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}) = \frac{1}{2}\alpha(V_{i+1}^n - 2V_i^n + V_{i-1}^n) + (1 - \frac{1}{2}\beta)V_i^n \tag{B.14}$$

$$\mathbf{A}\mathbf{V}^{n+1} = \mathbf{B}\mathbf{V}^n \tag{B.15}$$

where $\mathbf{A}$ is defined as

$$\mathbf{A} = \begin{pmatrix} (1 + \frac{1}{2}\beta + \alpha) & -\alpha & 0 & \dots & 0 \\ -\frac{1}{2}\alpha & (1 + \frac{1}{2}\beta + \alpha) & -\frac{1}{2}\alpha & \dots & 0 \\ 0 & -\frac{1}{2}\alpha & (1 + \frac{1}{2}\beta + \alpha) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\frac{1}{2}\alpha \\ 0 & 0 & \dots & -\alpha & (1 + \frac{1}{2}\beta + \alpha) \end{pmatrix} \tag{B.16}$$

and $\mathbf{B}$ as

$$\mathbf{B} = \begin{pmatrix} (1 - \frac{1}{2}\beta - \alpha) & \alpha & 0 & \dots & 0 \\ \frac{1}{2}\alpha & (1 - \frac{1}{2}\beta - \alpha) & \frac{1}{2}\alpha & \dots & 0 \\ 0 & \frac{1}{2}\alpha & (1 - \frac{1}{2}\beta - \alpha) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{1}{2}\alpha \\ 0 & 0 & \dots & \alpha & (1 - \frac{1}{2}\beta - \alpha) \end{pmatrix} \tag{B.17}$$

The boundary conditions are enforced with the double α contribution on the first term of the upper diagonal and the last term of the lower diagonal in both matrices. The method is implemented defining $\mathbf{C} = \mathbf{A}^{-1}\mathbf{B}$ which is evaluated only once and then $\mathbf{C}$ is applied to the vector to obtain the next iteration.

Appendix B.4. Analytical solution

For the unbounded problem in Eq. 3, there is an analytical solution given by

$$V(x,t) = \frac{\tilde{V}_0}{\sqrt{4\pi(\lambda^2/\tau)t}} \exp\left[-\frac{(x - x_0)^2}{4(\lambda^2/\tau)t} - \frac{t}{\tau}\right] \tag{B.18}$$

This was implemented by evaluating the gaussain at the discretized points x_i, using $\tilde{V}_0 = 1$. Since the initial impulse is normalized on $[0, 1]$, the normalization is chosen to also have a potential that integrates to 1 on the same interval. The units are $[V][L]^{-1}$, directly from dimensional analysis on the equation and also from the fact that its integral in space needs to be a potential.

Since this is an unbounded solution, it will be an approximation of the real potential that is valid for small times since the potential at the boundaries is basically zero initially. We can get an idea of the validity, by requiring that the potential at the boundaries is very small compared to the peak of the gaussian, that is $V(0,t) << V(x_0,t)$. Calling $C(t) = 4(\lambda^2/\tau)t$

$$\frac{V(0,t)}{V(x_0,t)} << 1 \implies e^{-x_0^2/C(t)} << 1 \implies x_0^2 >> \frac{4\lambda^2 t}{\tau} \implies t << \frac{\tau L^2}{16\lambda^2} = 0.0625 \tag{B.19}$$

Which means that looking at a simulation, the effects should already be visible from times that are a fraction of this value.

Appendix C. Transmission with ion channels

In this Appendix, the implementation of Crank-Nicolson to solve Eq. 4 is presented. We are assuming to work in $[0, L]$ with vectors discretized in N points $x_i = idx$ with $i = 0, 1, \ldots, N-1$, such that $dx = L/(N-1)$. Neumann boundary conditions are enforced. Also, the notation used is

$$V_i^n = V(t_n, x_i).$$

Some intermediate factors α and β are defined as

$$\alpha = \frac{\lambda^2 dt}{\tau dx^2} \qquad \beta = \frac{dt}{\tau}. \tag{C.1}$$

We recall the expression of the Na^+ channels permeability

$$g_{\mathrm{Na}}(V) = \frac{100}{1 + e^{\gamma(V^* - V)}} + 0.2\,, \tag{C.2}$$

and define $\tilde{g}_{\mathrm{Na}}(V) = g_{\mathrm{Na}}(V)/g_{\mathrm{K}}$. The relevant parameters are: $\lambda = 0.18$ mm, $\tau = 2.0$ ms, $\gamma = 0.5$ mV^{-1}, V^*=-40 mV, $\mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}} = 56$ mV and $\mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}} = -76$ mV, $g_{\mathrm{K}} = 5.0\,\Omega^{-1}\ \mathrm{m}^{-2}$. The initial impulse is generated as descripted in Eq. 6.

Appendix C.1. Crank-Nicolson implementation

We start expanding the second space derivative and the time derivative using central differences and taking the average of the results at times t_n and t_{n+1}.

$$\frac{\lambda^2}{2dx^2}(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1} + V_{i+1}^n - 2V_i^n + V_{i-1}^n) - \frac{\tau}{dt}(V_i^{n+1} - V_i^n) = \tag{C.3}$$

$$\frac{1}{2}\tilde{g}_{\mathrm{Na}}(V_i^{n+1})\left[V_i^{n+1} - \mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}}\right] + \frac{1}{2}\tilde{g}_{\mathrm{Na}}(V_i^n)\left[V_i^n - \mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}}\right] + \frac{1}{2}\left[V_i^{n+1} - \mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}}\right] + \frac{1}{2}\left[V_i^n - \mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}}\right] \tag{C.4}$$

$$\tag{C.5}$$

$$\frac{1}{2}\alpha(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1} + V_{i+1}^n - 2V_i^n + V_{i-1}^n) - V_i^{n+1} + V_i^n = \tag{C.6}$$

$$\frac{1}{2}\beta\tilde{g}_{\mathrm{Na}}(V_i^{n+1})\left[V_i^{n+1} - \mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}}\right] + \frac{1}{2}\beta\tilde{g}_{\mathrm{Na}}(V_i^n)\left[V_i^n - \mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}}\right] + \frac{1}{2}\beta\left[V_i^{n+1} - \mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}}\right] + \frac{1}{2}\beta\left[V_i^n - \mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}}\right] \tag{C.7}$$

$$\tag{C.8}$$

$$[1 + \frac{1}{2}\beta(1 + \tilde{g}_{\mathrm{Na}}(V_i^{n+1}))]V_i^{n+1} - \frac{1}{2}\alpha(V_{i+1}^{n+1} - 2V_i^{n+1} + V_{i-1}^{n+1}) = \tag{C.9}$$

$$\frac{1}{2}\alpha(V_{i+1}^n - 2V_i^n + V_{i-1}^n) + [1 - \frac{1}{2}\beta(1 + \tilde{g}_{\mathrm{Na}}(V_i^n))]V_i^n + \frac{1}{2}\beta(\tilde{g}_{\mathrm{Na}}(V_i^{n+1}) + \tilde{g}_{\mathrm{Na}}(V_i^n))\mathcal{V}_{\mathrm{Na}}^{\mathrm{Nernst}} + \beta\mathcal{V}_{\mathrm{K}}^{\mathrm{Nernst}} \tag{C.10}$$

$$\mathbf{A}^{[n]}\mathbf{V}^{[n+1]} = \mathbf{B}^{[n]}\mathbf{V}^{[n]} + \mathbf{q}^{[n]} \tag{C.11}$$

To define the problem as a linear system, we assume that $\tilde{g}_{\mathrm{Na}}(V_i^{n+1}) \approx \tilde{g}_{\mathrm{Na}}(V_i^n) \equiv \tilde{g}_i^n$ which is a good approximation if we take small enough time-steps such that V is varying slow enough. This gives $\mathbf{A}^{[n]}$ defined as

$$\mathbf{A}^{[n]} = \begin{pmatrix} (1 + \frac{1}{2}\beta(1 + \tilde{g}_1^n) + \alpha) & -\alpha & 0 & \ldots & 0 \\ -\frac{1}{2}\alpha & (1 + \frac{1}{2}\beta(1 + \tilde{g}_2^n) + \alpha) & -\frac{1}{2}\alpha & \ldots & 0 \\ 0 & -\frac{1}{2}\alpha & (1 + \frac{1}{2}\beta(1 + \tilde{g}_3^n) + \alpha) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\frac{1}{2}\alpha \\ 0 & 0 & \ldots & -\alpha & (1 + \frac{1}{2}\beta(1 + \tilde{g}_N^n) + \alpha) \end{pmatrix} \tag{C.12}$$

and $\mathbf{B}^{[n]}$ as

$$\mathbf{B}^{[n]} = \begin{pmatrix} (1-\frac{1}{2}\beta(1+\tilde{g}_1^n)-\alpha) & \alpha & 0 & \dots & 0 \\ \frac{1}{2}\alpha & (1-\frac{1}{2}\beta(1+\tilde{g}_2^n)-\alpha) & \frac{1}{2}\alpha & \dots & 0 \\ 0 & \frac{1}{2}\alpha & (1-\frac{1}{2}\beta(1+\tilde{g}_3^n)-\alpha) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{1}{2}\alpha \\ 0 & 0 & \dots & \alpha & (1-\frac{1}{2}\beta(1+\tilde{g}_N^n)-\alpha) \end{pmatrix} \tag{C.13}$$

Finally, $\mathbf{q}^{[n]}$ is a vector defined as

$$\mathbf{q}^{[n]} = \beta \begin{pmatrix} \tilde{g}_1^n \mathcal{V}_{\text{Na}}^{\text{Nernst}} + \mathcal{V}_{\text{K}}^{\text{Nernst}} & \tilde{g}_2^n \mathcal{V}_{\text{Na}}^{\text{Nernst}} + \mathcal{V}_{\text{K}}^{\text{Nernst}} & \dots & \tilde{g}_N^n \mathcal{V}_{\text{Na}}^{\text{Nernst}} + \mathcal{V}_{\text{K}}^{\text{Nernst}} \end{pmatrix}^T \tag{C.14}$$

The boundary conditions are enforced with the double α contribution on the first term of the upper diagonal and the last term of the lower diagonal in both matrices. Differently from the implementation in Appendix B, this time the matrices and the vector $\mathbf{q}$ are changing at each time-step n and thus needs to be updated at each step. To solve the system at $\mathbf{V}^{[n+1]}$, the intermediate $\tilde{\mathbf{V}}^{[n]} = \mathbf{B}^{[n]}\mathbf{V}^{[n]} + \mathbf{q}^{[n]}$ is evaluated and then $\mathbf{A}^{[n]}\mathbf{V}^{[n+1]} = \tilde{\mathbf{V}}^{[n]}$ is solved inverting the matrix.

Appendix C.2. Introducing termination of activation and afterhyperpolarization

Following the idea presented in section 3.4, a new implementation of the Crank-Nicolson scheme is introduced starting from the one in Appendix C.1, with the objective of introducing the termination of the activation. This is done by keeping a logger that keeps track of the time that is passed at each point from the activation. It is initialized as N-dimensional zero vector and once the permeability at some position reach a value of 15, that position is considered active and after 50 time-steps (which corresponds to 0.5 ms of activation, as found in Ref. [1]) the permeability is set to zero, simulating the closure of the Na^+ channels. The logger is still tracking the time and after 3 ms from the activation, when the potential is back under the threshold potential, the time in the logger is set to zero and the permeability follows again Eq. C.2. At any time, the permeability for $x < 1.0$ mm is kept constant to zero, as done previously to simulate an impulse coming from the body of the cell.
To add the effect of the afterhyperpolarization, a factor $1 + \mu_i^n$ is introduced in Eq. 4 on the term representing the K^+ channels. The coefficients is defined as $\mu_i^n = 0.03\, T_i^n$ where T_i^n is the number of time-steps in the logger, which means that is a vector dependent on position and time that is linearly increasing from the activation up to 300 steps later, when the logger is set back to zero. This is done with a small pre-factor to simulate the slow increased effect of the K^+ channels, that is still active for a few ms after the effect of the Na^+ channels is removed.

$$\lambda^2 \frac{\partial^2 V(x,t)}{\partial x^2} - \tau \frac{\partial V(x,t)}{\partial t} = \frac{g_{\text{Na}}(V(x,t))}{g_{\text{K}}} \left[V(x,t) - \mathcal{V}_{\text{Na}}^{\text{Nernst}} \right] + (1+\mu) \left[V(x,t) - \mathcal{V}_{\text{K}}^{\text{Nernst}} \right] \tag{C.15}$$

$$\lambda^2 \frac{\partial^2 V(x,t)}{\partial x^2} - \tau \frac{\partial V(x,t)}{\partial t} = \tilde{g}_{\text{Na}}(V(x,t)) \left[V(x,t) - \mathcal{V}_{\text{Na}}^{\text{Nernst}} \right] + \left[V(x,t) - \mathcal{V}_{\text{K}}^{\text{Nernst}} \right] \tag{C.16}$$

To reduce the amount of changes done to the code, the new coefficient is introduced as an additional term in the definition of $\tilde{g}_{\text{Na}}(V_i^n)$, which allow to use the same solver defined in Appendix C.1 with the modified

$$\tilde{g}_{\text{Na}}(V_i^n) = \frac{g_{\text{Na}}(V_i^n)}{g_{\text{K}}} + \mu_i^n \frac{V_i^n - \mathcal{V}_{\text{K}}^{\text{Nernst}}}{V_i^n - \mathcal{V}_{\text{Na}}^{\text{Nernst}}} \tag{C.17}$$

It is important to clarify that when the logger reach 50 time-steps of activation, only the first term in Eq. C.17 is set to zero, while the second keeps linearly increasing. After 300 steps the first term is effective again (even if now the potential is lower than the threshold) and the logger is brought back to zero, which means that the second term is now zero. Also, everything that was proposed in this scheme are only suggested changes based on the author interpretation and understanding of the phenomenon, based on experimental observations in Ref. [1]. A more in depth study is necessary to ensure the validity of the assumptions that are made.