NUMERICAL STUDY OF A ONE-DIMENSIONAL POISSON-NERNST–PLANCK ION CHANNEL MODEL BY FINITE ELEMENT BACKWARD AND FORWARD EULER METHODS

by

Michel Stanislas Korfhage

A Thesis Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Master of Science
in Mathematics

at
The University of Wisconsin-Milwaukee
May 2023

ABSTRACT

NUMERICAL STUDY OF A ONE-DIMENSIONAL POISSON-NERNST–PLANCK ION CHANNEL MODEL BY FINITE ELEMENT BACKWARD AND FORWARD EULER METHODS

by

Michel Stanislas Korfhage

The University of Wisconsin-Milwaukee, 2023
Under the Supervision of Professor Dexuan Xie

This thesis presents a numerical study of a one-dimensional Poisson-Nernst-Planck (PNP) ion channel model, which describes the transport of charged species in an electrolyte under the influence of an electric field. We develop a new numerical scheme for solving the PNP model by combining the method of lines with the finite element and Euler's forward and backward methods. We then implement the scheme based on the finite element library from the FEniCS project. To validate the accuracy of our numerical scheme, we construct an analytical solution of the PNP model with source terms. We find in numerical tests that the backward Euler method is more accurate and stable than the forward Euler method, especially for larger time steps. Furthermore, we use our numerical scheme to investigate the properties of the PNP model for an electrolyte with two ionic species. Our numerical results show that our numerical scheme can accurately capture the solution behavior of the PNP model.

Keywords: Poisson-Nernst-Planck equations, ion channel modeling, finite element method, Euler methods, FEniCS library.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to acknowledge and thank the members of my committee, Prof. Istvan Lauko, Prof. Lei Wang, and Prof. Dexuan Xie.

I would like to express my deepest gratitude to Prof. Dexuan Xie for providing me with the opportunity to work on this thesis project under his guidance. Throughout the work on my thesis, he provided me with lots of insights and was always available for help, when I needed assistance.

I am also grateful to my parents, for their emotional and financial support throughout my studies. It would have been impossible to finish my studies abroad without them.

1 Introduction

Cell membranes contain macromolecular pores, called ion channels. They are crucial for the electrical signaling in nerves, muscles, and synapses. A variety of different channels cooperate by opening and closing to support the nervous system [6, p. 1 f.]. These different channels are more or less permeable for some ions. E.g. Na channels transport many Na^+ ions. For many applications, it is important to model these channels by differential equations.

The Poisson-Nernst-Planck (PNP) equations are widely used to predict the behavior of electrochemical systems, such as ion channels and fuel cells, and the transport of ions in an electrolyte under the influence of an electric field [5], [10], [13]. It allows to numerically calculate the electrostatic potential and ionic concentration functions at a given point in space and time. In this thesis, we investigate the numerical solution of a one-dimensional PNP ion channel model using the method of lines, which is a technique for solving partial differential equations (PDEs) by discretizing them in space and then in time. Specifically, we use the finite element method (FEM) and the forward and backward Euler methods to discretize the spatial domain and time interval, respectively. We then develop a PNP numerical solver and implement it as a Python package based on a finite element library from the FEniCS project [3]. Moreover, we use this package to study the behavior of the PNP model under different conditions and compare the effects of various parameters on the solutions. To validate our Python package and numerical solver, we construct analytical solutions of the PNP model.

The thesis will start by reviewing the PNP model. We will then discretize it in the spatial domain as a variational problem using the finite element method, and in the time dimension by using the forward and backward Euler methods. We next introduce an iterative scheme for solving each nonlinear finite element system generated from the

backward Euler method. We further describe how we implement and analyze these methods. In the end, we conclude that the backward Euler method is able to produce better approximations of the PNP model's solution than the forward Euler method.

2 The PNP ion channel model

The transport of ions can be determined by Nernst-Planck equations and Poisson's equation, yielding the concentrations of ions and the electric field within the channel [6, p. 315 ff.].

In previous work [14, p. 4 ff.], the three-dimensional coordinate system in space was reduced to one dimension by selecting the normal direction of the membrane as the axis to be examined. The potential density function, $u(t,z)$, and the ionic concentration functions, $c_i(t,z)$, then became functions of one spatial variable, z, on an interval $0 \leq z \leq L$ with L representing the membrane channel length. The time variable, t, will be within the interval $[0,t_e]$ with t_e being the time after which the partial derivative of c_i with respect to t is almost zero, meaning that the model has reached the steady state with a stationary solution (i.e. independent of time). In this work, we consider a non-stationary potential u and n concentrations c_i as functions of z and t and define them by a one-dimensional PNP ion channel model as follows:

$$\frac{\partial c_i(t,z)}{\partial t} = \frac{\partial}{\partial z} D_i [Z_i c_i(t,z) \frac{\partial u}{\partial z} + \frac{\partial c_i(t,z)}{\partial z}], \quad i = 1,2,\ldots,n, \tag{1a}$$

$$-\epsilon_s \frac{\partial^2 u(t,z)}{\partial z^2} = \beta \sum_{i=1}^{n} Z_i c_i(t,z) + \rho(z), \tag{1b}$$

subject to the initial value conditions

$$c_i(0,z) = g_i(z), \quad 0 \leq z \leq L, \quad i = 1,2,\ldots,n, \tag{1c}$$

$$u(0,z) = g(z), \quad 0 \leq z \leq L, \tag{1d}$$

and the Dirichlet boundary value conditions

$$c_i(t,0) = c_{i,0}(t), \quad c_i(t,L) = c_{i,L}(t), \qquad 0 \le t \le t_e, \quad i = 1,2,\ldots,n, \tag{1e}$$

$$u(t,0) = u_0(t), \quad u(t,L) = u_L(t), \qquad 0 \le t \le t_e, \tag{1f}$$

where ρ denotes a permanent charge function; D_i is a diffusion function of species i; Z_i is the charge of species i; n is the number of species in the ionic solution; ϵ_s is the water permittivity constant; g_i and g are initial value functions; $c_{i,0}$, $c_{i,L}$, u_0, u_L are boundary value functions; and β is a physical constant.

3 Formulation of the finite element problem

In order to perform a full discretization of the PNP model, we use the method of lines. Following the method of lines, we first discretize the PNP model in the spatial dimension using a finite element method. Leaving the time variable t as a parameter, we reduced the PNP model to an ordinary differential equation (ODE) system [12, p.10]. This ODE system is then discretized by using the forward or backward Euler method.

3.1 Semi discretization in space

We start with a semi-discretization in space. Let $\Omega = (0, L)$ and $V = H^1(\Omega)$, where $H^1(\Omega)$ is a Sobolev Space on Ω. A function f of $H^1(\Omega)$ is in the Lebesgue space $L^2(\Omega)$ and its weak partial derivative $\frac{\partial f}{\partial z}$ is again in $L^2(\Omega)$ space [7, p. 49ff]. We also set $V_0 = H_0^1(\Omega)$, which is defined by

$$H_0^1(\Omega) = \{f \in H^1(\Omega) | f = 0 \text{ on the boundary } \partial\Omega \text{ of } \Omega\}$$

For any test function $v \in V_0$, we get a variational formulation of the PNP model (1) as follows:

$$\int_0^L \frac{\partial c_i(t,z)}{\partial t} v dz = \int_0^L \frac{\partial}{\partial z} D_i [Z_i c_i(t,z) \frac{\partial u}{\partial z} + \frac{\partial c_i(t,z)}{\partial z}] v dz, \quad i = 1, 2, \ldots, n \tag{2a}$$

$$-\epsilon_s \int_0^L \frac{\partial^2 u(t,z)}{\partial z^2} v dz = \int_0^L [\beta \sum_{i=1}^{n} Z_i c_i(t,z) + \rho(z)] v dz \tag{2b}$$

By applying integration by parts,

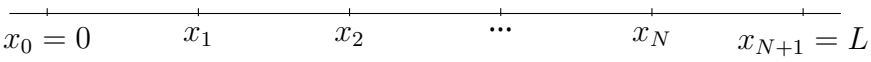
$$\int_0^L v du = -\int_0^L u dv$$

(2) is simplified as:

$$\int_0^L \frac{\partial c_i(t,z)}{\partial t} v dz = -\int_0^L v' D_i [Z_i c_i(t,z) \frac{\partial u}{\partial z} + \frac{\partial c_i(t,z)}{\partial z}] dz, \quad i = 1, 2, \ldots, n, \tag{3a}$$

$$\epsilon_s \int_0^L v' \frac{\partial u(t,z)}{\partial z} dz = \beta \sum_{i=1}^{n} Z_i \int_0^L c_i(t,z) v dz + \int_0^L \rho(z) v dz. \tag{3b}$$

We next construct a linear finite element function space, V_h, based on a mesh partition of Ω, $x_j = jh$ with the mesh size $h = \frac{L}{N+1}$ for $j = 0, 1, 2, \ldots, N, N+1$,



such that for each $u_h \in V_h$, u_h is a piecewise linear function,

$$u_h(x) = a_j x + b_j \quad \text{for } x_{j-1} < x < x_j, \quad j = 1, 2, \ldots, N+1$$

and is continuous in $\Omega = (0, L)$. We then obtain a system of nonlinear finite element equations as follows: Find $u(\cdot, t) \in V_h, \quad c_i(\cdot, t) \in V_h$, satisfying the boundary conditions

$$c_i(t, 0) = c_{i,0}(t), \quad c_i(t, L) = c_{i,L}(t), \quad 0 \le t \le t_e, \quad i = 1, 2, \ldots, n,$$

$$u(t, 0) = u_0(t), \quad u(t, L) = u_L(t), \quad 0 \le t \le t_e,$$

such that for any $v \in V_{h,0}$,

$$\begin{aligned} \int_0^L \frac{\partial c_i(t,z)}{\partial t} v dz &= -\int_0^L v' D_i [Z_i c_i(t,z) \frac{\partial u}{\partial z} + \frac{\partial c_i(t,z)}{\partial z}] v dz, \quad i = 1, 2, \ldots, n \\ \epsilon_s \int_0^L v' \frac{\partial u(t,z)}{\partial z} dz &= \beta \sum_{i=1}^{n} Z_i \int_0^L c_i(t,z) v dz + \int_0^L \rho(z) v dz, \end{aligned} \tag{4}$$

where $V_{h,0} = \{v \in V_h | v(0) = 0, v(L) = 0\}$.

3.2 Forward Euler method

We now consider time discretization of the finite element system (4). We construct a time interval partitioning, $t_k = k\tau$ for $k = 0, 1, 2, \ldots$, with τ being the time step size. In the following, the function at the k-th timestep, t_k, will be denoted in the following notation:

$$c_i^k(z) = c_i(t_k, z), \quad u^k(z) = u(t_k, z).$$

In the forward Euler method, the derivative at timestep t_k is approximated by

$$\frac{dc_i^k(z)}{dt} \approx \frac{c_i^{k+1}(z) - c_i^k(z)}{\tau} \tag{5}$$

We can apply (5) to (3a) to get

$$\int_0^L \frac{c_i^{k+1} - c_i^k}{\tau} v dz + D_i \int_0^L v'(Z_i c_i^k \frac{du^k}{dz} + \frac{dc_i^k}{dz}) dz = 0, \quad i = 1, 2, \ldots, n \tag{6a}$$

which can be reformulated as

$$\int_0^L c_i^{k+1} v dz = \int_0^L c_i^k v dz - \tau D_i \int_0^L v'(Z_i c_i^k \frac{du^k}{dz} + \frac{dc_i^k}{dz}) dz, \quad i = 1, 2, \ldots, n \tag{6b}$$

And for the potential function at $t = t_{k+1}$, $u^{k+1}(z)$, from (3b), we can get

$$\epsilon_s \int_0^L v' \frac{du^{k+1}(z)}{dz} dz = \beta \sum_{i=1}^n Z_i \int_0^L c_i^{k+1}(z) v dz + \int_0^L \rho(z) v dz \tag{7}$$

From the initial value conditions of (1c), and (1d), we can get

$$c_i^0(z) = g_i(z), \quad u^0(z) = g(z).$$

Therefore, the forward Euler method for solving the PNP model (1) at time step t_{k+1}, for $k = 0, 1, 2, \ldots$, is given as follows:

For $i = 1, 2, \ldots, n$, we first find $c_i^{k+1} \in V_h$, which satisfy the boundary value conditions,

$$c_i^{k+1}(0) = c_{i,0}, \quad c_i^{k+1}(L) = c_{i,L},$$

such that

$$\int_0^L c_i^{k+1} v dz = \int_0^L c_i^k v dz - \tau D_i \int_0^L v'(Z_i c_i^k \frac{du^k}{dz} + \frac{dc_i^k}{dz}) dz, \quad \forall v \in V_{h,0}. \tag{8a}$$

We then find $u^{k+1} \in V_h$ satisfying the boundary value conditions,

$$u^{k+1}(0) = u_0, \quad u^{k+1}(L) = u_L,$$

such that

$$\epsilon_s \int_0^L v' \frac{du^{k+1}(z)}{dz} dz = \beta \sum_{i=1}^n Z_i \int_0^L c_i^{k+1}(z) v dz + \int_0^L \rho(z) v dz, \quad \forall v \in V_{h,0}. \tag{8b}$$

3.3 Backward Euler method

In the backward Euler method, the derivative at timestep t_{k+1} is approximated by

$$\frac{dc_i^{k+1}(z)}{dt} \approx \frac{c_i^{k+1} - c_i^k}{\tau} \tag{9}$$

By setting $t = t_{k+1}$ and applying (9) to (3a), we get

$$\int_0^L \frac{c_i^{k+1} - c_i^k}{\tau} v dz + D_i \int_0^L v'(Z_i c_i^{k+1} \frac{du^{k+1}}{dz} + \frac{dc_i^{k+1}}{dz}) dz = 0, \qquad i = 1, 2, \ldots, n \tag{10a}$$

$$\Leftrightarrow \int_0^L c_i^{k+1} v dz + \tau D_i \int_0^L v'(Z_i c_i^{k+1} \frac{du^{k+1}}{dz} + \frac{dc_i^{k+1}}{dz}) dz = \int_0^L c_i^k v dz, \qquad i = 1, 2, \ldots, n \tag{10b}$$

Similarly, we get an equation of u^{k+1} from (3b) as follows:

$$\epsilon_s \int_0^L v' \frac{du^{k+1}(z)}{dz} dz - \beta \sum_{i=1}^{n} Z_i \int_0^L c_i^{k+1}(z) v dz = \int_0^L \rho(z) v dz \tag{11}$$

Hence, the backward Euler method for solving the PNP model (1) in the finite element method is obtained as follows:

For $k = 0, 1, 2, \ldots,$ find $u^{k+1} \in V_h$, and $c_i^{k+1} \in V_h, \quad$ for $i = 1, 2, \ldots, n$ satisfying the boundary value conditions

$$u^{k+1}(0) = u_0, \quad u^{k+1}(L) = u_L,$$

$$c_i^{k+1}(0) = c_{i,0}, \quad c_i^{k+1}(L) = c_{i,L},$$

such that for all $v \in V_{h,0}$,

$$\int_0^L c_i^{k+1} v dz + \tau D_i \int_0^L v' (Z_i c_i^{k+1} \frac{du^{k+1}}{dz} + \frac{dc_i^{k+1}}{dz}) dz = \int_0^L c_i^k v dz, \quad i = 1, 2, \ldots, n, \tag{12a}$$

$$\epsilon_s \int_0^L v' \frac{du^{k+1}(z)}{dz} dz - \beta \sum_{i=1}^{n} Z_i \int_0^L c_i^{k+1}(z) v dz = \int_0^L \rho(z) v dz, \tag{12b}$$

where the initial value conditions are given by

$$c_i^0(z) = g_i(z), \quad u^0(z) = g(z).$$

Hence, at each time step, a nonlinear system of (12) with $n + 1$ unknown functions, $\{c_i^{k+1}\}_{i=1}^n$ and u^{k+1}, is required to be solved numerically.

4 Numerical scheme for solving nonlinear finite element systems

In order to solve the nonlinear finite element system (12), which is generated from the backward Euler method, we construct a damped iterative scheme [14]. Let $u^{k,(l)}$ and $c_i^{k,(l)}$ denote the l-th iterates of u^k and c_i^k, at time step k respectively, for $k = 1, 2, \ldots$. With the initial iterates $u^{0,(0)}$ and $c_i^{0,(0)}$ being known from the initial value conditions, we define the updates $u^{k,(l+1)}$ and $c_i^{k,(l+1)}$ as follows: For $l = 0, 1, 2, \ldots,$

$$c_i^{k,(l+1)} = c_i^{k,(l)} + \omega(p_i - c_i^{k,(l)}), \quad i = 1, 2, \ldots, n, \tag{13a}$$

$$u^{k,(l+1)} = u^{k,(l)} + \omega(q - u^{k,(l)}), \tag{13b}$$

where ω denotes a damping parameter within the range $(0, 1)$. p_i is a solution to the linear finite element variational problem: Find $p_i \in V_h$ satisfying $p_i(0) = c_{i,0}$ and $p_i(L) = c_{i,L}$ such that:

$$\int_0^L p_i(z)v dz + \tau D_i \int_0^L v'(Z_i p_i(z)\frac{du^{k,(l)}(z)}{dz} + \frac{dp_i(z)}{dz})dz = \int_0^L c_i^{k,(l)}(z)v dz, \quad \forall v \in V_{h,0}, \tag{14}$$

q is a solution to the linear finite element variational problem: Find $q \in V_h$, satisfying the boundary value conditions $q(0) = u_0$ and $q(L) = u_L$ such that

$$\epsilon_s \int_0^L v' \frac{dq(z)}{dz} dz = \beta \sum_{i=1}^{n} Z_i \int_0^L c_i^{k,(l+1)}(z)v dz + \int_0^L \rho(z)v dz, \quad \forall v \in V_{h,0}. \tag{15}$$

The initial iterates $c_i^{k,(0)}$ and $u^{k,(0)}$ are defined by

$$c_i^{k,(0)} = c_i^{k-1,(l_c)}, \tag{16a}$$

$$u^{k,(0)} = u^{k-1,(l_c)}, \tag{16b}$$

where l_c is the number of iterations at which the following iteration termination rule is satisfied:

$$\max_{1\leq i\leq n} ||c_i^{k,(l+1)} - c_i^{k,(l)}|| < \epsilon, \quad ||u^{k,(l+1)} - u^{k,(l)}|| < \epsilon, \tag{17}$$

where ϵ denotes a tolerance (e.g. $\epsilon = 10^{-5}$). The iteration will be performed for a maximum of 300 times unless the iteration termination rule (17) is satisfied.

5 Implementation and numerical results

We implemented our numerical scheme based on a popular open-source finite element library, DOLFIN, which has a high-level Python interface [3]. This work used the legacy DOLFIN library instead of the more recent DOLFINx. To simplify the installation of the library and ensure consistent results when executing on different machines, the code was executed inside a docker container by using a provided docker image [2], which includes a working installation of FEniCS version 2019.1.0. The implementation process was done on a Jupyter notebook, which is a simple notebook interface for executing Python code step by step and showing the results [11].

The implemented solver was executed for different test cases. Here we set the permanent charge function ρ to be a constant (-10), $n = 2$ for a solution of table salt NaCl with a diffusion constant $D_1 = 0.133$ for Na^+ ions, $D_2 = 0.203$ for Cl^- ions, $\epsilon_s = 78$, $\beta \approx 4.2414$, and the spatial partition number $N = 256$.

5.1 Test 1

In this test case, we set $L = 40$ to get a mesh size of $h = 0.15625$. The boundary conditions were defined as follows: For $k = 0, 1, 2, \ldots,$

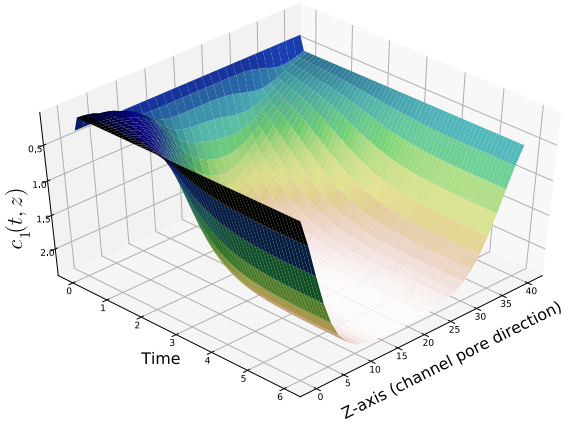
$$u^k(0) = -2, \quad u^k(L) = 2, \tag{18a}$$

$$c_1^k(0) = 0.1, \quad c_1^k(L) = 0.5, \tag{18b}$$

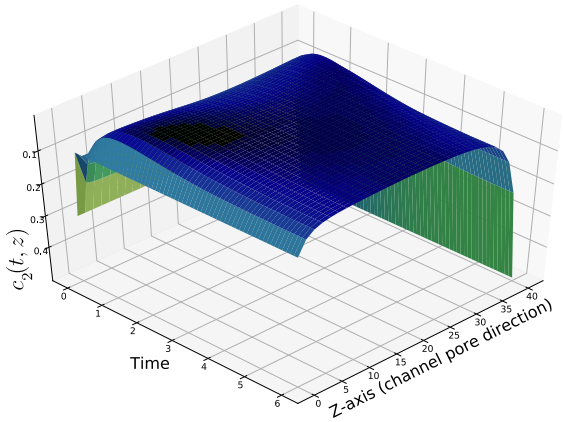
$$c_2^k(0) = 0.1, \quad c_2^k(L) = 0.5. \tag{18c}$$

The calculations were performed by the backward Euler method for time steps $\tau = 0.1$, $\tau = 0.05$, $\tau = 0.025$, $\tau = 0.0125$.

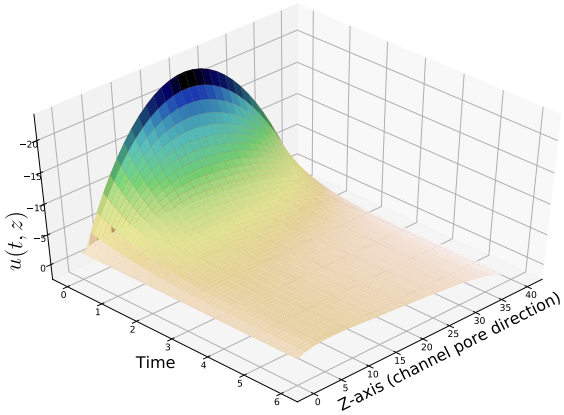
The finite element solution at $\tau = 0.1$ is reported in Figure 1, which converges to the steady solution of the PNP model, found in the previous work [14]. Independent of the



(a) Concentration of Na^+, $c_1(t,z)$, in Test 1



(b) Concentration of Cl^-, $c_2(t,z)$, in Test 1



(c) Potential, $u(t,z)$, in Test 1

Figure 1: Finite element solution (c_1,c_2,u) of the PNP model in Test 1 with $\tau = 0.1$.

selected τ, the solver converged to the mentioned steady-state solution. However, smaller time steps increased the time the scheme needed to calculate the solution. At $\tau = 0.1$, 70 seconds CPU time was used. It increased to 415 seconds for $\tau = 0.0125$.

5.2 Construction of analytical solutions for verification of numerical results

To be able to verify the correctness of numerical solutions over the entire time interval, we construct two analytic solutions of the PNP model with source terms. Similar work was done in [9] for the following PNP equations with source terms $f_i, \quad i = 1, 2, \ldots, n+1$:

$$\frac{\partial c_i(t,z)}{\partial t} = \frac{1}{A(z)}\frac{\partial}{\partial z}A(z)D_i[Z_i c_i \frac{\partial \psi}{\partial z} + \frac{\partial c_i}{\partial z}] + f_i(t,z), \quad i = 1, 2, ..., n, \tag{19a}$$

$$-\frac{1}{A(z)}\frac{\partial}{\partial z}(\epsilon_s A(z)\frac{\partial \psi}{\partial z}) = \beta \sum_{i=1}^{n} Z_i c_i + \rho(z) + f_{n+1}(t,z), \tag{19b}$$

where $A(z)$ is a cross-sectional area on a 3D geometry of the ion channel. We set $A(z) = 1$, $L = 1$, and $n = 2$ for our model,

$$\frac{\partial c_1(t,z)}{\partial t} = \frac{\partial}{\partial z}D_1[Z_1 c_1 \frac{\partial \psi}{\partial z} + \frac{\partial c_1}{\partial z}] + f_1(t,z), \tag{20a}$$

$$\frac{\partial c_2(t,z)}{\partial t} = \frac{\partial}{\partial z}D_2[Z_2 c_2 \frac{\partial \psi}{\partial z} + \frac{\partial c_2}{\partial z}] + f_2(t,z), \tag{20b}$$

$$-\epsilon_s \frac{\partial^2 \psi}{\partial z^2} = \beta[Z_1 c_1 + Z_2 c_2] + \rho(z) + f_3(t,z), \tag{20c}$$

where $0 \leq z \leq 1$ and $t \geq 0$. We need to determine the source terms f_1,f_2, and f_3, the boundary value conditions, and the initial value conditions to yield a PNP test model with an analytical solution.

5.2.1 Analytical solution A

We choose the analytical solution (c_1, c_2, u) to be the same as in [9, p.13]:

$$c_1(t,z) = z^2(1-z)e^{-t}, \tag{21a}$$

$$c_2(t,z) = z^2(1-z)^2e^{-t}, \tag{21b}$$

$$\psi(t,z) = -\frac{z^5(3-2z)}{60}e^{-t}. \tag{21c}$$

We can get their partial derivatives as follows:

$$\frac{\partial c_1(t,z)}{\partial t} = -c_1(t,z), \qquad \frac{\partial c_1(t,z)}{\partial z} = (2z-3z^2)e^{-t}, \tag{22a}$$

$$\frac{\partial c_2(t,z)}{\partial t} = -c_2(t,z), \qquad \frac{\partial c_2(t,z)}{\partial z} = (4z^3-6z^2+2z)e^{-t}, \tag{22b}$$

$$\frac{\partial \psi(t,z)}{\partial z} = -\frac{z^4(5-4z)}{20}e^{-t}, \qquad \frac{\partial^2 \psi(t,z)}{\partial z^2} = -z^3(-z+1)e^{-t}. \tag{22c}$$

Applying these derivatives to (20) yields the following source terms:

$$f_1 = (z^3 - z^2 + D_1(6z-2))e^{-t} + D_1Z_1\frac{32z^7-63z^6+30z^5}{20}e^{-2t}, \tag{23a}$$

$$\begin{aligned} f_2 =& [-z^4+2z^3-(1+12D_2)z^2+D_2(12z+2)]e^{-t} \\ &+ D_2Z_2\frac{-18z^8+52z^7-49z^6+15z^5}{10}e^{-2t}, \end{aligned} \tag{23b}$$

$$f_3 = [-(\epsilon_s+\beta Z_2)z^4 + (\epsilon_s+\beta(Z_1+2Z_2))z^3 - \beta(Z_1+Z_2)z^2]e^{-t} - \rho(z). \tag{23c}$$

From the exact solutions of (21), the boundary conditions, for $L = 1$, can be derived

as follows:

$$c_{1,0}(t) = 0, \qquad c_{1,L}(t) = (-L^3 + L^2)e^{-t} = 0, \tag{24a}$$

$$c_{2,0}(t) = 0, \qquad c_{2,L}(t) = (L^4 - 3L^3 + L^2)e^{-t} = -e^{-t}, \tag{24b}$$

$$\psi_0(t) = 0, \qquad \psi_L(t) = \frac{3L^5 - 2L^6}{60}e^{-t} = \frac{1}{60}e^{-t}. \tag{24c}$$

And the initial value conditions:

$$c_1(0, z) = z^2(1 - z), \tag{25a}$$

$$c_2(0, z) = z^2(1 - z)^2, \tag{25b}$$

$$\psi(0, z) = -\frac{z^5(3 - 2z)}{60}. \tag{25c}$$

A combination of (20) with (23), (24), (25) gives an initial value problem with an analytical solution. This initial value problem however has a zero steady solution $c_1 = 0$, $c_2 = 0$, $\psi = 0$, throughout the entire spatial domain, which does not make any sense in physics. We, therefore, need to find another initial value problem to evaluate the validity of the model.

5.2.2 Analytical solution B

We construct analytical solution B in the expressions:

$$c_1(t, z) = 2z^3 - 3z^2 + 5ze^{-t}, \tag{26a}$$

$$c_2(t, z) = 3z^3 - 2z + 4(L - z)e^{-t}, \tag{26b}$$

$$\psi(t, z) = 4z^3 - 7z^2 + z(3 - 2z)e^{-t}. \tag{26c}$$

Clearly,

$$\lim_{t\to\infty} c_1(t,z) = 2z^3 - 3z^2, \tag{27a}$$

$$\lim_{t\to\infty} c_2(t,z) = 3z^3 - 2z, \tag{27b}$$

$$\lim_{t\to\infty} \psi(t,z) = 4z^3 - 7z^2. \tag{27c}$$

which give a nonzero steady solution in cubic terms. A graph of the solution given in (26) is displayed in Figure 2.

With $L = 1$, the partial derivatives can be found as follows:

$$\frac{\partial c_1(t,z)}{\partial t} = -5ze^{-t}, \qquad \frac{\partial c_1(t,z)}{\partial z} = 6z^2 - 6z + 5e^{-t}, \tag{28a}$$

$$\frac{\partial c_2(t,z)}{\partial t} = 4(z-L)e^{-t} = 4(z-1)e^{-t}, \qquad \frac{\partial c_2(t,z)}{\partial z} = 9z^2 - 2 - 4e^{-t}, \tag{28b}$$

$$\frac{\partial \psi(t,z)}{\partial z} = 12z^2 - 14z + (3-4z)e^{-t}, \qquad \frac{\partial^2 \psi(t,z)}{\partial z^2} = 24z - 14 - 4e^{-t}. \tag{28c}$$

We then get the source terms of (20) as shown below:

$$\begin{aligned} f_1 =& -5ze^{-t} - D_1[Z_1(120z^4 - 32e^{-t}z^3 - 256z^3 + 234e^{-t}z^2 + \\ & 126z^2 - 40e^{-2t}z - 158e^{-t}z + 15e^{-2t}) + 12z - 6] \\ =& -D_1[Z_1(120z^4 - 256z^3 + 126z^2) + 12z - 6] \\ & + (-5z + 32D_1Z_1z^3 - 234D_1Z_1z^2 + D_1Z_1158z)e^{-t} \\ & + D_1Z_1(40z - 15)e^{-2t}, \end{aligned} \tag{29a}$$

$$
\begin{aligned}
f_2 =& 4(z-L)e^{-t} - D_2[Z_2(180z^4 - 48e^{-t}z^3 - 168z^3 - 117e^{-t}z^2 - 72z^2 + 32e^{-2t}z \\
&+ 128e^{-t}z + 96Le^{-t}z + 56z - 12e^{-2t} - 16Le^{-2t} - 6e^{-t} - 56Le^{-t}) + 18z] \\
=& - D_2[Z_2(180z^4 - 168z^3 - 72z^2 + 56z) + 18z] \\
&+ [4(z-L) + D_2Z_2(48z^3 + 117z^2 - 128z - 96Lz + 6 + 56L)]e^{-t} \\
&+ [D_2Z_2(-32z + 12 + 16L)]e^{-2t} \\
=& - D_2[Z_2(180z^4 - 168z^3 - 72z^2 + 56z) + 18z] \\
&+ [4(z-1) + D_2Z_2(48z^3 + 117z^2 - 128z - 96z + 6 + 56)]e^{-t} \\
&+ [D_2Z_2(-32z + 12 + 16)]e^{-2t},
\end{aligned}
\tag{29b}
$$

$$
\begin{aligned}
f_3 =& - \epsilon_s(24z - 14 - 4e^{-t}) - \beta[Z_1(2z^3 - 3z^2 + 5ze^{-t}) \\
&+ Z_2(3z^3 - 2z + 4(L-z)e^{-t})] - \rho(z) \\
=& - \epsilon_s(24z - 14 - 4e^{-t}) - \beta[Z_1(2z^3 - 3z^2 + 5ze^{-t}) \\
&+ Z_2(3z^3 - 2z + 4(1-z)e^{-t})] - \rho(z).
\end{aligned}
\tag{29c}
$$

From the exact solution (c_1, c_2, ψ) of (26) and $L = 1$, the boundary conditions can be derived as follows:

$$
c_{1,0}(t) = 0, \qquad c_{1,L}(t) = 2L^3 - 3L^2 + 5Le^{-t} = -1 + 5e^{-t}, \tag{30a}
$$

$$
c_{2,0}(t) = 4Le^{-t} = 4e^{-t}, \qquad c_{2,L}(t) = 3L^3 - 2L = 1, \tag{30b}
$$

$$
\psi_0(t) = 0, \qquad \psi_L(t) = 4L^3 - 7L^2 + L(3-2L)e^{-t} = -3 + e^{-t}. \tag{30c}
$$

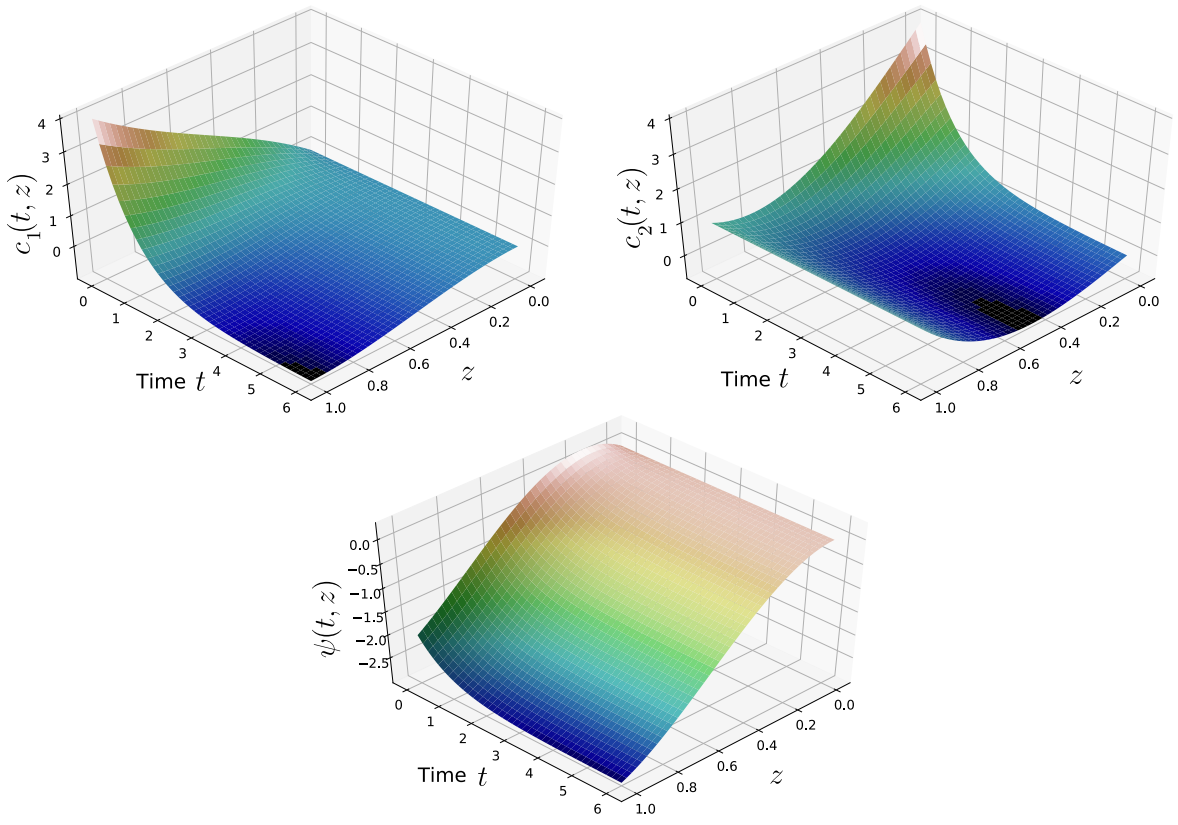


Figure 2: Surface plots of the analytical solutions (c_1,c_2,ψ) of (26) for our PNP test model defined by (20), (29), (30), and (31).

And the initial value conditions are given by

$$c_1(0,z) = 2z^3 - 3z^2 + 5z, \tag{31a}$$

$$c_2(0,z) = 3z^3 - 2z + 4(L-z) = 3z^3 - 2z + 4(1-z), \tag{31b}$$

$$\psi(0,z) = 4z^3 - 7z^2 + z(3-2z). \tag{31c}$$

Consequently, we have obtained another PNP test model defined by (20), (29), (30), and (31), whose analytical solution is given in (26).

5.2.3 Performance of our backward Euler method

We solved the PNP test model by our backward Euler method with $\tau = 0.1$. A solution was considered to have reached the steady state once the change of the concentra-

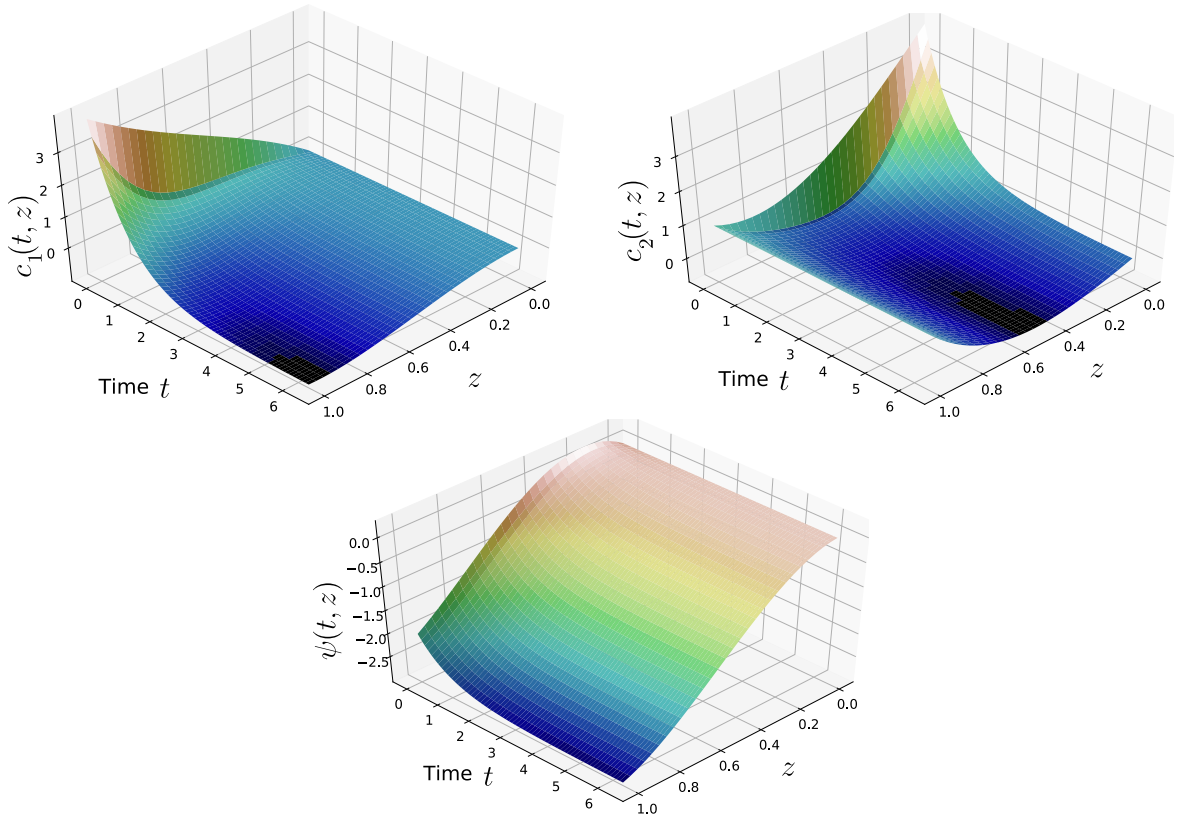


Figure 3: Surface plots of the numerical solution generated by our backward Euler method with $\tau = 0.1$, and $h = 0.00390625$ for the PNP test model with analytical solution (26).

tion and potential between two consecutive time steps was less than 10^{-3}. We used our damped iterative scheme to solve each related nonlinear system. We reached the threshold for the steady state at $t_k = 6.3$ with $k = 63$ for the constructed test case. For the calculation of the concentration and potential functions at the first 15 time steps, the scheme used the maximum of 300 iterations. Performing these 300 iterations took around 1 to 1.5 seconds on my laptop computer A. Afterwards the iteration was stopped earlier as the iteration termination rule was satisfied ($\epsilon = 10^{-5}$). For example, the scheme performed 78 iterations at $t_k = 4.9$ with $k = 49$ to find an acceptable solution, which took around 0.4 seconds in CPU time. In total, a little less than a minute was taken to find the numerical solution reported in Figure 3.

We compared the numerical solution with the analytical solution in Figure 4, showing that there exist large errors at the very first time steps and the further the distance to the boundaries, the worse the approximation of the numerical solution. The best accuracy is achieved for the potential function, where the error appears to be not changing with time and has a maximum value of about 0.08. The two concentration functions, on the other hand, show a spike right at the beginning. The function c_2 reaches its maximum error with about 1.6 at the first time step at $z \approx 0.3$. For the later time steps the error decreases constantly.

To get a more precise view of the precision per time step, we calculated the errors in the maximum norm of the spatial dimension as functions of time t in Figure 5. From the figure, we can see that the numerical accuracy can be improved significantly once the stable state is reached.

5.2.4 Test results by the forward Euler method

The forward Euler method, on the other hand, creates an explicit system of equations. Therefore, the calculation of each time step is faster than the backward Euler method. However, the solutions may be oscillating heavily towards the boundaries of the spa-

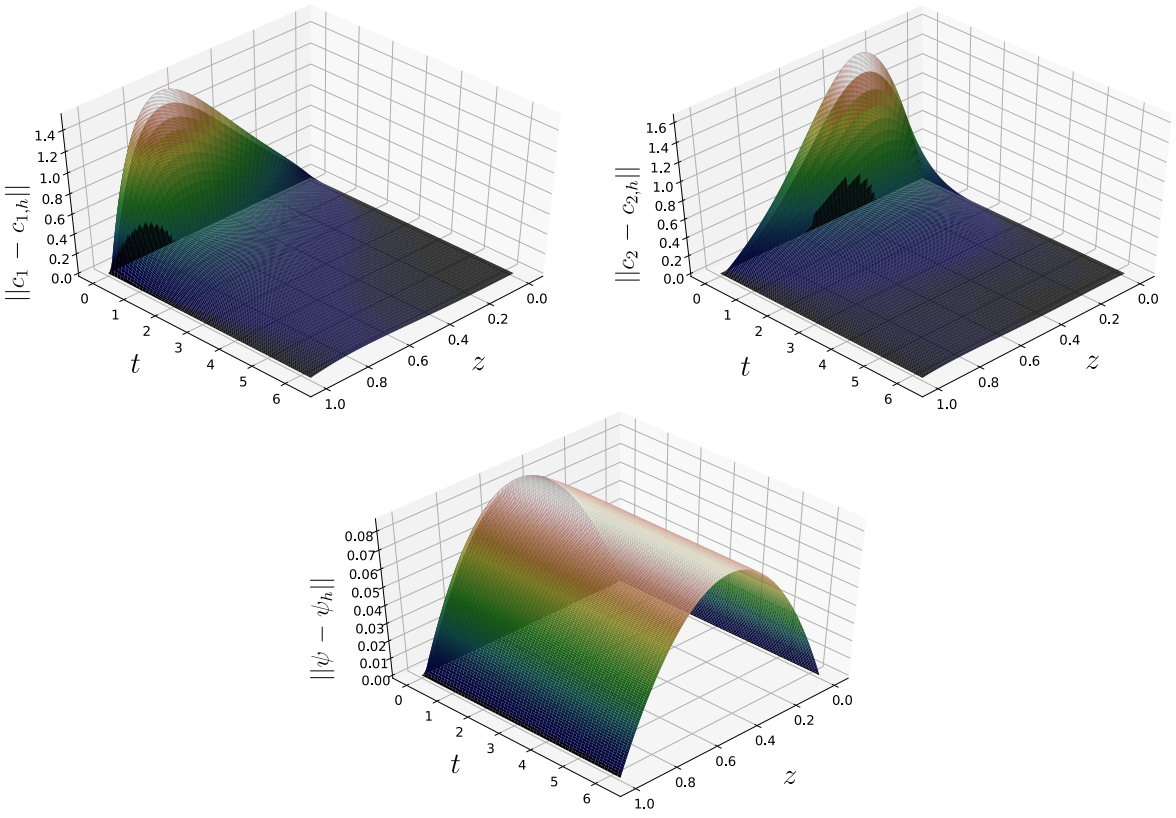


Figure 4: Surface plots of the absolute errors $||c_1 - c_{1,h}||$, $||c_2 - c_{2,h}||$, and $||\psi - \psi_h||$ of the numerical solution ($c_{1,h}$, $c_{2,h}$, ψ_h). The numerical solution ($c_{1,h}$,$c_{2,h}$,ψ_h) was found by the backward Euler method with $\tau = 0.1$, and $h = 0.00390625$ and the exact solution (c_1, c_2, ψ) is given in (26).

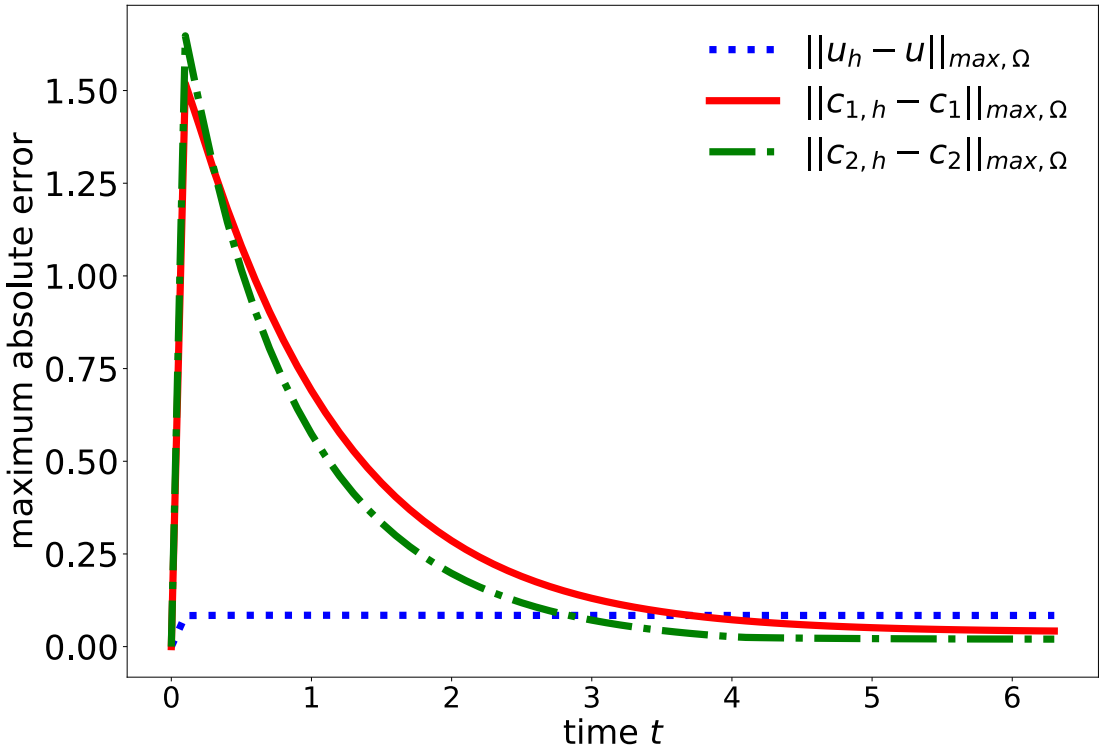


Figure 5: Errors between the numerical solution ($c_{1,h}$, $c_{2,h}$, ψ_h) generated by the backward Euler method and the exact solution (c_1, c_2, ψ) in the maximum norm as functions of time t.

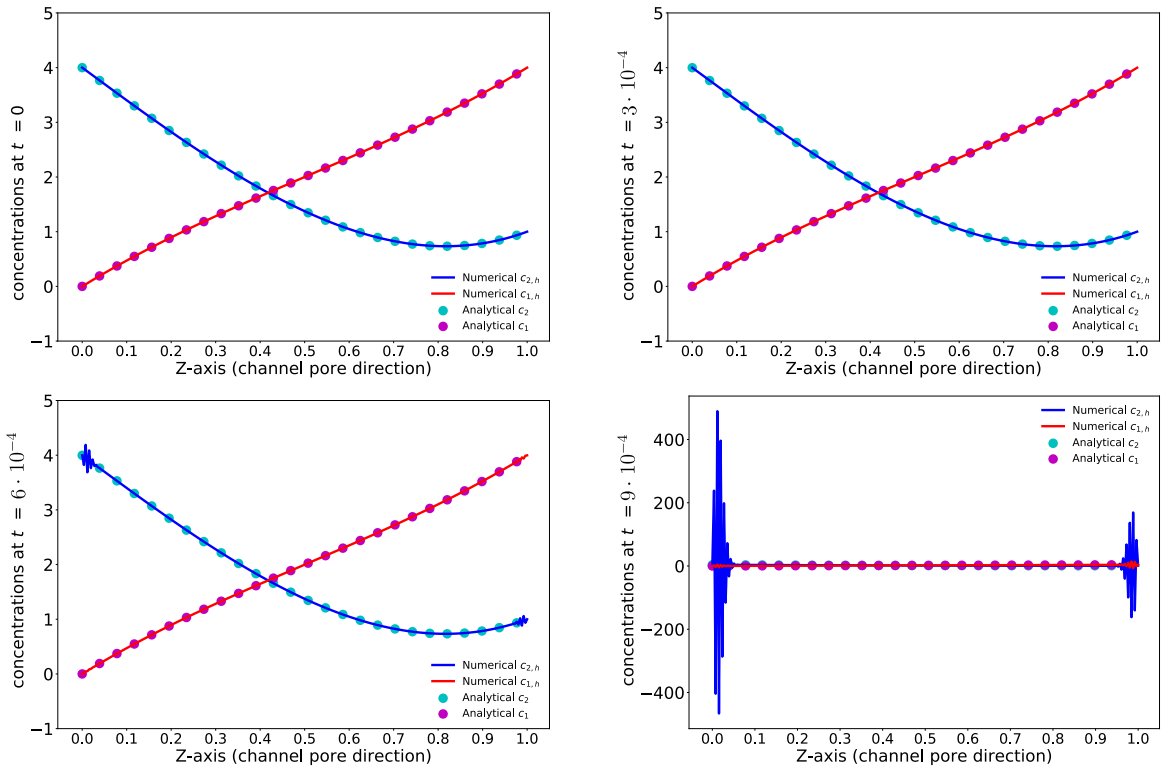


Figure 6: A comparison of the concentrations $c_{1,h}$ and $c_{2,h}$ calculated by the forward Euler method with time step $\tau = 10^{-4}$ with the analytical concentrations c_1 and c_2 for the PNP test model.

tial domain. This behavior is being analyzed by the stability zone of the algorithm. For the numerical solution to our example to be stable and a good approximation of the exact solution, the time step cannot exceed a given value, depending on the stiffness of the solution [1, p.304 f.].

This, however, causes issues even for small time steps τ. The Figure 6 shows the numerical solutions at time $t_k = k\tau$ with time step $\tau = 10^{-4}$ and $k = 0, 3, 6, 9$. The numerical solution of c_2 at $t_6 = 6 \cdot 10^{-4}$ shows some oscillation towards the boundaries, which the exact solution does not at all indicate. Note that the concentration c_2 approximately reaches the steady state for $t > 6$. Hence, the forward Euler method is not able to approximate it with the time step size $\tau = 10^{-4}$.

Further decreasing the time step size to $\tau = 10^{-5}$, we found that the forward Euler method produced a good numerical solution for $0 \leq t \leq 0.01$ as displayed in Figure

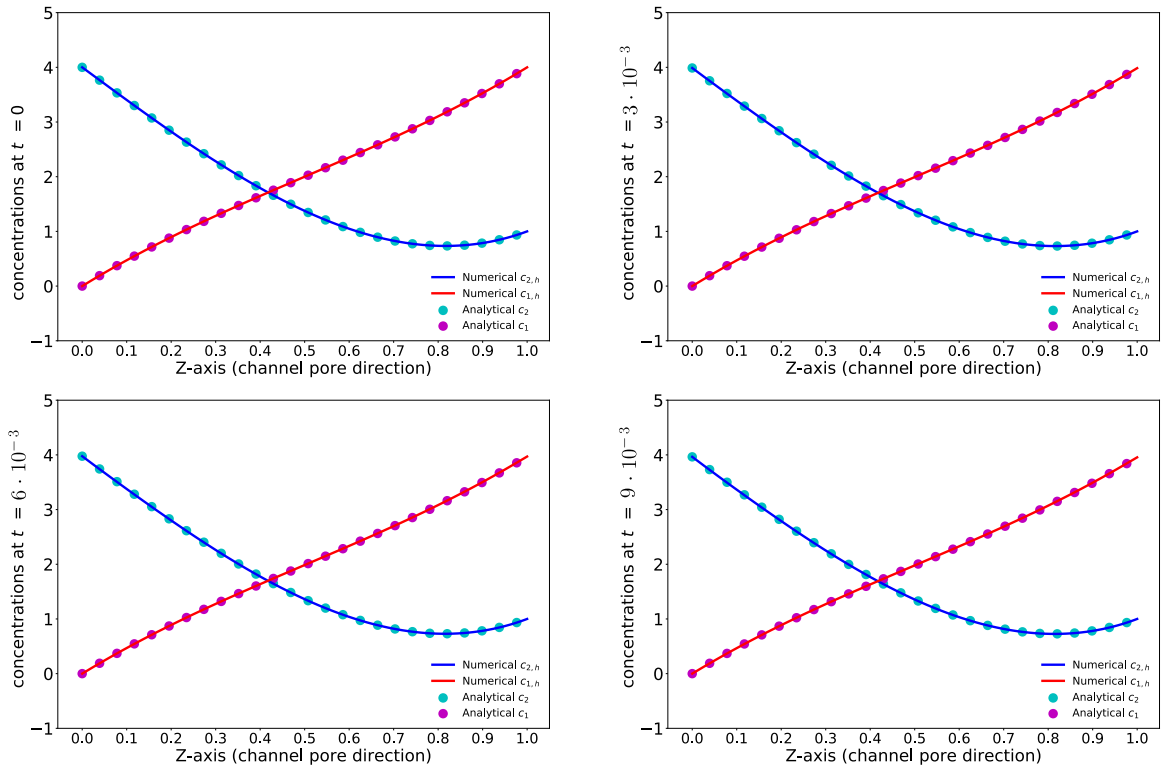


Figure 7: A comparison of the concentrations $c_{1,h}$ and $c_{2,h}$ calculated by the forward Euler method with time step $\tau = 10^{-5}$ with the analytical concentrations c_1 and c_2 for the PNP test model.

7. However, even though the calculation per time step can be done very fast by the forward Euler method compared to the backward Euler method, computing the solution in this small time interval already took about 10 seconds. Extrapolating this duration to the calculation over $0 \leq t \leq 6$, the calculation would take approximately 6000 seconds. This makes the forward Euler method too expensive in comparison to the backward Euler method.

6 Conclusions

The numerical study of the PNP model we present in this thesis provides valuable insights into the transport of charged species throughout an electrolyte under the influence of an electric field. We developed a new numerical scheme for solving a PNP model and evaluated its ability to approximate the potential function and the concentration functions of ions around an ion channel over time. We created a variational problem by the finite element method and the forward/backward Euler method. We successfully implemented our numerical scheme in Python using the DOLFIN library. In order to validate numerical solutions, we constructed a PNP test model with an analytical solution being given in algebraic expressions.

With the analytical solution, we found that the backward Euler method was able to find good approximations of the PNP test model in a reasonable amount of time. The forward Euler method, however, turned out to have stability issues. Decreasing the time step can improve the stability. However, the time for calculating the solution would be increased to an unfeasible level.

Overall, we have shown that the combination of the finite element method and the backward Euler method can be a promising approach for solving a PNP model. Our Python package can thereby be a valuable tool for investigating the behavior of ion transport across the cell membrane via an ion channel.

References

[1] BN Biswas et al. “A discussion on Euler method: A review”. In: *Electronic Journal of Mathematical Analysis and Applications* 1.2 (2013), pp. 294–317.

[2] *fenicsproject/stable*. URL: quay.io/fenicsproject/stable (visited on 04/26/2023).

[3] *FEniCSx | FEniCS Project*. URL: https://fenicsproject.org/ (visited on 04/26/2023).

[4] Dirk Gillespie. “A singular perturbation analysis of the Poisson Nernst Planck system: Applications to ionic channels”. In: (1999).

[5] Dirk Gillespie and Robert (Bob) Eisenberg. “Physical descriptions of experimental selectivity measurements in ion channels”. In: *European biophysics journal : EBJ* 31 (Nov. 2002), pp. 454–66. DOI: 10.1007/s00249-002-0239-x.

[6] Benoit van Hille. “Ionic channels of excitable membranes”. In: 2001.

[7] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 2012. ISBN: 9780486131597. URL: https://books.google.com/books?id=PYXjyoqy5qMC.

[8] A. Flavell M. Machen B. Eisenberg J. Kabre C. Liu and X. Li. “A conservative finite difference scheme for Poisson–Nernst–Planck equations”. In: (2014). DOI: 10.1007/s10825-013-0506-3. URL: https://doi.org/10.1007/s10825-013-0506-3.

[9] Hailiang Liu and Wumaier Maimaitiyiming. *Unconditional positivity-preserving and energy stable schemes for a reduced Poisson-Nernst-Planck system*. 2019. arXiv: 1909.13161 [math.NA].

[10] Benzhuo Lu et al. “Poisson–Nernst–Planck equations for simulating biomolecular diffusion–reaction processes I: Finite element solutions”. In: *Journal of Computational Physics* 229.19 (2010), pp. 6979–6994. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2010.05.035. URL: https://www.sciencedirect.com/science/article/pii/S0021999110002962.

[11] *Project Jupyter*. URL: https://jupyter.org/ (visited on 04/26/2023).

[12] W.E. Schiesser. *The Numerical Method of Lines: Integration of Partial Differential Equations*. Elsevier Science, 2012. ISBN: 9780128015513. URL: https://books.google.com/books?id=2YDNCgAAQBAJ.

[13] Isaac B. Sprague and Prashanta Dutta. “Modeling of Diffuse Charge Effects in a Microfluidic Based Laminar Flow Fuel Cell”. In: *Numerical Heat Transfer, Part A: Applications* 59.1 (2011), pp. 1–27. DOI: 10.1080/10407782.2010.523299. eprint: https://doi.org/10.1080/10407782.2010.523299. URL: https://doi.org/10.1080/10407782.2010.523299.

[14] Dexuan Xie. *An Extension of Goldman-Hodgkin-Katz Equations by Charges from Ionic Solution and Ion Channel Protein*. 2022. arXiv: 2208.11293 [q-bio.BM].

Appendices

A Hardware

HP Pavilion x360 14-dh0306ng

Microprocessor	Intel® Core™ i5-8265U (1.6 GHz base frequency, up to 3.9 GHz with Intel® Turbo Boost Technology, 6 MB cache, 4 cores)
Memory, standard	16 GB DDR4-2400 SDRAM (2 x 8 GB)
Hard drive	512 GB PCIe® NVMe™ M.2 SSD

B Software

OS	Arch Linux
OS-kernel	Linux 6.2.2-arch1-1 x86_64
Docker	Docker version 23.0.1, build a5ee5b1dfc