Summer 6-27-2023

SOLVING THE CABLE EQUATION, A SECOND-ORDER TIME DEPENDENT PDE FOR NON-IDEAL CABLES WITH ACTION POTENTIALS IN THE MAMMALIAN BRAIN USING KSS METHODS

Nirmohi Charbe

SOLVING THE CABLE EQUATION, A SECOND-ORDER TIME DEPENDENT PDE FOR NON-IDEAL CABLES WITH ACTION POTENTIALS IN THE MAMMALIAN BRAIN USING KSS METHODS

by

Nirmohi Charbe

A Thesis
Submitted to the Graduate School,
the College of Arts and Sciences
and the School of Mathematics and Natural Sciences
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Master of Science

Approved by:

Dr. James Lambers, Committee Chair
Dr. Bernd Schroeder
Dr. Huiqing Zhu

August 2023

ABSTRACT

In this thesis we shall perform the comparisons of a Krylov Subspace Spectral method with Forward Euler, Backward Euler and Crank-Nicolson to solve the Cable Equation. The Cable Equation measures action potentials in axons in a mammalian brain treated as an ideal cable in the first part of the study. We shall subject this problem to the further assumption of a non-ideal cable. Assume a non-uniform cross section area along the longitudinal axis. At the present time, the effects of torsion, curvature and material capacitance are ignored. There is particular interest to generalize the application of the PDEs including and other than Cable Equation to the study of Neurodegenerative diseases like multiple sclerosis, Alzheimer's, Parkinsons etc. The ultimate goal would be to be able to study a broad application of numerical methods to understand features of the human brain and its functions without involving medically invasive procedures.

ACKNOWLEDGMENTS

This is to thank all of those who have assisted me in this effort–my advisor Dr. Lambers and committee members Dr. Schroeder and Dr. Zhu for not giving up on me, my lovely parents and network of supportive friends I was fortunate to build.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure

LIST OF ABBREVIATIONS

c/s	-	Cross Section
CN	-	Crank Nicholson
Exp E	-	Explicit Euler
Imp E	-	Implicit Euler
KCL	-	Kirchoff's Current Law
KVL	-	Kirchoff's Voltage Law
KSS	-	Krylov's Subspace Spectral

NOTATION AND GLOSSARY

General Usage and Terminology

The notation used in this text represents fairly standard mathematical and computational usage. In many cases these fields tend to use different preferred notations to indicate the same concept, and these have been reconciled to the extent possible, given the interdisciplinary nature of the material. In particular, the notation for partial derivatives varies extensively, and the notation used is chosen for stylistic convenience based on the application.

The blackboard fonts are used to denote standard sets of numbers: $\mathbb{R}$ for the field of real numbers, $\mathbb{C}$ for the complex field, $\mathbb{Z}$ for the integers, and $\mathbb{Q}$ for the rationals. Capital letters, $A, B, \cdots$ are used to denote matrices, including capital greek letters, e.g., Λ for a diagnonal matrix. Functions which are denoted in boldface type typically represent vector-valued functions, and real-valued functions usually are set in lower case Roman or Greek letters. Calligraphic letters, e.g., $\mathcal{V}$, are used to denote spaces such as $\mathcal{V}$ denoting a vector space, $\mathcal{H}$ denoting a Hilbert space, or $\mathcal{F}$ denoting a general function space. Lower case letters such as i, j, k, l, m, n and sometimes p and d are used to denote indices.

Vectors are typset in square brackets, e.g., $[\cdot]$, and matrices are also typeset in square brackets. In general the norms are typeset using double pairs of lines, e.g., $||\cdot||$, and the absolute value of numbers is denoted using a single pairs of lines, e.g., $|\cdot|$. Single pairs of lines around matrices indicate the determinant of the matrix.

Chapter 1

Introduction

1.1 What is the Cable Equation?

The Cable Equation has been a point of interest in computational neuroscience [3] since the 1870s when it was first used to model neuronal electrotonic potential by Hermann. However, experimental evidence for its applicability to model the behaviour of axons surfaced prominently during the 1930s. Mainly due to its its mathematical advantages, over the next few decades it became of prime importance in the field. Since then there have been many important models built that contributed to the study of axonal potentials in a single cable like an axon or a bundle of neurons.

1.2 The Cable Equation

Our primary goal is to study changes in axonal potentials with the help of the Cable Equation [1] and extend this study by including changing parameters like cross-sectional area of a cable, curvature of the surface, torsion, membrane resistivity and so forth. For simplicity we shall start with the derivation of a second-order PDE for a non-uniform cable whose cross-sectional area changes relative to a given function. The solution will be able to predict potentials on the subsequent sections of the cable while analysing a smaller portion in consideration. The Cable Equation as represented below gives us its relationship with respect to space and time dependency

$$\frac{\partial V}{\partial T} = \frac{\partial^2 V}{\partial X^2} + f(V,t) \tag{1.1}$$

where V is voltage, t is time and $f(V,t)$= Source term that depends on the voltage and the time and

$$X = \frac{x}{\lambda_m}, \quad T = \frac{t}{\tau_m} \tag{1.2}$$

X and T are used to non-dimensionalize the equation. And

$$\lambda_m = \sqrt{\frac{R_m}{p(r_i + r_e)}}, \tau_m = R_m C_m \tag{1.3}$$

p	perimeter of the axon
r_i, re	membrane resistivity
C_m	membrane capacitance
R_m	membrane resistance

Here, V is our solution and it represents the potential in an axon at a particular point on the cable at the considered time. This will be computed by considering a small section of the cable i.e. finite cable that is divided in small sections for which the potential values are known at a given time. This information will be used to compute the potential along further points at the same given time.

The general assumption in our case would be that instead of a uniform linear cable with circular cross-sectional area, we will be assuming a non-uniform cable with changing circular cross-sectional area $A(r)$, which changes longitudinally. Let $r(x)$ be the radius of the cable at the point x. This method can be applied to non-circular cross-sections as well. However, to begin we have limited ourselves to assuming a circular cross-section for the cable.

The second-order PDE essentially takes into account the potential across a small section of longitudinal length and helps us predict the potential values at the given time on a point further than the considered section. The process can be iterated to predict potentials along the axonal length and upon comparison with real time potentials can be useful in detecting changes in the cross-sectional area of an axon.

This process will be carried out in two parts:

- Triple Stencil: This stencil can carry out

$$(I-(1-\theta)\Delta t L)\overrightarrow{V}^{n+1} = (I+\theta\Delta t L)\overrightarrow{V}^{n}$$

  - θ=1: Forward Euler
  - θ=0: Backward Euler
  - θ= $\frac{1}{2}$:Crank-Nicolson

- Krylov Subspace Spectral method (KSS) using Block Lanczos

Here we provide an outline of what to expect within this thesis. Chapter 2 will be focused on deriving the PDE that we shall be working on for the rest of the study. Chapter 3 will give us a detailed introduction to the methods that we are using i.e. the triple stencil and KSS using Block Lanczos. Chapter 4 contains a look at initial data and the numerical results for all methods. Here, we also get a chance to compare these results based on grid size, time stepping and method used. Lastly, we see the appendix broken down into main script and separate functions that we have used for computing the above results.

Chapter 2

DERIVATION CABLE EQUATION

2.1 Derivation of the Cable Equation

We assume that the cable in consideration is circular along its longitudinal axis and its cross-section will be varying according to some function $A(x)$ [7]. Consider the circuit below in figure

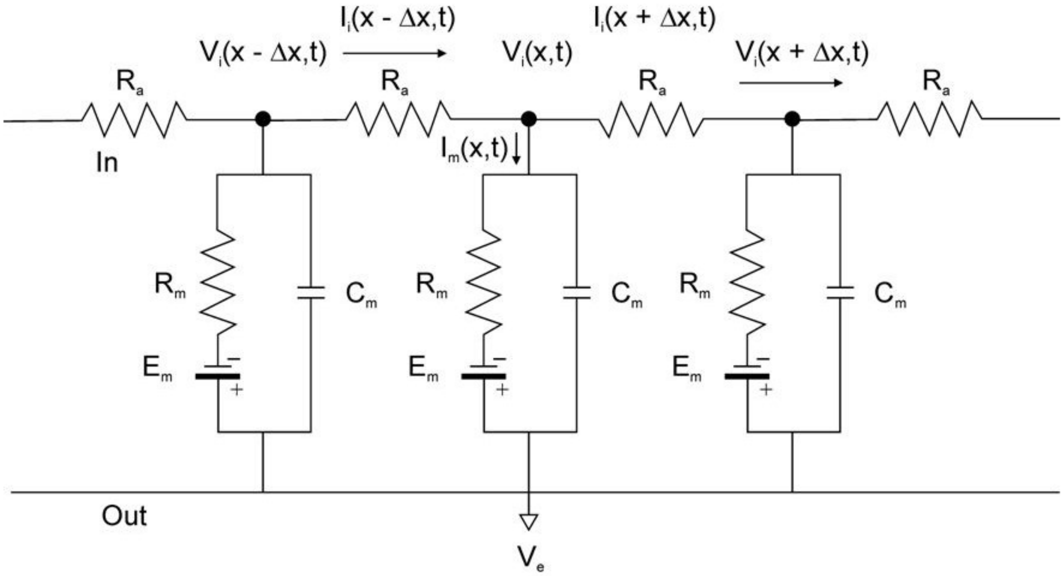


Figure 2.1: A Standard circuit set up resembling a small section of cable where the top most part is extra-cellular membrane, the circuit itself in the middle is the cell membrane/ wall and the bottom is the intra-cellular membrane

$$
\begin{aligned}
V_i(x,t) &= \text{Intra-cellular potential value w.r.t x and t} \\
V_e &= \text{Extra-cellular potential value} \\
E_m &= \text{Membrane potential value} \\
R_m &= \text{Membrane Resistance} \\
C_m &= \text{Membrane Capacitance} \\
I_m(x,t) &= \text{Membrane current w.r.t x and t} \\
I_i(x,t) &= \text{Intracellular current w.r.t x and t} \\
I_m(x,t) &= \text{Membrane current w.r.t x and t}
\end{aligned}
$$

The axial current I_e (extra-cellular current) and I_i (intra-cellular current) are ohmic, linear functions of the voltage. R_c and r_c are cytoplasmic resistance and cytoplasmic resistivity respectively (intra-cellular) and R_e and, R_c are sarcoplasmic resistance and sarcoplasmic resistivity respectively (extra-cellular) and lastly A_c and A_e are respective cross section area. We need both cross-section areas to imply that the extra-cellular resistance will be negligible and hence can be ignored.

$$r_i = \frac{R_c}{A_c}, \quad r_e = \frac{R_e}{A_e}. \tag{2.1}$$

Since A_e is extra-cellular area, it can attain large values, as it accounts for all area outside of the cable which is a large portion in comparison with the intra-cellular region. Hence, we can say that

$$r_e \approx 0. \tag{2.2}$$

Using Ohm's law of electrical resistance for a longitudinal cable, we have

$$V(x+\Delta x) - V(x,t) = -I_L(x,t)R_L \tag{2.3}$$

where

$$
\begin{aligned}
R_\mathrm{L} &= \frac{r_\mathrm{L}\Delta x}{A(x)}, \\
r_\mathrm{L} &= r_i, \\
R_\mathrm{L} &= R_i, \\
A(x) &= A_c.
\end{aligned}
$$

Now we have the following equation to represent the longitudinal current I_{L},

$$\begin{aligned} I_{\mathrm{L}} &= \frac{-A(x)}{r_L} \frac{[V(x+\Delta x,t)-V(x,t)]}{\Delta x} \\ &\approx \frac{-A(x)}{r_L} \frac{\partial V(x,t)}{\partial x} \end{aligned} \tag{2.4}$$

Longitudinal current is made of two kinds of currents. The first is membrane current based on its ionic movement I_{ion}, and the second is the membrane current based on its capacitance I_{cap} where

$$I_{\mathrm{ion}} = \frac{V(x,t)}{r_{\mathrm{m}}}, \tag{2.5}$$

$$I_{\mathrm{cap}} = \frac{C_{\mathrm{m}} \partial V(x,t)}{\partial t}, \tag{2.6}$$

with

$$\begin{aligned} r_m &= \text{membrane resistivity,} \\ C_m &= \text{membrane capacitance.} \end{aligned}$$

Considering that the total longitudinal current is the sum of the capacitor current and ionic current, we can say that

$$\begin{aligned} I_{\mathrm{L}}(x+\Delta x,t) - I_{\mathrm{L}}(x,t) &= I_{\mathrm{total}}, \\ I_{\mathrm{L}}(x+\Delta x,t) - I_{\mathrm{L}}(x,t) &= I_{\mathrm{cap}} + I_{\mathrm{ion}}. \end{aligned} \tag{2.7}$$

Substituting from (2.4) and (2.7) we obtain

$$\frac{-A(x)}{r_L} \frac{\partial V(x,t)}{\partial x} + \frac{-A(x+\Delta x)}{r_L} \frac{\partial V(x+\Delta x,t)}{\partial x} = C_{\mathrm{m}} \frac{\partial V(x,t)}{\partial t} + \frac{V(x,t)}{r_{\mathrm{m}}}.$$

Dividing the above equation by C_m and rearranging the terms yields

$$\frac{1}{r_{\mathrm{L}} C_{\mathrm{m}}} \left(A(x+\Delta x) \frac{\partial V(x+\Delta x)}{\partial x} - A(x+\Delta x) \frac{\partial V(x,t)}{\partial x} \right) = \frac{\partial V(x,t)}{\partial t} + \frac{V(x,t)}{r_{\mathrm{m}} C_{\mathrm{m}}}.$$

For $\Delta x \to 0$ the preceding equation converges to

$$\frac{\partial V(x,t)}{\partial t} = \frac{1}{C_{\mathrm{m}} r_{\mathrm{L}}} \frac{\partial}{\partial x} \left(A(x) \frac{\partial V(x,t)}{\partial x} \right) - \frac{V(x,t)}{r_{\mathrm{m}} C_{\mathrm{m}}}. \tag{2.8}$$

Above is the second-order PDE that we will be working with to solve for potential across the cable. It must be noted that since the equation is dependent on $A(x)$ we can work with any shape of the cross-section as in vito the cross section of an axon is rarely uniform all across its length.

Chapter 3

Methodology

3.1 Implicit Euler, Explicit Euler and Crank-Nicolson with Dirichlet BCs

We consider the derived cable equation and non-homogeneous Dirichlet BCs,

$$\frac{\partial V(x,t)}{\partial t} = \frac{1}{C_{\mathrm{m}} r_{\mathrm{L}}} \frac{\partial}{\partial x}\left(A(x)\frac{\partial V(x,t)}{\partial x}\right) - \frac{V(x,t)}{r_{\mathrm{m}} C_{\mathrm{m}}} \tag{3.1}$$

This equation can be rewritten in the following simplied form

$$V_t = \alpha(AV_x)_x - \beta V \tag{3.2}$$

We will start with (3.2) and discretize the PDE in space to obtain the system of ODEs

$$\overrightarrow{V}_{\mathrm{t}} = L\overrightarrow{V} \tag{3.3}$$

This is further discretized in time to obtain

$$(I-(1-\theta)\Delta t L)\overrightarrow{V}^{n+1} = (I+\theta\Delta t L)\overrightarrow{V}^{n}. \tag{3.4}$$

Depending on the value of θ, we can categorize the methods as

$$\begin{aligned}\theta &= 0, \text{Implicit Euler}\\ \theta &= 1, \text{Explicit Euler}\\ \theta &= \frac{1}{2}, \text{Crank-Nicholson}\end{aligned}$$

The coefficients are $\alpha = \frac{1}{c_{\mathrm{m}} r_{\mathrm{L}}}$ and $\beta = \frac{1}{r_{\mathrm{m}} c_{\mathrm{m}}}$ and L is defined as,

$$L = \alpha D^+ A D^- - \beta I \tag{3.5}$$

$$D^+ = \frac{1}{\Delta x}\begin{bmatrix} -1 & 1 & \cdots & 0\\ 1 & -1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & -1 \end{bmatrix}$$

$$D^- = -(D^+)^T$$

It is important to note that the relationship between D^+ and D^- must satisfy below condition [6],

$$D^+D^- = \frac{1}{(\Delta x)^2}\begin{bmatrix} -2 & 1 & \cdots & 0 \\ 1 & -2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 \end{bmatrix}$$

A is the cross section area which depends on the radius of the cable and the radius of the cable depends on x, these will be defined as

$$\begin{aligned} A(r) &= \pi r^2(x), \\ r(x) &= a\cos(x) + b\sin(x) + c, \\ \{a, b\} &= 0.1. \end{aligned}$$

Now, we consider a section of cable of Δx has unit length. This cable will be divided into $(N+1)$ sub-sections with $(N+2)$ points. Starting from point 0 to point $(N+1)$, we find mid-point of each section. Then radius is computed at the mid points and radius gives us the area values $\overrightarrow{A}$.

Considering (3.2), we compute for $(AV_x)_x$ as below using a centered difference stencil to obtain the following matrix by using $\vec{A}$ above,

$$D^+AD^- = \begin{bmatrix} -\left(A_{\frac{1}{2}} + A_{\frac{3}{2}}\right) & A_{\frac{3}{2}} & \cdots & 0 \\ A_{\frac{3}{2}} & -\left(A_{\frac{3}{2}} + A_{\frac{5}{2}}\right) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\left(A_{n-\frac{1}{2}} + A_{n+\frac{1}{2}}\right) \end{bmatrix}$$

Once we obtain A, we can substitute it in (3.5) to solve for L.
L now is input in (3.4) which gives us $\overrightarrow{V}^{n+1}$

3.2 Block Krylov Subspace Spectral Method

Krylov Subspace Spectral Methods were initially introduced to solve time-dependent, variable-coefficient problems. Hence, it is an organic choice for a second-order PDE that we have derived. It satisfies both important conditions– it is a time dependent problem and it also contains a variable coefficient. While we work with KSS, we must remember that the solution has a Fourier sine series expansion form and the coefficients of the solution have a spatially discretized form. We will be using a block KSS method to compute each of these coefficients individually to obtain our solution.

To begin understanding Krylov Subspace Spectral methods, let's consider the following one-dimensional initial boundary value problem:

$$\begin{aligned} u_t + Lu &= 0, \text{ on } (0,2\pi)\times(0,\infty) \\ u(x,0) &= f(x), 0 < x < 2\pi \\ u(0,t) &= u(2\pi,t), t > 0 \end{aligned}$$

Let $S(t) = \exp(-Lt)$ represent the exact solution operator. Then, by approximating the Fourier components, which are obtained if we apply the exact solution operator to the computed solution $\tilde{u}(x,t_n)$, the solution at time $t_{\text{n+1}}$ can be obtained as

$$\hat{u}(\omega,t_{n+1}) = \left\langle \frac{1}{2\pi}e^{i\omega x}, S(\Delta t)\tilde{u}(x,t_n) \right\rangle \tag{3.6}$$

We consider an N-point uniform grid with spacing $\Delta x = 2\pi/N$. Then, we discretize the functions on periodic domain $[0,2\pi]$, while the operator L and the solution operator $S(\Delta t)$ can be approximated by $N\times N$ matrices that represent linear operators on the space of grid functions. Next we can obtain the bilinear form for $\tilde{u}$,

$$\hat{u}(\omega,t_{n+1}) \approx \sqrt{\Delta x}\hat{e}_{\omega}^{H} S_N(\Delta t)u^n. \tag{3.7}$$

We have

$$
\begin{aligned}
[\hat{e}_w]_j &= \frac{1}{\sqrt{N}} e^{i\omega j\Delta x}, \quad [u^n]_j = u(j\Delta x, t_n), \\
S_N(t) &= exp(-L_N t), \quad [L_N]_{jk} = -p[D_N^2] + q(j\Delta x), \qquad (3.8)
\end{aligned}
$$

where D_N is a discretization of the differentiation operator that is defined on the space of grid functions.

Our primary task is to approximate (3.7) for $\hat{u}(\omega, t_{n+1})$:

$$
\hat{u}(\omega, t_{n+1}) = \hat{e}_\omega^H u^{n+1} = \hat{e}_\omega^H S_N(\Delta t) u^n \qquad (3.9)
$$

Considering the above form, we will use Golub and Meurant's method to approximate expressions of the form [5, 8]

$$
u^T f(A) v \qquad (3.10)
$$

where u and v are N-vectors and A is an $N \times N$ symmetric positive definite matrix. The function f is assumed to be analytic on the domain that contains the eigenvalues of A, implying that it has a converging Taylor series on that domain.

Our task is to implement this method with $A = L_N$, where L_N is the spatially discretized operator defined in (3.9). $f(\lambda) = \exp(-\lambda t)$ for some t, and vectors $\hat{e}_\omega$ and u^n play the roles of the vectors u and v

Considering the fact that the matrix A is symmetric positive definite, we can say that its eigenvalues will be all real. Hence we can say that

$$
b = \lambda_1 \geq \lambda_2 \cdots \geq \lambda_N = a > 0,
$$

and the eigenvectors which are orthogonal will be represented as $q_j, j = 1, 2 \ldots N$; they also correspond to the eigenvalues $\lambda_j, j = 1, 2 \ldots, N$ respectively. That allows us to redefine our initial expression as

$$
u^T f(A) v = \sum_{j=1}^{N} f(\lambda_j) u^T q_j q_j^T v.
$$

Assume that $a = \lambda_N$ is the smallest eigenvalue and $b = \lambda_1$ is the largest eigenvalue. We define the measure $\alpha(\lambda)$ as

$$\alpha(\lambda) = \begin{cases} 0, & \text{if } \lambda < a \\ \sum_{j=i}^{N} \alpha_j \beta_j, & \text{if } \lambda_i \le \lambda < \lambda_{i-1}, i = 2, \dots, N, \\ \sum_{j=1}^{N} \alpha_j \beta_j, & \text{if } b \le \lambda \end{cases} \tag{3.11}$$

and we have $\alpha_j = u^T q_j$ and $\beta_j = q_j^T v$. If this quantity is positive and increasing then we can express it as Riemann-Stieltjes integral

$$u^T f(A) v = I[f] = \int_a^b f(\lambda) d\alpha(\lambda).$$

We can bound the above integral using Gauss quadrature rules and obtain an approximation of the form

$$I[f] = \sum_{j=1}^{K} w_j f(t_j) + R[f]$$

where the nodes $t_j, j = 1, \dots K$, and the weights $w_j, j = 1 \dots, K$, can be obtained using the symmetric Lanczos algorithm if $u = v$ and if $u \neq v$ then we use the unsymmetric Lanczos algorithm [4]. In next section we are going to have a detailed look at the block formulation approach.

3.2.1 Block Gaussian Quadrature

Consider the generalization of the form (3.10) with two starting vectors u and v,

$$[u \quad v]^T f(A)[u \quad v].$$

we obtain below matrix of dimension 2 × 2,

$$\int_a^b f(\lambda)d\mu(\lambda) = \begin{bmatrix} u^T f(A)u & u^T f(A)v \\ v^T f(A)u & v^T f(A)u \end{bmatrix}$$

where $\mu(\lambda)$ is a matrix function of λ, and each of entry of it is a measure of the form $\alpha(\lambda)$. Consider that the integral $\int_a^b f(\lambda)d\mu(\lambda)$ is a 2×2 matrix and the most general K-node quadrature rule has the following formation,

$$\int_a^b f(\lambda)d\mu(\lambda) = \sum_{j=1}^{K} W_j f(T_j) W_j + \text{error} \tag{3.12}$$

where T_j and W_j are both symmetric. The above equation can be simplified using

$$T_j = Q_j \Lambda_j Q_j^T$$

where Q_j is the eigenvector matrix and Λ_j is the 2×2 matrix with the eigenvalues. Therefore

$$\sum_{j=1}^{K} W_j f(T_j) W_j = \sum_{j=1}^{K} W_j Q_j f(\Lambda_j) Q_j^T W_j.$$

Rewriting the equation,

$$W_j Q_j f(\Lambda_j) Q_j^T W_j = f(\lambda_1) z_1 z_1^T + f(\lambda_2) z_2 z_2^T,$$

for $z_k = W_j Q_j e_k$ for $k = 1, 2$, which yields

$$\int_a^b f(\lambda)d\mu(\lambda) = \sum_{j=1}^{K} f(t_j) v_j v_j^T + \text{error},$$

where v_j is a 2×2 vector and t_j is a scalar.

Now, we need the scalar nodes t_j and the associated vectors v_j. From [8], we know that there exists orthogonal matrix polynomials such that

$$\lambda p_{j-1}(\lambda) = p_j(\lambda)B_j + p_{j-1}(\lambda)M_j + p_{j-2}(\lambda)B_{j-1}^T,$$

with $p_0 = I_2$ and $p_{-1} = 0$. Rewriting the above equation, we obtain

$$\lambda[p_0(\lambda),\dots,p_{K-1}(\lambda)] = [p_0(\lambda),\dots,p_{K-1}(\lambda)]\tau_K + [0,\dots,0,p_K(\lambda)B_K],$$

where

$$\tau_K = \begin{bmatrix} M_1 & B_1^T & & & \\ B_1 & M_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-2} & M_{K-1} & B_{K-1}^T \\ & & & B_{K-1} & M_K \end{bmatrix} \tag{3.13}$$

where τ_K is a block-tridiagonal matrix. Hence we define the quadrature rule as

$$\int_a^b f(\lambda)d\mu(\lambda) = \sum_{j=1}^{2K} f(\lambda_j)v_j v_j^T + \text{error}. \tag{3.14}$$

where $2K$ is the order of the matrix τ_K, λ_j is one of its eigenvalues, and u_j is the vector with the first two elements of corresponding normalized eigenvector.

Now, we apply the block Lanczos algorithm to compute M_j and B_j. For X_1 as a $N \times 2$ known matrix, such that $X_1^T X_1 = I_2$, we assume $X_0 = 0$ for given dimensions. For $j = 1,\dots,$ compute

$$\begin{aligned} M_j &= X_j^T A X_j, \\ R_j &= AX_j - X_j M_j - X_{j-1}B_{j-1}^T, \\ X_{j+1}B_j &= R_j. \end{aligned}$$

The very last step is the QR decomposition for R_j such that it is coherent with the condition X_j is $N \times 2$, with $X_j^T X_j = I_2$. The matrix B_2 is 2×2 upper triangular. M_j is the coefficient matrix with same dimensions and is symmetric. R_j can eventually be rank deficient and if it turns out to be so, B_j will be singular in such situation.

3.2.2 Block KSS (Spectral) Method

Here, we dive into the block KSS method. It should be noted that $\hat{e}_\omega$ is a vector that discretizes a Fourier basis function $\hat{e}^{i\omega x}$). We define

$$R_0 = [\hat{e}_\omega \;\; u^n]$$

where ω is a wave number and $\omega = (\frac{-N}{2}+1), \ldots, \frac{N}{2}$. Upon computing the QR factorization we obtain

$$R_0 = X_1(\omega)B_0(\omega)$$

which gives us

$$X_1 = [\hat{e}_\omega \;\; u^n_\omega / \|u^n_\omega\|_2], \;\; B_0(\omega) = \begin{bmatrix} 1 & \hat{e}^H_\omega u^n \\ 0 & \|u^n_\omega\|_2 \end{bmatrix},$$

where

$$u^n_\omega = u^n - \hat{e}^H_\omega u^n.$$

Now, the goal is to obtain the Gauss Quadrature rule. The output of a Block Lanczos algorithm gives us the below block-tridiagonal matrix. The eigenvalues and eigenvectors of this matrix gives us nodes and weights, respectively, for the quadrature rule. The output matrix is

$$T_K(\omega) = \begin{bmatrix} M_1(\omega) & B_1(\omega)^H & & & \\ B_1(\omega) & M_2(\omega) & B_2(\omega)^H & & \\ & \ddots & \ddots & \ddots & \\ & & B_{K-1}(\omega) & M_{K-1}(\omega) & B_{k-1}(\omega)^H \\ & & & B_{K-1}(\omega) & M_K(\omega) \end{bmatrix}.$$

The Next step will be to find each of these Fourier components of the solution as

$$[\hat{u}^{n+1}]_\omega = [B_0^H E_{12}^H \exp(-T_K(\omega)\Delta t) E_{12} B_0]_{12}$$

where

$$E_{12} = [e_1 \;\; e_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}$$

Note that the above discussion is about a PDE with periodic boundary condition with a Fourier series in complex exponential form. Our PDE has Dirichlet boundary conditions with a Fourier sine series. Since our problem in consideration has different boundary conditions it is safe to assume that the solution will have a different form from as seen above.

It is to be emphasized that to compute the quantity $E_{12}^H \exp(-T_K(\omega)\Delta t)E_{12}$ we have to evaluate the eigenvalues and eigenvectors of $T_K(\omega)$ to effectively find the nodes and weights for the Gaussian Quadrature.

The KSS methods require $O(N \log N)$ flops per time step which is comparable to some of the other time stepping methods. In the next chapter, we are going to present and further discuss the process used to solve our PDE and its comparison with other methods.

Chapter 4

Numerical Results

4.1 Comparison of the methods

This section contains the numerical results for all the methods and their comparison. Forward Euler will be used to compute a reference solution because of the error convergence and other solutions will be compared to this one. Reference solution comes from using the Forward Euler method with the following parameters:

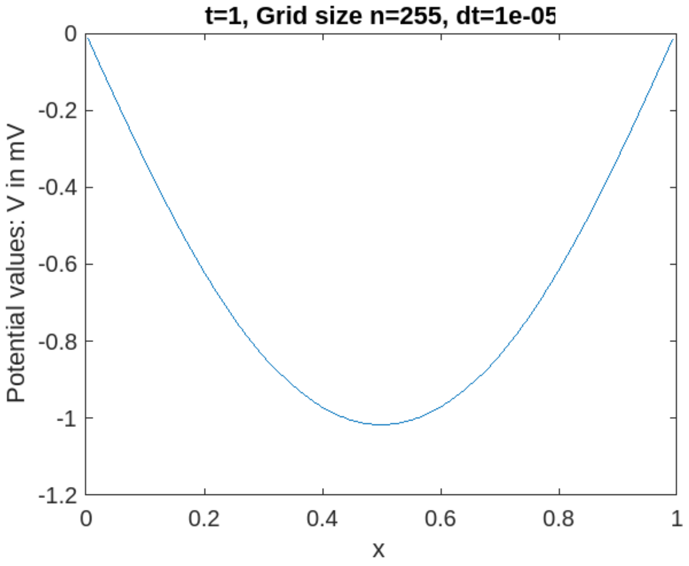


Figure 4.1: Forward Euler displaying a smooth solution for a very small time step

The initial data consists of potential values from a human brain and its graphical representation looks like:

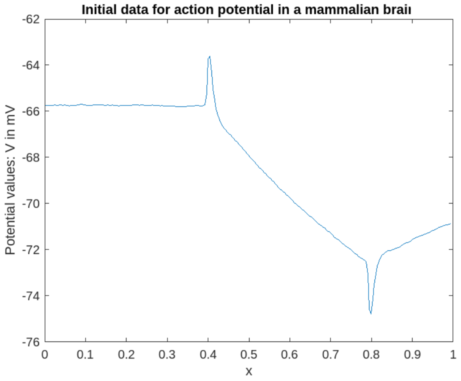
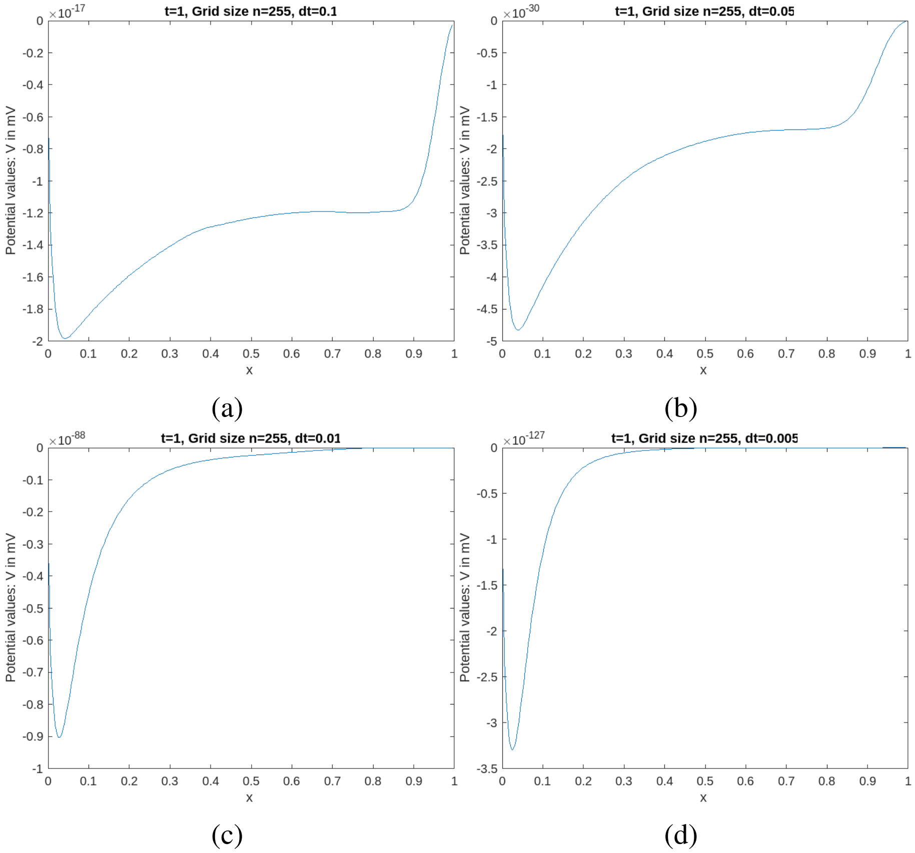


Figure 4.2: Usual range for potentials in human axon ranges from -75mV to -55mV as displayed in this graph

4.1.1 Results for Backward Euler

With given parameters, below are the results for Backward Euler:

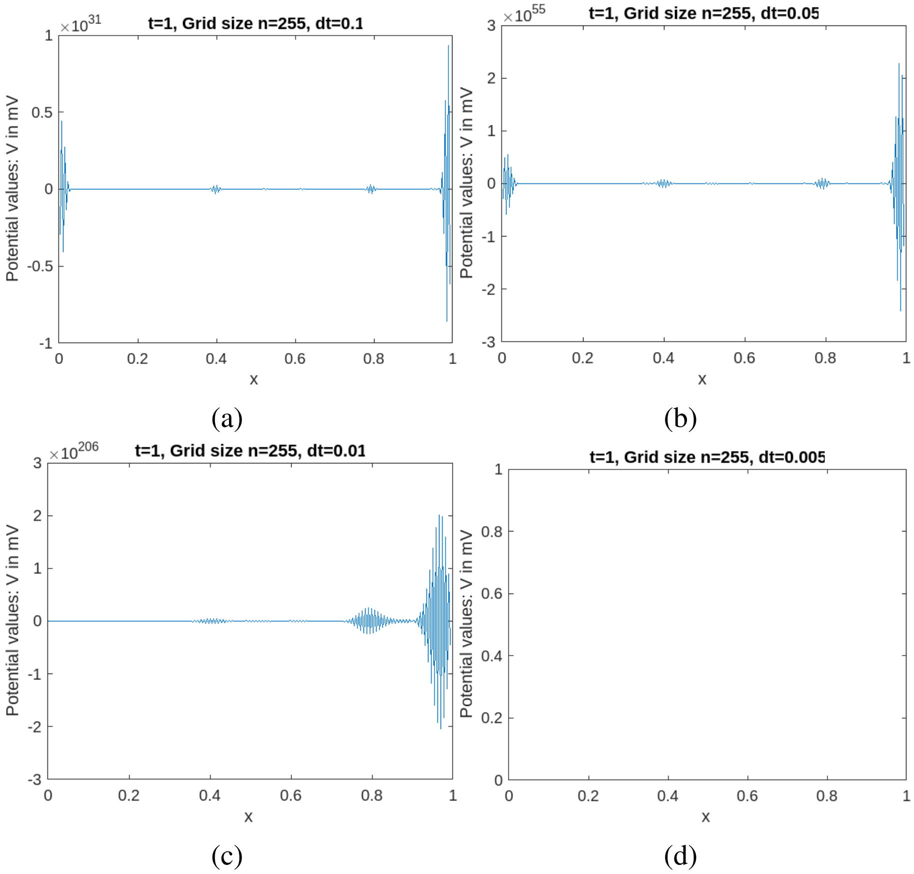


(a) (b)

(c) (d)

Figure 4.3: Backward Euler shows excessive damping with every reduced time step

4.1.2 Results for Forward Euler



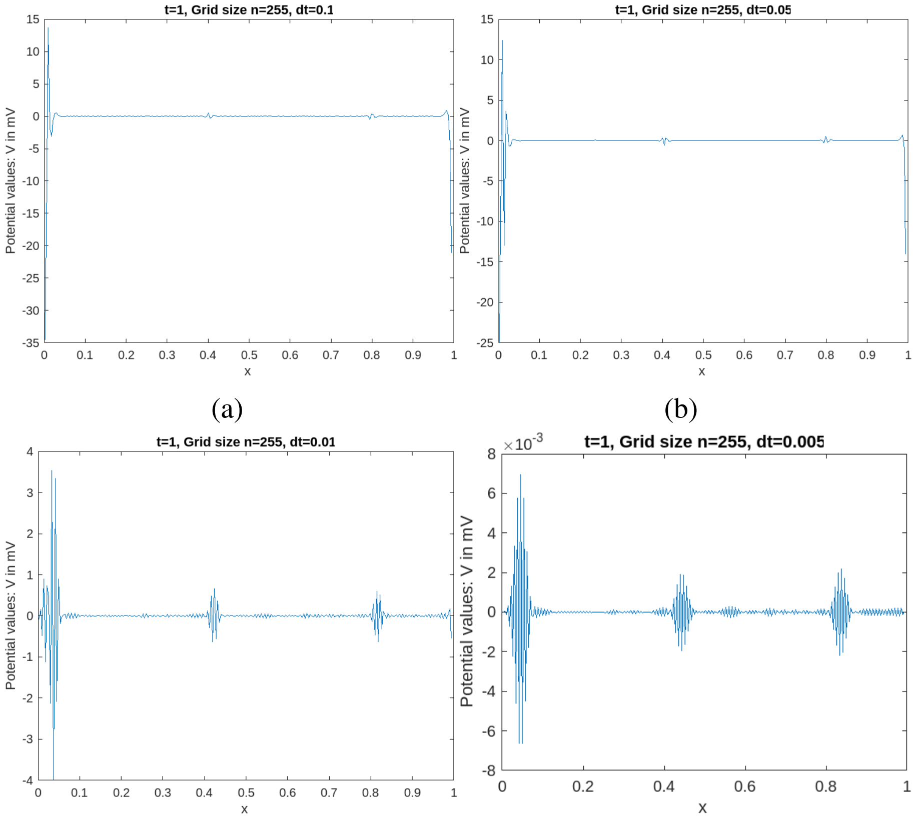
(a) (b)

(c) (d)

Figure 4.4: Forward Euler displays very unstable solution for moderately chosen time step. To obtain a desirable solution, we must choose a very small time step for this method

Forward Euler was expected to give us unstable behaviour for large time steps as compared to Backward Euler or Crank-Nicolson. This is easily observed from the solution whose order ranges up to 10^{200} and above. The behaviour however, changes as displayed in 4.1. Though the first three graphs show similar behaviour with instability the last graph given the small time step does not duplicate the same behaviour. However, we can confirm its undesirable nature from given order of 10^{170}.

4.1.3 Results for Crank-Nicolson



(a)

(b)

(c)

(d)

Figure 4.5: Crank Nicholson shows an oscillating behavior in the solution. This behaviour is explain below

Consider the Crank-Nicolson solution:

$$\overrightarrow{V}^{n+1} = \left(I - \frac{\Delta t}{2}L\right)^{-1}\left(I + \frac{\Delta t}{2}L\right)\overrightarrow{V}^{n}.$$

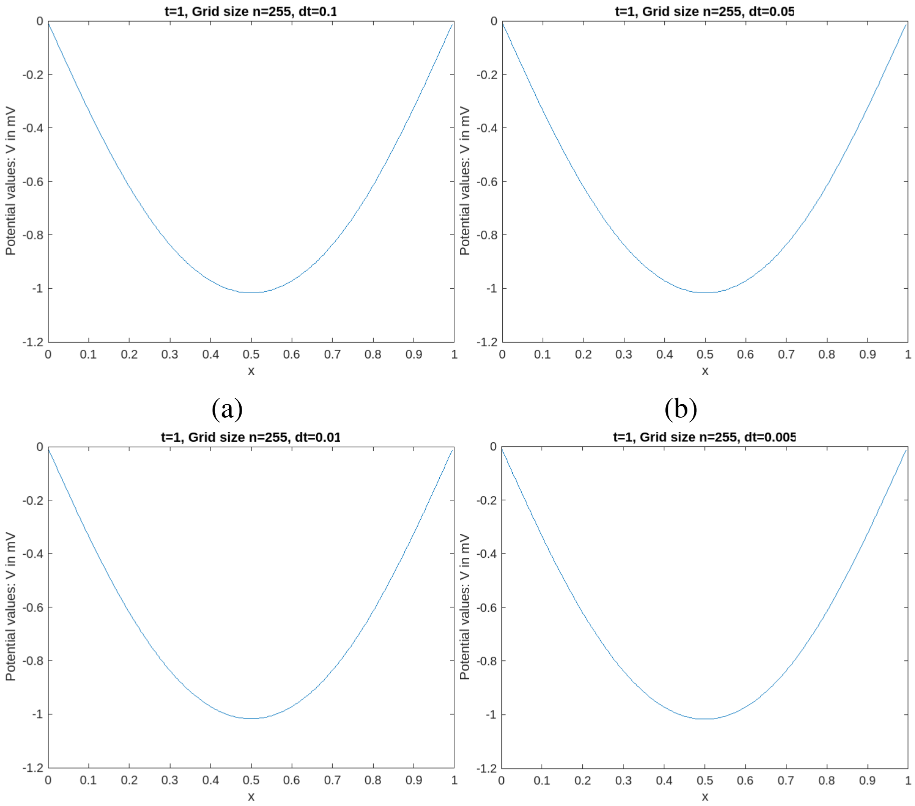
Let μ be an eigenvalue of the solution operator matrix L. We can define μ as,

$$\mu = \frac{1 + \frac{\Delta t}{2}\lambda}{1 - \frac{\Delta t}{2}\lambda}$$

where λ is an eigenvalue of L.

For a given basis function of the form $\sin(k\pi x)$, depending on the coefficient k the eigenvalues can have very large values. Assuming that λ is very large, the second term of each portion of μ becomes the dominating quantity. Given the opposite signs of numerator and denominator, this is what leads to the alternating effect as we have observed above.

4.1.4 Results for Krylov Subspace Spectral Method



(a) (b)

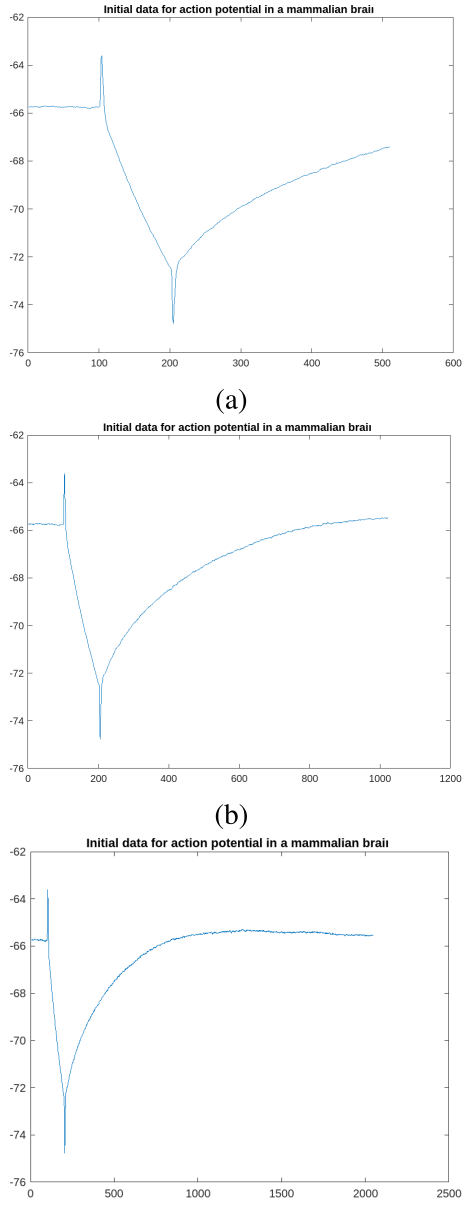
(c) (d)

Figure 4.6: We can see that KSS method produces a smooth solution that is directly comparable to the reference solution

KSS gives us a solution comparable to our reference solution. An advantage is that it can obtain similar accuracy with much larger time steps which saves us computational time and expense. We can clearly state that considering a problem of this nature, KSS is a far superior method as compared to its competitive methods.

4.2 Comparing the results for larger grid sizes

For following results, we will be using grid sizes of $n = 511$, $n = 1023$ and $n = 2047$, respectively. The initial data plot is represented as below:



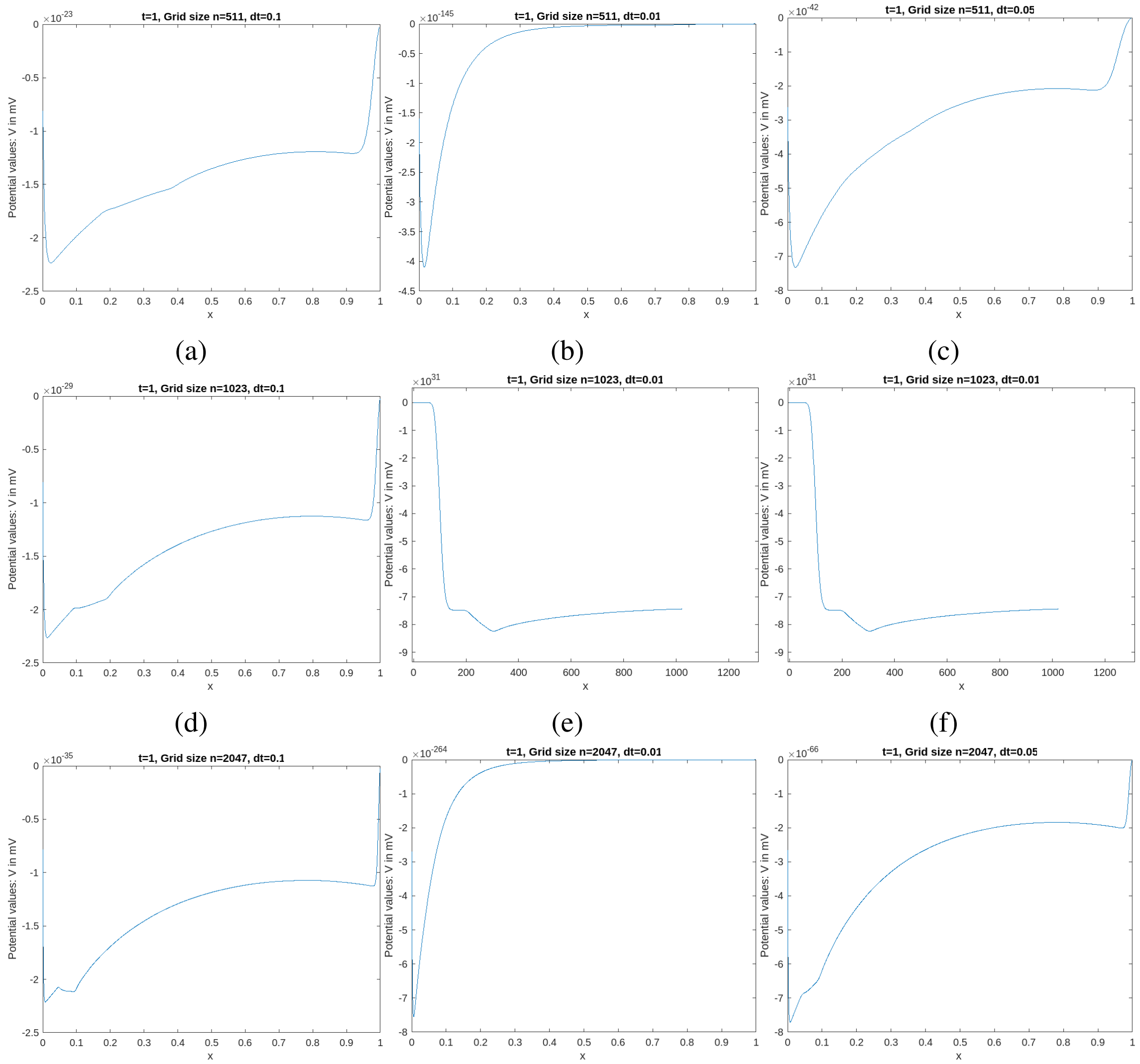
(a)

(b)

(c)

Figure 4.7: The initial data on larger grid sizes

4.2.1 Results for Backward Euler Method



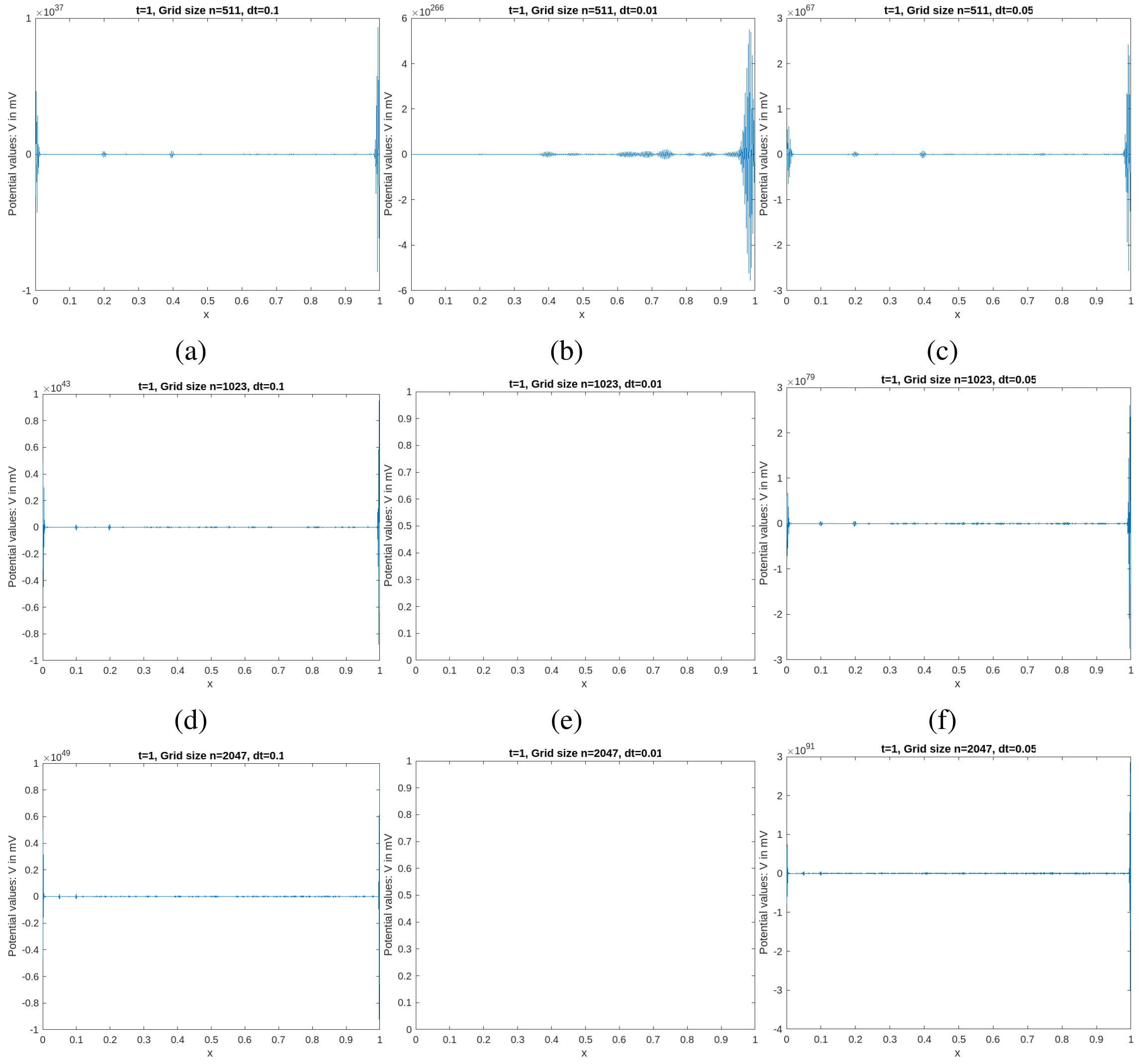
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.8: Backward Euler with a larger grid size becomes an increasingly unsuitable method for solving a PDE as ours

4.2.2 Results for Forward Euler Method



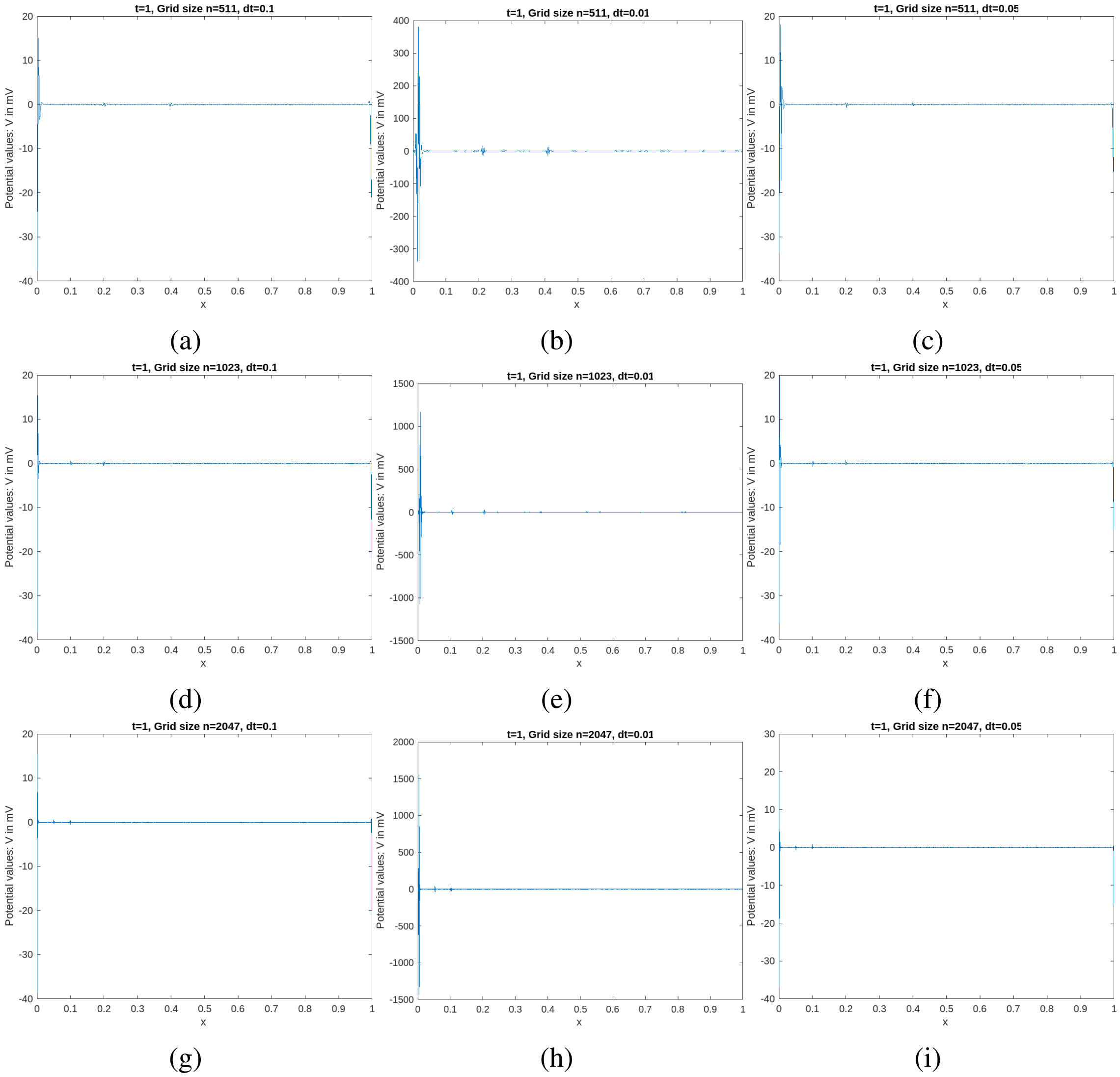
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.9: Forward Euler displays similar behaviour and can be considered unstable even at larger grid sizes unless the time steps in consideration are very small

4.2.3 Results for Crank-Nicolson Method



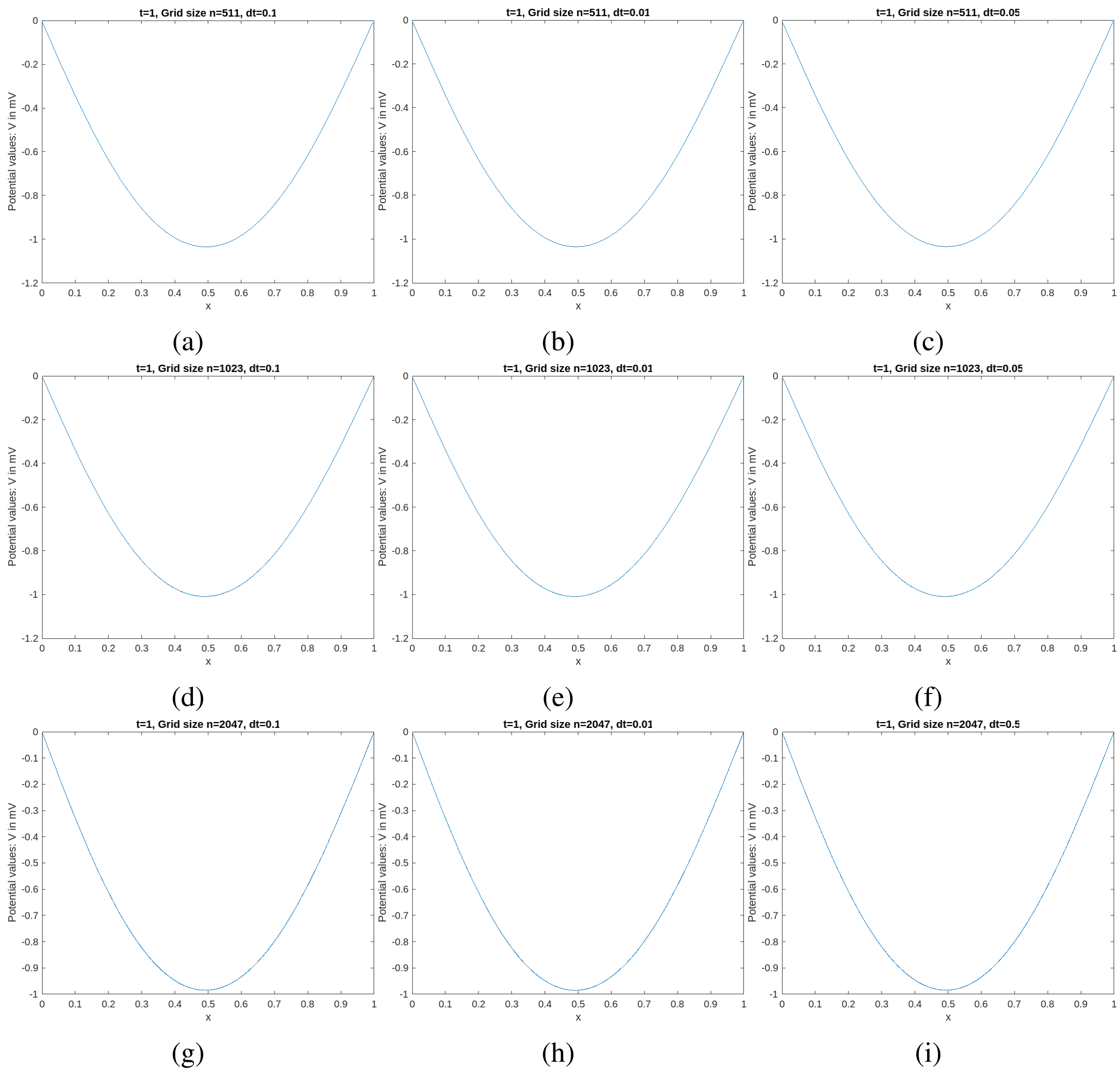
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.10: For larger grid sizes, we still observe the previously noted alternating behaviour of the solution

4.2.4 Results for Krylov Subspace Spectral Method



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4.11: For larger grid sizes, we still observe the previously observed smoothness of the solution

4.3 Error Comparison

Following are the observed errors for each method:

Backward Euler (Δt= 0.1, 0.05, 0.01. 0.005)			
m=8	m=9	m=10	m=11
1	1	1	1
1	1	1.0325×10^{6}	1
1	1	1.0803×10^{30}	1

Forward Euler (Δt= 0.1, 0.05, 0.01. 0.005)			
m=8	m=9	m=10	m=11
1.4787×10^{30}	1.3052×10^{36}	8.2565×10^{39}	6.2057×10^{45}
4.1864×10^{84}	3.095×10^{66}	NaN	2.0371×10^{88}
5.1830×10^{205}	9.8026×10^{265}	NaN	NaN

Crank Nicolson (Δt= 0.1, 0.05, 0.01. 0.005)			
m=8	m=9	m=10	m=11
3.9163	2.9964	0.9993	0.9997
3.1075	2.8802	0.9995	0.9997
1.1886	42.8691	1.4339	1.3993

KSS (Δt= 0.1, 0.05, 0.01. 0.005)			
m=8	m=9	m=10	m=11
2.6523×10^{-4}	3.7474×10^{-4}	0.9903	0.9904
7.3782×10^{-5}	1.2398×10^{-4}	0.9903	0.9904
4.8695×10^{-5}	4.1961×10^{-5}	0.9903	0.9904

The cases of $m = 10$ and $m = 11$ for KSS ended with unexpected results that have been speculated to improve if we further reduce the time-step for the reference solution and re-compute the error. However, at the present moment this is outside the scope of this thesis.

4.4 Conclusion

Considering the preceding results with respect to changing grid size and changing time step in comparison to the reference solution we have chosen, we can conclude that KSS gives us significant improvement as compared to other methods. Backward Euler being unconditionally stable, tends to lead to excessive damping. On similar note, Forward Euler is unstable unless time steps chosen are very small. Crank-Nicolson theoretically expected to achieve second-order accuracy has been inconclusive the this instance. However, this would not the most suitable method to deal with a second order Cable Equation.

4.5 Where do we go from here?

This thesis is helping us prepare a solid foundation for using numerical methods to be applied to problems based in Computational Neuroscience. Strictly speaking of the Cable equation, the following are some research ideas gathered from [2] and [9] that can be explored using our methods:

- Using the solution of Cable equation as $reverse\ functions$ to approximate the area of cross sections to detect anomalous changes for early detection in Neuro-degenerative proteins

- Building and solving first order PDEs to approximate pressure release in patients suffering stroke related swelling in the brain in order to minimize nerve damage

- Better understanding of fundamentals of certain conditions as Depression, ADD, ADHD, etc with study of behaviour of action potentials

- Comparing chemical changes (ionic charges, levels of chemicals, relative intake of medications) to behavioural changes in order to give better understanding of functioning of a healthy brain

Appendix A

MATLAB CODE

The first part of the code chooses grid points, time length, time steps, constant coefficients and builds the differential operator. Then it calls two different functions "triple stencil" and "kss

```
%loop this to compare reference Solution and computed solution
%this calls triplestencil.m

%enter all varying parameter informatoin
prompt1 = "theta=?"; %bwd euler=0, fwd euler=1, crank-nicholson=0.5
theta = input(prompt1);
prompt2 = "dt=?" %0.05, 0.01, 0.005
dt = input(prompt2);
prompt3 = "m=?"; %m=8,9,10,11
m = input(prompt3);

%constant coefficients of differential operator
a=0.1;
b=0.1;
c=1;
t=1;
cm=0.5;
rl=180;
rm=0.5;
alpha=1/(cm*rl);
beta=1/(cm*rm);

%choose number of points
n=((2^(m)-1));
h=1/(n+1);
x=linspace(0, 1, n+2);
```

```
x=x';


%defining zero vectors that fills in afterwards
xavg=zeros(n+1,1);
r=zeros(n+1,1);
A=zeros(n+1,1);
Adiag=zeros(n,1);

%for loop that will computer Am
for i=1:n+1
    xavg(i)=(x(i)+x(i+1))/2;
    for j=1:n+1
        r(j)=(a*(cos(xavg(j))))+(b*(sin(xavg(j))))+(c);
        A(j)=(pi)*((r(j))^2);
    end
    for k=1:n
        Adiag(k)=-(A(k)+A(k+1));
    end
end
Am=(spdiags([A(2:n+1) Adiag A(1:n)],-1:1,n,n))/(h^2);
L=alpha*Am-beta*speye(n,n);

%calling triple stencil+
V=readtable('voltage_allrep.txt');
V=table2array(V);
V=V(1:n,1);
Vt=triplestencil(L,V,t,dt,theta,n);
plot(Vt);

%calling fourier basis vectors
V_tnp1=fourierbasisvectors(V,n,L,dt,t);
```

The first function we have used is named "triple stencil" based on the fact that it implements Backward Euler, Forward Euler and Crank-Nicolson with respect to θ

```
%triple stencil function
function V=triplestencil(L,V,t,dt,theta,n)
timesteps=t/dt;
RHS=((eye(n,n)+(theta*dt*L)));
LHS=gausselimA((eye(n,n)-((1-theta)*dt*L)));
plot(V)
title('Initial data for action potential in a mammalian brain')
ax=gca;
ax.FontSize=10;
pause
for i=1:timesteps
    if theta==0
        V=forwsub(LHS,V);
        Vtnp1=backsub(LHS,V); %vref comes from BW euler
    else
        if theta==1
            Vtnp1=RHS*V;
        else
            V=forwsub(LHS,(RHS*V));
            Vtnp1=backsub(LHS,V);
        end
    end
    if rem(i,100)==0
        plot(V);
        xlabel('x')
        ylabel('Potential values: V in mV')
        title({[ 't=' num2str(t) ', Grid size n=' num2str(n) ',...
        dt=' num2str(dt) ]})
        ax=gca;
        ax.FontSize=10;
        pause(0.01)
    end
end
```

```
end
```

The next function we have build is called "kss" which used fourier basis vectors to compute fourier sine coefficients of the solution.

```
function V_tnp1=kss(V,n,L,dt,t)
timesteps=t/dt;
I=eye(n,n);
S=mydst(I);
S=S’;
V_np1=zeros(n,1);
for j=1:timesteps
    for i=1:n
        R0=[S(:,i) V];
        [X1, B0]=qr(R0, "econ");
        M1=X1’*L*X1;
        R1=L*X1-X1*M1;
        [X2,B1]=qr(R1,"econ");
        M2=X2’*L*X2;
        Tau2=[M1 B1’;B1 M2];
        Tauexp=expm(Tau2*dt);
        V_np1_t=B0’*Tauexp(1:2,1:2)*B0;
        V_np1(i,1)=V_np1_t(1,2);
    end
    V_tnp1=myidst(V_np1);
    if rem(i,100)==0
        plot(V_tnp1);
        xlabel(’x’)
        ylabel(’Potential values: V in mV’)
        title({[ ’t=’ num2str(t) ’, Grid size n=’ num2str(n) ’...
        dt=’ num2str(dt) ]})
        ax=gca;
        ax.FontSize=10;
        pause(0.01)
    end
end
```

BIBLIOGRAPHY

[1] Peter Dayan and L.F.Abbott. *Theoretical Neuroscience*. MIT Press, 2001.

[2] Asha Gopinathan. *Solving Cable equation with compact difference scheme*. 2013.

[3] Eric R. Kandel with James H. Schwartz and, Thomas M. Jessell. *Principles of Neural Science*. McGraw-Hill Companies, 2000.

[4] Bailey Rester with J.V. Lambers and, P. M. Jordan. *Acoustic singular surfaces in an exponential class of inhomogeneous gases: A new numerical approach based on Krylov subspace spectral methodologies*. 2022.

[5] J.V.Lambers. *Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration*. 2008.

[6] J.V.Lambers and Amber Sumner. *Explorations in Numerical Analysis*. 2019.

[7] Erick J. López-Sánchez and Juan M. Romero. *Cable Equation for general geometry*. 2017.

[8] Gene H. Golub and Meurant. G. *Matrices, Moments and Quadrature*. June-1993.

[9] Tjitse van der Molen Tal Sharf Paul K. Hansma Kenneth S Kosik Linda Petzold Kenneth R Tovar Zhuowei Cheng, Elmer Guzman. *Automated detection of extracellular action potentials from single neurons*. 2006.