Numerical Study of the Electrical Breakdown of a Non-Ohmic resistor

ARCC Scholar thesis

in partial fulfillment of the requirements
of the ARCC Scholar program

Mauricio A. Flores

April 26, 2013

Abstract

The present work is a numerical study of a highly nonlinear partial differential equation (PDE). This PDE is associated with an electrical configuration with a silicon carbide non-ohmic resistor and has the form of a diffusion equation with a diffusion constant that is dependent on field strength. We have numerically solved this nonlinear PDE through Fourier decomposition and an iterative boundary condition matching. The accuracy of our solution has been checked by comparing results against the linear limit of the nonlinear PDE and by comparing the results with a time evolution of the initial data for the system. A discussion is presented of the methods utilized, and the accuracy of the results obtained.

1 Acknowledgments

I am deeply grateful to my thesis advisor Dr.Richard H.Price, for his guidance through the course of this work. His scientific vision and dedication have constituted a wonderful model for a scientific career ahead of me. I would like to thank all members of the ARCC Executive Committee for giving me the opportunity to be a part of the ARCC Scholars program. I thank Dr. Fredrick A. Jenet for directing this program and being my research mentor during my freshman year. I give special thanks to Dr. Matthew J. Benacquista for being my research mentor during my sophomore and junior years. Dr. Benacquista's friendliness and encouragement helped me overcome difficult times.

I thank the Department of Physics & Astronomy at the University of Texas at Brownsville for providing the setting and support that allowed this work to be completed. I thank the National Science Foundation for grants AST-0750913 and HRD-0734800, as well as NASA for grant NNX09AV06A. This support funded the ARCC Scholars program, and hence supported my undergraduate education and research.

My life at UT Brownsville would not have been the same without the company and support of my friends. I would like to give special thanks to David Boon, Max Cornejo and Jesus Rivera. Additionally, I would like to thank my beloved girlfriend, Aura Salazar, for inspiring me to be a better version of myself.

Finally, I would like to thank my parents, Pedro and Patricia, for encouraging me to pursue a college education. Their life is a living example of perseverance, hard work and self-improvement. It is impossible for me to express in a few sentences my gratitude and admiration for them.

Contents

2 Background

This problem is related to a problem brought to our attention by Alexander Kazhanov of National Electric Coil. The objective of our work is to provide a numerical analysis of the steady state behavior of electrical coatings whose resistivity is non-ohmic, that is, non-constant.

Simulating this physical setup amounts to solving a one-dimensional diffusion-type nonlinear partial differential equation numerically. This equation is

$$\frac{\partial^2 U}{\partial x^2} = f(U)\frac{\partial U}{\partial t}, \tag{1}$$

where

$$f(U) = \frac{e^{-|E|}}{1+|E|}, \tag{2}$$

and where the relationship between variables E and U is given by

$$U = Ee^{|E|}. \tag{3}$$

In the equations above, x is a dimensionless space variable, t is a dimensionless time variable and E is a dimensionless description of the electric field in the non-ohmic resistor. We want to solve this problem on the domain $x \in [0, x_{max}]$, $t \in [0, t_{max}]$, subject to boundary conditions

$$\partial U/\partial t = V_0 \cos t \text{ at } x = 0, \tag{4}$$

$$U \to 0 \text{ as } x \to x_{max}. \tag{5}$$

Although most physical parameters have been absorbed into the variables, we are still left with two physical parameters, namely x_{max}, representing the length of the coating, and V_0. The size of V_0 determines how nonlinear our problem is. As our problem becomes more nonlinear it becomes more difficult to be solved. The nonlinearity of this problem presents itself through the nonlinear factor $f(U)$. In Fig. 1 we can see a sample plot of $f(U)$ for the particular choice $V_0 = 1$ and $x_{max} = 1$.

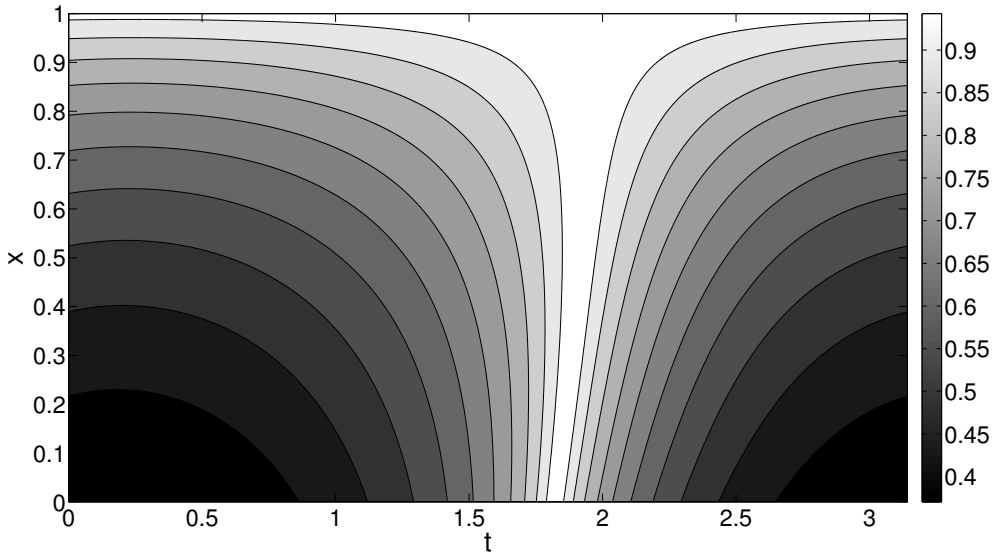

Figure 1: This plot shows the values that $f(U)$ takes on our solution domain, in logarithmic scale, for the case where $V_0 = 1$ and $x_{max} = 1$. As we can see from this plot, the smallest value of $f(U)$ is approximately 0.4.

3 Numerical Scheme

In this section we describe a method for solving Eq. (1) for $U(x,t)$ in the domain $x \in [0, x_{max}]$, $t \in [0, t_{max}]$.

3.1 Fourier series decomposition

We seek a solution $U(x,t)$ which is periodic in time, therefore our solution approach is based on expressing $U(x,t)$ as a Fourier series.

$$U = \sum_{n=1}^{\infty} A_n(x)\cos(nt) + B_n(x)\sin(nt). \tag{6}$$

The reason why we exclude the $n = 0$ term from the sum is the fact that the term $A_0(x)$ is the time average of the function $U(x,t)$, which must be zero because the input voltage has a time average of zero. It follows that the variable U is oscillatory and has a time average of zero. Now, the problem at hand is to solve for each $A_n(x)$ and $B_n(x)$.

The decomposition presented in Eq. (6) allows us to convert Eq. (1) into

$$\sum_{n=1}^{\infty}\left[\frac{d^2A_n}{dx^2}\cos(nt)+\frac{d^2B_n}{dx^2}\sin(nt)\right]=f(U)\sum_{n=1}^{\infty}\left[nB_n\cos(nt)-nA_n\sin(nt)\right]. \tag{7}$$

From this equation we can derive expressions for d^2A/dx^2 and d^2B/dx^2. The details of this derivation are relegated to Appendix 9.1. These expressions are

$$\frac{d^2A_n(x)}{dx^2}=\frac{1}{\pi}\int_{-\pi}^{+\pi}\cos(nt)f(U)\left(\sum_{m=1}^{\infty}mB_m\cos mt-mA_m\sin mt\right)dt, \tag{8}$$

$$\frac{d^2B_n(x)}{dx^2}=\frac{1}{\pi}\int_{-\pi}^{+\pi}\sin(nt)f(U)\left(\sum_{m=1}^{\infty}mB_m\cos mt-mA_m\sin mt\right)dt. \tag{9}$$

These expressions for d^2A_n/dx^2 and d^2B_n/dx^2 allow us to compute $A_n(x)$ and $B_n(x)$ given $A_n(x{=}0)$ and $B_n(x{=}0)$. The problem to be solved has been reduced to solving the system of ordinary differential equations stated in Eq. (8) and Eq. (9) for each $n = 1, 2, \ldots$

For numerical computations we must truncate our Fourier series after a certain number of modes have been included. This number of modes must be small enough that numerical solutions can be computed in a reasonable amount of time, yet not so small that the truncation error introduced makes the numerical solutions useless. The constant N will denote the number of modes to be kept in the Fourier expansion. One of the goals of this work is to determine an optimal value for N. For the remainder of this thesis, we keep only the first N terms.

3.2 Enforcing boundary conditions

In Sec. 3.1 we have determined how to obtain a numerical solution for $A_n(x)$ and $B_n(x)$ given $A_n(x{=}0)$ and $B_n(x{=}0)$. These values must be chosen in order to satisfy the boundary conditions of the nonlinear PDE subject of this work. The first condition, stated on Eq. (4) can be satisfied by simply choosing $dA_1(x{=}0)/dt = V_0$ and setting all other initial derivatives to zero. The boundary condition at x_{max}, stated in Eq. (5) can only be met by carefully choosing $A_n(x{=}0)$ and $B_n(x{=}0)$. A method for choosing these initial mode values is discussed in the following section.

3.3 Iterative mode start

We must choose $A_n(x{=}0)$ and $B_n(x{=}0)$ in order to enforce Eq. (4). This choice cannot be made directly because it is not initially known how $A_n(x)$ and $B_n(x)$ develop in x. Instead, a reasonable approach for solving this problem is to use a "shooting method"; we first perform a numerical experiment, observe the end behavior $A_n(x{=}x_{max})$ and $B_n(x{=}x_{max})$, and then adjust initial mode values accordingly. We interpret $A_n(x{=}x_{max})$ and $B_n(x{=}x_{max})$ as functions of $A_n(x{=}0)$ and $B_n(x{=}0)$.

For any smooth function $f(x, y)$, a first order Taylor expansion gives

$$f(x + \delta x, y + \delta y) \approx f(x, y) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y, \tag{10}$$

where δx and δy are assumed to be small. For suitable choices of δx and δy it is possible to make $f(x + \delta x, y + \delta y)$ go very close to zero. This idea can be applied to solve our problem, where instead of $f(x + \delta x, y + \delta y)$ we have functions $A_n(x{=}x_{max})$ and $B_n(x{=}x_{max})$ for each n.

We must make the assumption that $A_n(x{=}x_{max})$ and $B_n(x{=}x_{max})$ are continuous and well behaved functions of $A_n(x{=}0)$ and $B_n(x{=}0)$. Then, small changes in $A_n(x = 0)$ and $B_n(x = 0)$ will produce small changes in $A_n(x{=}x_{max})$ and $B_n(x{=}x_{max})$.

It is useful to consider $A_n(x)$ and $B_n(x)$ at any x to be the components of N dimensional vectors. We could choose vectors δA_n and δB_n to be small adjustments on vectors $A_n(x{=}0)$ and $B_n(x{=}0)$ such that these small adjustments bring $A_n(x{=}x_{max})$ and $B_n(x{=}x_{max})$ closer to the zero vector. This adjustment process is iterated until all A_n and B_n modes are small. Our computer implementation tests this condition by requiring

$$\sum_{n=1}^{N}(A_n^2 + B_n^2) \leq 10^{-12}. \tag{11}$$

The method we propose for obtaining optimal initial data for $A_n(x{=}0)$ and $B_n(x{=}0)$ consists of applying the following two equations iteratively

$$0 = A_n(x_{max}) = A_n^*(x_{max}) + \sum_{k=1}^{N}\left(\frac{\partial A_n^*(x_{max})}{\partial A_k(x{=}0)}\delta A_k + \frac{\partial A_n^*(x_{max})}{\partial B_k(x{=}0)}\delta B_k\right), \tag{12}$$

$$0 = B_n(x_{max}) = B_n^*(x_{max}) + \sum_{k=1}^{N}\left(\frac{\partial B_n^*(x_{max})}{\partial A_k(x{=}0)}\delta A_k + \frac{\partial B_n^*(x_{max})}{\partial B_k(x{=}0)}\delta B_k\right), \tag{13}$$

where $A_n^*(x_{max})$ and $B_n^*(x_{max})$ are the mode sizes at $x = x_{max}$ while $A_n(x_{max})$ and $B_n(x_{max})$ are the new mode sizes at $x = x_{max}$ At each iteration these expressions give a total of $2N$ equations for $2N$ unknowns. Solving this system allows us to obtain vectors δA_n and δB_n, which can then be applied to the initial mode values

$$A_n(x{=}0) = A_n^*(x{=}0) + \delta A_n \tag{14}$$

$$B_n(x{=}0) = B_n^*(x{=}0) + \delta B_n. \tag{15}$$

Equations(12)-(15) are applied iteratively until Eq. (11) is satisfied.

4 Computer Implementation

The solution scheme described in Sec. 3 has been implemented in C++. In the following subsections we discuss specific characteristics and techniques utilized in this implementation.

4.1 Domain discretization

The domain of our problem is $x \in [0, x_{max}]$ and $t \in [0, 2\pi]$. We have discretized this domain uniformly so that $x_i = i\Delta x$ where $\Delta x = x_{max}/(N_x - 1)$ is the uniform spacing between x grid points. The index i runs from 0 up to $N_x - 1$. Analogously, $t_j = j\Delta t$ where Δt is the uniform spacing $2\pi/(N_j - 1)$between time grid points, and j is an index running from 0 up to $N_j - 1$.

The choice of a uniform domain for time is based on the sinusoidal nature of our base functions $\sin(nt)$, $\cos(nt)$, $\sin(2nt)$, etc. The numerical integration of sinusoidal functions tends to be more accurate when the function is sampled on a uniform grid.

4.2 Spatial integration

After we have established our choice for $A_n(x{=}0)$ and $B_n(x{=}0)$ for each n we must advance these mode values in space according to Eq. (8) and Eq. (9). Assuming we have $A_n(x{=}x_i)$ and $B_n(x{=}x_i)$ for each n, we obtain, from Eqs .(8),(9)

$$A_m''(x_{i+1}) = \frac{1}{\pi}\sum_{j=1}^{N_t} \cos(mt_j) f(U) \left(\sum_{n=1}^{N} nB_n(x_i)\cos nt_j - nA_m(x_i)\sin nt_j \right) \Delta t, \tag{16}$$

$$B''_m(x_{i+1}) = \frac{1}{\pi}\sum_{j=1}^{N_t}\sin(mt_j)f(U)\left(\sum_{n=1}^{N} nB_n(x_i)\cos nt_j - nA_m(x_i)\sin nt_j\right)\Delta t, \tag{17}$$

where prime (') denotes d/dx. Then a simple Euler step allows to obtain $A'_n(x_{i+1})$, $B'_n(x_{i+1})$, $A_n(x_{i+1})$ and $B_n(x_{i+1})$ from

$$A'_n(x_{i+1}) = A'_n(x_i) + A''_n(x_{i+1})\Delta x, \tag{18}$$

$$A_n(x_{i+1}) = A_n(x_i) + A'_n(x_{i+1})\Delta x, \tag{19}$$

$$B'_n(x_{i+1}) = B'_n(x_i) + B''_n(x_{i+1})\Delta x, \tag{20}$$

$$B_n(x_{i+1}) = B_n(x_i) + B'_n(x_{i+1})\Delta x. \tag{21}$$

This Euler step is repeated until $i + 1 = N_x - 1$ after which the computation is complete.

5 Numerical Results

5.1 Decrease of mode magnitude

One of the most important practical problems to be considered in this study is the magnitude fall off of the modes, as the mode number gets large. For numerical computations, it is necessary to understand how these modes fall in magnitude in order to determine whether it is possible to neglect higher modes in order to reduce computational cost. A method for determining this fall off is to perform a simulation with a large number of modes, and then compare mode magnitudes in order to observe how significant each mode is with respect to the rest. In Fig. 2 and Fig. 3 we can see how modes A_n and B_n decay as the mode number increases.

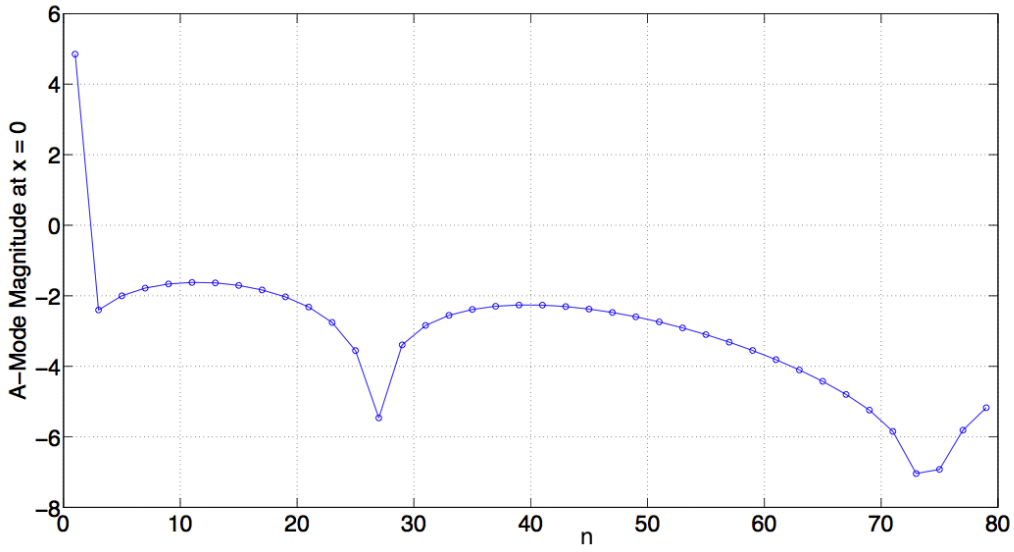

Figure 2: The decrease in $\log_{10}(A_n(x{=}0))$, the $x = 0$ mode amplitude, with increasing n. In this model $x_{max} = 10$ and $V_0 = 128$. Though there is an erratic behavior around $N \approx 20$, we clearly observe an overall drop in magnitude as n increases.

6 Quality Assessment

One of the most important aspects of numerical simulation is the ability to prove results are reliable. A given numerical scheme may provide fast and highly precise answers, but if we are unable to trust these results, the numerical scheme is useless. We have applied several different methods to verify the correctness of our numerical simulations.

6.1 Parameter adjustment

The first and most basic method for testing the reliability of our results is to make minor modifications to our computational parameters to test whether this *insignificant* parameter variation produces a significant variation in our numerical result. For example, when we discretize our space domain we arbitrarily choose N_x, which is the number of grid points in x. We could choose $N_x = 1000$ or $N_x = 999$, but there is no particular reason why we could prefer one option over the other. Then, if using N_x yields solution

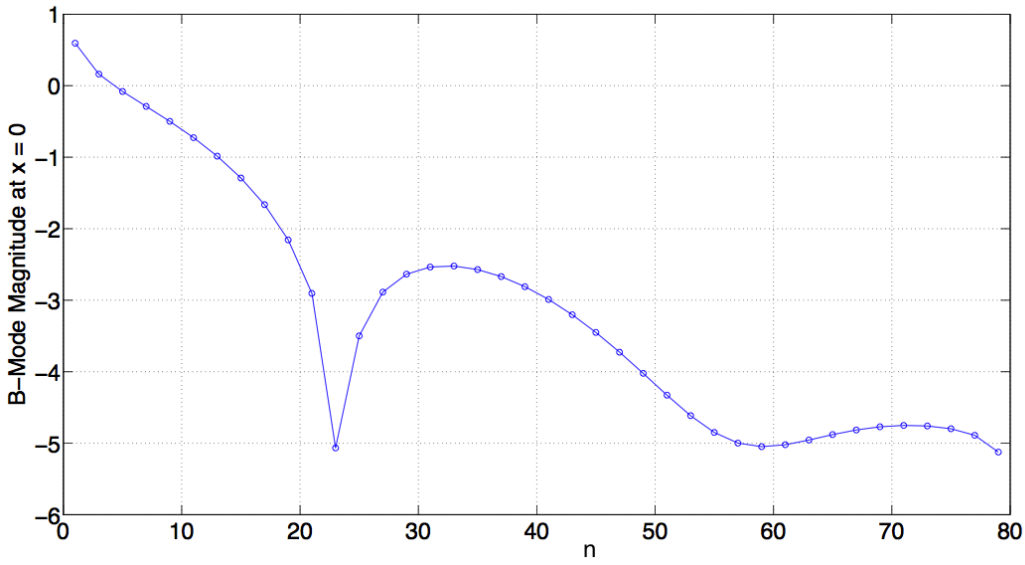

Figure 3: The decrease in $\log_{10}(B_n(x{=}0))$, the $x = 0$ mode amplitude, with increasing n. In this model $x_{max} = 10$ and $V_0 = 128$. Though there is an erratic behavior around $N \approx 20$, we clearly observe an overall drop in magnitude as n increases.

$U_1(x,t)$ and using $N_x = 999$ yields solution $U_2(x,t)$ we would hope that $U_1 \approx U_2$. Otherwise, how could we choose which numerical solution is correct between U_1 and U_2? If U_1 and U_2 are very different numerical solutions this would show our solution method is unstable and therefore ineffective. Varying parameters does not prove a method is correct, but it may provide useful evidence if a method is incorrect. Among the parameters we varied to test the precision of our results are time and space grid sizes, number of modes, initial voltages and the value of parameter x_{max}.

Fortunately, our solution method proved to be stable under all these tests. One of these tests is included as an example. In Fig. 4 we can see a comparison between the numerical solution $U(x,t)$ obtained using 2 and 3 modes.

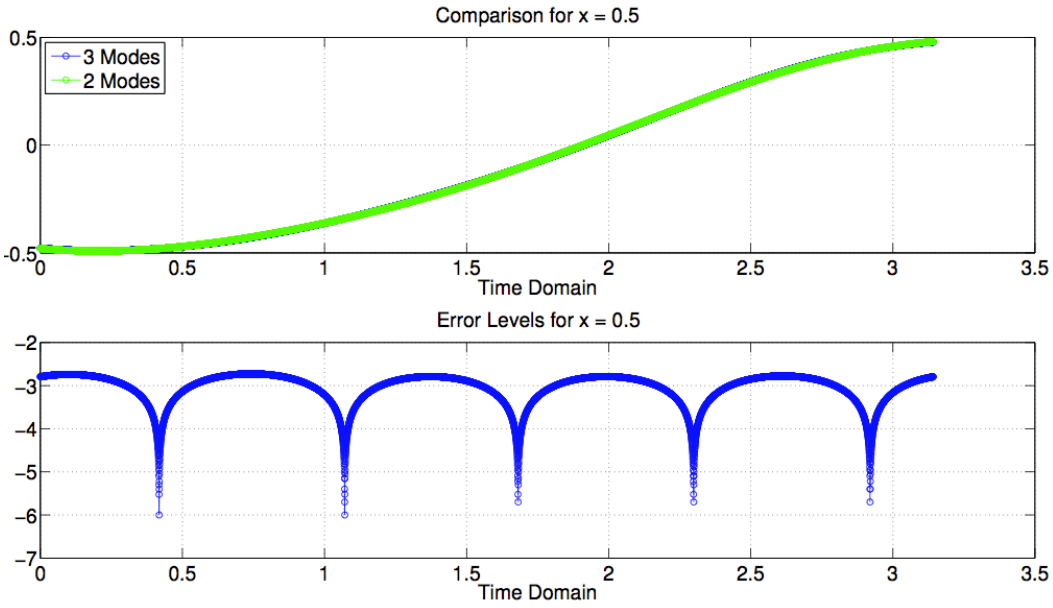

Figure 4: The top figure shows numerical solutions using $N = 3$ and $N = 2$, while the bottom plot shows the difference between these two solutions, in logarithmic scale. As we can see, our solution scheme is stable to changing mode numbers; the two results are extremely close.

6.2 Linearized limit

The difficulty of solving the problem in question is its high nonlinearity, caused by the term

$$f(U) = \frac{e^{-|E|}}{1+|E|}.$$

Note that $f(U) \to 1$ as $E \to 0$, which turns our problem into a linear PDE, that can be readily solved analytically. The magnitude of E is controlled by the initial driving potential V_0. For sufficiently small choices of V_0 our simulation should render a solution $U(x,t)$, which is very similar to the solution of Eq. (1) when we set $f(U) = 1$, that is $\partial^2 U/\partial x^2 = \partial U/\partial t$. We can follow the procedure suggested in Sec. 3.1 to obtain

$$\frac{d^2 A_n(x)}{dx^2} = nB_n(x), \tag{22}$$

$$\frac{d^2 B_n(x)}{dx^2} = -nA_n(x). \tag{23}$$

The details of this simplification are left to Appendix 9.2. Then we choose that all modes other than A_1 are set to 0 at $x = 0$, and then it is easy to see that higher modes will never be introduced in this calculation. We are left with the task of solving the system of equations:

$$A_1''(x) = B_1(x) \qquad \text{and} \qquad B_1''(x) = -A_1(x), \tag{24}$$

subject to the initial conditions $A_1(x=0) = V_0$, $B_1(x=0) = 0$, and the conditions at x_{max}, $A_1(x) \to 0$ and $B_1(x) \to 0$ as $x \to \infty$. This system has been solved with the use of Wolfram Mathematica 8.0 software. The analytical solution is:

$$A_1(x) = \sqrt{2}V_0\left[e^{-z}(a\sin z + b\cos z) + e^{z}(c\sin z + d\cos z)\right] \tag{25}$$

$$B_1(x) = \sqrt{2}V_0\left[e^{-z}(b\sin z - a\cos z) + e^{z}(c\cos z - d\sin z)\right], \tag{26}$$

where:

$$a \equiv \frac{1}{h}\left[e^{2z_m} + \cos(2z_m) - \sin(2z_m)\right]$$

$$b \equiv -\frac{1}{h}\left[e^{2z_m} + \cos(2z_m) + \sin(2z_m)\right]$$

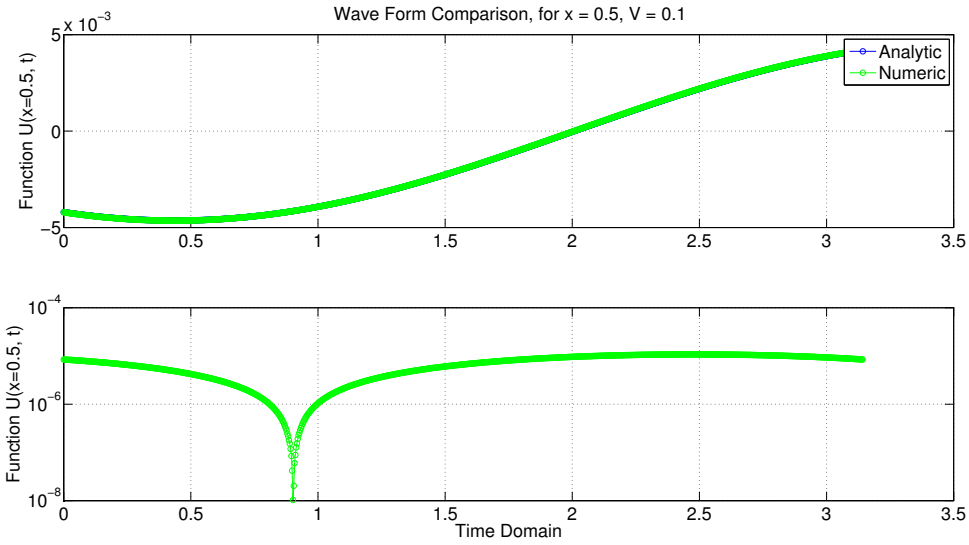

Figure 5: Comparison between an analytical result and a numerical result (top) and an error plot with the difference between these two solutions (bottom), for the almost linear case, where $V_0 = 0.01$. As we can see, the match between these two solutions is nearly perfect.

$$c \equiv \frac{1}{h}\left[e^{-2z_m} + \cos(2z_m) + \sin(2z_m)\right]$$

$$d \equiv \frac{1}{h}\left[e^{-2z_m} + \cos(2z_m) - \sin(2z_m)\right],$$

where

$$z \equiv \frac{x}{\sqrt{2}} \quad , \quad z_m \equiv \frac{x_{max}}{\sqrt{2}} \quad \text{and} \quad h \equiv 4(\cosh(2z_m) + \cos(2z_m)).$$

We performed a series of numerical experiments to verify the quality of our results using these methods. Our results proved to be convincing, for the choices of $V_0 = 0.01$ and $V_0 = 0.1$. We can see these results for a fixed x-value, in Fig. 5 and Fig. 6.

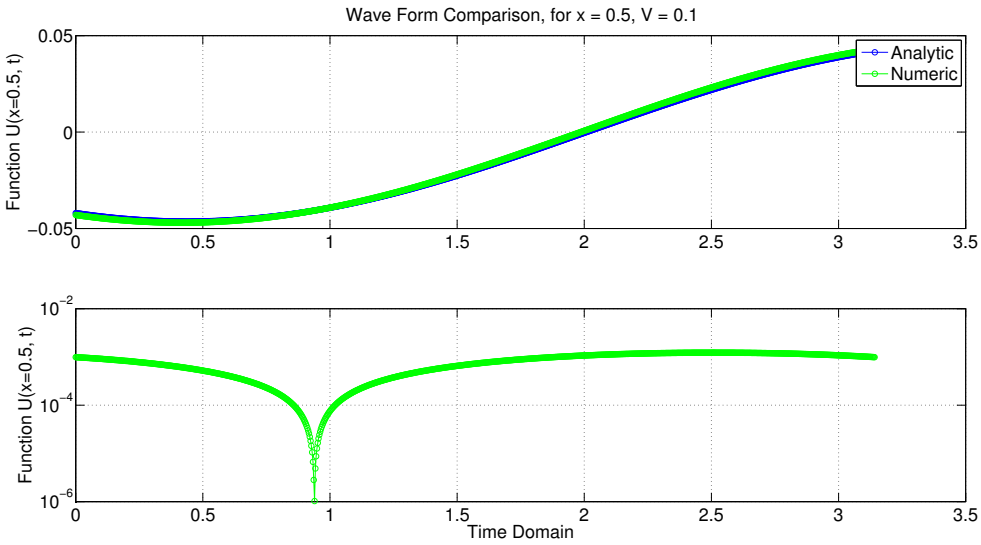

Figure 6: Solution for $V_0 = 0.1$. We can see from the top graph, the analytical and the numerical solutions start to show some deviation from one another. This is to be expected, as $f(U) \approx 1$ is only a good approximation for very small V_0. As we can see from the bottom graph, changing V_0 from 0.01 to 0.1 increases the error by a factor of 100.

6.3 Finite difference post-processing

An alternative method for determining the success of the solution obtained is to simply verify whether Eq. (1) has been satisfied at each point in our x, t grid. For a point (x_i, t_j) in our discretized domain, the condition to check is

$$\frac{U(x_{i+1}t_j) - 2U(x_i, t_j) + U(x_{i-1}, t_j)}{\Delta_x^2} - f(U(x_i, t_j))\frac{U(x_i, t_{j+1}) - U(x_i, t_{j-1})}{2\Delta t} \leq C, \tag{27}$$

where

$$f(U(x_i, t_j)) = \frac{e^{-|E(x_i,t_j)|}}{1 + |E(x_i, t_j)|}. \tag{28}$$

One of the limitations of this method is to determine what is an acceptable error bound C, and to take into account the magnitude fluctuations of $U(x_i, t_j)$. In certain regions our errors may appear to be quite large (or quite small), yet this simply means $U(x_i, t_j)$ is larger (or smaller) in this region. Normalizing the error presented in Eq. (27) by $U(x_i, t_j)$ seems to be an appropriate alternative, although this introduces a new problem which is how to evaluate the quality of our solution when $U(x_i, t_j)$ is a very small number.

In Fig. 7 we can see an example of a normalized finite difference error. In Fig. 6.3 we can see the corresponding numerical solution $U(x, t)$, which serves as a reference to understand where normalized errors should be larger (e.g. where $U(x, t)$ is very small).

6.4 Finite difference evolution

As a final numerical test, we decided to test whether our numerical solution could reproduce itself under a finite difference evolution scheme. As in Sec. 6.3, we solve Eq. (27) for $U(x_i, t_{j+1})$, to obtain

$$U(x_i, t_{j+1}) = \frac{\Delta t}{\Delta x^2}\frac{U(x_{i+1}, t_j) - 2U(x_i, t_j) + U(x_{i-1}, t_j)}{f(U(x_i, t_j))} + U(x_i, t_j). \tag{29}$$

Using Eq. (29) we may derive a simple yet effective method for verifying the quality of the numerical solution method we have proposed. Our solution scheme produces initial data for $U(x = 0, t)$, using this data, alongside our boundary conditions we can simply evolve $U(x_i, t_j)$ in time, and then check whether this solution coincides with the numerical results we have obtained. These two solution strategies are quite different in their methodology; if they

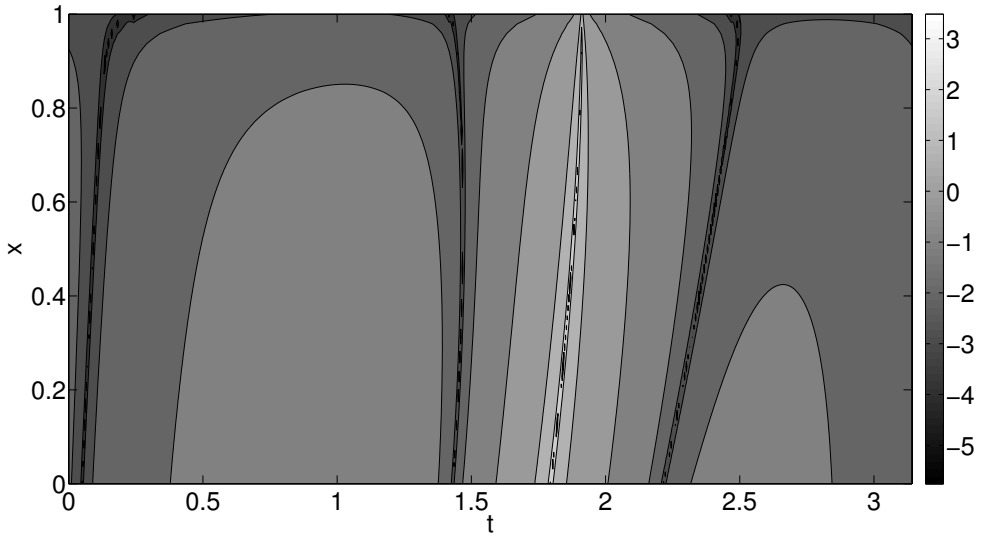

Figure 7: Finite difference error as described in Eq. (27). As we can see from this plot, the percentage error on our finite difference test is no more than 10% for the vast majority of our solution domain. The only exception occurs around the vertical line $t \approx 1.7$, which coincides with the region where $U(x,t)$ is very small, as can be seen in Fig. 6.3.

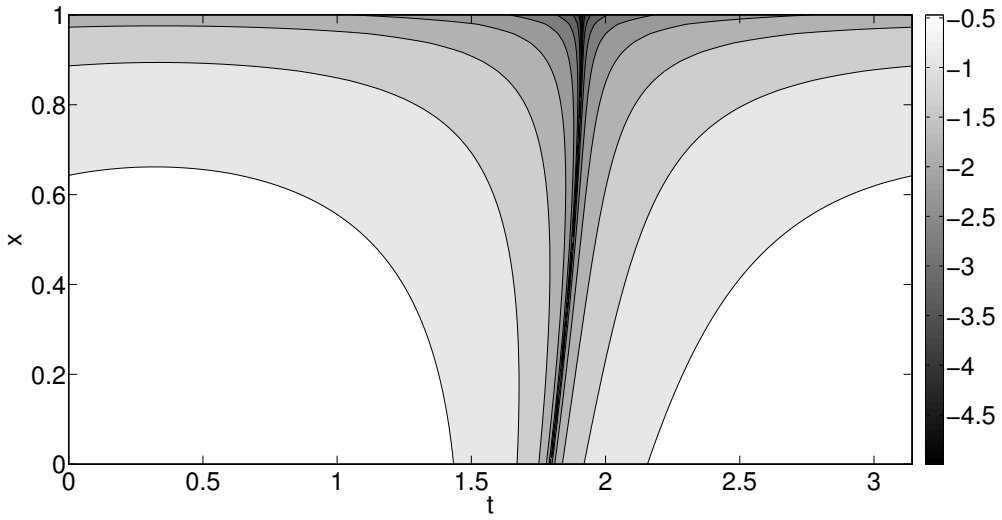

Figure 8: A logscale plot of $U(x,t)$ for the same simulation that produced Fig. 7. As we can see, $U(x,t)$ is very small where $t \approx 1.7$.

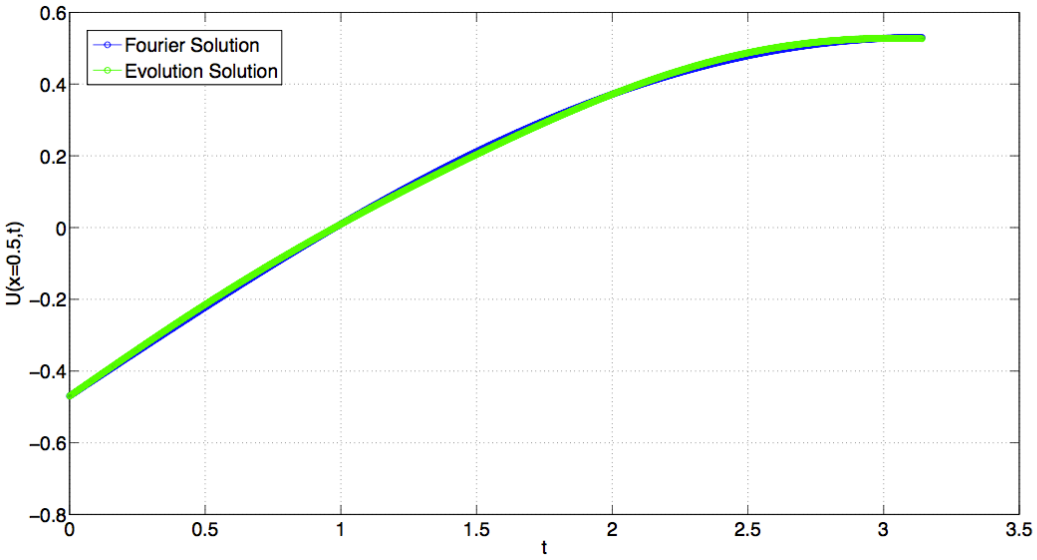

Figure 9: A comparison at $x = 0.5$ between the solutions obtained by Fourier analysis and the solution obtained by finite difference evolution in time. As we can see, the two solutions match almost perfectly.

match this would decisively prove the reliability of our results. We can see one such comparison for $x = 0.5$.

7 Conclusions

In this work we have presented a numerical solution scheme for solving Eq. (1). Most of our efforts have been directed toward testing the precision and accuracy of our solution.

All of our numerical experiments indicate that the behavior of $U(x, t)$ is dominated by the modes A_1 and B_1. This result is quite surprising, yet positive, as it allows us to compute our numerical solutions much more quickly.

The numerical solution scheme we have proposed has proven to be effective against most of our tests. We successfully showed this solution is compatible with the linearized limit case, where $E \to 0$. For the vast majority of our solution domain we managed to prove this solution is consistent with Eq. (1) when using finite differences as a check.

Unfortunately, our finite difference evolution results only give us a partial understanding of the behavior of our physical problem. While we were able to show that evolving our initial data for $U(x = 0, t)$ does replicate our numerical solution, we were not able to truly understand what happens after we continue to evolve our results for more than one cycle. As a matter of fact, we would have liked to observe how this solution replicates itself over and over in cycles of 2π. Although we were able to see such behavior, this solution replicated itself exactly, which is very suspicious numerically. We could not devise a test for proving this replication was in fact correct, and not just a computational mistake.

8 Future Work

The work presented in this thesis could be extended and improved in several ways. Perhaps one of the most important improvements to be performed is to implement a better time evolution scheme, in order to provide faster and more reliable results for the test presented in Sec. 6.4. Any implicit time-stepping, such as Crank-Nicolson, could achieve this goal.

Additionally, it would be nice to see how much nonlinearity our numerical solution scheme can handle effectively. Most of our work was done using voltages on the range between 1 and 5, while our x_{max} was usually set to either 1 or 3. The use of more clever numerical schemes (or faster computers) would allow to handle larger parameter values with finer grid sizes; thus giving us a better understanding of the nonlinearity present in this problem.

An alternative option for improving the results we have obtained would be to implement adaptive step sizes in our spatial stepping, thus making our results more accurate in regions where Eq.(1) is more nonlinear.

9 Appendices

9.1 Fourier analysis derivation

In Sec. 3 we obtained

$$\sum_{n=1}^{N}\left[\frac{d^2A_n}{dx^2}\cos(nt)+\frac{d^2B_n}{dx^2}\sin(nt)\right]=f(U)\sum_{n=1}^{N}\left[nB_n\cos(nt)-nA_n\sin(nt)\right]. \tag{30}$$

The objective is to solve for $d^2A_n(x)/dx^2$ and $d^2B_n(x)/dx^2$ from Eq. (30). We must first recall the following identities, valid for $n, m \in \mathbb{N}$

$$\frac{1}{\pi}\int_{-\pi}^{+\pi}\sin(nt)\sin(mt)dt=\delta_{mn}, \tag{31}$$

$$\frac{1}{\pi}\int_{-\pi}^{+\pi}\cos(nt)\cos(mt)dt=\delta_{mn}, \tag{32}$$

$$\frac{1}{\pi}\int_{-\pi}^{+\pi}\sin(nt)\cos(mt)dt=0. \tag{33}$$

We can multiply both sides of Eq. (30) by $\cos(mt)$, and then integrate over $[-\pi,+\pi]$. The left-hand side of this new equation is

$$\int_{-\pi}^{+\pi}\cos(mt)\sum_{n=1}^{N}\left[\frac{d^2A_n}{dx^2}\cos(nt)+\frac{d^2B_n}{dx^2}\sin(nt)\right]dt \tag{34}$$

$$=\sum_{n=1}^{N}\left[\frac{d^2A_n}{dx^2}\int_{-\pi}^{+\pi}\cos(mt)\cos(nt)dt+\frac{d^2B_n}{dx^2}\int_{-\pi}^{+\pi}\cos(mt)\sin(nt)dt\right].$$

We apply Eq. (33) to the first part of this expression, and Eq. (32) to the second part. Then the expression on Eq. (34) is equivalent to

$$\frac{d^2A_m(x)}{dx^2}\int_{-\pi}^{+\pi}\cos^2(mt)dt=\pi\frac{d^2A_m(x)}{dx^2}. \tag{35}$$

Thus, from Eq. (30) we derive an expression for the second spatial derivative of $A_m(x)$,

$$\frac{d^2A_m(x)}{dx^2}=\frac{1}{\pi}\int_{-\pi}^{+\pi}\cos(mt)f(U)\left(\sum_{n=1}^{N}nB_n(x)\cos(nt)-nA_n(x)\sin(nt)\right)dt. \tag{36}$$

By multiplying Eq. (30) by $\sin(mt)$ and then following the same procedure as before, we obtain an expression for the second spatial derivative of $B_n(x)$,

$$\frac{d^2B_m}{dx^2} = \frac{1}{\pi}\int_{-\pi}^{+\pi} \sin(mt)f(U)\left(\sum_{n=1}^{N} nB_n\cos(nt) - nA_n\sin(nt)\right)dt \quad (37)$$

We now notice that $f(U)$ is an even function, therefore the integrand on Eq. (36) and Eq. (37) is also even. These equations simplify to

$$\frac{d^2A_m}{dx^2} = \frac{2}{\pi}\int_{0}^{+\pi} \cos(mt)f(U)\left(\sum_{n=1}^{N} nB_n(x)\cos(nt) - nA_n(x)\sin(nt)\right)dt \quad (38)$$

$$\frac{d^2B_m}{dx^2} = \frac{2}{\pi}\int_{0}^{+\pi} \sin(mt)f(U)\left(\sum_{n=1}^{N} nB_n\cos(nt) - nA_n\sin(nt)\right)dt. \quad (39)$$

9.2 Linearized limit derivation

In Sec. 6.2 we consider the linearized limit $f(U) = 1$. Applying this assumption to Eq. (8) and Eq. (9) and assuming a truncated sum of N modes we obtain

$$\frac{d^2A_n(x)}{dx^2} = \frac{1}{\pi}\int_{-\pi}^{+\pi} \cos(nt)\left(\sum_{m=1}^{N} mB_m\cos mt - mA_m\sin mt\right)dt, \quad (40)$$

$$\frac{d^2B_n(x)}{dx^2} = \frac{1}{\pi}\int_{-\pi}^{+\pi} \sin(nt)\left(\sum_{m=1}^{N} mB_m\cos mt - mA_m\sin mt\right)dt. \quad (41)$$

Using orthogonality properties presented in Eq. (31), Eq. (32) and Eq. (33) these equations simplify to

$$\frac{d^2A_n(x)}{dx^2} = \frac{nB_n}{\pi}\int_{-\pi}^{+\pi} \cos(nt)\cos(nt)dt = nB_n(x), \quad (42)$$

$$\frac{d^2B_n(x)}{dx^2} = \frac{-nA_n}{\pi}\int_{-\pi}^{+\pi} \sin(nt)\sin(nt)dt = -nA_n(x). \quad (43)$$

9.3 Computation of the Lambert W function

The Lambert W function is an implicit multivalued complex function defined for any complex z as

$$z = W(z)e^{W(z)}. \quad (44)$$

For z real, the Lambert W function is complex multivalued for $z < -e^{-1}$, it is real multivalued for $-e^{-1} \leq z < 0$, and it is real single valued for $z \geq 0$. The branch which satisfies $W(z) \geq -1$ is called the principal branch of $W(z)$ and is denoted by W_0. The other branch, which satisfies $W(z) \leq -1$ is denoted by W_{-1}. Both W_0 and W_{-1} are represented in Fig. 10.

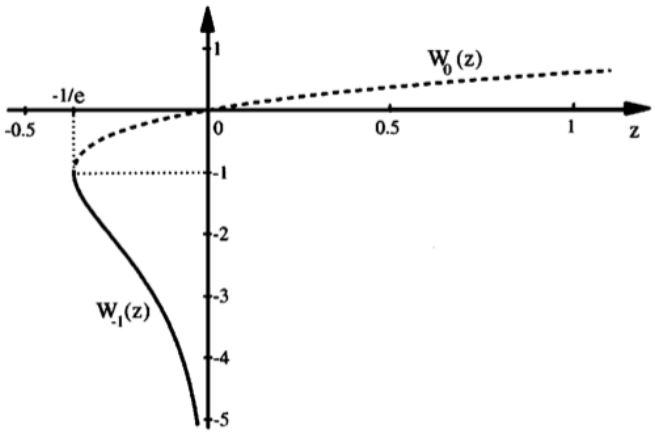

Figure 10: Two real branches of the lambert W function. The dashed line corresponds toW_{-1}, defined for $-e^{-1} < z < 0$. The solid line corresponds to W_0, defined for $z \geq -\frac{1}{e}$. This figure was obtained from Ref.[7]

For the purpose of this particular numerical study, the function of interest is $z = W(z)e^{|W(z)|}$. Therefore, the sign of $W(z)$ is determined by the sign of z. If z was negative, then $W(z) = -W(|z|)$. Thus, we are only interested in the evaluation of the branch W_0, for $z \geq 0$. A numerical implementation of the lambert function is readily available on the GNU Scientific Library, as well as on commercial software, such as Maple or MATLAB. Two of the most common numerical techniques utilized for solving for the Lambert W function are Newton's method, and Halley's method. A description of these methods can be found in several numerical analysis textbooks, such as Ref.[3] or Ref.[4].

References

[1] M. L. Boas, in *Mathematical Methods in the Physical Sciences, 3rd Edition*, (Wiley, New York, 2006).

[2] A. Iserles, in *A First Course in the Numerical Analysis of Differential Equations, 2nd Edition*, (Cambridge University Press, New York, 2009).

[3] F. B. Hildebrand, in *Introduction to Numerical Analysis, 2nd Edition*, (Dover, New York, 1974).

[4] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, in *Numerical Recipes, 3rd Edition*, (Cambridge University Press, New York, 2007).

[5] D. J. Griffiths, in *Introduction to Electrodynamics, 3rd Edition*, (Benjamin Cummings, New York, 1999).

[6] D. A. Patterson, and J. L. Hennessy, in *Computer Organization and Design, 4th Edition*, (Kauffman, Burlington, 2009).

[7] F. Chapeau-Blondeau and A. Monir, IEEE Transactions on Signal Processing **50**, 9 (2002).