

Interactive Mobile Simulation of Classical Electrodynamics using Finite Difference Methods

Ehsan Dadgar-Kiani
Computer Science, Boston College
Advised by Professor Sergio Alvarez

Abstract

Unlike other branches of physics that can be simplified to lower spatial dimensions in order to improve understanding for an introductory student, classical electrodynamics is innately a three-dimensional science. This introduces the need for an educational utility to simulate the otherwise complicated phenomena that occur in three dimensions. This thesis aims to address this need through an iPad application that allows users to construct custom simulations and interact with them through touch gestures. The application allows the user to simultaneously explore electromagnetic induction, circuit theory, electrostatic fields, and elementary particle systems. Since the application allows for a high level of customization, a versatile modeling technique was needed to numerically approximate the laws of electrodynamics, called Maxwells Equations. The finite difference time domain method is a general method of approximating differential equations by discretizing the space and time domain. Since this applications primary purpose is as an explorative and potentially educational tool rather than a lab utility, it uses techniques to favor temporal as opposed to numerical efficiency. Furthermore, the circuit implementation made use of various elementary graph algorithms such as depth first search in order to solve a circuit's equivalent matrix equation.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 3 |
| 1.1 | Maxwell's Equations | 3 |
| 1.2 | Circuit Theory | 3 |
| 1.2.1 | Circuit Elements | 3 |
| 1.2.2 | Kirchhoff's Laws | 4 |
| 2 | Algorithms | 4 |
| 2.1 | Finite Difference Time Domain | 4 |
| 2.1.1 | Example: Simple RLC circuits | 5 |
| 2.2 | Nodal Analysis | 6 |
| 3 | Implementation | 7 |
| 3.1 | Design | 8 |
| 3.2 | Supporting Touch Gestures | 8 |
| 3.3 | Constructing Graph | 10 |
| 3.4 | Example | 11 |
| 3.5 | Particle System | 11 |
| 4 | Conclusion | 11 |
| 4.1 | Acknowledgements | 12 |
| 4.2 | Future plans | 12 |
| 5 | References | 13 |

1 Introduction

1.1 Maxwell's Equations

All of electrodynamics can be explained using Maxwell's four Equations, which describe the creation of electric and magnetic fields, in addition to the Lorentz Equation, which describes how a charged particle moves subject to these fields [8]. Maxwell's equations can be expressed in both differential (microscopic) and integral (macroscopic) form; the differential form is useful for finding fields at a point in space, whereas the integral form is more useful in calculating other quantities such as charge, current, or electric potential (voltage). For the purposes of this thesis, only the macroscopic forms are of use. They are as follows:

$$\oint_S \mathbf{E}_n \cdot d\mathbf{A} = \frac{1}{\epsilon_0} Q_{inside} \quad (1)$$

$$\oint_S \mathbf{B} \cdot d\mathbf{A} = 0 \quad (2)$$

$$\oint_C \mathbf{E} \cdot d\mathbf{l} = -\frac{d}{dt} \int_S \mathbf{B}_n \cdot d\mathbf{A} \quad (3)$$

$$\oint_C \mathbf{B} \cdot d\mathbf{l} = \mu_0 I_C \quad (4)$$

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (5)$$

As is evident by the fact that the above equations contain vectors and vector operations (dot product, cross product), it is clear that electrodynamics is an innately three-dimensional science.

This thesis places on emphasis on Equations 3 and 5. Equation 3 is called Faraday's Law of Induction, and it essentially states that the electromotive force (or voltage) in a circuit is equal to the negative change in magnetic flux through the inner surface of the circuit with respect to time. It is also the fundamental principle behind how electrical generators, motors, transformers, and inductors operate [12]. Furthermore, Equation 5 basically defines the force that a single charged particle experiences in an electric and magnetic field.

1.2 Circuit Theory

Much of one's experiences with electricity and magnetism come through their application in circuits. Although the concepts of voltage, current, and resistance seem out of the realm of electrodynamics, they are merely the results of many particles (electrons) subject to electric and magnetic fields at a large scale.

1.2.1 Circuit Elements

A voltage, or an electrical potential difference between two points, is what drives charged particles from one point to another. We define electric current as the flow of electric charge with respect to time, $I = \frac{dq}{dt}$. We can also define impedance as the opposition to this current when there is an applied voltage: $Z = R + i(\omega L - \frac{1}{\omega C})$. The impedance of a wire is essentially zero. Capacitance ($C = Q/V$), Resistance (R), and Inductance ($L = \frac{V}{dI/dt}$) are all intrinsic attributes of the different circuit elements called capacitors, resistors, and inductors, respectively [9]. Ohm's Law defines the relationship between current and voltage: $V = IZ$. Although Ohm's Law is merely a linear approximation and does not apply in certain scenarios, it is taken to always be exact for the purposes of this thesis.

When presented with a circuit and given the impedance values of all the circuit elements in addition to the applied voltage signal (emf), the main problem at hand is to find the current and voltage values for each of the elements in the circuit.

1.2.2 Kirchhoff's Laws

For complicated circuits consisting of many loops, Kirchhoff's Laws are necessary to help solve the problem. Kirchhoff's first law, which follows from the principle of conservation of charge, states that the current entering a junction of wires must equal to the current leaving the junction; or in other words, the total current must equal zero [9]:

$$\sum_{k=1}^n \tilde{I}_k = 0 \quad (6)$$

Kirchhoff's second law similarly follows from the conservation of energy and states that the sum of the electric potential differences around any loop in a circuit must equal zero [9]:

$$\sum_{k=1}^n \tilde{V}_k = 0 \quad (7)$$

These laws will be crucial when it comes to implementing Nodal analysis and solving a circuit problem in this thesis.

2 Algorithms

When it comes to simulating electrodynamics on a computer, the main problem that arises is determining which numerical technique to use to approximate the above equations. There are a myriad of techniques that have been developed over time, but many of them are suitable for specific geometries or scenarios. Because the nature of this thesis is such that the end-user designs the problem themselves, I have opted to use the most versatile of methods, the Finite Difference Time Domain method (FDTD).

2.1 Finite Difference Time Domain

Finite Difference Time Domain is a very simple yet elegant technique. It basically works by discretizing space and time, thus turning equations involving derivatives into difference equations. The derivative of a function can be approximated by the definition of the slope between two points. There are several ways of doing so (see Equations 8 through 10).

$$f'(x_0) \simeq \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} \quad (8)$$

$$f'(x_0) \simeq \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \quad (9)$$

$$f'(x_0) \simeq \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \quad (10)$$

In this thesis, the independent variable is typically time, so only Equation 8 is useful because you don't have the function value in the future, so the other two are not applicable.

Figure 6 shows a visual representation of the finite difference time domain method. In most cases, simply decreasing the size of the time-step will increase the accuracy of the approximation until a point is reached at which the finite precision of the floating-point representation in computers leads to significant round-off error [3]. Since numerical accuracy was not a major concern in the scope of this thesis, the time step was simply taken from the framerate of the application.

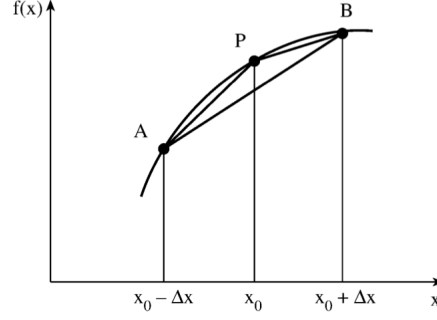


Figure 1: Finite Difference Time Domain Method [3]

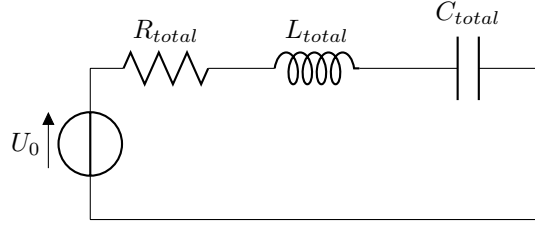


Figure 2: RLC Circuit

2.1.1 Example: Simple RLC circuits

An example that can show the power of the FDTD method is a RLC circuit which simply consists of a resistor, inductor, and capacitor in series with a voltage source (see Figure 2 for a schematic). In this case, the voltage source is an electric generator: a device that converts mechanical to electrical energy by obeying Equation 3, Faraday's Law. The device is essentially a rotating loop of wire between a north and a south magnetic pole.

When used in conjunction with circuits, Faraday's Law is more commonly expressed as the following equivalent equation (with ξ being the electromotive force and Φ_B being the magnetic flux) [12]:

$$\xi = -\frac{d\Phi_B}{dt} \quad (11)$$

In the application, the only way to generate an electromotive force is using a square loop of wire. If we consider the square loop of wire small enough such that the magnetic field is uniform across its inside surface (with area A and normal \hat{n}), the flux is simply given by Equation 12. This simplifies the calculation of the electromotive force at a given time by simply comparing the dot product of the normal to the surface and the magnetic field to that of the previous timestep.

$$\Phi_B = \mathbf{B} \cdot (A\hat{n}) \quad (12)$$

$$\xi_t = -\frac{d\Phi_B}{dt} \simeq -\frac{(\Phi_B)_t - (\Phi_B)_{t-1}}{\Delta t} \quad (13)$$

$$\xi_t = -A \cdot \frac{(\mathbf{B} \cdot \hat{n})_t - (\mathbf{B} \cdot \hat{n})_{t-1}}{\Delta t} \quad (14)$$

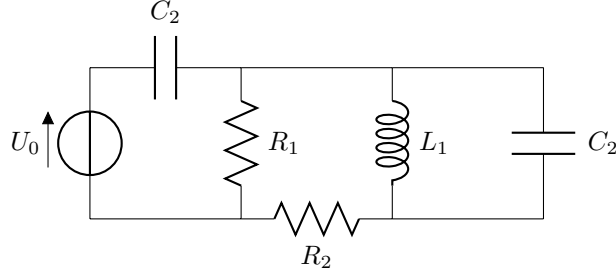


Figure 3: Sample Circuit

Using Kirchhoff's Voltage Law and the definitions of capacitance and inductance, one can arrive at the following second order differential equation that includes the current as a function of time. Since this is a single loop, the current going through all the circuit elements is the same.

$$Ri(t) + L\frac{di}{dt} + \frac{1}{C} \int_0^t i(\tau)d\tau = v(t) \quad (15)$$

When in series, the effective values of the resistance, capacitance, and inductance are given by the following equations:

$$R_{total} = R_1 + R_2 + \dots + R_n \quad (16)$$

$$L_{total} = L_1 + L_2 + \dots + L_n \quad (17)$$

$$\frac{1}{C_{total}} = \frac{1}{C_1} + \frac{1}{C_2} + \dots + \frac{1}{C_n} \quad (18)$$

Even though Equation 15 is a second order differential equation, we only need to use the FDTD method to the first order in this problem because the integral can be replaced by a running count of the total charge being stored in each capacitor, where $Q_{total} = Q_1 + Q_2 + \dots + Q_n$. Using this fact in conjunction with the FDTD method, Equation 19 can be discretized and rewritten as:

$$Ri_t + L\frac{i_t - i_{t-1}}{\Delta t} + \frac{1}{C}Q_{total} = (\xi_{total})_t \quad (19)$$

This can be rearranged such that i_t is on one side, which allows one to calculate the current going through the circuit using its value from the previous timestep. The later section on the implementation of this algorithm will include details on a working demo of this problem. For the most part, this approach using the FDTD method works very well and produces very realistic results, with the only downside being that it is only applicable to simple single-looped circuits. A more general method, called Nodal analysis, allows one to solve more complicated lumped circuits using a matrix method.

2.2 Nodal Analysis

When analyzing a complex circuit consisting of multiple loops, one possible method is to use Kirchhoff's Law (Equation 6) at each junction of wires, which would result in a system of n equations, where n is the number of junctions. Using Ohm's Law, this system could then be rewritten in terms of the electric potential at each point. Solving these equations would complete the problem and give the voltage and current through each circuit element.

For demonstration purposes, we will consider the circuit in Figure 3. If we insert a node at junctions, or between adjacent circuit elements that are connected, and share as many nodes as possible, the circuit can be represented by the graph in Figure 4

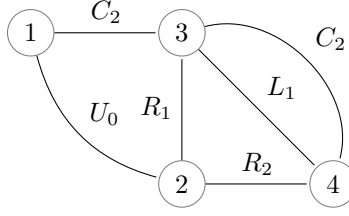


Figure 4: Graph formed from Figure 3

We know that according to Kirchoff's Current Law (Equation 6), that the sum of all the currents entering a junction must equal zero. That means that the sum of the currents through the edges connected to a particular node in the graph must be zero. Thus, we get the following system of linear equations, where I_{kj} is the current flowing from the k th to the j th node.

$$\begin{aligned} I_{13} + I_{12} &= 0 \\ I_{31} + I_{32} + I_{34} &= 0 \\ I_{21} + I_{23} + I_{24} &= 0 \\ I_{42} + I_{43} &= 0 \end{aligned} \tag{20}$$

Using the definition of impedance results in the relationships stated below [2]:

$$I_{kj} = \begin{cases} R^{-1}(U_k - U_j), & \text{Ohmic resistor} \\ i\omega C(U_k - U_j), & \text{Capacitor} \\ (i\omega L)^{-1}(U_k - U_j), & \text{Inductor/Coil} \end{cases} \tag{21}$$

Substituting these in yields the following system of linear equations:

$$\begin{aligned} i\omega C_1(U_1 - U_3) + U_0 &= 0 \\ i\omega C_1(U_3 - U_1) + R_1^{-1}(U_3 - U_2) + (i\omega L)^{-1}(U_3 - U_4) + i\omega C(U_3 - U_4) &= 0 \\ -U_0 + R_1^{-1}(U_2 - U_3) + R_2^{-1}(U_2 - U_4) &= 0 \\ R_2^{-1}(U_4 - U_2) + (i\omega L)^{-1}(U_4 - U_3) + i\omega C(U_4 - U_3) &= 0 \end{aligned} \tag{22}$$

This system of linear equations can be represented by the following matrix:

$$\begin{pmatrix} i\omega C_1 & 0 & -i\omega C_1 & 0 \\ -i\omega C_1 & -R_1^{-1} & (i\omega C_1 + R_1^{-1} + (i\omega L)^{-1} + i\omega C) & -((i\omega L)^{-1} + i\omega C) \\ 0 & (R_1^{-1} + R_2^{-1}) & -R_1^{-1} & -R_2^{-1} \\ 0 & -R_2^{-1} & -((i\omega L)^{-1} + i\omega C) & (R_2^{-1} + (i\omega L)^{-1} + i\omega C) \end{pmatrix} \times \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} -U_0 \\ 0 \\ U_0 \\ 0 \end{pmatrix} \tag{23}$$

This resultant matrix equation can then be solved using a numerical library (such as LAPACK [13]) or by a combination of Gaussian Elimination and Back Substitution [11], which is what was done in this thesis' implementation. Perhaps the biggest problem with this matrix method is that for larger problems, where the number of nodes that needs to be introduced is quite large, the matrix increases in dimension and becomes impractical to solve.

3 Implementation

The aforementioned algorithms were implemented using the Objective-C language [5] for use on an iPad device (manufactured by Apple, Inc.). Some of the basic functionality of the application includes:

(a) Popup for adding a new object

(b) Manipulating objects in 3D

(c) Edit screen for an Lightbulb

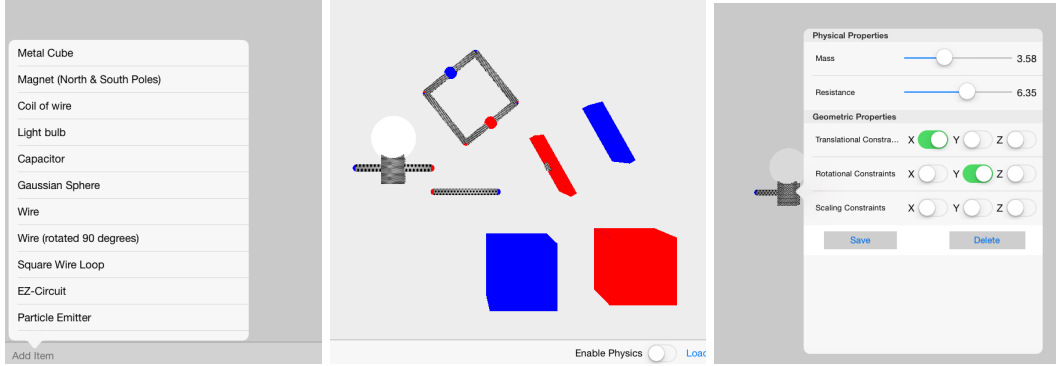


Figure 5: Various screenshots from iPad application

- Move objects around in a three dimensional environment and apply geometric transformations (translate, rotate, scale) using touch gestures
- Modify the various properties of objects (such as magnetization, mass, charge, resistance, capacitance, etc.)
- Enable 'physics mode,' where physical laws are enforced using finite difference and matrix methods

3.1 Design

Figure 6 shows a UML diagram of the class hierarchy used in the iPad implementation. The base class *Object* represents a physical object with vectors for position, velocity, rotation, and scale. *Magnets* are represented by a pair of blocks (north and south pole) that can be moved around such that in the parallelogram between their opposite faces there is a constant magnetic field (directly proportional to the its magnetization) pointing from the north to the south block. Instances of type *CircuitElement* are objects that additionally have an impedance and an array of vectors representing where the circuit element's electrodes are. Lightbulbs are unique in that they indicate how much current is passing through them by how much they light up (ranging from white to yellow).

3.2 Supporting Touch Gestures

Figure 7 shows the various touch gestures that are supported by the application. 1-Finger pan is used to change an object's position in the current viewing plane. It's implementation was fairly straightforward using the iPad's built-in libraries [7]; the 2D touch point is merely unprojected from window-coordinates to the object-coordinates and assigned to the object's position. Similarly, the 2-Finger Rotate and 2-Finger Pinch gestures were trivial using the iPad's build in libraries [7]. The 2-Finger rotate, on the other hand, was slightly more complicated. Each frame, the 2D delta values of the gesture were used to construct the euler angles around the x,y, and z axes. But before applying these angles to the object's rotation matrix, they are converted into the object's local coordinate system using the inverse of its rotation matrix. The pseudocode for this can be seen in Algorithm 1.

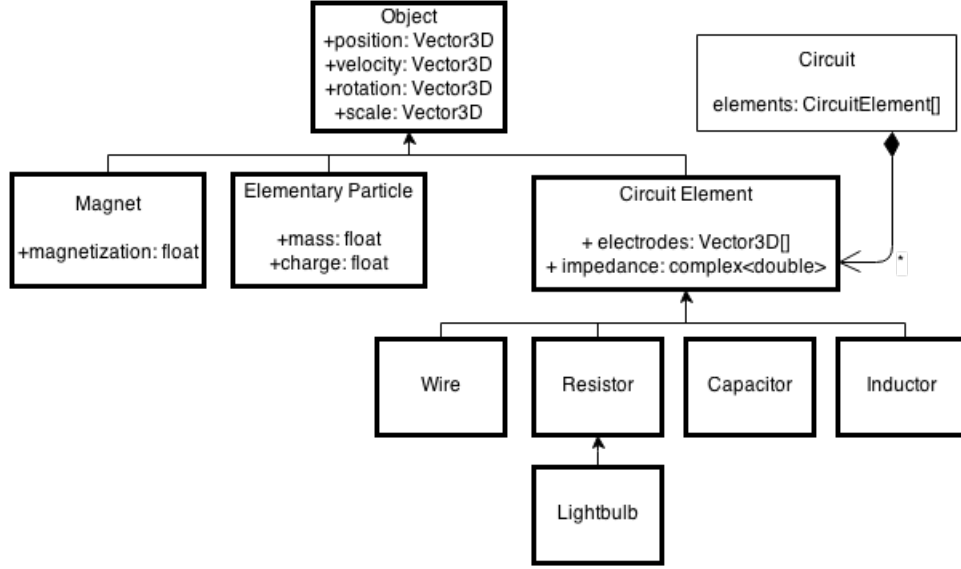


Figure 6: UML Diagram of iPad Implementation

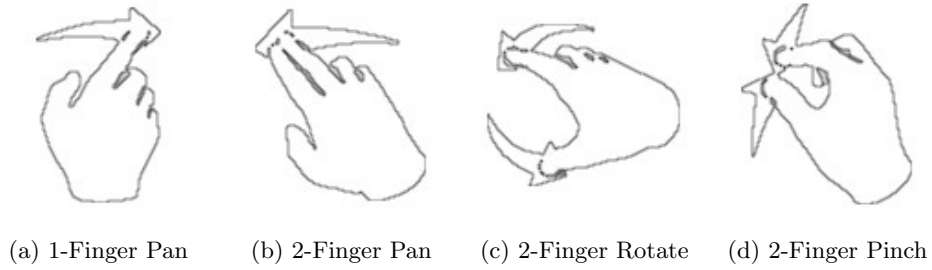


Figure 7: Object-manipulation gestures supported by iPad application

Algorithm 1 Procedure for handling 2-Finger rotate gesture

```

procedure ONGESTURECHANGED(object, delta)
    rotAngles  $\leftarrow m * \{\text{delta.y}, \text{delta.x}, 0\}$ 
    rotMatrix  $\leftarrow \text{object.rotationMatrix}()$ 
    xAxis  $\leftarrow \text{rotMatrix}^{-1} \times \{1, 0, 0\}$ 
    yAxis  $\leftarrow \text{rotMatrix}^{-1} \times \{0, 1, 0\}$ 
    zAxis  $\leftarrow \text{rotMatrix}^{-1} \times \{0, 0, 1\}$ 
    rotMatrix  $\leftarrow \text{MatrixRotate}(\text{rotMatrix}, \text{rotAngles.x}, \text{xAxis})$ 
    rotMatrix  $\leftarrow \text{MatrixRotate}(\text{rotMatrix}, \text{rotAngles.y}, \text{yAxis})$ 
    rotMatrix  $\leftarrow \text{MatrixRotate}(\text{rotMatrix}, \text{rotAngles.z}, \text{zAxis})$ 
    object.rotationMatrix  $\leftarrow \text{rotMatrix}$ 

```

3.3 Constructing Graph

When it came to the circuit implementation, one of the main challenges was how to turn a user's custom circuit, which is stored as simply an array of circuit elements, into a corresponding graph that could be used in nodal analysis.

Algorithm 2 Algorithm for converting list of circuit element objects into a graph

```

function SHOULD SNAPPOINTS(P1, P2)
    snapDistance  $\leftarrow$  0.25 ▷ min distance between points in contact
    return ((P2 - P1).length() < snapDistance)

function CREATEGRAPHFROMCIRCUITELEMENTS(elements)
    elementToNodes  $\leftarrow$  {} ▷ nodes connected to a circuit element
    nodeToElements  $\leftarrow$  {} ▷ circuit elements connected to a node

    for (e1, e2) in elements.allPairs() do ▷ create nodes between circuit elements
        for (index1  $\rightarrow$  electrode1) in e1.electrodes() do
            for (index2  $\rightarrow$  electrode2) in e2.electrodes() do
                if ShouldSnapPoints(electrode1, electrode2) then
                    node  $\leftarrow$  elementToNodes[e1][index1]
                    if node is null then ▷ check if a node already exists here
                        node  $\leftarrow$  elementToNodes[e1][index2]
                    if node is null then
                        node  $\leftarrow$  Node()
                    elementsToNodes[e1][index1]  $\leftarrow$  node
                    elementsToNodes[e2][index2]  $\leftarrow$  node
                    nodesToElements[node].addAll([e1, e2])

    graph  $\leftarrow$  Graph()

    for (currentNode  $\rightarrow$  currentNodeElements) in nodeToElements do ▷ create the edges
        for element in currentNodeElements do
            for neighborNode in elementToNodes[element] do
                if neighborNode  $\neq$  currentNode then
                    edge  $\leftarrow$  Edge(currentNode, neighborNode, element)
                    graph.addEdge(edge)

    return graph

```

Essentially what this algorithm does is go through all pairs of circuit elements and introduce nodes where their electrodes come into contact within a certain threshold distance. The edges of the graphs then represent circuit elements.

The complexity of this algorithm is determined by the number of circuit elements (N) and the number of electrodes (e) for each element (which is currently only 2, so it is basically negligible). The big-O of this algorithm becomes:

$$O(n) = e^2 * N^2 \simeq N^2 \quad (24)$$

In this application, wires are considered circuit elements, unlike the nodal analysis algorithm that completely ignored them since they have no effect on the current or voltage. Their inclusion would result in an excess number of unnecessary variables that merely increase the size of the matrix but

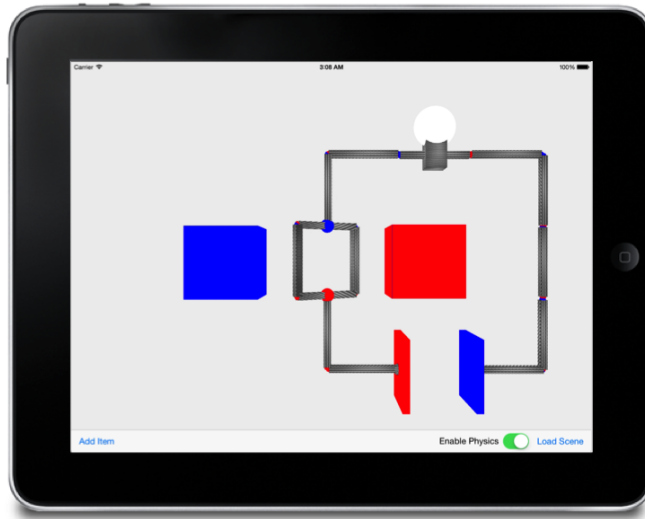


Figure 8: RC Circuit in Application

do not change the final answer. Therefore, when using the graph and KCL to construct the matrix, edges for wires are ignored in order to minimize the number of variables.

3.4 Example

Figure 8 is a still image from a video that was used for demonstration purposes, where the user creates the circuit piece by piece using a magnet, wires, light-bulb, and capacitor. When physics mode is enabled, and the user rotates the square loop in the xz -plane and notices that this results in an alternating current. Much like how a generator works, the mechanical energy of the rotating loop is converted to electrical energy in the circuit. The frequency of the alternating current should be the same as the angular frequency of the square loop rotating about the y -axis, which is the case.

3.5 Particle System

In addition to creating circuits, users of the application can create particle systems that emit particles of a customizable charge, velocity, mass, and lifetime. These particles are really only subject to the Lorentz Force (Equation 5), so they can be used to test for the presence of electric or magnetic fields. For example, if a positively charged particle is in the presence of a uniform magnetic field (as in Figure 9), it behaves as a cyclotron and follows a spiral path whose radius depends on the particle charge, mass, and external magnetic field.

4 Conclusion

This was a great experience as an interdisciplinary thesis project, for it brought together physics, numerical methods, graph theory, mobile development, and 3D rendering altogether in a practical application.

Although much was accomplished in the two semesters of both exploring scientific computation and developing this application, there were several shortcomings. For instance, when dealing with large circuits, the aforementioned Nodal Analysis algorithm only applies if the net emf voltage has a single harmonic time-dependence; if the net emf is a superposition of multiple harmonics then the algorithm will not apply. This is the case even if there are just two square loops rotating at different

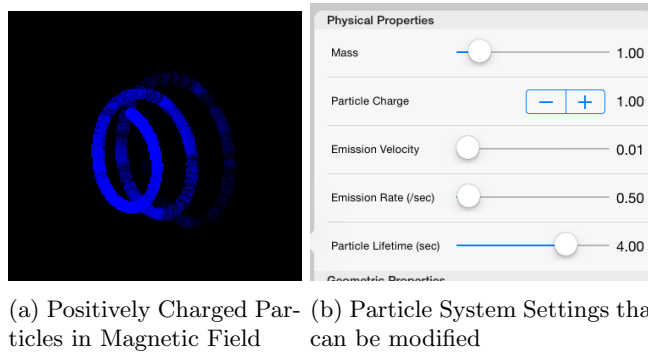


Figure 9: Particle Systems in Application

angular velocities because their induced currents will have different frequencies. Furthermore, the magnetic flux can most likely be calculated in a more precise way by breaking up the inner surface of the loop into fractional pieces and summing the flux over them, instead of assuming that the magnetic field is uniform across the entire surface of the loop; due to time constraints this was not implemented. Finally, the graph representing a circuit is only constructed (using Algorithm 2) when the physics mode is turn on; thus, if any circuit elements are moved such that their electrodes are no longer touching, the graph data structure will not be updated in real time to reflect this change. This means that in physics mode, a circuit can be broken by moving an intermediate wire, but the circuit will still behave normally. This is obviously a shortcoming that could be fixed by intelligently running the graph algorithm only when a circuit element has moved, and this functionality is currently in progress.

4.1 Acknowledgements

Many thanks to Professor Alvarez for his help and advice throughout the past year with regards to this thesis, and to the entire Computer Science faculty at Boston College who have made the past four years an enlightening experience.

4.2 Future plans

There is still further work to be done on this physics engine and simulation tool to get it to a fully stable and usable state. Although it currently allows for the creation of circuits and demonstrates several of the laws of both electrodynamics and circuit theory, it could definitely be improved to explore more areas of interest that were missed due to time constraints. Some future plans include:

- Use a numerical method or technique for more complicated circuits that will still apply for induced voltages that are not single harmonics.
- Possibility of using iterative methods to converge to a solution for more complicated circuit problems as opposed to using the matrix solution to solve for an exact solution each time
- Currently, the only source of emf is the preset coil of wire; it would be much more realistic and powerful to consider every loop of wire constructed from individual circuit elements as a possible source of electromotive force if the magnetic field through its surface is changing. Furthermore, it would be interesting to consider the self-inductance and mutual-inductance of loops or coils of wire, for this would allow for a simulation involving transformers.

The future plan is to continue development and post to an online repository such as *github.com* for public use.

5 References

References

- [1] John B. Schneider, *Understanding the Finite Difference Time Domain Method* Washington State University, October 22, 2013.
- [2] *Numerische Methoden fur CSE*, ETHZ, May 17, 2013, 17 May 2013, http://www.math.ethz.ch/education/bachelor/lectures/hs2012/math/nummath_cse/Scripts_and_more.
- [3] Matthew N. O. Sadiku, *Numerical Techniques in Electromagnetics*, CRC Press, 2nd Edition, 2001.
- [4] *Nodal Analysis*, Syncleus, December 1, 2010, http://wiki.syncleus.com/index.php/Nodal_Analysis.
- [5] *Programming with Objective-C*, Apple, December 13, 2012, <https://developer.apple.com/library/mac/documentation/cocoa/conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>.
- [6] *OpenGL ES Programming Guide for iOS*, Apple, March 10, 2013, https://developer.apple.com/library/ios/documentation/3ddrawing/conceptual/opengles_programmingguide/introduction/introduction.html.
- [7] *Event Handling Guide for iOS*, Apple, January 28, 2013, https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/GestureRecognizer_basics/GestureRecognizer_basics.html.
- [8] Sophocles J. Orfanidis, *Maxwell's Equations*, Rutgers University, Electromagnetic Waves and Antennas, August 25, 2013, <http://www.ece.rutgers.edu/~orfanidi/ewa/ch01.pdf>.
- [9] David Cory, *Introduction to Electronics, Signals, and Measurement*, Massachusetts Institute of Technology, Spring 2006, <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-071j-introduction-to-electronics-signals-and-measurement-spring-2006/>
- [10] Weisstein, Eric W., *Gaussian Elimination*, MathWorld—A Wolfram Web Resource, <http://mathworld.wolfram.com/GaussianElimination.html>.
- [11] Jim Lambers, *Gaussian Elimination and Back Substitution*, The University of Southern Mississippi, Summer 2010, <http://www.math.usm.edu/lambers/mat610/sum10/lecture4.pdf>
- [12] Rod Nave, *Faraday's Law*, Department of Physics and Astronomy, Georgia State University, <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/farlaw.html>.
- [13] LAPACK - Linear Algebra PACKage, November 16, 2013, <http://www.netlib.org/lapack/lapacke.html>