Computational Models of Film and Bulk Superconductors in Time Varying Magnetic Fields

Alden R. Pack

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Mark K. Transtrum, Advisor

Department of Physics and Astronomy

Brigham Young University

August 2015

ABSTRACT

Computational Models of Film and Bulk Superconductors in
Time Varying Magnetic Fields

Alden R. Pack
Department of Physics and Astronomy, BYU
Bachelor of Science

We numerically study the time-dependent Ginzburg-Landau equations of superconductivity using a Galerkin method implemented in FEniCS, an automated differential equation solver. We consider geometries for both a bulk material (line from zero to infinity) and a film (half-line), corresponding to mixed and Neumann boundary conditions respectively. We simulate quenching by switching on an external magnetic field, allowing the material to approach a steady state, and then switching on a greater field. Our solutions exhibit the Meissner effect, convergence to the steady state solution, and quenching of superconductors.

Keywords: Ginzburg-Landau equations, superconductivity, time dependence, numerical approximations, finite element method, FEniCS

ACKNOWLEDGMENTS

I express appreciation to Dr. Mark Transtrum and all of his efforts to help me understand this material and work me through the research. I also thank the BYU Department of Physics and Astronomy for funding me over the years.

Contents

List of Figures

Chapter 1

1.1 Motivation for this Work

Physicists today continue to explore the infinitesimal in order to understand the building blocks of the universe. A key tool to measuring subatomic phenomena are particle accelerators. Particle accelerators function with the help of superconducting resonant cavities. Understanding the properties of superconductors—negligible internal resistance and expulsion of internal magnetic fields (Meissner effect)—allows engineers and scientists to improve the performance of superconducting resonance cavities.

Resonant cavities use large AC currents to create standing radio frequency waves (see Fig. 1.1). At each anti-node of the standing wave, the particle beam receives a boost of kinetic energy. [1] Internal resistance causes the cavities to heat up, requiring cooling. With lower internal resistance, superconductors dissipate much less power then materials like copper. [2] For that reason, most resonant cavities are made of superconducting materials.

So much progress has been made in this field that these resonant cavities are approaching the theoretical limits of superconductor performance. One problem, quenching, occurs when the external magnetic field forces itself into the superconductor. [2] The metal then enters a nonsuperconducting state, followed by rapid overheating and power loss. By applying numerical methods, we can simulate electromagnetic fields in these cavities and improve their design avoid quenching

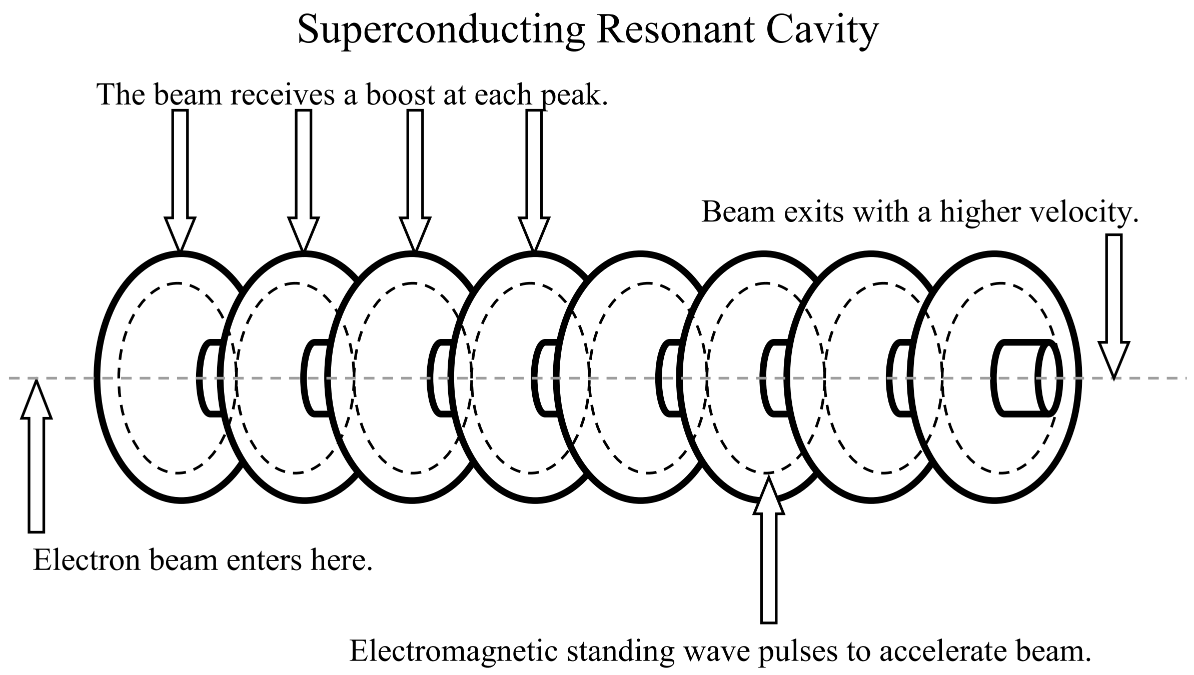
Superconducting Resonant Cavity
The beam receives a boost at each peak.
Beam exits with a higher velocity.
Electron beam enters here.
Electromagnetic standing wave pulses to accelerate beam.

Figure 1.1 Superconducting resonant cavities are a modern application of superconductors. They use electromagnetic standing waves to accelerate particles to high speeds.

in larger magnetic fields.

The objective of this thesis is not to give a full three dimensional model of superconducting resonant cavities, but provide foundational methods and strategies for modeling superconductors in simplified geometries. We will analyze the Ginzburg-Landau equations (GLE) of superconductivity [3] using a Galerkin method implemented in FEniCS, an automated differential equation solver [4]. The geometries considered are bulk and film superconductors. This thesis serves for two purposes: it summarizes my efforts to model the GLE in one dimension so that I can refer to it in future research, and it guides future students who desire to learn about numerically modeling superconductors.

1.2 Organization of the Thesis

Chapter 1 introduces the reader to the GLE, the Galerkin Method, and FEniCS. If the reader desires to learn about the history and theory of superconductors, I refer them to Michael Tinkham's book, "Introduction to Superconductivity". [3]

Chapter 2 has three objectives: to explain in detail the boundary conditions and symmetry of film and bulk superconductors, to explain how to apply those conditions to the Ginzburg Landau equations, and to show how the Galerkin method is implemented with the help of FEniCS.

Chapter 3 illustrates how our solutions relate to physical phenomena. We show how the time-dependent solution approaches the steady state, illustrate the dynamics of quenching, and explain how our results can extend to more complex geometries.

1.3 The Ginzburg-Landau Equations

The Ginzburg-Landau equations (GLE) minimize the free energy in a superconductor [3]. In other words, due to the second law of thermodynamics, a superconductor will approach the conditions given by the GLE. Following the work of S. Jonathan Chapman [5] and assuming the absence of an external electric field, we obtain the GLE which we will analyze:[1]

$$-\alpha\frac{\partial f}{\partial t}+\xi^2\nabla^2 f = f^3 - f + f|\mathbf{q}|^2 \text{ in } \Omega, \tag{1.1}$$

$$-\lambda^2(\nabla\times)^2\mathbf{q} = \alpha\frac{\partial \mathbf{q}}{\partial t} + f^2\mathbf{q} \text{ in } \Omega. \tag{1.2}$$

Eqs. (1.1) and (1.2) are coupled, nonlinear, partial differential equations that depend on two functions f and $\mathbf{q}$. The scalar function f defines a dimensionless order parameter that gives the state of the material and is calculated by dividing the free energy at each point by the free energy

[1]Chapman has these equations in terms of κ which is the ratio $\frac{\xi}{\lambda}$. Since we are more interested in the units from ξ and λ we have inserted them with the appropriate adjustments.

of a perfectly superconducting material with no magnetic field. If f equals zero, the material acts as a normal metal. If f is nonzero, the material exhibits superconductivity. The vector function $\mathbf{q}$ defines the dimensionless magnetic potential vector. The curl of $\mathbf{q}$ is the magnetic field in the superconductor.

Most superconducting materials are defined by the ratio κ of the coherence length and the penetration depth. This is written as $\kappa = \frac{\xi}{\lambda}$. The penetration depth λ is how far the external magnetic field penetrates the surface of the superconductor. The coherence length ξ is the distance from the surface of the superconductor that remains superconducting, assuming there is an external magnetic field.

We need to keep track of units. The term α gives the time scale. The magnetic field has units of $\sqrt{2}H_c$ where H_c is the magnitude of the magnetic field at which the superconductor quenches (also known as the critical magnetic field). The order parameter and magnetic potential vectors are dimensionless, but we will measure distance in λ, the penetration depth. For simplicity we set the values of κ and α to one.

1.4 Galerkin Method

This section explains how to implement a Galerkin method.

Partial differential equations like the GLE have infinitely many solutions. By defining boundary conditions (done in Chapter two) we restrict ourselves to one exact solution. Unfortunately there are no closed form solutions to the GLE; however, numerical methods can find a close approximation. We will use a Galerkin method to find our numerical solution. First we define a finite set of continuous, and piecewise differentiable functions, known as basis functions. By taking a linear combination of these basis functions we can create a wide variety of functions. Projecting the exact solution onto these basis functions gives us the closest approximation.

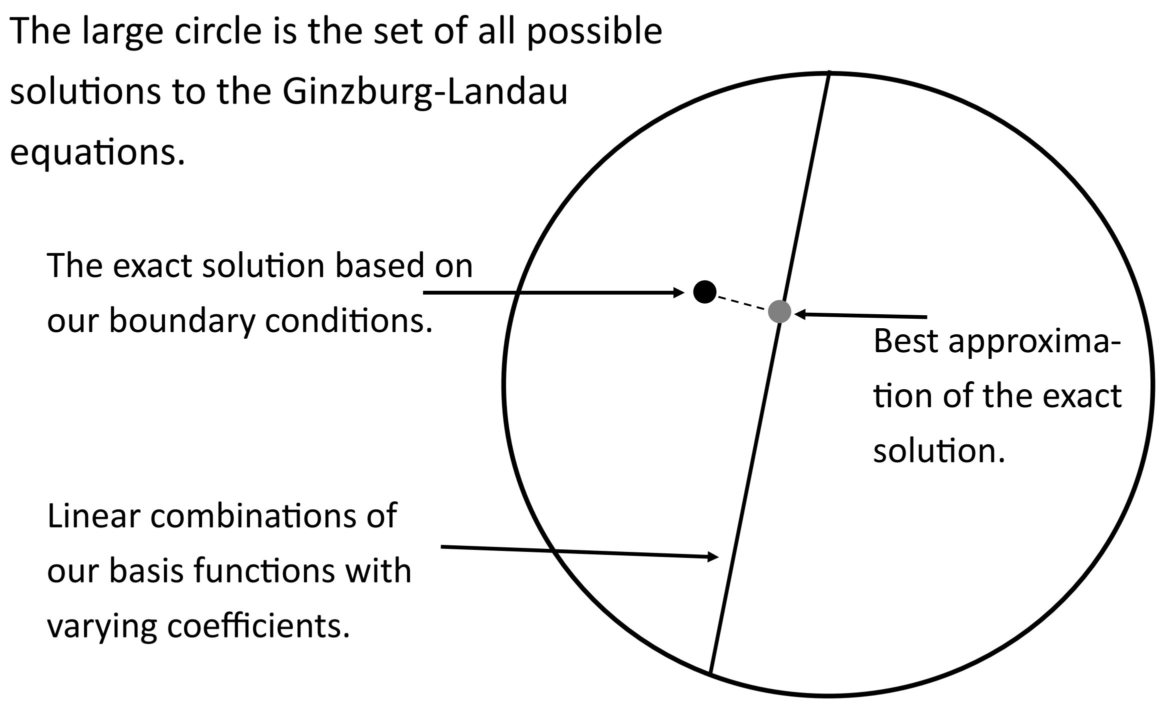
The large circle is the set of all possible solutions to the Ginzburg-Landau equations.
The exact solution based on our boundary conditions.
Best approximation of the exact solution.
Linear combinations of our basis functions with varying coefficients.

Figure 1.2 This cartoon illustrates the Galerkin Method. Each point within the circle represents a possible solution to the Ginzburg-Landau equations. Defining boundary conditions restricts the solutions to one exact solution (the black dot.) By adding basis functions together, represented by the line, we can approximate the exact solution (the gray dot.)

As an illustration I refer to Fig 2.1. The large circle represents the set of all functions that satisfy the GLE. The black dot represents the exact solution that matches our boundary conditons. The black line shows all the possible linear combinations of our basis functions. We move along this line by changing the coefficients of each basis function and then taking their sum. The gray point represents the closest our basis functions can get to the exact solution.

Since we want to consider a general case where the boundary conditions vary with time, we use the time-dependent GLE. To do so we define time steps, adjusting the coefficients of our basis functions with time.

Let's apply the Galerkin method to the GLE. We denote the basis functions by ϕ_i where i is an indexing term. In Chapter 2 we use symmetry to limit $\mathbf{q}$ to its y component q_y or simply q. With this notation, we denote the approximation for the order parameter f by the sum $\sum a_i(t)\phi_i$ and the approximation for q by $\sum b_i(t)\phi_i$ where $a_i(t)$ and $b_i(t)$ are their coefficients respectively. These sums are known as the trial solutions. For ease of reading, we assume they are time dependent without explicitly writing (t).

Now we project the infinite solution onto our basis. To do so we take the inner product of each term in the GLE with another basis function ϕ_j. This new ϕ_j is known as a test function, and it uses the same basis as the trial functions.

Assuming $\alpha = \xi = \lambda = 1$, we apply the Galerkin method to Eqs. (1.1) and (1.2) and obtain:

$$\begin{aligned}
-\sum_{ij}\frac{\partial a_i}{\partial t}\int \phi_i\phi_j d\Omega + \sum_{ij} a_i \int (\nabla^2\phi_i)\phi_j d\Omega = \sum_{i,j,k,l} a_i a_j a_k \int \phi_i\phi_j\phi_k\phi_l d\Omega \\
-\sum_{jk} a_i \int \phi_i\phi_j d\Omega + \sum_{i,j,k,l} a_i b_j b_k \int \phi_i\phi_j\phi_k\phi_l d\Omega, \qquad (1.3) \\
-\sum_{i,j} b_i \int (\nabla\times)^2 \phi_i\phi_j d\Omega = \sum_{i,j}\frac{\partial b_i}{\partial t}\int \phi_i\phi_j d\Omega + \sum_{i,j,k,l} a_i a_j b_k \int \phi_i\phi_j\phi_k\phi_l d\Omega. \qquad (1.4)
\end{aligned}$$

In the future we omit the summation symbols for ease of reading and use the convention that repeated indices are summed.

Subtracting the terms on the right hand side to the left hand side calculates the accuracy of the approximation, known as the residual. By solving for the coefficients a and b that minimize the residual we find an effective numerical approximation to the GLE. We can then plot the basis functions with their corresponding coefficients and see how the state evolves with time.

1.5 FEniCS

To understand the intricacies of the Galerkin Method we first coded from scratch using python. We then switched to FEniCS, a software package that has a built in differential equation solver. This was done in anticipation of future work involving more complex geometries. A. Loggs, co-founder of FEniCS, explained the meaning of the acronym, “FE is Finite Element(s). CS is Computational Science, Computer Science or Computational Software. ni sits nicely in the middle.” [6] This software automates the Galerkin method. For the exact process I refer to the book “Automated Solution of Differential Equations by the Finite Element Method” by A.Logg et al. [4]

Chapter 2

Chapter 2

2.1 Boundary Conditions and Symmetry

Up to this point we have not deviated from the general form of the GLE. We now define the geometry of the system and the boundary conditions so that we can obtain a well-defined solution. Distance is measured in λ and time in α.

We consider two geometries of superconductors: a bulk superconductor and a film superconductor. In the case of the bulk superconductor, we assume that for x less than zero, a vacuum extends to negative infinity, and for x greater than zero, a bulk superconductor extends to positive infinity. The yz-plane does not vary, so all of the variations are seen as x changes. The film superconductor is similar, but the metal ends at some finite distance L, and a vacuum extends for x larger than L.

A bulk superconductor has Neumann and Dirichlet boundaries for both the magnetic potential vector and the order parameter. Let us consider the boundary conditions for the magnetic potential vector. We assume there is an external magnetic field with units $\sqrt{2}H_c$. Let $\mathbf{q}$ have a $\hat{y}$ component dependent on x so that $\mathbf{q} = (0, q_y, 0)$. The curl of $\mathbf{q}$ is $\nabla \times \mathbf{q} = q_y'\hat{z}$, which leads to the boundary condition $q'_y(0) = H_a$. For convenience we denote q_y by q. This is a Neumann boundary at $x = 0$. We know that the Meissner effect forces the magnetic field to zero as x approaches infinity. This can be satisfied by setting $q(\infty) = 0$, a Dirichlet boundary condition.

Now we look at the order parameter. Because of the Meissner effect, the farther we get from the external magnetic field the more superconducting the metal becomes. This means that the order parameter approaches 1 as x increases. This is written as $f(\infty) = 1$. We also make the assumption that at $x = 0$, no surface currents leak into the vacuum, meaning $f'(0) = 0$ (see Tinkham's book for more detail on the relation between the order parameter and current) [3].

For a film superconductor, the boundary conditions for $x = 0$ remain the same, but the right boundary changes. We can no longer claim that for $f(\infty)$, the metal is superconducting. Instead, we assume conditions similar to the left hand side. We set $f'(L) = 0$ and $q'(L) = H_b\hat{z}$, where L is the length of the superconductor, and $H_b\hat{z}$ is the external magnetic field strength on the right boundary. These are purely Neumann boundary conditions.

2.2 Implementation of FEniCS

Up to this point we have focused on the theory. Running the simulation on a computer can be done in many ways, but we chose to use FEniCS (see section 1.5 for background information). We ran FEniCS in python; however, FEniCS is also compatible with C++. The complete code that we used in producing some of our plots is given in appendix A. Instead of going into the details of the inner workings of our code, we present a summary of the most important steps.

Our first step verified FEniCS gave accurate results. We used FEniCS to solve a simple heat equation (see appendix B) and we were pleased to see that it matched the exact analytic solution. Adding nonlinearities and comparing it with a leapfrog approach we found FEniCS handles nonlinearities nicely as well (see appendix C).

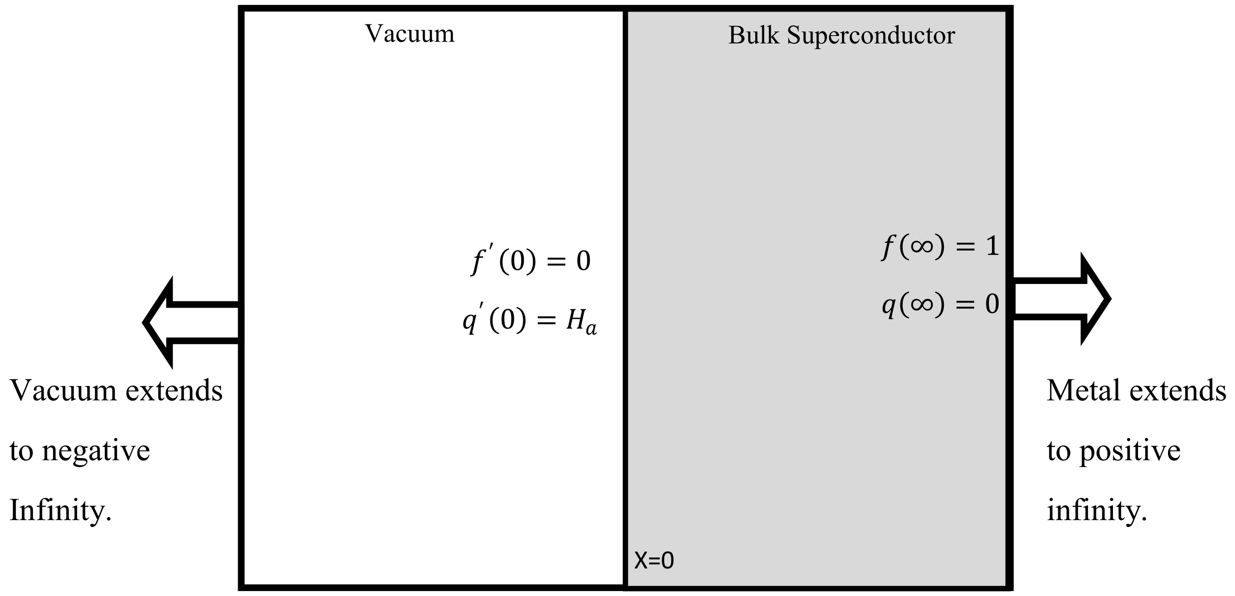

Figure 2.1 A bulk superconductor can be represented as filling half of space. We let the boundary between the superconductor and vacuum be at $x = 0$. The superconductor extends to positive infinity. The boundary conditions for the dimensionless order parameter f, and the dimensionless magnetic potential q are $f'(0) = 0$, $f(\infty) = 1$, $q'(0) = H_a$ and $q(\infty) = 0$.

Having verified FEniCS, we extend those preliminary results to the GLE. To do so requires defining an initial condition. To simulate a real life situation we assume the superconductor starts in a superconducting state (order parameter is one and magnetic potential is zero for all x) and then we turn on an external magnetic field.

We begin by ignoring time dependence and finding a steady state solution. We can find a steady state solution by setting the time dependence to zero, or we can approximate that solution by letting the program run for a long period of time. By using the first method on a bulk superconductor we

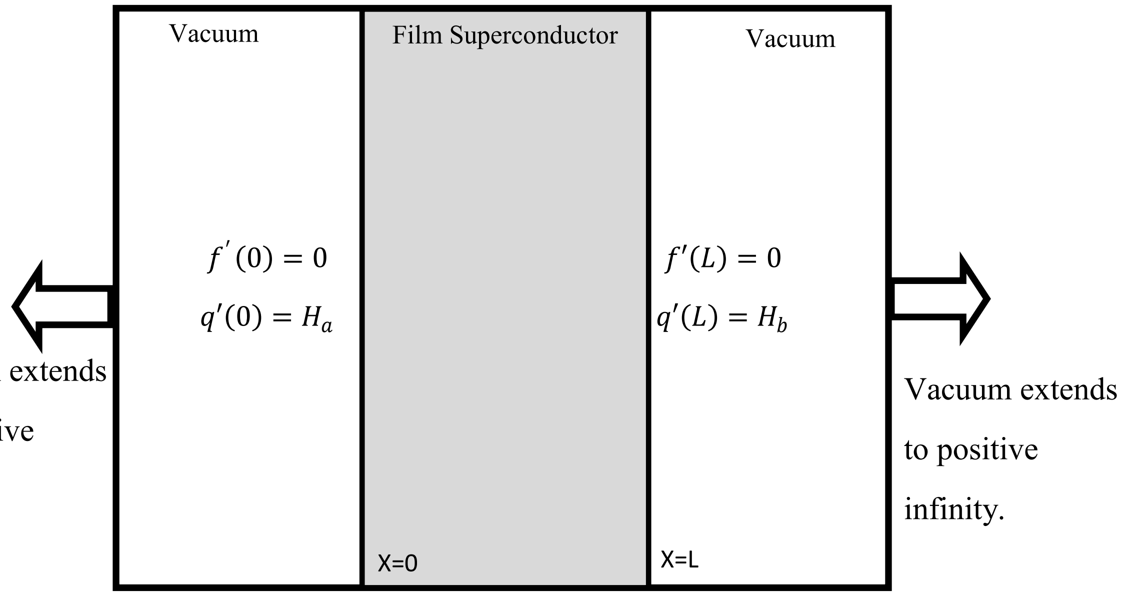

Figure 2.2 A film superconductor can be represented as existing between $x = 0$ and $x = L$. The boundary conditions for the order parameter f and the magnetic potential q are $f'(0) = 0$, $f(\infty) = 1$, $q'(0) = H_a$ and $q(\infty) = 0$.

obtain the plot in Fig 3.1. The second method can be seen in Figs. 3.2-3.5.

After solving the steady state we explore time dynamics. Time dependence is accounted for using a forward difference method. The forward difference method assumes that we know the state of the system in the past and take a small time step forward based on the partial derivative with time. The partial derivative $\frac{\partial f}{\partial t}$ can be approximated by $\frac{f-f_0}{dt}$ where f defines the current f solution, f_0 defines the previous f solution, and dt is the time step. We start at $t = 0$ and use a while loop to step through time, saving the different f and q solutions.

Chapter 3

Chapter 3: Results and Conclusions

This chapter verifies that our numerical approximations are effective and accurate. We discuss how the system evolves from an initial condition to the appropriate steady state. We indicate dynamics such as the Meissner effect, approach to the steady state, and quenching. Finally, we end by reviewing how these methods can extend to more complex geometries.

3.1 Final Solutions

In Fig. 3.1 we plot the steady state solution of a bulk superconductor. We first verify that this solution satisfies the boundary conditions. Let us consider the order parameter. Observe that at $x = 0$ the slope of the order parameter flattens, indicating the slope approaches zero and that there is no current seeping into vacuum. For the left hand boundary we assume that at a distance $L = 10$ the material is sufficiently shielded from the external magnetic field at $x = 0$. Seeing how the order parameter comes close to one, we have a good approximation.

We can also see the Meissner effect. At the surface the external magnetic field is at .5 and the magnetic potential is $-.5$. The magnetic potential and magnetic field decrease the deeper we penetrate the superconductor until they become negligible.

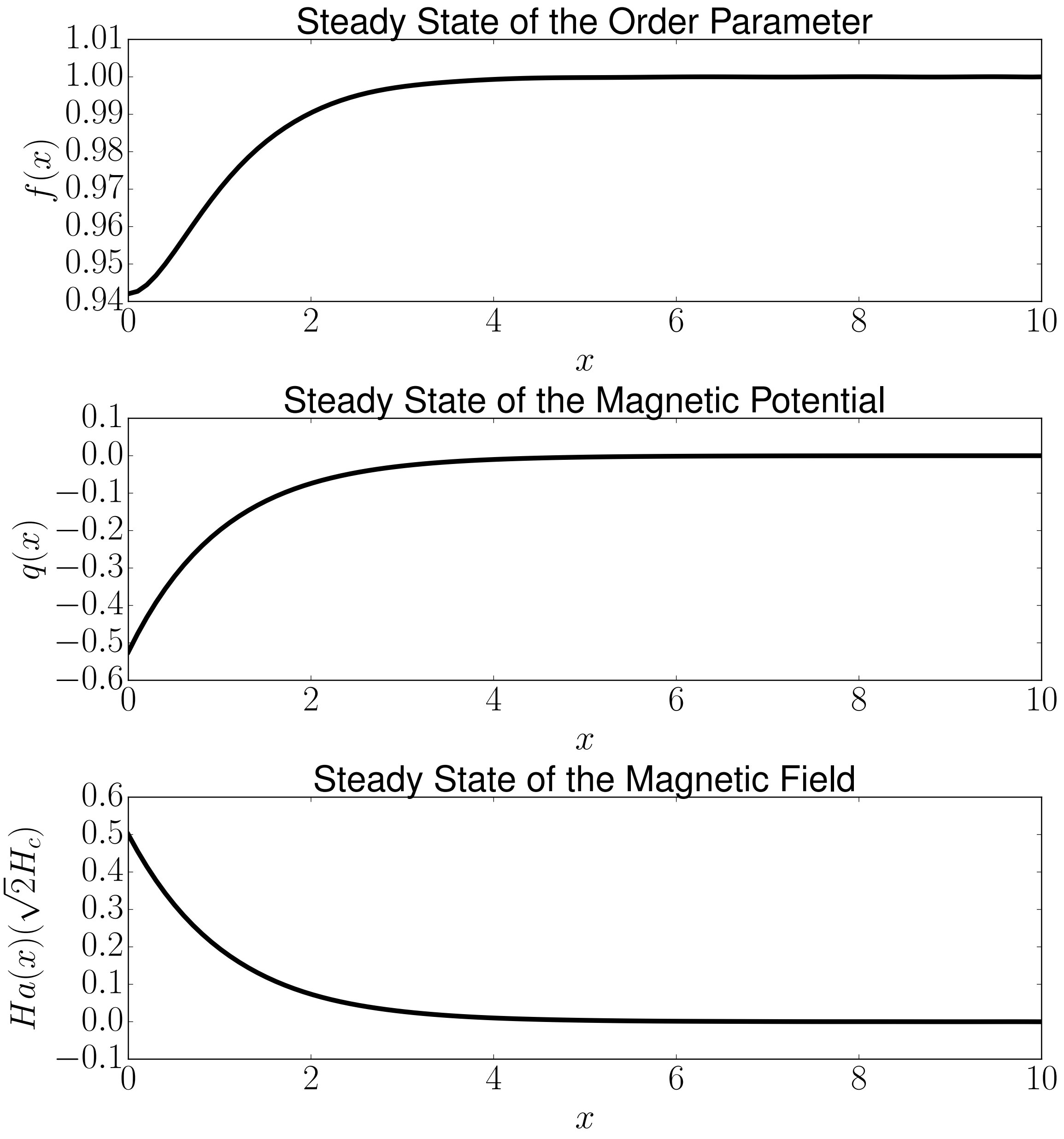

Figure 3.1 The plot of the steady state solution of the order parameter, magnetic potential vector, and magnetic field gives a better understanding of how superconductors work. These plots demonstrate the Meissner effect or gradual decrease of the magnetic potential and magnetic field deep in the superconductor.

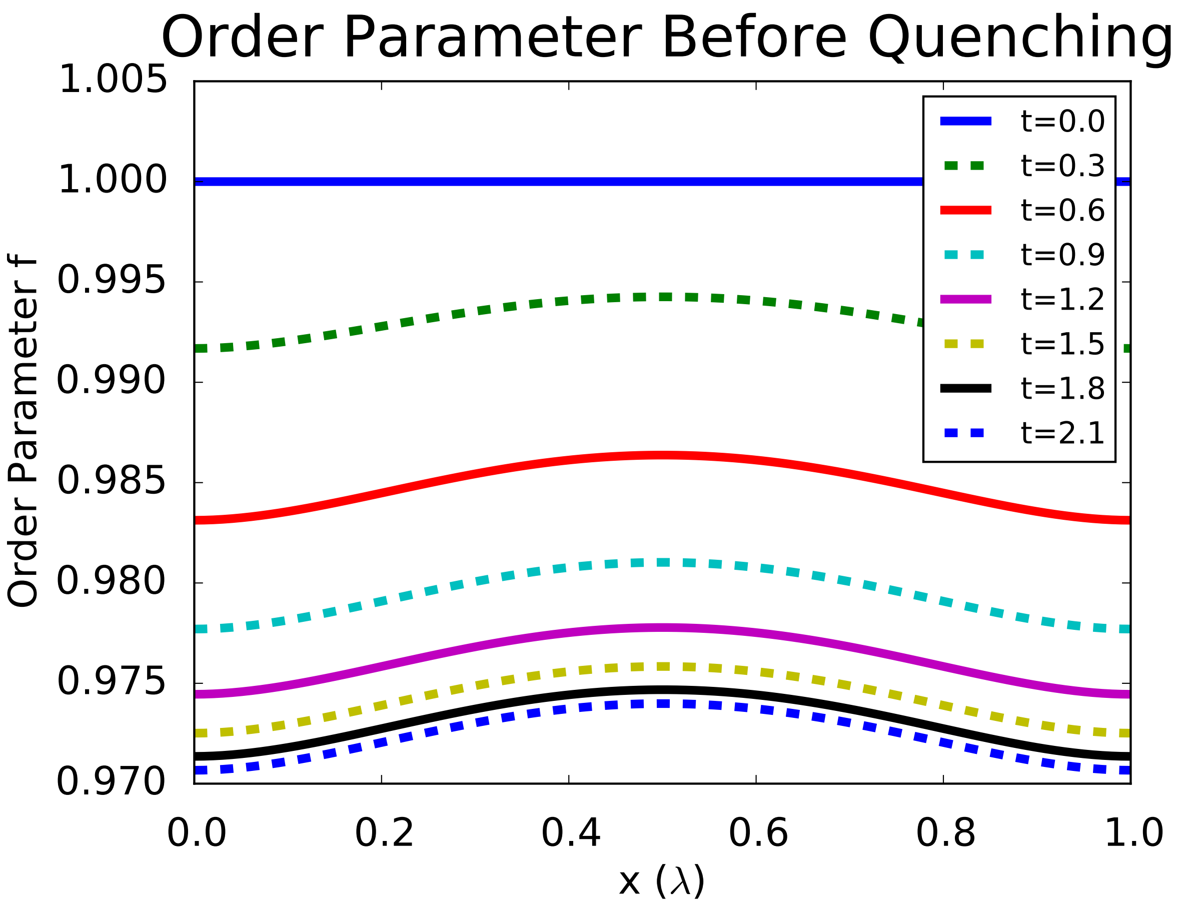

Figure 3.2 The order parameter changes in time increments of .3 when a magnetic field with a strength of .7 is switched on. Observe that the order parameter approaches a nonzero steady state, indicating the material is still superconducting. This is a superconducting film.

The more interesting phenomena occurs when we include time dependence. Figs. (3.2)-(3.5) show what happens when we have an initial superconducting state and then turn on an external magnetic field. Throughout the material the order parameter is set to one and the magnetic potential to zero, representing a perfectly superconducting state. An external magnetic field is instantaneously switched on to .7, given time to settle, and then increased to 10.

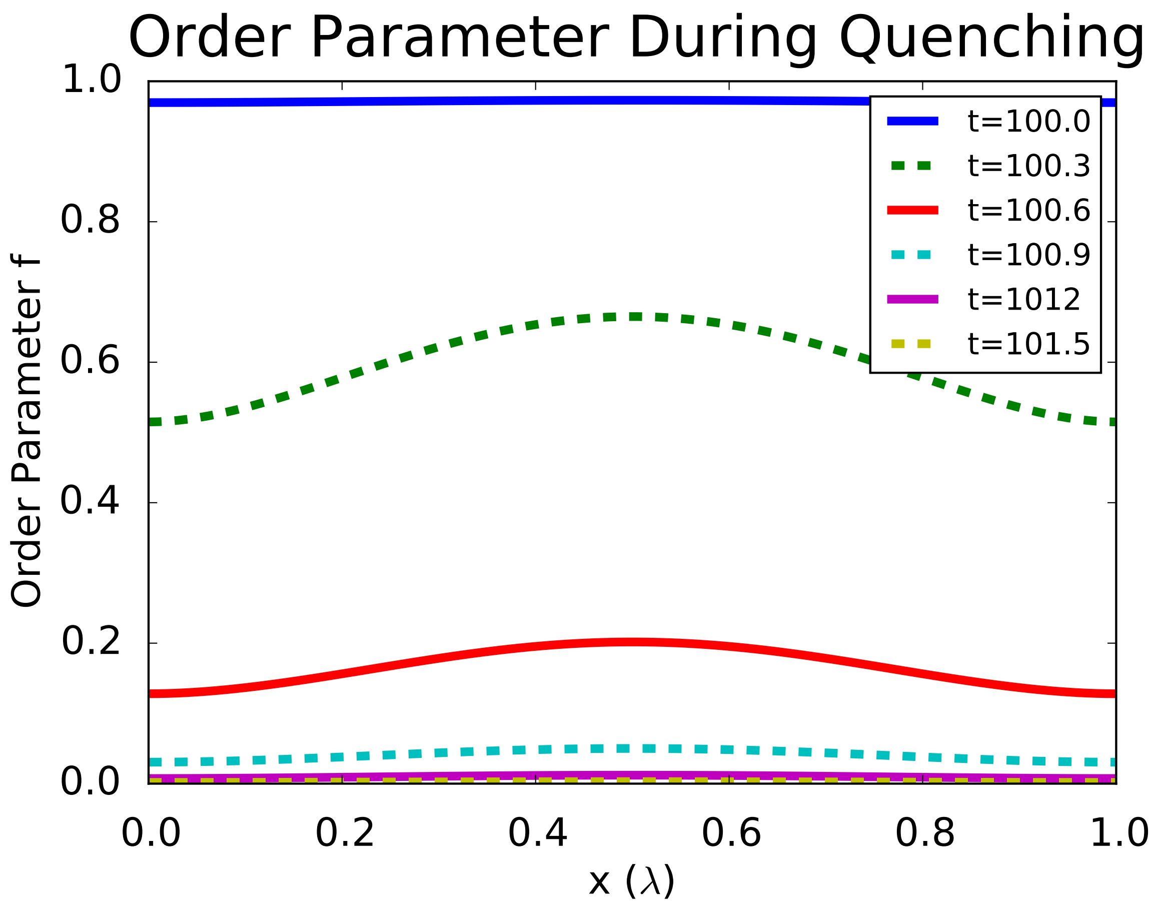

Figure 3.3 Having increased the external magnetic field from .7 to 10, we observe quenching. The order parameter quickly drops to zero, indicating the material loses superconductivity. Each line is the order parameter incremented by .3 time units. This is a superconducting film.

Observe in Fig. 3.2 that the order parameter is initially one throughout. Once we turn on the field we can see how the order parameter drops but remains nonzero. Each line represents a time increment of .3. Similarly in Fig. 3.3, the magnetic field initially starts at zero. When the field is turned on it raises to .7 on the boundaries, but has a dip in the middle. This shows the Meissner effect is trying to expel the internal magnetic field.

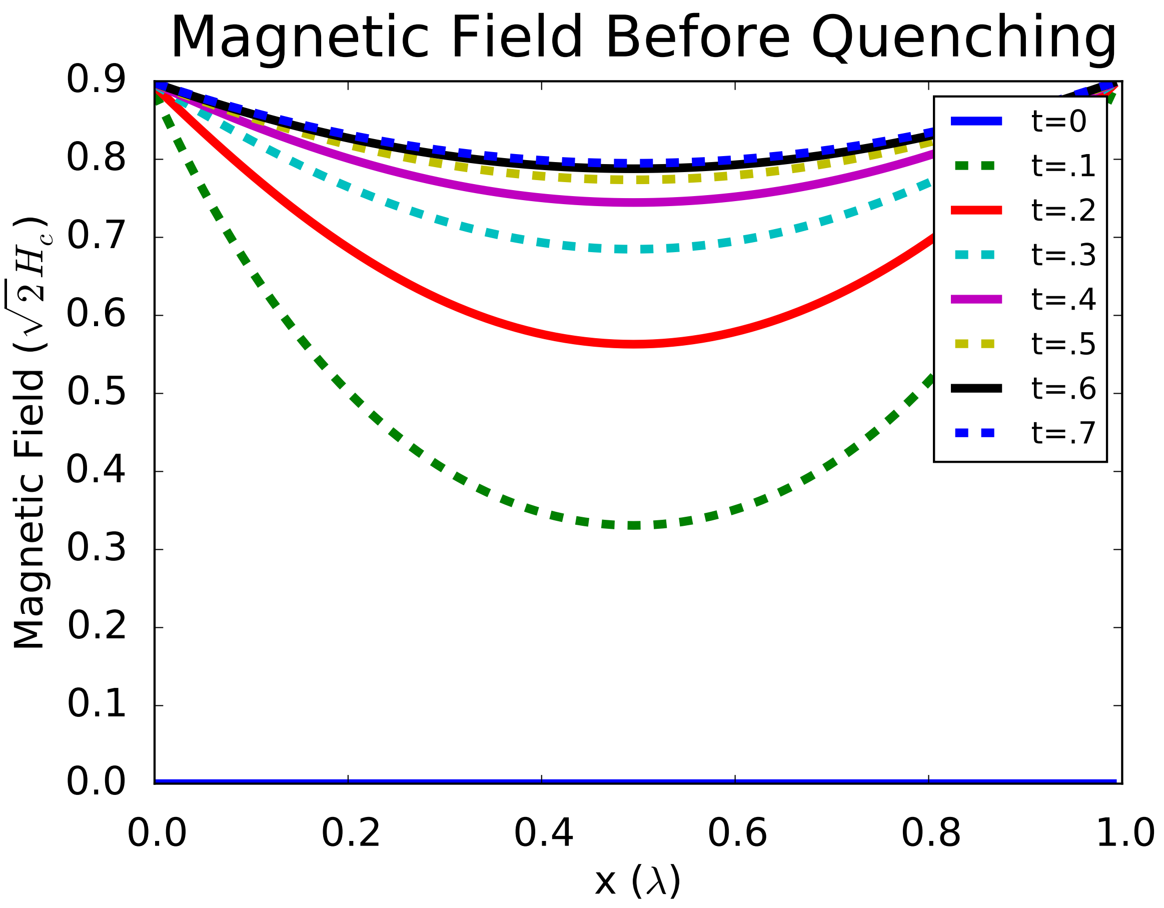

Figure 3.4 When an external magnetic field is switched on to .7, the magnetic field increases inside the superconductor; however, the Meissner effect expels this field, causing a dip in the curve. This is a superconducting film.

Quenching occurs when the magnetic field is raised even higher. After letting the superconductor approach the steady state, the magnetic field is raised to 10. Notice that in Fig. 3.2 the order parameter not only decreases, but drops to zero. The magnetic field increases from .7 to 10. There is no dip in the field, indicating a full penetration of the field in the material. If we were to make a video, we would see that the transition from superconducting and nonsuperconducting happens very rapidly. Through computational modeling, we have successfully recreated important phenomena in superconductors.

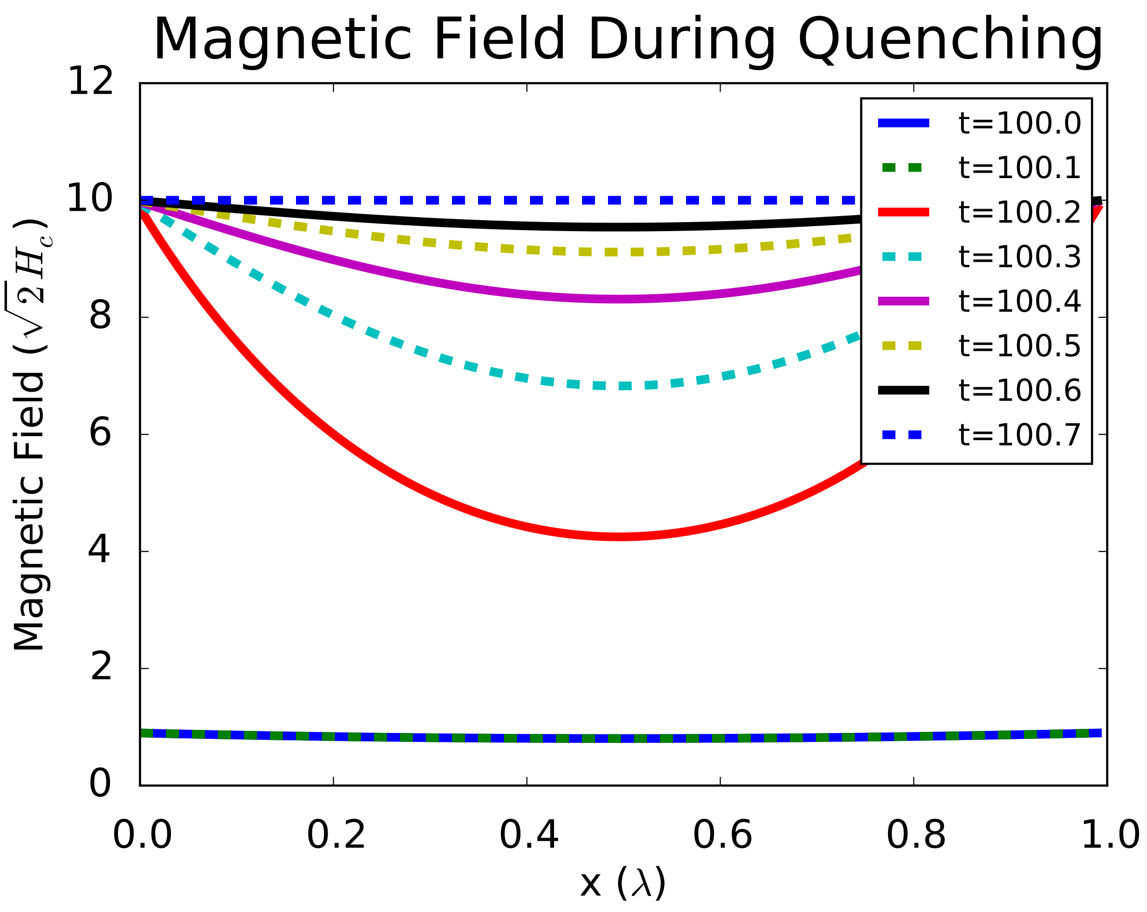

Figure 3.5 The external magnetic field is raised from .7 to 10. This causes quenching, or the complete penetration of the field into the superconductor. Note that at $t = 100.7$ the superconductor is no longer able to expel the magnetic field (the magnetic is 10 throughout.) This models a superconducting film.

3.2 Future Work

The next step in our research will be to generalize our results to more complex geometries. Instead of assuming transverse symmetry, we can look at how a cylindrical superconductor behaves in a magnetic field. This will allow us to see the surface currents, which were not observable with just one spatial variation. The superconducting Meisner state is metastable. Superconductivity can persist above the critical magnetic field up to a superheating field. Time dependent solutions in multiple dimensions will allow us to study this metastable transition in greater detail. After looking at the cylinder we can take one step further to simulate the geometries in a resonant cavity, hopefully giving us an better understanding of how to design them.

Another point to consider is that in chapter one we assumed κ and α were one. This is generally not true. Depending on the material used κ and α can vary, which will change the solution that we have found. As we move forward, continuing this research project, we will use κ and α values that correspond to materials used in resonant cavities.

To conclude, computational methods are a powerful tool. Once mastered they can save money and time by providing models with which experimentalists can receive insight. In the future, computational models will play an important role for the development of new accelerator technologies that are critical for many areas of science.

Appendix A

The code used in the plots of the film superconductor are given as an example of how to use FEniCS in python.

```
###Code begins here

#This solves the time dependent Ginzburg-Landau equations of superconductivity

#assuming Neumann conditions on the boundary.


from dolfin import *

import numpy as np


##Define a mesh

L = 1.0

x = np.linspace(0,L,100)

dt = 0.1

t = 0

T = 1000*dt

mesh = IntervalMesh(99,0,L)

V = FunctionSpace(mesh, "CG", 1)
```

```
#Define function spaces, test functions, and trial functions
ME = V*V
fq = Function(ME)
dfq = TrialFunction(ME)
df, dq = split(dfq)
f, q = split(fq)
r, p = TestFunctions(ME)

#Define initial conditions and boundary conditions
fq0 = Expression(('1.0','0.0'), L = L)
fq_1 = interpolate(fq0,ME)
f_1,q_1 = fq_1.split()
fpL = 0.0
fpR = 0.0
g = Expression('-fpL + (fpR + fpL)/L*x[0]', fpL = fpL, fpR = fpR, L = L)
HL = 0.9
HR = 0.9
h = Expression('-HL + (HR + HL)/L*x[0]', HL=HL, HR = HR, L = L)

#Set up GLE
F1= f*r*dx-f_1*r*dx+dt*(inner(grad(f),grad(r))
    +inner(f**3,r)-inner(f,r)+inner(f*q**2,r))*dx-dt*g*r*ds
F2= q*p*dx-q_1*p*dx+dt*(inner(grad(q),grad(p))+inner(f**2*q,p))*dx-dt*h*p*ds
F=F1+F2
```

```
a = derivative(F,fq,dfq)


#Define the problem and the appropriate solver
class TDGLequation(NonlinearProblem):
    def __init__(self, a, L):
        NonlinearProblem.__init__(self)
        self.L=L
        self.a = a
    def F(self, b, x):
        assemble(self.L, tensor=b)
    def J(self, A, x):
        assemble(self.a, tensor=A)


problem = TDGLequation(a,F)
solver = NewtonSolver()


fsols,qsols = [f_1.compute_vertex_values()],[q_1.compute_vertex_values()]
ts = [0]


#Use a while look to cycle through time steps. This is the first magnetic field
while t<=T:
    t += dt
    ts.append(t)
    h.HL = HL
    h.HR = HR
    solver.solve(problem, fq.vector())

    fq_1.vector()[:]=fq.vector()

    f_1,q_1 = fq.split()

    fsols.append(f_1.compute_vertex_values())

    qsols.append(q_1.compute_vertex_values())


#Turn on a second higher magnetic field

T = 2*T

HL = 10.0

HR = 10.0

while t<=T:

    t += dt

    ts.append(t)

    h.HL = HL

    h.HR = HR

    solver.solve(problem, fq.vector())

    fq_1.vector()[:]=fq.vector()

    f_1,q_1 = fq.split()

    fsols.append(f_1.compute_vertex_values())

    qsols.append(q_1.compute_vertex_values())
```

Appendix B

In order to verify FEniCS gives accurate results we solved the heat equation in one dimension and compared FEniCS results with an analytic solution.

```
#This solves the diffusion equation in 1 dimension by using tools in FEniCS.
#Dirichlet boundary conditions. Compared with an analytic solution.


from dolfin import *
import numpy as np
from scipy.integrate import odeint
from scipy.optimize import fsolve
from scipy.integrate import trapz
import pylab as p
path="basisfunctions/"


#First we find an analytic solution to compare FEniCS with.
#These are predefined basis functions and their matrices.
N = 8
N2=N-2
```

```
PhiPhiMatrix = np.load(path+"PhiPhi%i.npy" %N)

Phi4Matrix = np.load(path+"Phi4%i.npy" %N)

dPhidPhiMatrix = np.load(path+"dPhidPhi%i.npy" %N)

dPhiBCPhiMatrix = np.load(path+"dPhiBCPhi%i.npy" %N)

Philist = np.load(path+"Philist%i.npy" %N)

dPhilist = np.load(path+"dPhilist%i.npy" %N)


#This contain the boundary conditions from another problem.

AfPhi = np.load(path+"AfPhi%i.npy" %N)

AfdPhi = np.load(path+"AfdPhi%i.npy" %N)

L=1

x=np.linspace(0,L,len(Philist[:,1]))


#Using the singular value decomposition we force the boundaries to zero.

Afnull = np.linalg.svd(AfPhi)[2][2:]

Afdnull = np.linalg.svd(AfdPhi)[2][2:]

Philist2=np.dot(Philist, Afnull.T)

PhiPhiMatrix = np.dot( Afnull,np.dot( PhiPhiMatrix,Afnull.T))

PhiPhiMatrixInverse = np.linalg.inv(PhiPhiMatrix)

dPhidPhiMatrix = np.dot( Afnull,np.dot( dPhidPhiMatrix,Afnull.T))

dPhilist2=np.dot(dPhilist, Afdnull.T)


#This is the initial state (coefficients) of the heat equation.

a0 = np.zeros(N)

a0[-2] = 1.0
```

```
a0 = np.dot(Afnull, a0)


def Res(a):
   return  np.dot(PhiPhiMatrixInverse, np.dot(dPhidPhiMatrix, a))



def RHS(a, t):
   return -Res(a)


dt = 0.01/10
ts = np.arange(0,0.1, dt)
As = odeint(RHS, a0, ts) ## Coefficients with N-2 basis funcions



initial = np.array(np.sin(2*np.pi*x))


#Formulate coefficients for analytic solution.
def bn(a):
    return 2/L*trapz(initial*np.sin(np.pi*x*a/L),x)



M=10 #numberof basis for analytic soln
soln = np.empty( (M,len(x)) )
damp = np.empty( (M,len(ts)) )
```

```
for i in range(M):

    soln[i,:] = bn(i)*np.sin(np.pi*x*i/L)
    damp[i,:] = np.exp(-i**2*np.pi**2*ts/L**2)

solnf = np.dot(damp.T, soln)



#Now fenics starts here.
#Create mesh and define function space

#define a mesh and function space.
mesh = IntervalMesh(99,0,1)
V = FunctionSpace(mesh, "CG", 1)

#Define boundary conditions

# u0 = Expression(' x[0]*(1-x[0]) ') ## Initial conditions
u0 = Expression('sin(x[0]*2*pi)', pi = np.pi)
u0.t = 0
def boundary(x, on_boundary):
    return on_boundary
##Set Dirichlet conditions
bc = DirichletBC(V, u0, boundary)
u_1 = interpolate(u0,V)
```

```
#Define variational problem
#dt = ts[1] - ts[0]


f = Constant(0)
u = TrialFunction(V)
v = TestFunction(V)
a = u*v*dx+dt*inner(grad(u),grad(v))*dx
#L = (u_1+dt*f)*v*dx
L = u_1*v*dx


A = assemble(a)


u = Function(V)
T = ts[-1]
t = dt



sols = [u_1.compute_vertex_values()]
ts = []#a = u*v*dx+dt*inner(grad(u),grad(v))*dx
#Solve for best fit.
while t<= T:
    b = assemble(L)
    bc.apply(A,b)
    solve(A, u.vector(), b)
```

```
t += dt

u_1.assign(u)

sols.append(u.compute_vertex_values())

ts.append(t)
```

Appendix C

We generalize the results from Appendix B to account for nonlinearities. We compare our process with a leapfrog approach.

```
#This solves the nonlinear diffusion equation in 1 dimension by using tools in fenics. D

from dolfin import *

import numpy as np

import pylab as p

#Create mesh and define function space


x = np.linspace(0,1,100)

mesh = IntervalMesh(99,0,1)

V = FunctionSpace(mesh, "CG", 1)


#Define boundary conditions


#u0 = Expression('2*sin(x[0]*pi)', pi = np.pi)

u0 = Expression('-96.5*x[0]*pow((x[0] - 1),5)')
```

```
def boundary(x, on_boundary):

    return on_boundary


bc = DirichletBC(V, u0, boundary)

u_1 = interpolate(u0,V)


#Define variational problem

dt = 1e-2

f = Constant(0)

u = TrialFunction(V)

v = TestFunction(V)


L = (u_1 + dt*f)*v*dx

u = Function(V)

L

a = u*v*dx+dt*inner(grad(u),grad(v))*dx+dt*inner(u,v)*dx-dt*inner(u**3,v)*dx-L


T = dt*10

t = dt



sols = [u_1.compute_vertex_values()]

ts = []


J = derivative(a,u)
```

```
problem = NonlinearVariationalProblem(a, u, bc, J)

solver = NonlinearVariationalSolver(problem)

solver.solve()


while t<= T:

    F = a

    solve(F==0, u, bcs = bc)

    t += dt

    u_1.assign(u)

    sols.append(u.compute_vertex_values())

    ts.append(t)


###

#This script solves fdot = d2f/dx2 + f - f**3 Dirichlet conditions with leapfrog approac

###


##set up ghost point mesh

from numpy import *

import pylab as p


a = 0.0

b = 1.0

N = 20

h = b/N

#x = linspace(a-h/2,b+h/2,N+2)
```

```
x = linspace(a,b,N)

dt = .0001

f = sin(pi*x)

#f  = -96.5*x*(x - 1)**5

f0 = f

fsols = [f0]

#fold = zeros(len(x))

#fold2[1:N] = dt/h**2*(-1*f0[0:N-1]-1*f0[2:N+1]+2*f0[1:N])-dt*(f0[1:N]-f0[1:N]**3)+f0[1:


ynew = zeros( N )

ynewlist = []

tf = 1.0

nsteps =int(tf/dt)

#for n in range(0,nsteps-1):

t = 0.0

count = 0


#def Laplacian(y, h):

#    return (y[:-2] + y[2:] - 2*y[1:-1])/h/h


while t < dt*1000:

# for n in range(5):

    ynew[1:-1] = f0[1:-1]+dt/h**2*(f0[0:-2]+f0[2:]-2*f0[1:-1])#+dt/h**2*(f0[1:-1]-f0[1:-

    ynew[0] = ynew[1]

    ynew[-1] = ynew[-2]
```

```
#ynew[0] = 0.0

#ynew[-1] = 0.0


# ynewlist.append(ynew.copy())

f0 = ynew.copy()

if count == 10:

    fsols.append(f0)

    count = 0

t += dt

count += 1
```

Bibliography

[1] E. Collings, M. Sumption, and T. Tajima, “Magnesium diboride superconducting RF resonant cavities for high energy particle acceleration,” Superconductor Science and Technology **17,** S595 (2004).

[2] H. Padamsee, K. Shepard, and R. Sundelin, “Physics and accelerator applications of RF superconductivity,” Annual Review of Nuclear and Particle Science **43,** 635–686 (1993).

[3] M. Tinkham, *Introduction to superconductivity* (Courier Dover Publications, New York, 2012).

[4] G. Wells, K.-A. Mardal, and A. Logg, *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book* (Springer, New York, 2012).

[5] S. J. Chapman, “Superheating field of type II superconductors,” SIAM Journal on Applied Mathematics **55,** 1233–1258 (1995).

[6] A. Logg, “Origin of the name FEniCS,”, https://answers.launchpad.net/fenics/+question/204935, 2012, (accessed: 2014-03-28).

Index