Thermal Detection of Inaccessible Corrosion*

Matt Charnley† and Andrew Rzeznik‡

July 1, 2013

Abstract

In this paper, we explore the mathematical inverse problem of detecting corroded material on the reverse side of a partially accessible metal plate. We provide a novel formulation of the two-dimensional problem using a heat source as the detection method, developing a numerical method for performing these reconstructions. The reconstruction is performed via integration against test functions, and we will show how a linearization can be used to simplify the initial problem and explain a regularization method used to obtain acceptable results for the corrosion profile. Results will be shown for a variety of corrosion profiles and system thermal parameters and error will be quantified for each case. It is shown that the reconstruction of small corrosion profiles is within a reasonable amount of error for real-world applications (20%), and while larger corrosion is less accurately reconstructed, this method will allow for detection of corrosion of any size. Possibilities for future work are also outlined, including the definition of regularization parameters and extending the procedure to different domains.

Contents

*Rose-Hulman REU 2012, NSF Grant DMS-1003924
†University of Notre Dame
‡Cornell University

1 Introduction and Motivation

Corrosion in metal can be caused by a variety of factors. Rusting is a common form of corrosion, which occurs when metals, such as iron, are exposed to the air in the presence of water. Rusted materials are significantly weaker than their unrusted counterparts, and this change in strength can lead to significant problems as the rust takes over more of the metal surface. Other defects in materials can be caused by acids, wear over time, or physical damage, and all of these can also be treated as types of corrosion, so long as the damaged region is given proper strength parameters.

Corrosion and defects in material can lead to serious consequences in engineering applications. In mild cases, the presence of corrosion or damaged material may alter the efficiency of a system or slightly change how the system behaves without affecting the overall outcome. In the most severe case, corrosion that has developed over a long period of time can lead to material failure, as well as personal injury to anyone in the area. Material failure could result in anything from a pipe leak in a chemical plant to the collapse of a bridge, all of which we want to avoid. For this reason, it is desirable to be able to find and analyze corrosion in the field before it has time to develop into a more serious, and potentially deadly, problem.

In order to model the corrosion, and hopefully avoid the costly consequences of material failure, we attack the situation as a mathematical inverse problem. We assume that we have a metal plate with two disjoint regions, regular material and corroded material, with the boundary between the two regions defined by a function of horizontal position in the plate, supported away from the edges of the plate. Using the heat equation with temperature and heat flux continuity on the boundary between the two regions, we seek to develop a method of determining where the boundary is between the regular and corroded regions. In order for this method to be effective in real-world situations, it must be able to reconstruct the corrosion profile from a limited amount of temperature data, which would be available on one surface of the plate, and do so within a reasonable amount of error. All of this was achieved for small corroded regions in the two-dimensional problem with the method described in this paper.

Previous attempts at solving this problem include one that uses electricity, following most of the same analysis carried out here. There are two main differences between using electricity and heat. Firstly, the heat equation has a time component, which makes the calculations more complicated, but also allows for more data to be extracted from the problem. Secondly, from a practical standpoint, electricity is more difficult to use because it requires direct contact with the plate in order to apply a voltage drop across it. Heat, on the other hand, does not require such a contact. Lasers and infra-red sensors can be used to apply heat fluxes and measure temperature from a distance, thereby allowing the corrosion in the plate to be analyzed without attaching anything to it, and potentially causing more damage. The paper from this REU in 2009 carries out the electrical analysis, including the linearization and regularization methods [1]. Most of our process follows their procedure, with several modifications for the heat equation as necessary. Analysis with the heat equation has been carried out before (see [2] and [4]), but we present a novel linearization method to reconstruct the corrosion in the material in the two-dimensional case.

Our method proceeds by using Green's Identities on the temperature in the plate and a set of test functions that are developed to have specific properties. These identities allow us to use what we know about the inside of the plate to help analyze the boundary. By assuming that all but one of the boundaries of the plate are insulated, we can reduce to using what we know about the temperature and heat flux at the top of the plate (where we can measure temperature) in order to find the boundary of the corroded region. We then assume that the corroded region is 'small,' and use a linearization approximation to simplify the problem and generate a system of equations that a computer can be used to solve. A variety of different heat fluxes and corrosion profiles were tested in the hopes of developing a process that could be used to accurately reconstruct any corrosion profile. We also hope that this analysis will grant more insight into the problem and how the corrosion affects the transfer of heat within the body.

This paper will cover the entire process described above. Section 2 will lay out the general setup used for this paper and discuss the forward problem, which is solved by a heat equation solver and was used to generate the data that would be plugged into our reconstruction algorithm. Section 3 defines and discusses inverse problems, along with the specific problem we are attacking here: finding the corrosion profile from temperature measurements. Section 4 discusses various issues with attempting to solve an inverse problem, namely uniqueness and ill-posedness. Section 5 will outline the process of generating a solution, while the full details of the algorithm generation can be found in the Appendices. Section 6 will discuss different methods used to improve the resulting profiles, and Section 7 will show some of our results. Finally, the paper will close with some conclusions and possibilities for future work in Sections 8 and 9.

1.1 Terms and Definitions

The following terms and symbols will be used frequently throughout the paper

Ω	The entire region where heat transfer is taking place.
Ω_1	The section of Ω where the material has not been corroded.
Ω_2	The section of Ω where the material has been corroded.
u_0	The temperature profile in the uncorroded plate Ω made of material (1), using the same input flux used to compute u_1, u_2.
u_1	The temperature profile in Ω_1, function of (x, y, t).
u_2	The temperature profile in Ω_2, function of (x, y, t).
$C(x)$	The corrosion profile; the function of the curve that separates Ω_1 from Ω_2.
α	Thermal diffusivity, a material property. Determines how fast heat can diffuse through the medium.
α_i	Thermal diffusivity for material i, which is present in region Ω_i.
k	Thermal conductivity, a material property. Determines how well heat can be conducted in from the environment.
k_i	Thermal conductivity for material i, which is present in region Ω_i.

2 Forward Problem

In general, the forward problem assumes that we know the boundary and constraints of the system and need to predict how it will respond to an outside stimulus. For our particular problem, which deals with corrosion in a metal plate, we will assume that the metal plate is a finite rectangle Ω of length L and height 1, as shown in Figure 1. The rectangle Ω is set in the Cartesian plane $\mathbb{R}^2$ so that $x = 0$ marks the left side of the rectangle, $x = L$ is the right edge, $y = 0$ denotes the bottom of the sample and $y = 1$ indicates the top.

We also assume there are two regions, separated by the curve $C(x)$, each having different thermal properties, which include thermal conductivity (k) and thermal diffusivity (α). We will also assume that all external boundaries except the boundary at $y = 1$ are perfectly insulated, so no heat can enter or escape, while some defined heat flux $g(x)$ is applied on the top boundary, $y = 1$. Temperature and heat flux are assumed to be

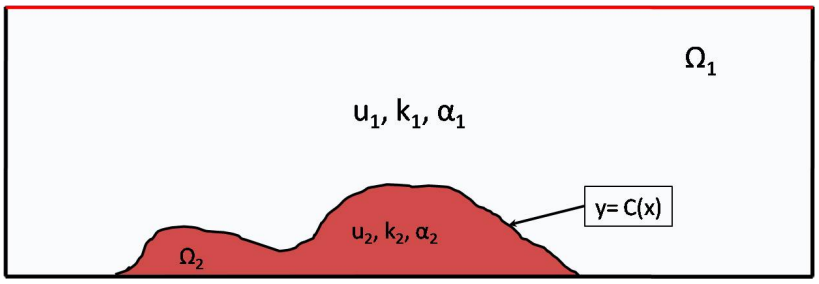


Figure 1: General setup for the problem

continuous over any interface, including the curve $C(x)$. All of these conditions can be formally stated as follows.

We assume that u_1 satisfies

$$\begin{aligned} \frac{\partial u_1}{\partial t} - \alpha_1 \nabla^2 u_1 &= 0 \text{ on } \Omega_1 \\ \frac{\partial u_1}{\partial x} &= 0 \text{ on } x = 0,\ x = L \\ \frac{\partial u_1}{\partial y} &= g(x) \text{ on } y = 1 \\ u_1(x, y, 0) &= 0 \text{ on } \Omega_1, \end{aligned}$$

while it is assumed that u_2 satisfies

$$\begin{aligned} \frac{\partial u_2}{\partial t} - \alpha_2 \nabla^2 u_2 &= 0 \text{ on } \Omega_2 \\ \frac{\partial u_2}{\partial x} &= 0 \text{ on } x = 0,\ x = L \\ \frac{\partial u_2}{\partial y} &= 0 \text{ on } y = 0 \\ u_2(x, y, 0) &= 0 \text{ on } \Omega_2, \end{aligned}$$

and the continuity conditions on $C(x)$ give us

$$\begin{aligned} u_1 &= u_2 \text{ on } C(x) \\ k_1 \frac{\partial u_1}{\partial \vec{n}} &= k_2 \frac{\partial u_2}{\partial \vec{n}} \text{ on } C(x). \end{aligned}$$

The forward problem would be stated: Given the input flux $g(x)$, the thermal properties of both materials, and the curve $C(x)$ dividing Ω_1 and Ω_2, find the temperature profiles u_1 and u_2 that satisfy these equations.

If the domain is simple, *e.g.* a rectangle, Fourier Series can be used to solve this problem. However, in our case, the corrosion profile makes the domain somewhat irregular, and numerical methods become the main tool for reaching a solution. COMSOL Multiphysics was our program of choice for solving the forward problem. It uses a finite element method to generate the solution to the heat equation given a geometry and any specified combination of heat fluxes, insulated surfaces, and material properties. From this data, it will generate the temperature values for a mesh of points (x, y) in the plate at a set of specified timesteps. The program also contains a functionality to export different sections of the data to text files, which can be used in MATLAB or other numerical software. This software is very useful to us since without any experimental apparatus, this forward solver is our only source for the data needed to test our methods for solving the inverse problem.

3 Inverse Problem

The forward problem is a deterministic differential equation that can be solved numerically. However, in engineering applications, it is assumed that we will be able to measure temperature and an input flux, but do not know what the inside of the material looks like. This type of formulation is what characterizes a mathematical inverse problem. Specifically for our problem, we do not know, and would like to determine, the function $C(x)$ that divides the two regions of different materials. Assuming that the second region is corroded or missing material, knowing this function $C(x)$ could help determine the safety of that component in a structure and help to see if it needs to be replaced.

Intuition says that the change in material properties in a specific area of the plate should affect the way the heat flows through the object. It was hoped that this would cause a significant change in the temperature profile of the plate, specifically on the top surface. Figure 2 shows a COMSOL image of the temperature in two identical plates after a given amount of time. The only difference between the two plates is the missing material on the bottom of the left plate.

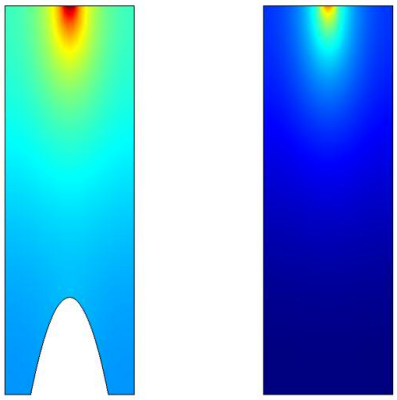


Figure 2: COMSOL images showing changes in temperature profile due to corrosion

As can be seen in Figure 2, there is clearly a difference in the temperature at the top of the plate caused by the corrosion at the bottom. This gives us hope that we should be able to recover at least some information about the corrosion profile from the accessible surface temperature. As will be seen in the later sections of the paper, this temperature difference gives us enough information to get an approximation to the corrosion on the back surface of the plate, identifying both the location and size of the corroded region.

4 Uniqueness and Ill-Posedness

For most differential equations problems, there are two main conditions that factor into finding a solution: existence and uniqueness. For inverse problems, however, a third condition comes into play, the fact that the inverse problem must be well-posed. Existence for our problem is fairly easy to show: all of the temperature data used in our reconstruction algorithm came from solving the forward problem for a given corrosion profile, so there is clearly a corrosion profile that can generate the given temperature data. Uniqueness, however, is much more important to us. The ability of any reconstruction method to generate an accurate representation of the actual corrosion in the system depends on the mathematical inverse problem having a unique solution; *i.e.* for any given set of data on the top surface and thermal parameters for the two plate regions, there is only one possible corrosion profile $C(x)$ that can generate that data. A formalization of this requirement is stated in the following conjecture:

Conjecture 1 (Inverse Problem Uniqueness)**.** If, for any input flux $g(x)$ on the top surface, two corrosion profiles $C(x)$ and $C^*(x)$ generate the same temperature solution u_1 on the top boundary of Ω_1 for all time, then $C(x) = C^*(x)$.

While this is still an unproven conjecture, we will assume it for this problem in order to move forward with our reconstruction algorithm. A previous paper proves this claim for the case that Ω_2 is missing material [4], which is our problem in the special case that $k_2 = \alpha_2 = 0$. However, this result has not been extended to the case of a corroded region on the plate with non-zero α_2 and k_2. We will continue on with the problem with the assumption that our uniqueness conjecture holds, and the results displayed later seem to empirically verify this assumption.

Finally, the inverse problem must also be well-posed for there to be a chance of developing this kind of reconstruction algorithm. Well-posedness means that a small change in the given data of the problem results in a small change in the reconstructed solution, while ill-posedness is the strong dependence of the solution on the specific initial data. The inverse problem we are trying to solve here is ill-posed, which, as described in Section 6.1, presents itself in the fact that the matrix $\mathbf{B}$ is ill-conditioned. Thus, taking the inverse of $\mathbf{B}$ in order to solve the problem results in a matrix with extremely large values that is multiplied by the initial data to get a solution. Thus, if the initial data changes even a little bit, the large values in the inverse of $\mathbf{B}$ will magnify this small change to significantly alter the solution. Even a small amount of noise or numerical error in initial data can lead to a reconstructed solution that is very far from the truth. Therefore, extra steps need to be taken to ensure that the solution generated to this problem is reasonable and fits to the experimentally generated profiles. The steps taken to deal with the ill-posedness of this problem will be discussed in Section 6.1, and the results of this procedure can be seen in Section 7.

5 Solving the Inverse Problem

The approach to solving this inverse problem involves applying Green's Identity to the heat equation, and a linearization method to generate a simplified set of equations that can be solved by a computer. Only a summary of the calculations is presented; the full version may be found in the Appendices. The layout and formulation of the problem can be seen in Section 2, with the equations as stated there. The goal is to find the curve $C(x)$ that can generate the given data for u_1 along the top surface $y = 1$.

5.1 Computations

In order to attempt to solve this inverse problem, we employed the so-called 'Reciprocity Gap' approach, which is an application of Green's Identities to this specific problem. The idea is to extract information about $C(x)$ from boundary data by integrating u by parts against strategically chosen test functions. We begin with

Theorem 5.1 (Green's Second Identity)**.** *For any bounded region $D \subset \mathbb{R}^2$ with piecewise smooth boundary ∂D, and any two functions $u, v \in C^2(\bar{D})$, we have*

$$\int_D (u\nabla^2 v - v\nabla^2 u)\, dA = \int_{\partial D} \left(u\frac{\partial v}{\partial \vec{n}} - v\frac{\partial u}{\partial \vec{n}} \right)\, ds.$$

We want to use Green's Identity with what we know about both the interior and the boundary of Ω to generate an approximation for the function $C(x)$. This analysis will also use a collection of 'test functions' ϕ_k, $1 \leq k \leq M$. Each test function ϕ_k will satisfy (see Section C in the Appendix for the construction of

specific ϕ_k)

$$\begin{aligned} \frac{\partial \phi_k}{\partial t} + \alpha_1 \nabla^2 \phi_k &= 0 \text{ on } \Omega, \\ \frac{\partial \phi_k}{\partial \vec{n}} &= 0 \text{ on } y = 0,\ x = 0, \text{ and } x = L, \\ \phi_k(x, y, T) &= 0 \text{ on } \Omega. \end{aligned}$$

We start with the two equations

$$\begin{aligned} \int_0^T \int_{\Omega_1} u_1 \left(\frac{\partial \phi}{\partial t} + \alpha_1 \nabla^2 \phi \right) \, dA \, dt &= 0 \\ \int_0^T \int_{\Omega_2} u_2 \left(\frac{\partial \phi}{\partial t} + \alpha_1 \nabla^2 \phi \right) \, dA \, dt &= 0 \end{aligned}$$

and, using Green's Identity and integration by parts, we can reduce these equation to

$$RG(\phi) = \left(\frac{k_2}{k_1} - 1 \right) \int_0^T \int_0^L C'(x) \phi \frac{\partial u_2}{\partial x} \bigg|_{C(x)} - \phi \frac{\partial u_2}{\partial y} \bigg|_{C(x)} dx \, dt + \frac{\alpha_2 - \alpha_1}{\alpha_1 \alpha_2} \int_0^T \int_{\Omega_2} \phi \frac{\partial u_2}{\partial t} \, dA \, dt \tag{5.1}$$

where we have defined

$$RG(\phi) = \int_0^T \int_{top} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right) \, ds \, dt \tag{5.2}$$

for one of the test functions ϕ. Crucial steps in this reduction, as can be seen in the Appendices, involve use of the heat equation and the specific boundary conditions assigned to both the temperature profile in the plate and ϕ. Continuity along the boundary, as discussed previously, allows us to combine the two equations into a single expression.

5.2 Linearization

The next step to solving this inverse problem is linearization. Specifically, we assume that the corrosion profile is a small perturbation of the uncorroded back surface $y = 0$, and then use this assumption to formally linearize the relationship between the function $C(x)$ and the top surface temperature data. For this procedure, we focus on the heat equation in Ω_1 and allow the linearization assumption to take care of the terms from Ω_2.

Making the linearization assumption as stated above, we assume that

$$C(x) = \epsilon C_0(x)$$

where ϵ is some small positive constant, and $C_0(x)$ is an order 1 function. When we perform a power series expansion on the terms in our equation, this will allow us to ignore terms of order ϵ^2, effectively linearizing the equation. This linearization assumption will be valid as long as the maximum height of the corrosion is less than around 0.1. If it is bigger than that, terms of order ϵ^2 may not be 'small' and could play a role in the solution to the resulting equations. However, we will assume that this holds, and move forward with the calculations.

By applying this linearization assumption to each term in (5.1) and integrating by parts in a few different terms to make the expression easier for a computer to handle (see Section B in the Appendices), shows that we are looking for solutions to the system of integral equations,

$$RG(\phi_k) = \int_0^L C(x) w_k(x) dx \tag{5.3}$$

with $1 \leq k \leq M$, where we have defined

$$w_k(x) := \int_0^T \left(1 - \frac{k_2}{k_1}\right) \frac{\partial \phi_k}{\partial x} \frac{\partial u_0}{\partial x}\bigg|_{y=0} + \left(\frac{k_2}{k_1} - \frac{\alpha_2}{\alpha_1}\right) \frac{u_0}{\alpha_2} \frac{\partial \phi_k}{\partial t}\bigg|_{y=0} dt.$$

From equation (5.3), we can see that plugging in $C(x) = 0$ gives

$$RG_0(\phi_k) = \int_0^T \int_{top} u_0 \frac{\partial \phi_k}{\partial \vec{n}} - \phi_k \frac{\partial u_0}{\partial \vec{n}} \; ds \; dt = 0. \tag{5.4}$$

Subtracting equation (5.4) from (5.3) gives

$$\tilde{RG}(\phi_k) = \int_0^T \int_{top} \tilde{u_1} \frac{\partial \phi_k}{\partial \vec{n}} \; ds \; dt = \int_0^L C(x) w_k(x) \; dx \tag{5.5}$$

since

$$\frac{\partial u_0}{\partial \vec{n}}\bigg|_{y=1} = \frac{\partial u_1}{\partial \vec{n}}\bigg|_{y=1}$$

where $\tilde{u_1} = u_1 - u_0$ is the disturbance from the uncorroded temperature profile. This is the expression used to compute the function $C(x)$ in MATLAB.

5.3 Finding $C(x)$: Least 2-Norm

From the previous section, we are looking for a solution to the system of equations

$$RG_k = \int_0^L C(x) w_k(x) \; dx \tag{5.6}$$

for $1 \leq k \leq M$. In order to specify a single solution, we look for the function $C(x)$ with the smallest L^2 norm. As shown in [1], the approximation for the function $C(x)$ with this property must be a linear combination of the w_k functions, or

$$C(x) = \sum_{i=1}^{M} \lambda_i w_i(x).$$

Plugging this into equation (5.6) gives

$$\begin{aligned} RG_k &= \int_0^L \sum_{i=1}^{M} \lambda_i w_i(x) w_k(x) \; dx \\ &= \sum_{i=1}^{M} \lambda_i \int_0^L w_i(x) w_k(x) \; dx. \end{aligned}$$

Defining the coefficient matrix $\mathbf{B}$ by

$$\mathbf{B}_{ik} = \int_0^L w_i(x) w_k(x) \; dx$$

gives

$$RG_k = \sum_{i=1}^{M} \mathbf{B}_{ik} \lambda_i$$

for $1 \leq k \leq M$, or

$$\vec{RG} = \mathbf{B}\vec{\lambda}.$$

Then calculating $\vec{\lambda}$ via $\mathbf{B}^{-1}\vec{RG}$ and letting

$$C(x) = \sum_{i=1}^{M} \lambda_i w_i(x)$$

gives an approximation for the corrosion profile in the system.

6 Refining the Solution

Performing the procedure in Section 5.3 and using it to approximate the function $C(x)$ gives a graph similar to that in Figure 3.

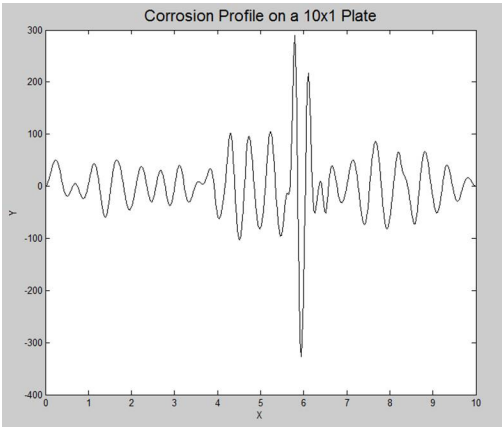


Figure 3: Graph of the raw approximation of $C(x)$

Since our plate is only one unit thick, this is not a feasible result. We need some way to refine our solution to get something that more closely matches the curve $C(x)$.

6.1 Regularization

The high amplitude values and oscillatory nature of the approximation of $C(x)$ in Figure 3 suggest that this inverse problem is ill-posed. In our case, this presents itself in the fact that the matrix $\mathbf{B}$ is ill-conditioned: it has singular values that vary greatly in magnitude. Figure 4 shows the singular values of $\mathbf{B}$ on a log scale, emphasizing the ill-conditioned nature of the matrix. Following the analysis in [1], we implemented a regularization process using the singular value decomposition of the matrix $\mathbf{B}$.

A singular value decomposition, or SVD, is a decomposition of a matrix B of the form $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^{-1}$ where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and Σ is a diagonal matrix [6]. This breaks down the transformation induced by $\mathbf{B}$ into its rotational and extensional parts. We call the diagonal values of Σ the singular values of $\mathbf{B}$. They are similar to the eigenvalues of the matrix in that they dictate how certain components extend or shrink under the transformation induced by $\mathbf{B}$. Since $\mathbf{U}$ and $\mathbf{V}$ are unitary, if $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^{-1}$, then $\mathbf{B}^{-1} = \mathbf{V}\Sigma^{-1}\mathbf{U}^{-1}$. Therefore, the singular values of $\mathbf{B}^{-1}$ are the multiplicative inverses of the singular values of $\mathbf{B}$, and a singular value of $\mathbf{B}$ close to zero will result in a large singular value for $\mathbf{B}^{-1}$. As shown in Figure 3, these large values for $\mathbf{B}^{-1}$ can cause results that are infeasible for the given problem. Therefore, in order to account for this, for any singular value in $\mathbf{B}$ that is less than a given threshold, we will set the corresponding singular value in $\mathbf{B}^{-1}$ to zero. This will eliminate the extremely large singular values in $\mathbf{B}^{-1}$, removing some numerical instability. This calculation does remove the possibility of getting a completely accurate solution, but this optimal result

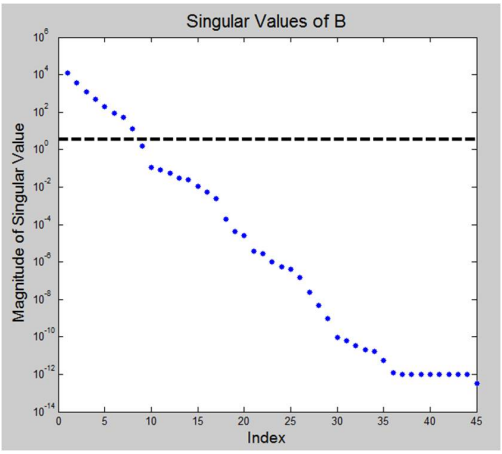


Figure 4: Singular values of the matrix $\mathbf{B}$

is not possible with the numerical data that come out of this process. The exact implementation of this process is described below.

As in [1], we begin by defining ϵ_∞ as the maximum absolute error in the system and ϵ_{thresh} as the relative noise on a 0 to 1 scale. These two parameters can be changed to adjust the amount of regularization applied to the system. For our system, we initially set $\epsilon_{thresh} = 0.2$ and varied ϵ_∞ to improve the regularization. We then set, as defined in [1],

$$\epsilon_d = \frac{\sqrt{M} h_w C_{max} \epsilon_\infty}{\epsilon_{thresh}} \tag{6.1}$$

where M is the number of test functions used, h_w is the average maximum value of the w_k functions, and

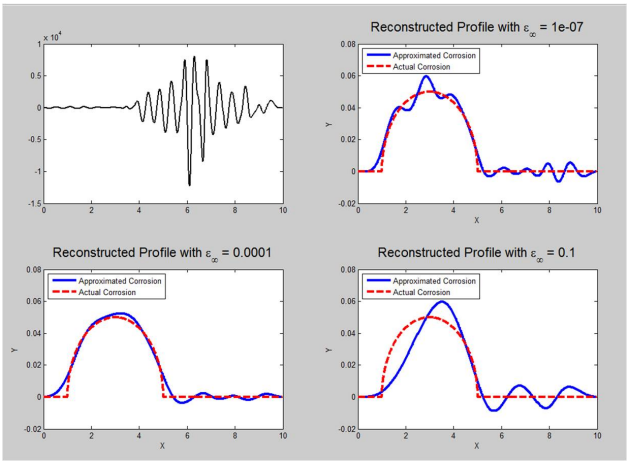
$$C_{max} = \max_k \int_0^L \int_0^T \left| \frac{\partial \phi_k}{\partial y} \right| \, dt \, dx.$$

However, as is discussed in Section D of the Appendices, this definition would not allow us to pick a single value for ϵ_∞ to minimize the error in all of the reconstructions. Therefore, we will redefine ϵ_d to be

$$\epsilon_d = \sqrt{M} h_w C_{max} \xi(k, \alpha) \tag{6.2}$$

where ξ incorporates the ϵ_{thresh} and ϵ_∞ terms and a dependence on the thermal parameters of the system. With this ϵ_d defined, we now look at the singular value decomposition of $\mathbf{B}$. If any of the singular values are less than ϵ_d, we set the corresponding value in $\mathbf{B}^{-1}$ to zero, which eliminates the high magnitude terms in $\mathbf{B}^{-1}$. This is shown graphically in Figure 4 by the black dashed line. Any singular value below that line would be set to zero in $\mathbf{B}^{-1}$. Computing the corrosion profile with this modified $\mathbf{B}^{-1}$ matrix gives much better results that approximate the actual corrosion in the plate. Figure 5 shows how the process of regularization changes the approximation for the curve $C(x)$ for different values of ϵ_∞, which correspond to different values of ξ. In this figure, the solid line shows the reconstructed profile after the stated amount of regularization, and the dashed line is the actual corrosion profile that was used to generate the data used for RG_k.

As can be seen in Figure 5, even a small amount of regularization greatly helps in seeing what the corrosion profile actually looks like. However, it is also possible to over-regularize the process, which is what has happened in the bottom right graph. With this high of a regularization coefficient, many of the singular



		L^2 Error
$\epsilon_\infty = 10^{-7}$	$\xi = 5 * 10^{-7}$	0.0114
$\epsilon_\infty = 10^{-4}$	$\xi = 5 * 10^{-4}$	0.0090
$\epsilon_\infty = 0.1$	$\xi = 0.5$	0.0270
		Error in Max
$\epsilon_\infty = 10^{-7}$	$\xi = 5 * 10^{-7}$	0.0094
$\epsilon_\infty = 10^{-4}$	$\xi = 5 * 10^{-4}$	0.0024
$\epsilon_\infty = 0.1$	$\xi = 0.5$	0.0097

Figure 5: Approximations of the curve $C(x)$ at different regularization levels

values in the decomposition of $\mathbf{B}^{-1}$ have been set to zero, which eliminates most of the w_k functions from the final approximation of $C(x)$. With so few functions, it is impossible for the curve to be modeled accurately. Figure 6 shows a plot of the L^2 error and error in maximum value for each of the reconstructions as a function of the ϵ_∞ parameter. The three fluxes and the average presented here are from the second method of improving the solution, which is described in Section 6.2.

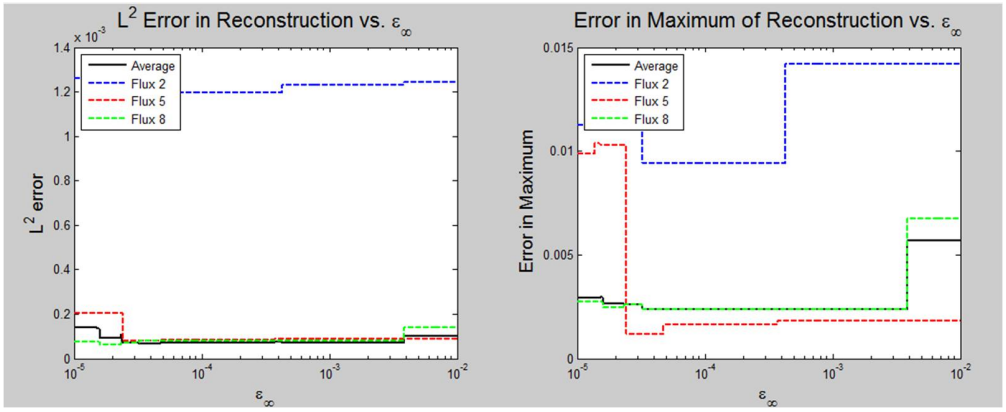


Figure 6: Graph of the L^2 error of the reconstruction as a function of the regularization parameter, ϵ_∞

As seen in the graph, a value of $\epsilon_\infty = 10^{-4}$ seems to be the optimal value for this type of reconstruction. However, as can be seen in the other graphs in Section D of the Appendices, different values of k_2 and α_2 lead to different values of ϵ_∞ that will give the optimal regularization. Therefore, the optimal parameter for each regularization was chosen for the data presented in Section 7. More discussion of the regularization parameters and how they were chosen can be found in Section D of the Appendices.

6.2 Variation of Input Flux

There is one more significant problem with the w_k functions generated above, specifically when $\frac{k_2}{k_1} = \frac{\alpha_2}{\alpha_1}$. In this case, the second term of the integral drops out, and all of the functions will be exactly zero wherever

$\frac{\partial u_0}{\partial x} = 0$. The most notable point where this is going to occur is directly under the symmetry point of the heat source. The problem is that no matter the choice of ϕ_k, all of the w_k functions are zero at the center of the heat source, meaning that the approximation for $C(x)$ will also go to zero there. This method can not model any corrosion at this point or in the surrounding area because the functions are continuous. Figures 7 and 8 show the w_k functions for a heat flux centered at $x = 5$ and the resulting $C(x)$ profile, illustrating the problem.

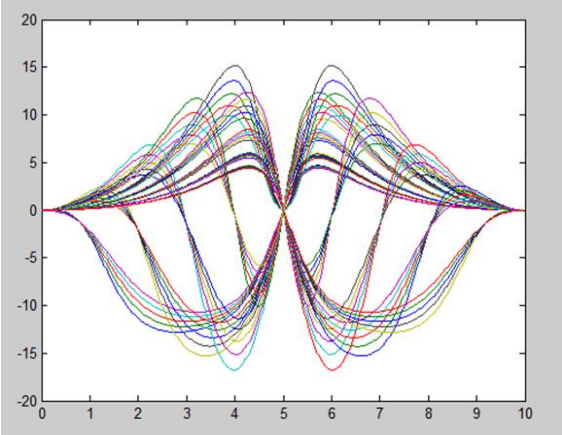


Figure 7: w_k functions for a flux centered at 5

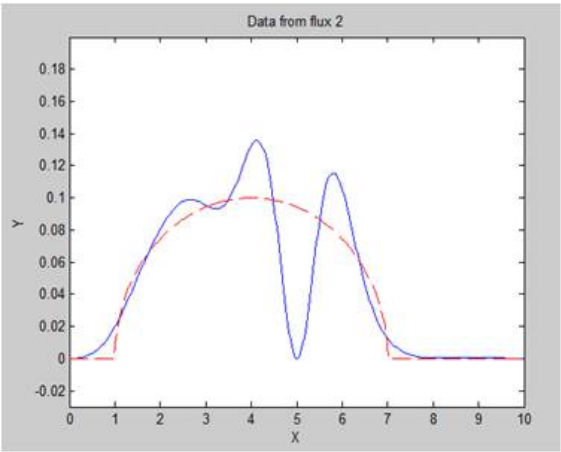


Figure 8: Corrosion profile using these w_k functions, with an error at 5

This problem was resolved by moving the input heat flux around the plate and meshing the different solutions together. We attempted to use a time-dependent flux in order to accomplish this, but the reconstructions from these data were not reliable. Therefore, we elected to use three separate time-independent heat fluxes, and combined the three solutions to get a final reconstruction. Each corrosion profile was tested with a heat flux centered at 2, 5, and 8. The three different $C(x)$ solutions were then meshed together using a set of conditional statements, ignoring each function in the interval of radius 2 around the center of the flux, with a smoothing region of 0.3. This method of reconstruction led to a smooth representation of the curve $C(x)$ that followed the actual profile fairly closely over the entire plate. For a further description of this, and the equation used to mesh the three sets of data, see Section D of the Appendix. Multiple images showing all three of the approximated corrosion profiles with different input fluxes, as well as the average curve, can be seen in Section 7.

This idea of moving the source of the heat flux around the plate could be incorporated into an actual physical experiment to improve the reconstruction. Three separate trials could be run with the flux at different locations on the plate. The only other condition that would need to be met is that the temperature of the plate needs to be uniform at the start of the trial. This could be accomplished either by waiting a defined amount of time to allow the temperature to equilibrate, or by flowing water over the surface of the plate to expedite the heat transfer. The three data sets could be gathered individually, and entered into the reconstruction program as was done here.

7 Results

Here we show numerical examples displaying the ability of this reconstruction method to match size, location, and complexity of a corrosion profile. These reconstructions were generated by setting up the corrosion profile geometry in a COMSOL file and obtaining data on the top boundary for the plate with corrosion. Another

COMSOL file was used to model a plate made of the same material that had not been corroded, and temperature data was collected from both the top and bottom boundaries of the uncorroded plate. If this procedure were to be implemented in the field, the data from the uncorroded plate would still be collected from a numerical solver. Since it does not involve any corrosion, the only information needed to generate this data is the type of metal and dimensions of the plate, which will be known in advance. On site, the only data that will need to be collected is temperature data on the accessible surface of the possibly corroded metal plate. These data sets were then imported into MATLAB and the approximation for $C(x)$ was generated by the methods explained in Section 5.

The thermal parameters and corrosion profiles displayed below were chosen because they sample a wide variety of scenarios for corrosion that may be studied in the field. Parameters α_2 and k_2 for the corroded region are set to be about an order of magnitude less than those of the uncorroded material to model the extra resistance to heat flow provided by a corroded region. The specific shape of the profile was constructed of elliptical arcs to simplify the procedure of comparing the reconstructed data to the actual profile. Some profiles are small and localized, while others are widespread over the plate, and a variety of heights are presented to see when the linearization assumption breaks down and how the reconstruction changes when this happens. The overall dimensions of the domain were chosen to approximate the width to depth ratio of a structural member that may be seen in an actual experiment. The α and k parameters were left at arbitrary values because they gave reasonable reconstructions, and this paper is a first attempt at developing a reconstruction method. The focus was thus on refining the solution and generating a viable reconstruction method for arbitrary parameters, which could then be set to physical values by future investigations. Also, as this was only a two-dimensional study, exploration into actual parameters for k and α did not seem necessary at this point.

The error between the reconstructed profile and the actual profile is presented in two ways. The first is an L^2 error, which is, if we define the actual corrosion profile as $a(x)$ and the reconstructed profile as $r(x)$, calculated by the formula

$$E_{L^2} = \left(\int_0^{10} |r(x) - a(x)|^2 dx \right)^{1/2} \approx \left(\sum_{i=0}^{N} |r(x_i) - a(x_i)|^2 \Delta x. \right)^{1/2}$$

The calculation is done numerically, but it approximates the analytical L^2 norm. The second error presented is the error in the maximum value of the profile. This would be important in determining whether a piece of metal is about to corrode through and deciding to replace part of the structure. This error is calculated by the formula

$$E_{max} = |\max_x r(x) - \max_x a(x)|.$$

The relative error for the reconstruction can be computed by dividing the L^2 error in reconstrution by the L^2 norm of the corrosion profile. This is computed by the formula

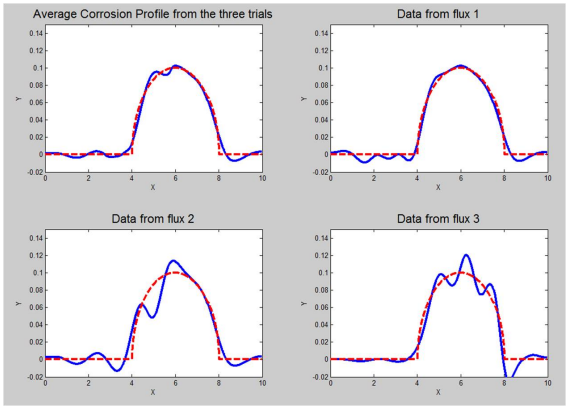
$$A = \left(\int_0^{10} |a(x)|^2 \, dx \right)^{1/2}$$

The L^2 norm (A) of the three types of corrosion shown in the paper can be seen in Table 1.

For all of the results, the solid line is the reconstructed profile, and the dashed line is the actual profile programmed into COMSOL. The graph in the top left is the average of the other three corrosion profiles, while the other graphs represent the individual reconstructions, with fluxes centered at 2, 5, and 8 respectively. For all of these reconstructions, the thermal parameters of material 1 are set to $k_1 = 1$ and $\alpha_1 = 1$, and the parameters for material 2 are specified under each graph. The regularization procedure and parameters of Section 6 were also used in generating these results.

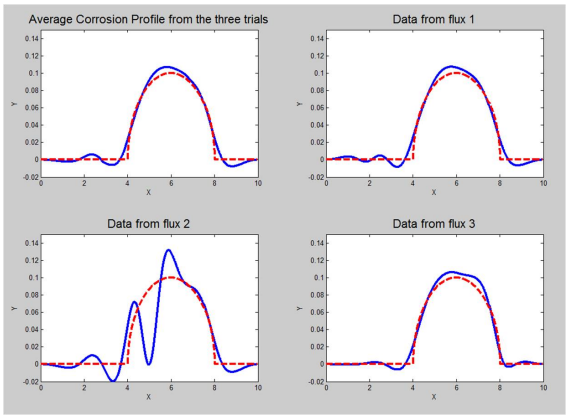
Corrosion Type	Figures	L^2 norm
Left Side, Height 0.05	(5, 22)	0.0817
Right Side, Height 0.1	(9 - 13)	0.1633
Multiple	(14 - 16)	0.7523

Table 1: L^2 norms for the different corrosion types



	L^2 Error
Flux at 2	0.0146
Flux at 5	0.0342
Flux at 8	0.0287
Meshed Data	0.0143
	Error in Max
Flux at 2	0.0022
Flux at 5	0.0138
Flux at 8	0.0204
Meshed Data	0.0024
Relative Error	8.757%

Figure 9: Reconstruction of corrosion profile with $k_2 = 0.1$, $\alpha_2 = 0.05$.



	L^2 Error
Flux at 2	0.0180
Flux at 5	0.0698
Flux at 8	0.0163
Meshed Data	0.0173
	Error in Max
Flux at 2	0.0072
Flux at 5	0.0318
Flux at 8	0.0062
Meshed Data	0.0069
Relative Error	10.594%

Figure 10: Reconstruction of corrosion profile with $k_2 = 0.1$, $\alpha_2 = 0.1$.

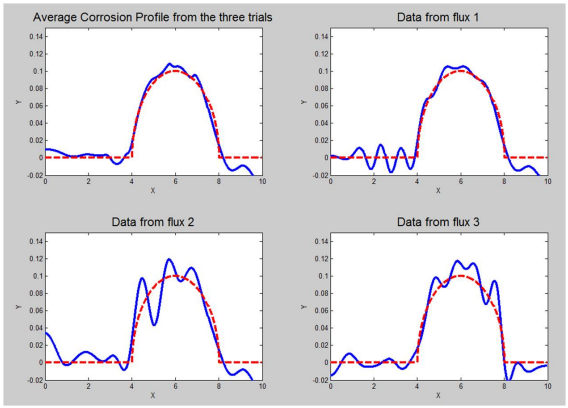


Figure 11: Reconstruction of corrosion profile with $k_2 = 0.1$, $\alpha_2 = 0.2$.

	L^2 Error
Flux at 2	0.0299
Flux at 5	0.0517
Flux at 8	0.0329
Meshed Data	0.0277
	Error in Max
Flux at 2	0.0053
Flux at 5	0.0192
Flux at 8	0.0172
Meshed Data	0.0084
Relative Error	16.963%

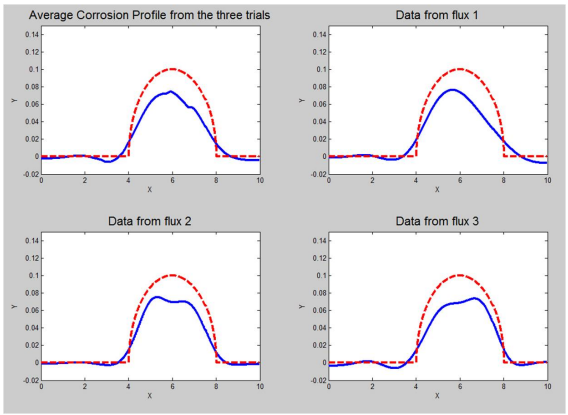


Figure 12: Reconstruction of corrosion profile with $k_2 = 1$, $\alpha_2 = 0.1$.

	L^2 Error
Flux at 2	0.0558
Flux at 5	0.0474
Flux at 8	0.0502
Meshed Data	0.0553
	Error in Max
Flux at 2	0.0236
Flux at 5	0.0248
Flux at 8	0.0264
Meshed Data	0.0257
Relative Error	33.864%

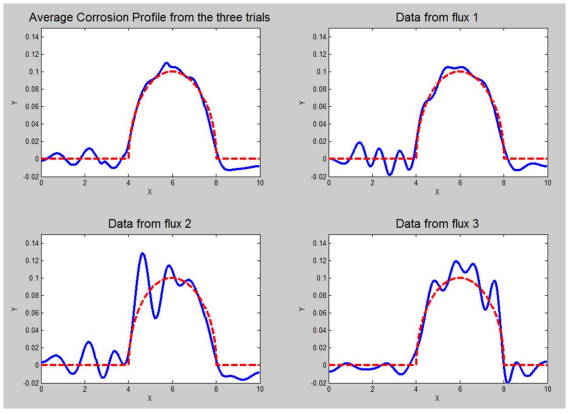


Figure 13: Reconstruction of corrosion profile with $k_2 = 0.1$, $\alpha_2 = 1$.

	L^2 Error
Flux at 2	0.0245
Flux at 5	0.0523
Flux at 8	0.0331
Meshed Data	0.0224
	Error in Max
Flux at 2	0.0051
Flux at 5	0.0283
Flux at 8	0.0192
Meshed Data	0.0101
Relative Error	13.717%

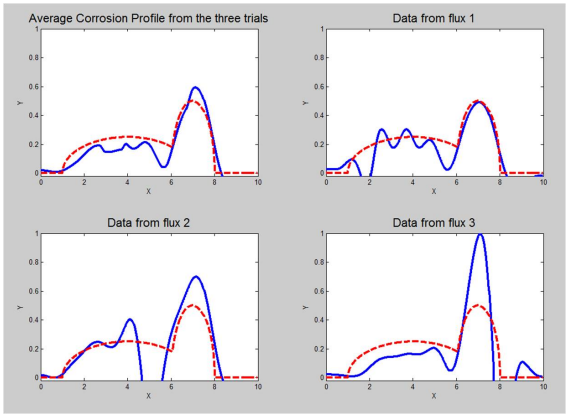


Figure 14: Reconstruction of large corrosion profile with $k_2 = 0.1$, $\alpha_2 = 0.05$.

	L^2 Error
Flux at 2	0.2661
Flux at 5	0.5674
Flux at 8	0.7015
Meshed Data	0.2244
	Error in Max
Flux at 2	0.0089
Flux at 5	0.1993
Flux at 8	0.4967
Meshed Data	0.0935
Relative Error	29.829%

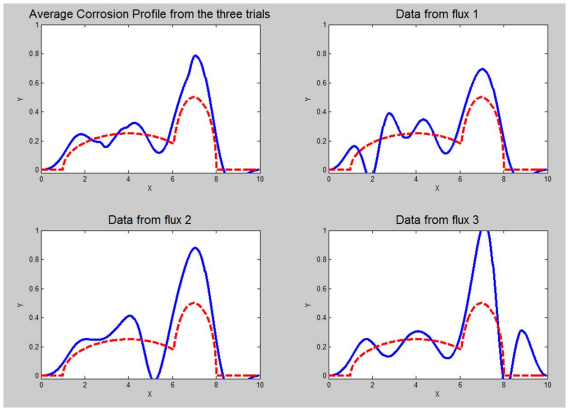


Figure 15: Reconstruction of large corrosion profile with $k_2 = 0.1$, $\alpha_2 = 0.1$.

	L^2 Error
Flux at 2	0.3236
Flux at 5	0.4952
Flux at 8	0.5505
Meshed Data	0.3169
	Error in Max
Flux at 2	0.1939
Flux at 5	0.3796
Flux at 8	0.5502
Meshed Data	0.2850
Relative Error	42.124%

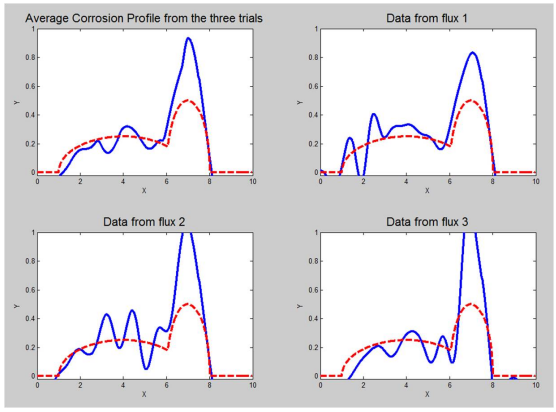


Figure 16: Reconstruction of large corrosion profile with $k_2 = 0.1$, $\alpha_2 = 0.2$.

	L^2 Error
Flux at 2	0.4654
Flux at 5	0.6440
Flux at 8	0.6412
Meshed Data	0.5203
	Error in Max
Flux at 2	0.3336
Flux at 5	0.5581
Flux at 8	0.7246
Meshed Data	0.4335
Relative Error	69.161%

Corrosion Type	k_2	α_2	L^2 Error	Relative L^2 Error
Right Side, Height 0.1	0.1	0.05	0.0143	8.757%
	0.1	0.1	0.0173	10.594%
	0.1	0.2	0.0277	16.963%
	1	0.1	0.0553	33.864%
	0.1	1	0.0224	13.717%
Multiple	0.1	0.05	0.2244	29.829%
	0.1	0.1	0.3169	42.124%
	0.1	0.2	0.5203	69.161%

Table 2: Summary table of L^2 error and relative error for all of the corrosion profiles

8 Conclusions/Discussion

As the results in Section 7 have shown, this reconstruction method allows for detection and fairly accurate imaging of small plate corrosion with varying material parameters. When the height of the corrosion is less than 10% of the height of the plate, the relative error of the reconstruction is less than 20% (ignoring the case where $k_1 = k_2$, which would not be seen in a physical system). As the corrosion size increases, the linearization assumptions are not as valid and the relative L^2 error grows significantly, but the reconstruction still finds the corrosion in the plate. However, in looking at those results, it can be seen that the reconstructed profile generally oscillates around the actual profile. Under some interpretation (which may be able to be done numerically), one could use these reconstructions to draw a very reasonable estimate to the corrosion in the plate. This could result in an easy way to reduce the L^2 error of most of these reconstructions to reasonable levels, and shows the potential for this method to be used to image unknown corrosion.

Another error caused by violating the linearization assumption is in the height of the corrosion profile. While the height of the reconstructed profile may be significantly different from that of the actual corrosion, the location is very accurate (as in Figure 15). Even though the overshoot can be around 25% of the actual corrosion profile height, this is not necessarily a bad thing for engineering applications, since it may cause replacement of a part of a structure well before material failure, as opposed to waiting until it is too late. Also, knowing the location of the corrosion accurately can be very helpful in attempting to address the cause of the corrosion, possibly allowing the new replacement part to last longer than the original.

The results also show, in a physical sense, what situations lead to improved imaging of corrosion. By looking at the three individual profiles that were averaged to create the final reconstruction, it can be seen that the individual profiles that are more effective at modeling the corrosion profile have heat source off to the side of the corrosion. This can be explained by analyzing the way in which heat diffuses. If the heat source is on the side of the corrosion, the heat is going to diffuse over the whole plate, and thus must flow towards the corrosion profile. When it hits the corrosion profile, it can not continue forward in the same way, since there is a change in thermal conductivity at the boundary. Heat flux is preserved at the boundary, but the change in thermal conductivity will induce a change in the surrounding temperature profile. Since we have assumed the corrosion profile is a function of x, the excess heat that needs to diffuse can only move one way: up the plate towards the top edge. As the heat continues to diffuse across the plate, the corrosion profile forces more of it up towards the top of the plate, where the data is gathered. Therefore, this type of trial will give a lot of information about the corrosion profile, leading to an improved reconstruction.

Another error that shows up in the reconstruction occurs when the corroded region is near the side of the plate. When this happens, the reconstruction often detects a small area of corrosion between the actual profile and the edge of the plate. This is most prevalent when the heat source is far away from the corrosion profile, and can also be explained in the same vein as the last paragraph. In this case, the heat is diffusing laterally across the corrosion profile, and is getting pushed up towards the top of the plate. When the

corrosion profile becomes smaller near the edge of the plate, the heat must diffuse down to the corrosion profile in order to properly image it. However, if this is very close to the edge, the heat is unable to diffuse to the bottom of the plate at the same rate that it would normally. Therefore, the reconstruction mechanism assumes there is something there stopping it from diffusing downward, and adds corrosion to the section. This error may be systematic and could be corrected for, but has not yet been fully analyzed.

9 Future Work

There are many possibilities for future work on this problem. One of the most applicable is the process of converting these calculations to the three dimensional problem. The mathematics for these higher-dimensional problems are almost identical to what has been presented here, but the computation time greatly increases for these problems. In attempting to run code that would generate test functions for a three-dimensional problem, it was discovered that the code would take approximately 30 days to run to completion, while the two-dimensional code could run in approximately an hour. A more efficient implementation may help with this issue, but more computing power would definitely be necessary for this to be done in a reasonable amount of time. An easier way to approach this problem might be to consider a cylindrical pipe or circular plate, as using the axisymmetric nature of the region may help to reduce and simplify calculations. The analysis of the full three dimensional problem would allow for this method to be used in engineering applications to visualize corrosion.

The two dimensional problem could also be explored in a variety of different ways from here. One way is with the material and experimental parameters. In the applied case, the metals will not have thermal properties of 1 and 0.1, and the heat source will not have an arbitrary flux of 100. As this was an initial study into the feasibility of this kind of reconstruction, these arbitrary parameters were chosen. Future work could consider an actual plate of steel 10 meters by 1 meter, and put the thermal conductivity and thermal diffusivity parameters for steel into COMSOL, add in a heat flux that has been calculated from devices that would be used to apply heat in the field, and see how the plate reacts. Investigations would lead to discovering what kinds of heat fluxes are actually necessary to test this procedure on physical systems, figuring out if this reconstruction can be done in an engineering sense, and learning how accurate these reconstructions could be on actual materials.

In extending this problem to physical applications, an important issue to be considered is the ill-posedness of the inverse problem (see Section 4) and how it relates to noise. If the inverse problem is highly dependent on the initial conditions and temperature data, a small amount of noise could cause a significant change in the reconstructed corrosion profile. Since noise and equipment precision errors are always going to be present in physical systems, this could completely invalidate this method as a possibility for detecting corrosion on the reverse side of a metal plate. Studies will need to be done on how the reconstruction algorithm responds to noise, either artificially generated noise or error from the precision of temperature-measurement devices, would need to be done before this procedure could be implemented for a physical system.

Some other analysis that can be done centers around the regularization process and reconstruction of the corrosion profile. There are many other regularization methods that could be implemented, as well as other ways to pick a unique corrosion profile as a solution to the system of integral equations. As shown in Section 6.1 and Section D of the Appendices, the regularization depends on the thermal parameters of the system. This issue could be investigated further by taking more data with a more varied set of thermal parameters, and see if some kind of function could be fit to predict the best regularization parameter ξ for a given set of thermal parameters. Additional refinement methods could also be added to this process to smooth out the reconstructed curve and attempting to extend the method beyond the linearization assumption to model corrosion of all sizes accurately. Other work could also be done in analyzing the situation discussed in Section 6.2 with the symmetry of the heat source. Nothing we have found attempts to address this issue, or

even encounters this problem. Our solution was to move the flux around the plate, but there may be many other solutions. Perturbation analysis could develop a different solution that would give more insight into the problem. Time-dependent fluxes are also an option; these would be easy to use in physical situations, should avoid the symmetry problem, and would allow the time-dependence of the heat equation to play a role in the reconstruction algorithm. All of these ideas would give a more reliable reconstruction of the corrosion profile, which could then be extended to the three dimensional case for applications.

References

[1] Court Hoang and Katherine Osenbach. *Electrical Impedance Imaging of Corrosion on a Partially Accessible 2-Dimensional Region: An Inverse Problem in Non-destructive Testing.* Rose-Hulman REU 2009

[2] Kurt Bryan and Lester Caudill. *Reconstuction of an unknown boundary portion from Cauchy data in* n *dimensions.* IOP Publishing, Inverse Problems **21** (2005) 239-255

[3] Olivier Poisson. *Uniqueness and Holder stability of discontinuous diffusion coefficients in three related inverse problems for the heat equation.* IOP Publishing, Inverse Problems **24** (2008) 025012

[4] Kurt Bryan and Lester Caudill Jr. *Uniqueness for a Boundary Identification Problem in Thermal Imaging.* Electronic Journal of Differential Equations, C-1, (1997), pp. 23-39

[5] A.N. Tikhonov and V.Y. Arsenin *Solutions of Ill-Posed Problems.* New York: Winston (1977)

[6] Gilbert Strang *Introduction to Linear Algebra* Wellesley-Cambridge Press. (1998)

[7] Walter A. Strauss *Partial Differential Equations: An Introduction* Wiley, New York (2008)

[8] Bruce R. Kusse and Erik A. Westwig *Mathematical Physics: Applied Mathematics for Scientists and Engineers* Wiley, New York (1998)

Appendices

A Computations and Combining Equations

This section will show all of the calculations used in Section 5.1 to simplify the problem. The two main techniques we are going to use in this process are Green's Identity and integration by parts.

Theorem A.1 (Green's Second Identity)**.** *For any bounded region $D \subset \mathbb{R}^2$ with piecewise smooth boundary ∂D, and any two functions $u, v \in C^2(\bar{D})$, we have*

$$\int_D (u\nabla^2 v - v\nabla^2 u)\ dA = \int_{\partial D} \left(u\frac{\partial v}{\partial \vec{n}} - v\frac{\partial u}{\partial \vec{n}} \right)\ ds.$$

Integration by parts allows us to take an integral of a product of two functions, and change which one is differentiated by a specific variable:

$$\int_a^b u\frac{\partial v}{\partial z}\ dz = uv|_a^b - \int_a^b \frac{\partial u}{\partial z} v\ dz.$$

As described in Section 5.1, we start with the equation

$$\int_0^T \int_{\Omega_1} u_1 \left(\frac{\partial \phi}{\partial t} + \alpha_1 \nabla^2 \phi \right)\ dA\ dt \quad = \quad 0$$

or

$$\int_0^T \int_{\Omega_1} u_1 \frac{\partial \phi}{\partial t}\ dA\ dt + \alpha_1 \int_0^T \int_{\Omega_1} u_1 \nabla^2 \phi\ dA\ dt \quad = \quad 0.$$

Integrating the first term by parts in time and using Green's Identity on the second term gives

$$\int_{\Omega_1} \left[u_1\phi|_0^T - \int_0^T \phi \frac{\partial u_1}{\partial t}\ dt \right]\ dA + \alpha_1 \int_0^T \int_{\Omega_1} \phi \nabla^2 u_1\ dA + \int_{\partial \Omega_1} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right)\ ds\ dt \quad = \quad 0$$

or

$$\int_{\Omega_1} u_1\phi|_0^T\ dA - \int_{\Omega_1} \int_0^T \phi \left(\frac{\partial u_1}{\partial t} - \alpha_1 \nabla^2 u_1 \right)\ dt\ dA + \alpha_1 \int_0^T \int_{\partial \Omega_1} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right)\ ds\ dt \quad = \quad 0.$$

The first two terms in the above expression are zero because u_1 solves the heat equation and vanishes at $t = 0$, while ϕ vanishes at $t = T$. Canceling the α_1 gives:

$$\int_0^T \int_{\partial \Omega_1} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right)\ ds\ dt = 0$$

or, factoring in the boundary of Ω_1 and the conditions on the functions there,

$$\int_0^T \int_{top} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right)\ ds\ dt + \int_0^T \int_{C(x)} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right)\ ds\ dt = 0 \tag{A.1}$$

where $C(x)$ is defined with the downward normal.

Similarly, we can look at the region Ω_2 and start with

$$\int_0^T \int_{\Omega_2} u_2 \frac{\partial \phi}{\partial t}\ dA\ dt + \alpha_1 \int_0^T \int_{\Omega_2} u_2 \nabla^2 \phi\ dA\ dt \quad = \quad 0.$$

Integrating the first term by parts in time and using Green's Identity on the second term gives

$$\int_{\Omega_2}\left[u_2\phi|_0^T-\int_0^T\phi\frac{\partial u_2}{\partial t}\,dt\right]\,dA+\alpha_1\int_0^T\left[\int_{\Omega_2}\phi\nabla^2u_2\,dA+\int_{\partial\Omega_2}\left(u_2\frac{\partial\phi}{\partial\vec{n}}-\phi\frac{\partial u_2}{\partial\vec{n}}\right)\,ds\right]\,dt=0.$$

The first term is zero because ϕ and u_2 vanish at the endpoints in time. Since u_2 solves the heat equation in Ω_2, we know that

$$\frac{\partial u_2}{\partial t}=\alpha_2\nabla^2u_2.$$

Plugging this in above gives

$$\begin{aligned}
-\alpha_2\int_{\Omega_2}\int_0^T\phi\nabla^2u_2\,dt\,dA+\alpha_1\int_0^T\int_{\Omega_2}\phi\nabla^2u_2\,dA\,dt+\alpha_1\int_0^T\int_{\partial\Omega_2}\left(u_2\frac{\partial\phi}{\partial\vec{n}}-\phi\frac{\partial u_2}{\partial\vec{n}}\right)\,ds\,dt &= 0\\
(\alpha_1-\alpha_2)\int_0^T\int_{\Omega_2}\phi\nabla^2u_2\,dA\,dt+\alpha_1\int_0^T\int_{\partial\Omega_2}\left(u_2\frac{\partial\phi}{\partial\vec{n}}-\phi\frac{\partial u_2}{\partial\vec{n}}\right)\,ds\,dt &= 0\\
\frac{\alpha_1-\alpha_2}{\alpha_1}\int_0^T\int_{\Omega_2}\phi\nabla^2u_2\,dA\,dt+\int_0^T\int_{\partial\Omega_2}\left(u_2\frac{\partial\phi}{\partial\vec{n}}-\phi\frac{\partial u_2}{\partial\vec{n}}\right)\,ds\,dt &= 0.
\end{aligned}$$

Taking the boundary of Ω_2 into consideration gives

$$\frac{\alpha_1-\alpha_2}{\alpha_1}\int_0^T\int_{\Omega_2}\phi\nabla^2u_2\,dA\,dt+\int_0^T\int_{C(x)}\left(u_2\frac{\partial\phi}{\partial\vec{n}}-\phi\frac{\partial u_2}{\partial\vec{n}}\right)\,ds\,dt+\int_0^T\int_{bottom}\left(u_2\frac{\partial\phi}{\partial\vec{n}}-\phi\frac{\partial u_2}{\partial\vec{n}}\right)\,ds\,dt=0 \tag{A.2}$$

where $C(x)$ has the upward normal. $C(x)$ has two different normals in these two equations because Green's Identity treats the boundary of the region as a positively oriented curve with outward normal. Therefore, as seen in Figure 17, the integration with Green's identity around Ω_1 results in (A.1), traversing the $C(x)$ curve from left to right, with the normal pointing downward. Similarly, integrating on Ω_2 results in (A.2), traversing $C(x)$ from right to left, but with the normal pointing upward.

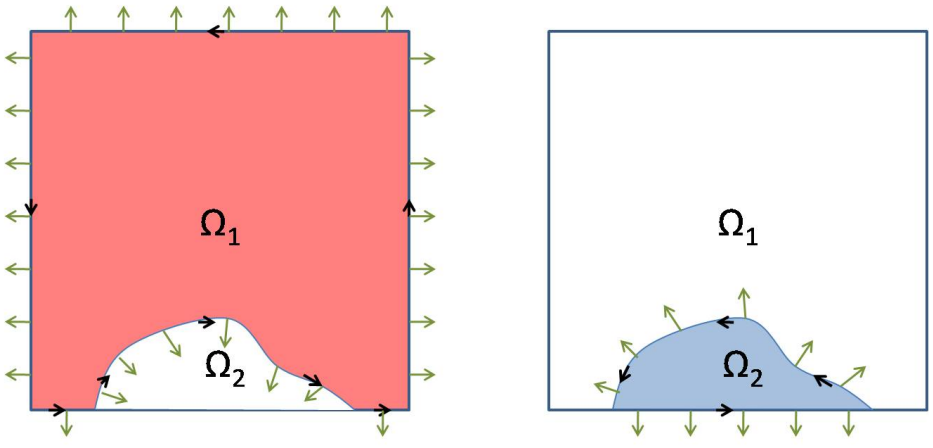


Figure 17: Illustration of the different normals generated by Green's Identity

Since we are eventually going to combine these equations, the two integrals over $C(x)$ need to refer to the same path along $C(x)$. The integral with respect to ds is independent of direction, and reversing the direction of the normal introduces a negative sign to the integral. Therefore, we need to change the sign of the integral

over $C(x)$ in (A.2) so that the two integrals over $C(x)$ can be added together. Adding this negative sign and canceling the last term since both functions have zero normal derivative at $y = 0$ gives:

$$\frac{\alpha_1 - \alpha_2}{\alpha_1} \int_0^T \int_{\Omega_2} \phi \nabla^2 u_2 \; dA \; dt - \int_0^T \int_{C(x)} \left(u_2 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_2}{\partial \vec{n}} \right) \; ds \; dt = 0 \tag{A.3}$$

where $C(x)$ is defined with the downward normal.

We now seek to combine equations (A.1) and (A.3) to generate a relation that will allow us to approximate the function $C(x)$. We will use the temperature profile u_2 in Ω_2 and the continuity conditions on $C(x)$ to do this. As described in the body of the paper, the continuity conditions tell us that along the curve $C(x)$,

$$\begin{aligned} u_1 &= u_2 \\ k_1 \frac{\partial u_1}{\partial \vec{n}} &= k_2 \frac{\partial u_2}{\partial \vec{n}}. \end{aligned}$$

To start, modify equation (A.1) using the continuity conditions on $C(x)$, to give

$$\int_0^T \int_{top} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right) \; ds \; dt + \int_0^T \int_{C(x)} \left(u_2 \frac{\partial \phi}{\partial \vec{n}} - \frac{k_2}{k_1} \phi \frac{\partial u_2}{\partial \vec{n}} \right) \; ds \; dt = 0. \tag{A.4}$$

Then, adding equations (A.3) and (A.4) gives

$$\int_0^T \int_{top} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right) \; ds \; dt + \left(1 - \frac{k_2}{k_1} \right) \int_0^T \int_{C(x)} \phi \frac{\partial u_2}{\partial \vec{n}} \; ds \; dt + \frac{\alpha_1 - \alpha_2}{\alpha_1} \int_0^T \int_{\Omega_2} \phi \nabla^2 u_2 \; dA \; dt = 0.$$

Defining

$$RG(\phi) = \int_0^T \int_{top} \left(u_1 \frac{\partial \phi}{\partial \vec{n}} - \phi \frac{\partial u_1}{\partial \vec{n}} \right) \; ds \; dt \tag{A.5}$$

and rearranging gives

$$RG(\phi) = \left(\frac{k_2}{k_1} - 1 \right) \int_0^T \int_{C(x)} \phi \frac{\partial u_2}{\partial \vec{n}} \; ds \; dt + \frac{\alpha_2 - \alpha_1}{\alpha_1} \int_0^T \int_{\Omega_2} \phi \nabla^2 u_2 \; dA \; dt.$$

Knowing that u_2 solves the heat equation on Ω_2 and that on $C(x)$

$$\vec{n} = \frac{< C'(x), -1 >}{\sqrt{C'(x)^2 + 1}} \qquad \text{and} \qquad ds = \sqrt{C'(x)^2 + 1}$$

this expression can be written as

$$RG(\phi) = \left(\frac{k_2}{k_1} - 1 \right) \int_0^T \int_0^L C'(x) \phi \frac{\partial u_2}{\partial x} \bigg|_{C(x)} - \phi \frac{\partial u_2}{\partial y} \bigg|_{C(x)} dx \; dt + \frac{\alpha_2 - \alpha_1}{\alpha_1 \alpha_2} \int_0^T \int_{\Omega_2} \phi \frac{\partial u_2}{\partial t} \; dA \; dt \tag{A.6}$$

which is the fully simplified non-linear problem.

B Linearization

We now look to linearize the problem to find a way to numerically solve it. We assume that the corrosion profile, $C(x)$, is small, or

$$C(x) = \epsilon C_0(x)$$

where ϵ is a small positive constant, and $C_0(x)$ is an order 1 function.

The last term in equation (5.1) or (A.6) may be written as

$$\frac{\alpha_2-\alpha_1}{\alpha_1\alpha_2}\int_0^T\int_0^L\int_0^{C(x)}\phi\frac{\partial u_2}{\partial t}\;dy\;dx\;dt.$$

If $C(x)$ is small, we can approximate the function $\phi\frac{\partial u_2}{\partial t}$ by a constant over the innermost integral in y. This term can then be written as approximately equal to

$$\frac{\alpha_2-\alpha_1}{\alpha_1\alpha_2}\int_0^T\int_0^L C(x)\;\phi\frac{\partial u_2}{\partial t}\bigg|_{y=0}dx\;dt$$

where we choose to evaluate the functions at $y=0$.

We also want to linearize the parts of the first integral in (A.6) about the line $y=0$, ignoring all terms of $O(\epsilon^2)$. In linearizing these terms, we will use a power series expansion around $y=0$, namely

$$f|_{y=\gamma}=f|_{y=0}+\gamma\;\frac{\partial f}{\partial y}\bigg|_{y=0}+\gamma^2\;\frac{\partial^2 f}{\partial y^2}\bigg|_{y=0}+\cdots,$$

and we will ignore terms of higher orders of ϵ that will vanish because of the linearization assumption. Looking at the terms in (A.6), we have

$$\begin{aligned}
C'(x)\;\phi\frac{\partial u_2}{\partial x}\bigg|_{C(x)} &= C'(x)\;\phi\frac{\partial u_2}{\partial x}\bigg|_{y=0}+C(x)C'(x)\;\frac{\partial\phi}{\partial y}\frac{\partial u_2}{\partial x}\bigg|_{y=0}+C(x)C'(x)\;\phi\frac{\partial^2 u_2}{\partial x\partial y}\bigg|_{y=0}\\
&= C'(x)\;\phi\frac{\partial u_2}{\partial x}\bigg|_{y=0}+O(\epsilon^2)\\
\phi\frac{\partial u_2}{\partial y}\bigg|_{C(x)} &= \phi\frac{\partial u_2}{\partial y}\bigg|_{y=0}+C(x)\;\frac{\partial\phi}{\partial y}\frac{\partial u_2}{\partial y}\bigg|_{y=0}+C(x)\;\phi\frac{\partial^2 u_2}{\partial y^2}\bigg|_{y=0}+O(\epsilon^2)\\
&= 0+0+C(x)\;\phi\frac{\partial^2 u_2}{\partial y^2}\bigg|_{y=0}+O(\epsilon^2)
\end{aligned}$$

Plugging these terms into equation (A.6) gives

$$RG(\phi)=\left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L C'(x)\;\phi\frac{\partial u_2}{\partial x}\bigg|_{y=0}-C(x)\;\phi\frac{\partial^2 u_2}{\partial y^2}\bigg|_{y=0}dx\;dt+\frac{\alpha_2-\alpha_1}{\alpha_1\alpha_2}\int_0^T\int_0^L C(x)\;\phi\frac{\partial u_2}{\partial t}\bigg|_{y=0}dx\;dt. \tag{B.1}$$

Integrating the first term in equation (B.1) by parts in x (integrating $C'(x)$) gives

$$\begin{aligned}
RG(\phi) &= \left(\frac{k_2}{k_1}-1\right)\int_0^T\left[C(x)\phi\frac{\partial u_2}{\partial x}\bigg|_{x=0}^{x=L}+\int_0^L -C(x)\;\frac{\partial\phi}{\partial x}\frac{\partial u_2}{\partial x}\bigg|_{y=0}-C(x)\;\phi\frac{\partial^2 u_2}{\partial x^2}\bigg|_{y=0}-C(x)\;\phi\frac{\partial^2 u_2}{\partial y^2}\bigg|_{y=0}dx\right]dt\\
&\quad+\frac{\alpha_2-\alpha_1}{\alpha_1\alpha_2}\int_0^T\int_0^L C(x)\;\phi\frac{\partial u_2}{\partial t}\bigg|_{y=0}dx\;dt
\end{aligned}$$

where the first term is zero because $\frac{\partial u_2}{\partial x}=0$ at both sides of the rectangle. Combining the two second derivatives into a Laplacian and separating the first term of the integral gives

$$\begin{aligned}
RG(\phi) &= \left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L -C(x)\;\frac{\partial\phi}{\partial x}\frac{\partial u_2}{\partial x}\bigg|_{y=0}dx\;dt-\left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L C(x)\;\phi\nabla^2 u_2\big|_{y=0}\;dx\;dt\\
&\quad+\frac{\alpha_2-\alpha_1}{\alpha_1\alpha_2}\int_0^T\int_0^L C(x)\;\phi\frac{\partial u_2}{\partial t}\bigg|_{y=0}dx\;dt.
\end{aligned}$$

Converting the second term to a time derivative by the heat equation and reorganizing some terms gives

$$\begin{aligned} RG(\phi) &= \left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L -C(x)\,\frac{\partial\phi}{\partial x}\frac{\partial u_2}{\partial x}\bigg|_{y=0} dx\,dt - \left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L C(x)\,\frac{\phi}{\alpha_2}\frac{\partial u_2}{\partial t}\bigg|_{y=0} dx\,dt \\ &\quad + \frac{\alpha_2-\alpha_1}{\alpha_1}\int_0^T\int_0^L C(x)\,\frac{\phi}{\alpha_2}\frac{\partial u_2}{\partial t}\bigg|_{y=0} dx\,dt \end{aligned}$$

or

$$RG(\phi) = \left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L -C(x)\,\frac{\partial\phi}{\partial x}\frac{\partial u_2}{\partial x}\bigg|_{y=0} dx\,dt - \left[\left(\frac{k_2}{k_1}-1\right) - \frac{\alpha_2-\alpha_1}{\alpha_1}\right]\int_0^T\int_0^L C(x)\,\frac{\phi}{\alpha_2}\frac{\partial u_2}{\partial t}\bigg|_{y=0} dx\,dt.$$

Simplifying the coefficient of the second term gives

$$\begin{aligned} \left(\frac{k_2}{k_1}-1\right) - \frac{\alpha_2-\alpha_1}{\alpha_1} &= \frac{k_2}{k_1} - 1 - \frac{\alpha_2}{\alpha_1} + 1 \\ &= \frac{k_2}{k_1} - \frac{\alpha_1}{\alpha_2} \end{aligned}$$

Thus, our expression becomes

$$RG(\phi) = \left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L -C(x)\,\frac{\partial\phi}{\partial x}\frac{\partial u_2}{\partial x}\bigg|_{y=0} dx\,dt - \left(\frac{k_2}{k_1}-\frac{\alpha_2}{\alpha_1}\right)\int_0^T\int_0^L C(x)\,\frac{\phi}{\alpha_2}\frac{\partial u_2}{\partial t}\bigg|_{y=0} dx\,dt. \quad \text{(B.2)}$$

Now, making the assumption that the temperature profile u_1 is close to the uncorroded temperature profile u_0, which is reasonable in the case that $C(x)$ is small, we get

$$u_1 = u_0 + \epsilon\tilde{u_1}.$$

We evaluate this expression on the curve $C(x)$ to obtain

$$u_2|_{C(x)} = u_1|_{C(x)} = u_0|_{C(x)} + \epsilon\ \tilde{u_1}|_{C(x)}\,.$$

Linearizing the far left and far right about $y = 0$ gives

$$u_2|_{y=0} + C(x)\,\frac{\partial u_2}{\partial y}\bigg|_{y=0} + O(\epsilon^2) = u_0|_{y=0} + C(x)\,\frac{\partial u_0}{\partial y}\bigg|_{y=0} + O(\epsilon^2) + \epsilon\ \tilde{u_1}|_{y=0} + O(\epsilon^2).$$

Since

$$\frac{\partial u_2}{\partial y}\bigg|_{y=0} = \frac{\partial u_0}{\partial y}\bigg|_{y=0} = 0$$

we are left with

$$u_2|_{y=0} = u_0|_{y=0} + O(\epsilon).$$

Integrating the last term of equation (B.2) by parts in time will generate a set of functions which can be used to numerically solve for the function $C(x)$.

$$\begin{aligned} RG(\phi) &= \left(\frac{k_2}{k_1}-1\right)\int_0^T\int_0^L -C(x)\,\frac{\partial\phi}{\partial x}\frac{\partial u_0}{\partial x}\bigg|_{y=0} dx\,dt - \left(\frac{k_2}{k_1}-\frac{\alpha_2}{\alpha_1}\right)\left[u_0\phi|_0^T - \int_0^T\int_0^L C(x)\,\frac{u_0}{\alpha_2}\frac{\partial\phi}{\partial t}\bigg|_{y=0} dx\,dt\right] \\ &= \left(1-\frac{k_2}{k_1}\right)\int_0^T\int_0^L C(x)\,\frac{\partial\phi}{\partial x}\frac{\partial u_0}{\partial x}\bigg|_{y=0} dx\,dt + \left(\frac{k_2}{k_1}-\frac{\alpha_2}{\alpha_1}\right)\int_0^T\int_0^L C(x)\,\frac{u_0}{\alpha_2}\frac{\partial\phi}{\partial t}\bigg|_{y=0} dx\,dt \\ &= \int_0^L C(x)\left[\int_0^T \left(1-\frac{k_2}{k_1}\right)\frac{\partial\phi}{\partial x}\frac{\partial u_0}{\partial x}\bigg|_{y=0} + \left(\frac{k_2}{k_1}-\frac{\alpha_2}{\alpha_1}\right)\frac{u_0}{\alpha_2}\frac{\partial\phi}{\partial t}\bigg|_{y=0} dt\right] dx \end{aligned}$$

and, defining

$$w_k(x) := \int_0^T \left(1 - \frac{k_2}{k_1}\right) \frac{\partial \phi_k}{\partial x} \frac{\partial u_0}{\partial x}\bigg|_{y=0} + \left(\frac{k_2}{k_1} - \frac{\alpha_2}{\alpha_1}\right) \frac{u_0}{\alpha_2} \frac{\partial \phi_k}{\partial t}\bigg|_{y=0} dt$$

we are looking for solutions to the system of integral equations,

$$RG(\phi_k) = \int_0^L C(x) w_k(x) dx \tag{B.3}$$

with $1 \leq k \leq M$.

C Generation of Test Functions

As described in Section 5, we want to numerically generate test functions ϕ_k such that

$$\begin{aligned} \frac{\partial \phi_k}{\partial t} + \alpha_1 \nabla^2 \phi_k &= 0 \text{ on } \Omega \\ \frac{\partial \phi_k}{\partial \vec{n}} &= 0 \text{ on } y = 0,\ x = 0, \text{ and } x = L \\ \phi_k(x, y, T) &= 0 \text{ on } \Omega \end{aligned}$$

In order to do this, we look for a related function ψ_k, which satisfies

$$\begin{aligned} \frac{\partial \psi_k}{\partial t} - \alpha_1 \nabla^2 \psi_k &= 0 \text{ on } \Omega \\ \frac{\partial \psi_k}{\partial \vec{n}} &= 0 \text{ on } y = 0,\ x = 0, \text{ and } x = L \\ \psi_k(x, y, 0) &= 0 \text{ on } \Omega \end{aligned}$$

Then, letting $\phi_k(x, y, t) = \psi_k(x, y, T - t)$ will give the desired functions ϕ_k. In order to find ψ, we will utilize the Green's Function for heat, and model a situation with point heat sources.

The Green's Function for heat is

$$K_{(x_0, y_0)}(x, y, t) = \frac{1}{4\pi\alpha t} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{4\alpha t}},$$

which satisfies

$$\begin{aligned} \frac{\partial K}{\partial t} - \alpha \nabla^2 K &= 0 \text{ on } \mathbb{R}^2 \\ K(x, y, 0) &= \delta(x - x_0, y - y_0) \end{aligned}$$

where δ is the Kronecker delta function. The above differential equation and initial condition is the definition of a Green's Function for the heat equation. As long as the point (x_0, y_0) is outside of our domain Ω, this function K will satisfy the heat equation in Ω with zero initial condition. We will be using this to build up the desired functions ψ. This function models the spread of a point impulse heat source at the point (x_0, y_0).

By Duhamel's Principle, and by carrying out the calculations, it can be shown that

$$\int_0^t K_{(x_0, y_0)}(x, y, \tau)\ d\tau$$

is also a solution to the heat equation with zero initial condition. It models the spread of a constant heat source at the point (x_0, y_0). Since the initial condition is zero, any sum of these integrals will also solve the heat equation with zero initial condition.

In order to specify the zero Neumann data conditions for ψ, we will use the method of images [7] [8]. The idea is if we make the array of point sources symmetric around some line, then heat has no reason to flow across the line, resulting in zero normal derivative on that line. We will take an initial point (x_0, y_0), and reflect it over $x = 0$ to get $(x_0, -y_0)$. These two point sources together form a temperature profile with zero Neumann data at $x = 0$. Since we also want zero Neumann Data at $x = L$, we reflect these two points over that line, but to maintain symmetry about $x = 0$, we also reflect over $x = -L$. Repeating this process gives a set of points of the form $(x_0, 2jL - y_0)$ and $(x_0, 2jL + y_0)$ for $j \in \{-N, ..., N\}$. This is demonstrated in Figure 18.

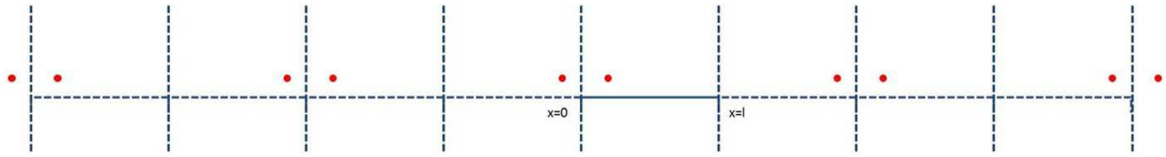


Figure 18: Reflections in the x-direction for the method of images

This set of points is not completely symmetric about the line $x = L$, but the variance from symmetry is 2 point sources more than NL units away. Since the Green's function for the heat equation drops off as distance squared, these points have almost no effect on the heat flux at $x = L$, so we can take that flux to be approximately zero. The equation describing the temperature profile in this situation is

$$\sum_{j=-N}^{N} \int_0^t K_{(2jl+x_0,y_0)}(x, y, \tau) + K_{(2kl-x_0,y_0)}(x, y, \tau)\ d\tau.$$

Finally, to ensure zero Neumann data at $y = 0$, we reflect all of these points over the line $y = 0$, giving a diagram similar to that in Figure 19.

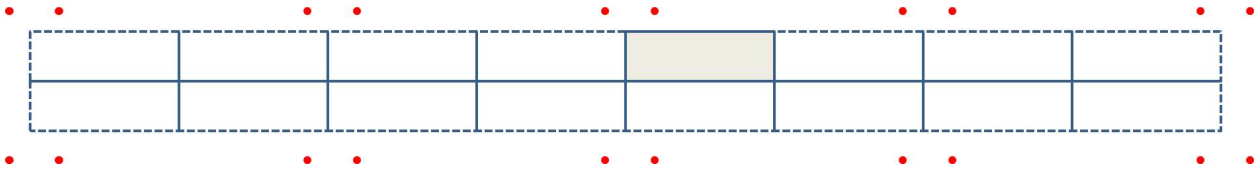


Figure 19: Full set of reflections for the method of images

The equation describing the temperature profile given these point sources, which satisfies the heat equation, zero initial conditions, and the desired Neumann boundary conditions is

$$\psi(x, y, t) = \sum_{j=-N}^{N} \int_0^t K_{(2jl+x_0,y_0)}(x, y, \tau) + K_{(2kl-x_0,y_0)}(x, y, \tau) + K_{(2jl+x_0,-y_0)}(x, y, \tau) + K_{(2kl-x_0,-y_0)}(x, y, \tau)\ d\tau.$$

Defining $\phi(x, y, t) = \psi(x, y, T-t)$ satisfies all the desired conditions. The necessary derivatives were computed analytically and then used in the MATLAB code to create the numerical ϕ functions. In order to generate the set of functions used in the calculations, we defined the set of points for the first heat source, $(x_0, y_0)_k$

where $x_0 \in \{1, ..., 9\}$ and $y_0 \in \{1.1, ..., 1.5\}$. Working out the derivatives for ϕ gives:

$$\begin{aligned}
\phi_{x_0,y_0} &= \int_0^{T-t} \frac{1}{4\pi\alpha\tau}\left(e^{-\frac{(y-y_0)^2}{4\alpha\tau}} + e^{-\frac{(y+y_0)^2}{4\alpha\tau}}\right)\sum_{j=-N}^{N}\left(e^{-\frac{(x-(2jL+x_0))^2}{4\alpha\tau}} + e^{-\frac{(x-(2jL-x_0))^2}{4\alpha\tau}}\right)\\
\left.\frac{\partial\phi}{\partial t}\right|_{y=0} &= \frac{-1}{2\pi\alpha(T-t)}e^{\frac{y_0^2}{4\alpha(T-t)}}\sum_{j=-N}^{N}\left(e^{-\frac{(x-(2jL+x_0))^2}{4\alpha(T-t)}} + e^{-\frac{(x-(2jL-x_0))^2}{4\alpha(T-t)}}\right)\\
\left.\frac{\partial\phi}{\partial x}\right|_{y=0} &= \frac{-e^{\frac{-y_0^2}{4\alpha(T-t)}}}{\pi\alpha}\sum_{j=-N}^{N}\left(\frac{x-(2jL+x_0)}{(x-(2jL+x_0))^2+y_0^2}e^{-\frac{(x-(2jL+x_0))^2}{4\alpha(T-t)}} + \frac{x-(2jL-x_0)}{(x-(2jL-x_0))^2+y_0^2}e^{-\frac{(x-(2jL-x_0))^2}{4\alpha(T-t)}}\right)\\
\left.\frac{\partial\phi}{\partial y}\right|_{y=1} &= \frac{-1}{2\pi\alpha}\sum_{j=-N}^{N}\left[(1-y_0)e^{-\frac{(1-y_0)^2}{4\alpha(T-t)}}\left(\frac{e^{-\frac{(x-(2jL+x_0))^2}{4\alpha(T-t)}}}{(x-(2kL+x_0))^2+(1-y_0)^2} + \frac{e^{-\frac{(x-(2jL-x_0))^2}{4\alpha(T-t)}}}{(x-(2kL-x_0))^2+(1-y_0)^2}\right)\right.\\
&\quad \left.+(1+y_0)e^{-\frac{(1+y_0)^2}{4\alpha(T-t)}}\left(\frac{e^{-\frac{(x-(2jL+x_0))^2}{4\alpha(T-t)}}}{(x-(2kL+x_0))^2+(1+y_0)^2} + \frac{e^{-\frac{(x-(2jL-x_0))^2}{4\alpha(T-t)}}}{(x-(2kL-x_0))^2+(1+y_0)^2}\right)\right]
\end{aligned}$$

D Meshing Solution Data and Regularization

In Section 6, we discuss two methods that we used to refine our solution and get a reconstruction of the corrosion profile that could actually be used to understand how corrosion was affecting the metal plate. First, in Section 6.1 we discuss choosing regularization parameters that could be used to adjust the solution via a singular value decomposition. Figures 20 and 21 show two more plots of the error in L^2 norm and maximum value as a function of ϵ_∞, with references back to the corrosion profile that they represent in the results section.

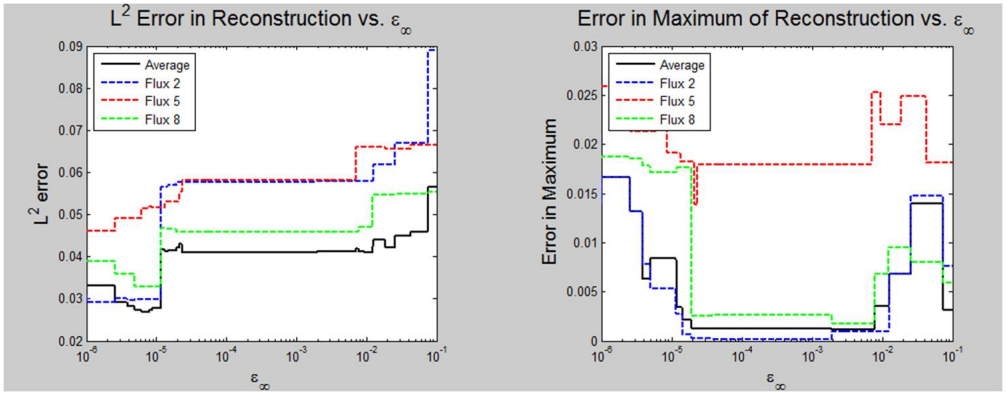


Figure 20: Error in L^2 norm and maximum value for Figure 11.

As seen in the different profiles, the value for ϵ_∞ that minimizes the L^2 error is not consistent between the graphs presented. Therefore, more analysis was done into the choice of the ϵ_∞ parameter. Table 3 contains all of the different corrosion profiles and thermal parameters studied in this paper, along with the approximate value or range of values for ϵ_∞ that would lead to the minimum error in L^2 norm of the reconstruction.

As can be seen in Table 3, the value for ϵ_∞ that resulted in the least L^2 norm was not consistent across the trials. However, it was generally consistent for a given set of thermal parameters across a multiple corrosion profiles. Therefore, instead of using the ϵ_∞ as suggested in [1], we will use a different formula for ϵ_d as

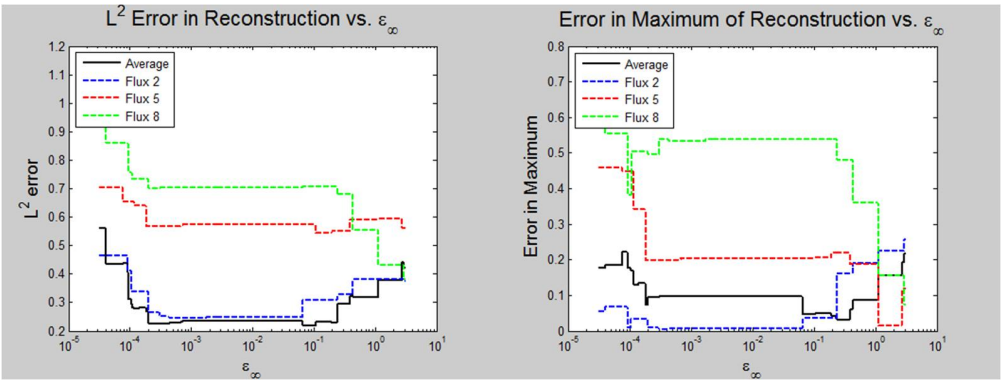


Figure 21: Error in L^2 norm and maximum value for Figure 14.

Corrosion Type	(Example)	k_2	α_2	ϵ_∞
Left, Height 0.05	(Figure 5)	0.1	0.1	10^{-4}
Right, Height 0.1	(Figure 9)	0.1	0.05	$2*10^{-4}-3*10^{-4}$
		0.1	0.1	$4*10^{-4}-3*10^{-3}$
		0.1	0.2	10^{-5}
		0.1	1.0	$4*10^{-6}-5*10^{-6}$
		1.0	0.1	$10^{-3}-10^{-2}$
Multiple	(Figure 14)	0.1	0.05	$2*10^{-4}-4*10^{-4}$
		0.1	0.1	$2*10^{-2}$
		0.1	0.2	10^{-5}

Table 3: Values of the ϵ_∞ parameter that give the minimum L^2 error in reconstruction

opposed to (6.1). We will define

$$\epsilon_d = \sqrt{M} h_w C_{max} \xi(k, \alpha) \tag{D.1}$$

where ξ incorporates the ϵ_{thresh} and ϵ_∞ terms and accounts for the dependence of the regularization on the thermal parameters of the system. More data will need to be collected in order to explicitly determine this dependence, so for this paper, we have assumed various values for this ξ parameter. Table 4 lists the different values we have chosen for each set of thermal parameters of region Ω_2 in the report.

k_2	α_2	ϵ_∞	$\xi(k,\alpha)$
0.1	0.05	$3*10^{-4}$	$1.5*10^{-3}$
0.1	0.1	10^{-3}	$5*10^{-3}$
0.1	0.2	10^{-5}	$5*10^{-5}$
0.1	1.0	$4*10^{-6}$	$2*10^{-5}$
1.0	0.1	$5*10^{-3}$	$2.5*10^{-2}$

Table 4: List of the ξ values used in the regularization

Second, in Section 6.2, we present the idea of meshing the reconstruction from three distinct heat fluxes in order to more accurately model the corrosion in the plate. This section will show more detail in that concept. Although we originally decided to move the input flux to avoid the symmetry issue, it would help to give a better approximation to the corrosion in any case. Averaging different results to the reconstruction adds more data to the system, and due to the numerical nature of the reconstruction, more data will always be helpful in approximating a wide range of possible corrosion profiles. As presented in Figure 22, averaging the

three profiles helps reduce the L^2 error between the approximated profile and the actual corrosion profile. More examples of these data can be seen in the results section of the paper.

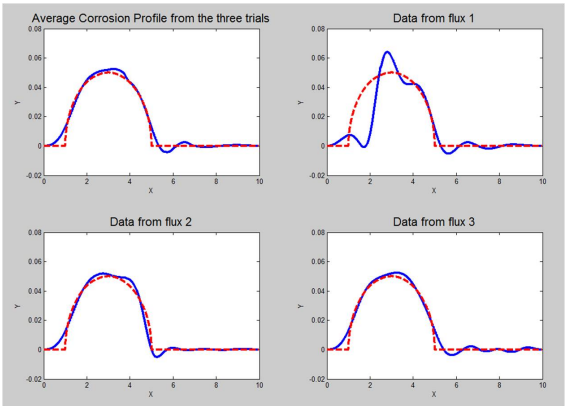


Figure 22: Three different reconstruction profiles and the average.

	L^2 Error
Flux at 2	0.0351
Flux at 5	0.0094
Flux at 8	0.0089
Meshed Data	0.0084
	Error in Max
Flux at 2	0.0142
Flux at 5	0.0018
Flux at 8	0.0024
Meshed Data	0.0024
Relative Error	10.282%

The actual function used to mesh the solutions together is (D.2). It is a piecewise defined function that averages any of the solutions that are more than 2 unit away in the x-direction, and contains an exponential smoothing function to connect the different regions. In the definition below f_2, f_5, and f_8 refer to the reconstruction function from the heat flux centered at 2, 5, and 8 respectively.

The definition uses two parameters: s, which determines the width of the smoothing region, and v, which determines how quickly the exponential function goes to 0 or 1 in the smoothing region.

$$r(x) = \begin{cases} \frac{1}{2}(f_5 + f_8) & 0 < x < 3 - s \\ \frac{1}{2}\left(f_5 e^{-v*(x-(3-s))^2} + \left(2 - e^{-v*(x-(3-s))^2}\right) f_8\right) & 3 - s < x < 3 + s \\ f_8 & 3 + s < x < 4 - s \\ \frac{1}{2}\left(\left(1 - e^{-v*(x-(4-s))^2}\right) f_2 + \left(1 + e^{-v*(x-(4-s))^2}\right) f_8\right) & 4 - s < x < 4 + s \\ \frac{1}{2}(f_2 + f_8) & 4 + s < x < 6 - s \\ \frac{1}{2}\left(\left(2 - e^{-v*(x-(6-s))^2}\right) f_2 + f_8 e^{-v*(x-(6-s))^2}\right) & 6 - s < x < 6 + s \\ f_2 & 6 + s < x < 7 - s \\ \frac{1}{2}\left(\left(1 - e^{-v*(x-(7-s))^2}\right) f_5 + \left(1 + e^{-v*(x-(7-s))^2}\right) f_2\right) & 7 - s < x < 7 + s \\ \frac{1}{2}(f_2 + f_5) & 7 + s < x < 10 \end{cases} \tag{D.2}$$

The parameters s and v used in our model were determined empirically by looking at the graphs of the reconstructed corrosion profile and choosing values that gave a smoothly connected profile. For all of the reconstructions presented in the paper, we have chosen $s = 0.3$ and $v = 30$.