

**Numerical Scheme for the Solution to Laplace's Equation using Local Conformal
Mapping Techniques**

by

Cynthia Anne Sabonis

A Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in Applied Mathematics

by

April 2014

Approved:

Professor Burt S. Tilley, Capstone Advisor

Professor Luca Capogna, Department Head

Abstract

This paper introduces a method to determine the pressure in a fixed thickness, smooth, periodic domain; namely a lead-over-pleat cartridge filter. Finding the pressure within the domain requires the numerical solution of Laplace's equation, the first step of which is approximating, by interpolation, the curved portions of the filter to a circle in the xy plane. A conformal map is then applied to the filter, transforming the region into a rectangle in the uv plane. A finite difference method is introduced to numerically solve Laplace's equation in the rectangular domain. There are currently methods in existence to solve partial differential equations on non-regular domains. In a method employed by Monchmeyer and Muller, a scheme is used to transform from cartesian to spherical polar coordinates. Monchmeyer and Muller stress that for non-linear domains, extrapolation of existing cartesian difference schemes may produce incorrect solutions, and therefore, a volume centered discretization is used. A difference scheme is then derived that relies on mean values. This method has second order accuracy. [7] The method introduced in this paper is based on a 7-point stencil which takes into account the unequal spacing of the points. From all neighboring pairs, a linear system of equations is constructed, which takes into account the periodic domain. This method is solved by standard iterative methods. The solution is then mapped back to the original domain, with second order accuracy. The method is then tested to obtain a solution to a domain which satisfies $y = \sin(x)$ at the center, a shape similar to that of a lead-over-pleat cartridge filter. As a result, a model for the pressure distribution within the filter is obtained.

Acknowledgements

I would like to thank Professor Burt Tilley and Marisa Zemsky without whose patience and guidance I could not have completed this project. Both their advice and willingness to share their knowledge has been an integral part of my success at WPI. I would also like to thank the Pall Corporation and Dr. Mark Hurwitz for presenting the problem at the MPI 2013 Workshop. I would also like to thank Professor Goulet, whose constant guidance, support, and kindness has helped me in more ways than he could ever know. He is a true gem, and has set an example that I aspire to live by in my everyday life. In addition, I would like to thank my family for their support through this process, as well as Neil deGrasse Tyson, whose passion for life and the mathematical sciences inspires me to greatness every day.

Contents

1	Introduction	3
2	Problem Formulation	5
2.1	Application of Conformal Map	6
2.2	Derivation of 7-Point Stencil	7
2.3	Application of 7-Point Stencil	10
3	Results	12
4	Conclusion	15
5	Appendix	17
5.1	Derivation of 5 Point Stencil	17
5.2	Numerical Solution to Toy Problem	18
5.3	Exact Solution to Toy Problem	19
5.4	Error Analysis	21
5.5	Matlab code for 5-Point Stencil	25
5.6	Derivation 'Scheme 2' of 7-Point Stencil	28
5.7	Application of 7-Point Stencil	29
5.8	Matlab code for computing the radius and theta for the 7-Point Stencil	31
5.9	7-Point Stencil Finite Difference Method	34
5.10	Matlab code for Mapping the centerlines of the rectangular domain onto the original domain	36

List of Figures

1	LOP Cartridge Filter [3]	3
2	Shape of filter	5
3	Filter Discretization	5
4	Conformal Map $w = \ln z$ [2]	7
5	Discretization the curved domain into the mesh	8
6	7-Point Stencil	8
7	Application of 7-point stencil to a conformally mapped domain	10
8	Annulus Domain to test 7-point stencil	12
9	Mesh on physical domain [8]	12
10	Solution to Laplacian on an annulus using 7-point stencil with constant pressure at boundaries	13
11	Solution to Laplacian on an annulus using 7-point stencil with variable pressure at boundaries	13
12	Test domain, with the centerline of the filter defined by $y = \sin(x)$	14
13	Solution to Laplacian using the 7-point stencil with $y = \sin(x)$ as centerline, constant pressure at boundaries	14
14	5 Point Stencil	17
15	Mesh Using 5 Point Stencil	18
16	Level Curves of approximated solution to Laplacian using the 5-point stencil, step size 1/10	22
17	Level curves of exact solution to the Laplacian using, 5-point stencil step size 1/1000	22
18	3D plot of Approximated Solution to Laplacian using the 5-point stencil, step size 1/10	22
19	3D plot of Exact Solution to Laplacian using using the 5-point stencil, step size 1/1000	22
20	Level curves of approximated solution to the Laplacian using the 5-point stencil, step size 1/100	23
21	Level curves of exact solution to the Laplacian using 5-point stencil step size 1/1000	23
22	3D plot of Approximated Solution to Laplacian using the 5-point stencil, step size 1/100	23
23	3D plot of Exact Solution to Laplacian using using the 5-point stencil, step size 1/1000	23
24	Level curves of approximated solution to the Laplacian using the 5-point stencil, step size 1/1000	24
25	Level curves of exact solution to the Laplacian using, 5-point stencil step size 1/1000	24
26	3D plot of Approximated Solution to Laplacian using the 5-point stencil, step size 1/1000	24
27	3D plot of Exact Solution to Laplacian using using the 5-point stencil, step size 1/1000	24

List of Tables

1	Approximation of the Solution to Laplace's Equation	21
---	---	----

1 Introduction

Treatment of water to improve its cleanliness began as early as the 1800s in the United States. [6] Water treatment has remained a prominent issue; the removal of visible toxins and pollutants as well as pathogens and bacteria will always be a necessity. To remove these particles, we rely upon filtration. This process removes the solids suspended in water. These particles may be as large as grit or sand, or as small as pathogens such as bacteria or viruses.

Depending on what use the water will be put to, different filter designs and sizes are used. This paper examines in depth the behavior and function of the Lead-Over-Pleat (LOP) cartridges filter, which is used primarily in food and beverage applications. The LOP cartridge works to clarify and clean fluids whose contaminant levels are less than 0.01% by weight. The LOP cartridge filter is characterized by its cylindrical housing, in which sits the replaceable filter element, which is the cartridge. The cartridge is cylindrical in shape as well, and is made to a set of standard lengths and diameters, so that it may be easily replaceable. The cartridge is a pleated sheet to maximize filtration area, and is long with respect to diameter. The LOP cartridge has an inside to outside flow pattern to protect filtrate from trapping particles inside the cartridge, as pictured below.

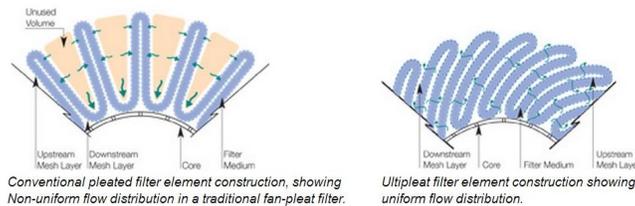


Figure 1: LOP Cartridge Filter [3]

Compared with standard cartridges, the pleated cartridge design has been shown to lower capital and installations costs, have a longer service life, and to lower operating costs by 30%. Additional features of the LOP cartridge are that it has a fixed fiber matrix with no adhesives or surfactants. This allows for constant filtrate quality, highly stable structure, higher product yields, and process reliability.

The pleated LOP cartridge forms a cylinder around the center core. To maximize the performance of the filter, there is an optimal number of pleats per unit length. Too many pleats may result in inter-pleat gaps getting pinched closed, resulting in portions of the cartridge being unusable or unused entirely. An option to counteract this effect is to inter-weave the pleats with strips of corrugated material (at right angles to the direction of the pleating). This will help to reduce any obstructions caused by overpleating.

Our goal is to construct a mathematical model to understand how a curvilinear domain affects pressure through the membrane. To do this we need not only an understanding of filters, but of porous media as well; the equations that describe the fluid flow within them as well as the different classifications and types. A porous medium is said to be homogeneous with respect to a certain property if that property is independent of position within the medium; otherwise the medium is heterogeneous. In addition, a medium in which a certain property is independent of direction is said to be isotropic. If that property were to vary with direction, say for example, the permeability, the medium would then be anisotropic. We consider only

homogeneous isotropic media.[1] There are several fundamental fluid transport equations which govern the behavior of a fluid in a porous medium. A pertinent equation to our topic at hand is Darcy's law, which describes the velocity of a fluid in a porous media. For a homogeneous, incompressible fluid, Darcy's law states;

$$\mathbf{q} = -\frac{k}{\mu}\nabla p$$

where \mathbf{q} is the volumetric flow rate (velocity), k is the permeability constant, μ is the viscosity, and ∇p is the pressure gradient [1].

Since the filter is homogeneous and isotropic, the permeability, k , is constant. This assumption may be altered in the future, as the filtered fluid continuously brings solutes to the membrane, building a layer of rejected solutes, increasing the resistance to filtration. [4]

In addition to Darcy's law, we also have a continuity condition; $\nabla \cdot \mathbf{q} = 0$. Using these two equations, the problem is transformed to

$$\nabla \cdot \mathbf{q} = \nabla \cdot \left(-\frac{k}{\mu}\nabla p\right) = 0$$

$$\nabla^2 p = 0$$

Therefore, to obtain an approximation of the pressure within the filter, we must find the solution to Laplace's equation on the given domain.

This paper outlines how to approach and solve the above problem. Initially, known x and y coordinates are interpolated to obtain an approximation to the equation of a circle with radius r and θ value from the axis for the given curve. A conformal map is then applied to the circle. Curved portions of the filter are mapped to a rectangular domain in the u, v plane. A finite difference method is used to find the solution to Laplace's equation on the rectangular domain. Given the nature of the domain, a finite difference stencil is derived which solves for the Laplacian at a point in terms of the six surrounding points. Using the stencil in conjunction with the periodicity of the domain, a system of n equations and n unknowns may be solved using a direct method. These points are then mapped back to the original domain.

2 Problem Formulation

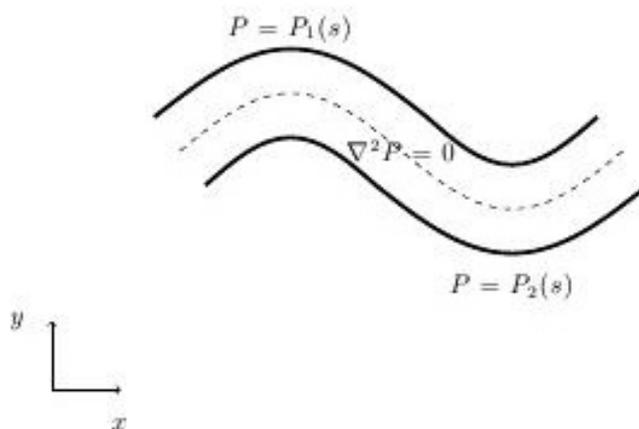


Figure 2: Shape of filter

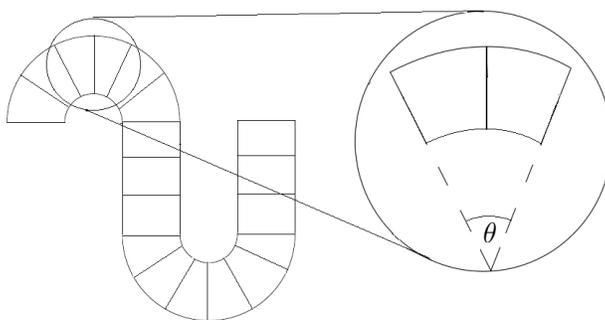


Figure 3: Filter Discretization

The problem at hand is to solve for the Laplacian within a curvilinear filter. To accomplish this, the problem is broken down into a series of smaller, more manageable problems. One can see that the filter may be discretized into small 'wedges' (see figure 3) throughout the domain. We begin, therefore, with a method to simplify the geometry of each 'wedge'. To start, we focus on the boundary conditions, which are given as coordinates (x, y) .

The geometry of the LOP cartridge filter presents the largest challenge at the curved segment. This domain may be approximated by the equation of a circle. In order to form an approximation, we first look to see if there is a way that we may approximate the radius value from the center of the approximated circle, as well as the angle, θ , of the curved segment.

Given a data point; $p(x_i, y_j)$, on the circle, we would like to output radius values, r , and theta values, θ , with respect to the center points, (a, b) . Given three points, $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, each point on the same circle with the center (a, b) and radius R . Therefore, the following equations must be satisfied;

$$\begin{aligned}(x_1 - a)^2 + (y_1 - b)^2 &= R^2 \\(x_2 - a)^2 + (y_2 - b)^2 &= R^2 \\(x_3 - a)^2 + (y_3 - b)^2 &= R^2\end{aligned}\tag{1}$$

Since each of these points are on the same circle, each equation may be set equal.

$$\begin{aligned}(x_1 - a)^2 + (y_1 - b)^2 &= (x_2 - a)^2 + (y_2 - b)^2 \\(x_2 - a)^2 + (y_2 - b)^2 &= (x_3 - a)^2 + (y_3 - b)^2\end{aligned}$$

After simplifying these equations, we receive a system of equations which may then use to solve for the centerpoint, (a, b) , which we then use to approximate the radius, r and θ .

$$\begin{aligned}a(x_2 - x_1) + b(y_2 - y_1) &= \frac{1}{2}(x_2^2 - x_1^2 + y_2^2 - y_1^2) \\a(x_3 - x_2) + b(y_3 - y_2) &= \frac{1}{2}(x_3^2 - x_2^2 + y_3^2 - y_2^2)\end{aligned}$$

$$\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix} \times \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(x_2^2 - x_1^2 + y_2^2 - y_1^2) \\ \frac{1}{2}(x_3^2 - x_2^2 + y_3^2 - y_2^2) \end{bmatrix}$$

The above system of equations may then be solved to obtain the center, (a, b) of the approximated circle which, with the given three points, $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, may be substituted into equation 1 to obtain the radius of the approximated circle. Once can see that as the curvature of the filter decreases, the matrix will be close to singular. The radius of curvature is therefore computed for each approximated for each circle approximated. If the curvature is found to be below a certain level, that portion of the domain will not be approximated to the equation of a circle. [8]

In order to find the angle, θ , from the horizontal to each of the approximated points on the circle, the length of the chord connecting each of the points on is calculated. Using this distance, which may be called CL , the θ value may be approximated using $\theta = \cos^{-1}\left(1 - \frac{CL^2}{2R^2}\right)$.

2.1 Application of Conformal Map

After approximating the curved portions of the filter, the problem consists of solving Laplace's equation on the given domain. The complicated geometry of the filter provides a challenge computationally. Computing the solution in a simpler domain would be simpler for calculations as well as accuracy.[5] We therefore consider ways to simplify the domain in which to solve the equation $\nabla^2 p = 0$. There are a variety of ways to achieve this, among them being, building up a flow by superimposing a distribution of sources and sinks, applying perturbation techniques assuming the body is thin, and lastly, solving Laplace's equation numerically. [5] For our purposes, we consider breaking the domain into of series of smaller 'wedges', considering the curved poritons, which were approximated to the equation of a circle. The width of these portions is dependent on the θ value. These wedges, or semi-circles satisfy the equation $\nabla^2 p = 0$ and are therefore harmonic functions. Therefore, if $w = f(z)$ is an analytic function mapping D onto a domain D' , p is carried over as well. Because the solution will be harmonic,

it will be of the form $z = x + iy$. We choose a well known mapping $w = f(z) = \ln(x + iy)$ which will transform the domain from an annulus to a rectangle, and therefore, a much simpler computational problem.

$$\begin{aligned} \ln(x + iy) &= \ln(re^{i\theta}) = \ln(r) + i\theta \\ u &= \ln(r) \quad v = \theta + \frac{\pi}{2} \end{aligned}$$

where r and θ have been previously calculated.

This relation provides a one-to-one correspondance between our domain D and our transformed domain; looking at the problem locally, this is true. Globally, looking at the shape of the filter as a whole, we can see that this is not true. Therefore, to solve Laplace's equation in the domain, we section the domain into a sequece of one to one functions to solve locally on each one. As a result, each section of the domain will have inverse, $z = f^{-1}(w)$.

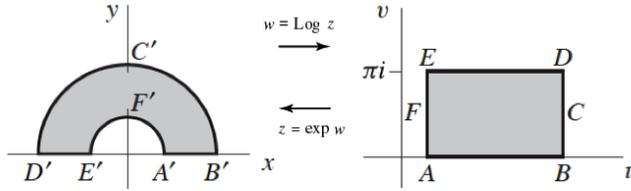


Figure 4: Conformal Map $w = \ln z$ [2]

That is, each point of the mapping will be associated with a given point in D . This inverse, $f^{-1}(w)$ will map the solution back to the original domain. After computing the solution locally on each of these regions, the final solution may be reached by placing each local map together, reforming the shape of the filter. The goal is to get a linear relation of the pressure at specific grid points. The local conformal map with a finite difference discretization scheme in the uv plane accomplishes this.

2.2 Derivation of 7-Point Stencil

After calculating the r and θ and applying the conformal map, $\nabla^2 p = p_{uu} + p_{vv} = 0$ in the transformed region. Our solution has thus been further simplified to solving Laplace's equation in each local mapping. To solve this numerically, we rely on finite difference methods. To do this, we use Taylor's polynomial to approximate the derivative of a function around a point. This method is commonly used to formulate approximate difference equations to estimate the solution of differential equations. One of the most well known is the 5-point stencil. (See Appendix). Here, we find the pressure at each point in terms of the surrounding points. The deviation here from standard methods occurs because of the geometry of the filter. Whereas more conventional methods may have equally spaced points in the uv plane, the geometry of the filter has unequal spacing in the u and v direction.

We seek a solution to the Laplacian about a point $p(u_0, v_0) = P_1$. We do this by using a Taylor expansion about the chosen point. Each point is approximated to the second derivative, producing a scheme with second order accuracy. Within the derivation, changes in the u and v axis are defined using h_i and k_j respectively, where $i = 1, 2, 3$ and $j = 1, 2$. h is defined as

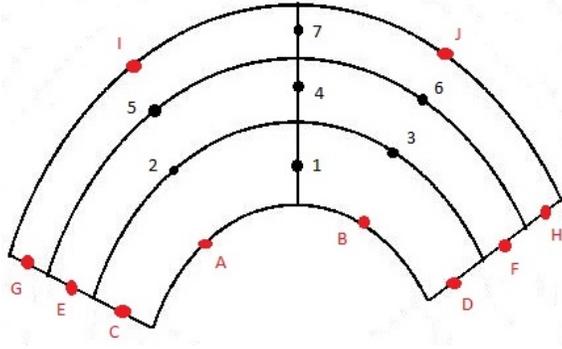


Figure 5: Discretization the curved domain into the mesh

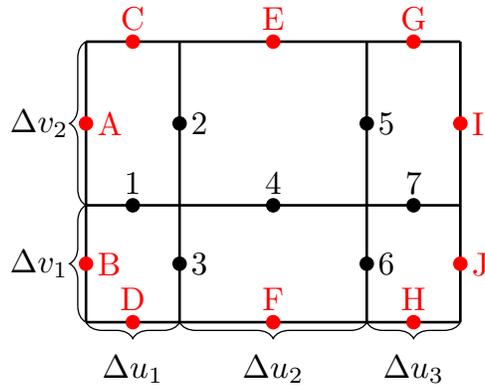


Figure 6: 7-Point Stencil

$h_i = \frac{\Delta u_i}{2}$ and k as $k_j = \frac{\Delta v_j}{2}$, as shown in y. Therefore, to find points P_1, P_4, P_7 , the following equations are used;

$$p(u_0 - h_1, v_0 + k_2) \approx p(u_0, v_0) - p_u(u_0, v_0)h_1 + p_v(u_0, v_0)k_2 + \frac{1}{2} [p_{uu}(u_0, v_0)h_1^2 - 2p_{uv}(u_0, v_0)h_1k_2 + p_{vv}(u_0, v_0)k_2^2] \quad (2)$$

$$p(u_0 - h_1, v_0 - k_1) \approx p(u_0, v_0) - p_u(u_0, v_0)h_1 - p_v(u_0, v_0)k_1 + \frac{1}{2} [p_{uu}(u_0, v_0)h_1^2 + 2p_{uv}(u_0, v_0)h_1k_1 + p_{vv}(u_0, v_0)k_1^2] \quad (3)$$

$$p(u_0 + h_1, v_0 + k_2) \approx p(u_0, v_0) + p_u(u_0, v_0)h_1 + p_v(u_0, v_0)k_2 + \frac{1}{2} [p_{uu}(u_0, v_0)h_1^2 + 2p_{uv}(u_0, v_0)h_1k_2 + p_{vv}(u_0, v_0)k_2^2] \quad (4)$$

$$p(u_0 + h_1, v_0 - k_1) \approx p(u_0, v_0) + p_u(u_0, v_0)h_1 - p_v(u_0, v_0)k_1 + \frac{1}{2} [p_{uu}(u_0, v_0)h_1^2 - 2p_{uv}(u_0, v_0)h_1k_1 + p_{vv}(u_0, v_0)k_1^2] \quad (5)$$

We combine equations (2)-(5) by eliminating terms proportional to p_u and p_v to find

$$k_1 p(u_0 - h_1, v_0 + k_2) + k_2 p(u_0 - h_1, v_0 - k_1) + k_1 p(u_0 + h_1, v_0 + k_2) + k_2 p(u_0 + h_1, v_0 - k_1) \approx 2(k_1 + k_2)p(u_0, v_0) + (k_1 + k_2)h_1^2 p_{uu}(u_0, v_0) + k_1 k_2 (k_1 + k_2) P_{vv}(u_0, v_0)$$

or,

$$k_1 P_A + k_2 P_B + k_1 P_2 + k_2 P_3 \approx 2(k_1 + k_2)P_1 + (k_1 + k_2)h_1^2 P_{uu} + k_1 k_2 (k_1 + k_2) P_{vv} \quad (6)$$

For points C and D the pressure is approximated by

$$\begin{aligned} p(u_0, v_0 + 2k_2) &\approx p(u_0, v_0) + p_v(u_0, v_0)2k_2 + \frac{1}{2}p_{vv}(u_0, v_0)(2k_2)^2 \\ p(u_0, v_0 - 2k_1) &\approx p(u_0, v_0) - p_v(u_0, v_0)2k_1 + \frac{1}{2}p_{vv}(u_0, v_0)(-2k_1)^2 \end{aligned}$$

Which, put in terms of our points, is

$$\begin{aligned} P_C &\approx P_1 + 2k_2 P_v + 2k_2^2 P_{vv} \\ P_D &\approx P_1 - 2k_1 P_v + 2k_1^2 P_{vv} \end{aligned}$$

We now take $k_1 P_C + k_2 P_D$ and solve for P_{vv}

$$P_{vv} \approx \frac{k_1 P_C + k_2 P_D - (k_1 + k_2)P_1}{2(k_1 k_2^2 + k_2 k_1^2)} \quad (7)$$

We now return to (6), where, in order to obtain the desired result, add a factor of zero, in the form of $(k_1 + k_2)h_1^2 P_{vv} - (k_1 + k_2)h_1^2 P_{vv}$.

$$\begin{aligned} (k_1 + k_2)h_1^2 P_{uu} + k_1 k_2 (k_1 + k_2) P_{vv} &\approx k_1 P_A + k_2 P_B + k_1 P_2 + k_2 P_3 - 2(k_1 + k_2)P_1 \\ (k_1 + k_2)h_1^2 [P_{uu} + P_{vv}] + (k_1 + k_2)k_1 k_2 P_{vv} - (k_1 + k_2)h_1^2 P_{vv} &\approx k_1 P_A + k_2 P_B + k_1 P_2 + k_2 P_3 - 2(k_1 + k_2)P_1 \\ (k_1 + k_2)h_1^2 [P_{uu} + P_{vv}] + (k_1 + k_2) [k_1 k_2 - h_1^2] P_{vv} &\approx k_1 P_A + k_2 P_B - 2(k_1 + k_2)P_1 + k_1 P_2 + k_2 P_3 \end{aligned}$$

Substituting for P_{vv} using 7, we solve for the Laplacian.

$$(k_1 + k_2)h_1^2 [\nabla^2 P] \approx k_1 P_A + k_2 P_B - 2(k_1 + k_2)P_1 + k_1 P_2 + k_2 P_3 - (k_1 + k_2) [k_1 k_2 - h_1^2] \left[\frac{k_1 P_C + k_2 P_D - (k_1 + k_2)P_1}{2(k_1 k_2^2 + k_2 k_1^2)} \right]$$

And our 7 point stencil for the Laplacian, (the subscript denotes the scheme used);

$$\begin{aligned} \nabla^2 P_1 &\approx \left[k_1 P_A + k_2 P_B + k_1 P_2 + k_2 P_3 - 2 \left[(k_1 + k_2) - \frac{(k_1 + k_2)^2 [k_1 k_2 - h_1^2]}{4(k_1 k_2^2 + k_2 k_1^2)} \right] P_1 \right. \\ &\quad \left. - \frac{(k_1 + k_2) [k_1 k_2 - h_1^2] [k_1 P_C + k_2 P_D]}{2(k_1 k_2^2 + k_2 k_1^2)} \right] / (k_1 + k_2)h_1^2 \end{aligned} \quad (8)$$

Using similar techniques, a stencil for $P_2, P_5, P_3,$ and P_6 may be constructed. (For the full derivation, see the Appendix.)

$$\begin{aligned} \nabla^2 P_2 &\approx \left[h_2 P_C + h_1 P_E + h_2 P_1 + h_1 P_4 - 2 \left[(h_2 + h_1) - \frac{h_2 h_1 (h_1 + h_2)^2 - (h_2 + h_1)^2 k_2^2}{4(2h_2 h_1^2 + h_1 h_2^2)} \right] \right. \\ &\quad \left. - \frac{(h_1 + h_2)(h_2 h_1 - k_2^2) [h_2 P_A + h_1 P_5]}{2h_2 h_1^2 + h_1 h_2^2} \right] / (h_2 + h_1)k_2^2 \end{aligned} \quad (9)$$

When used in conjunction, these two schemes, (9) and (10), will solve for the pressure on the interior of the given domain. Note that the the stencil may be refined in order to make the approximation more accurate.

2.3 Application of 7-Point Stencil

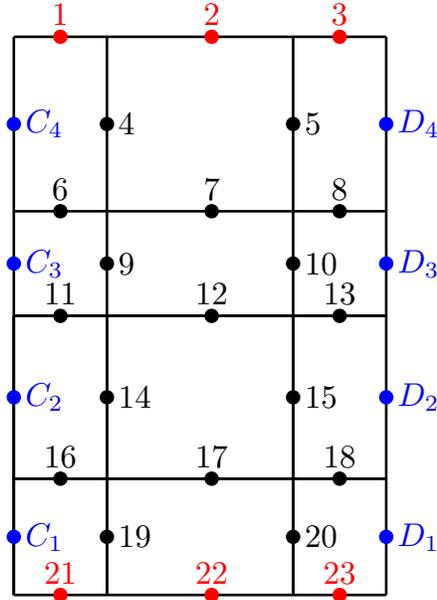


Figure 7: Application of 7-point stencil to a conformally mapped domain

The established stencil may now be used solve for the pressure inside in the domain. For the given problem, a three layer filter is considered. The stencil, however, is generalized to take into account multiple wedges at a time, per the figure above. When using the stencil, the boundary points present an additional challenge. Of the mapped region, boundary values are known for two of the sides; specifically, the left and right hand side of the rectangular domain. The three layer stencil points C_4 , C_3 , C_2 and C_1 may be approximated using the methods discussed earlier, and it may be assumed points D_4 , D_3 , D_2 , and D_1 are a set distance from them. Using the stencil to calculate each of the unknown points, there are 17 equations and 23 unknowns. Using the geometry of the filter as an advantage, one can see that the structure is periodic. That is, for each 'wedge' of the filter in which we are trying to solve for the pressure, the leftmost and rightmost points will be equal. Looking at the conformal mapping, this will translate to the top and bottom pressures. That is, points 1, 2, 3 will be equivalent to points 21, 22, 23. This allows points 1, 2 and 3 to be written using the seven point stencil, and thus provides an additional six equations to the system. The problem is now contains 23 equations representing 23 unknowns, and may be represented as a linear system, $Ax = b$. The A matrix will consist of the coefficients of each of the points in the 23 equations, while the b matrix will contain the known boundary values of the domain. This system may be solved for the vector x , thus providing a solution to the Laplacian at each of the points 1, 2...23. (For the calculated equations, please see Appendix.)

While solving for the vector x in Matlab, one can use the traditional direct method. However, because the matrix is sparse, we want to use an interative solver.

The way in which the mesh is numbered is an important aspect of the problem formulation;

each point is labeled so that when compared with the conformal map, the points are within a set, bounded distance of each other. As a result, the matrix A is a diagonally dominant matrix, which provides stability to the method as well as more guaranteed convergence of solution.

3 Results

The above system of equations may be used in conjunction with the interpolated points and the conformal mapping to solve for the pressure distribution within a curved domain. To test the method for accuracy, the code was tested on an annulus. In this way, the method was compared with the exact solution, which is computed using the method of separation of variables. The method was tested first for an annulus with an inner function of $p = 1$ and an outer function of $p = 3$. It was then tested with an inner function of $p = 1$ and an outer function of $p = 3 + \cos(x)$.

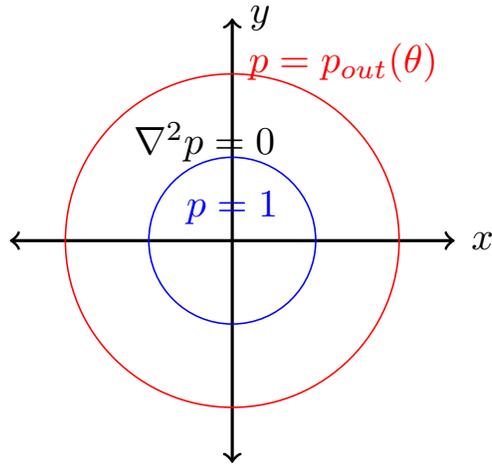


Figure 8: Annulus Domain to test 7-point stencil

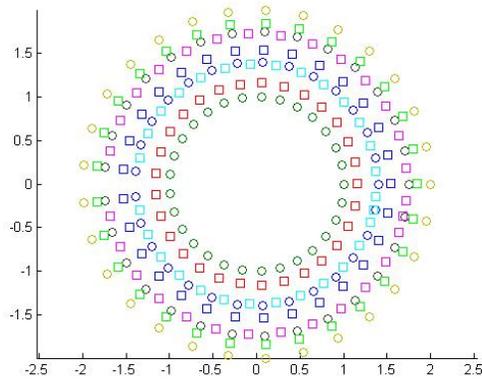


Figure 9: Mesh on physical domain [8]

The above figures display the domain as well as the mesh on the annulus at which the solution to the Laplacian, $\nabla^2 p = 0$ was calculated. For the first test, both the interior and exterior function were given constant values, with the interior being $p = 1$ and the exterior being $p = 3$. Because the boundary of the domain is a perfect circle, and the pressure values are constants, it is expected the the calculated solution will be exact. This is due to the second order accuracy of the method. When compared with the solution method using separation of variables, as expected, the results showed a relative error of $1.470943125797088e-15$. Figure (10) shows the relative error of the approximated solution vs the exact solution at each layer within the filter. One can see that at the innermost layer, closest to the boundary function $p = 1$, the error is greatest. The error in solution decreases at the layers approach the outer filter, $p = 3$. The method was then tested with the outer function changed to $p = 3 + \cos(x)$. Due to conformal mapping, which does not preserve distance, it is expected that this solution would not be exact. This calculated solution showed an error of 0.001061201797963 . This demonstrates quadratic convergence, as is consistent with the derivation of the method. Figure (11) displays the exact pressure vs. the approximated pressure in the domain in which the outgoing pressure is the function $y = \cos(x)$.

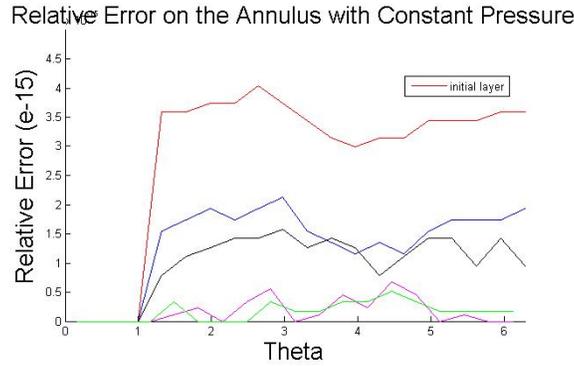


Figure 10: Solution to Laplacian on an annulus using 7-point stencil with constant pressure at boundaries

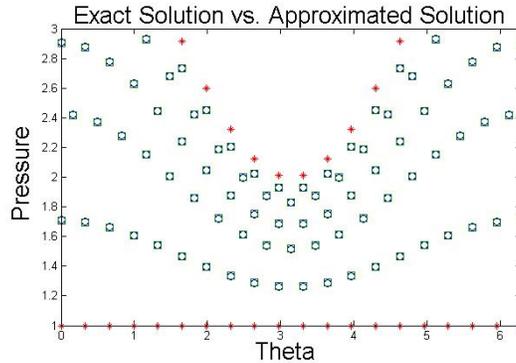


Figure 11: Solution to Laplacian on an annulus using 7-point stencil with variable pressure at boundaries

Once the accuracy and convergence of the method had been tested successfully, the method was tested for a centerline which fit the function $y = \sin(x)$ with constant pressures on the boundary, $p = 1$ and $p = 3$. One can see that the pressure distribution within the filter resembles the sine curve. For $\theta = \frac{\pi}{2} = -\frac{\pi}{2}$, the values at which the curvature is highest, the change in the pressure is greatest.

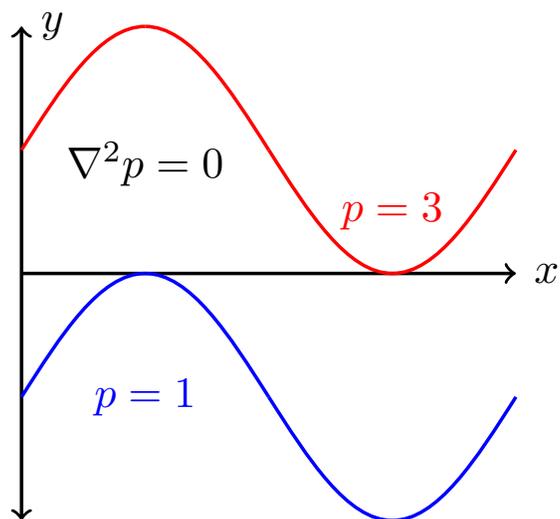


Figure 12: Test domain, with the centerline of the filter defined by $y = \sin(x)$.

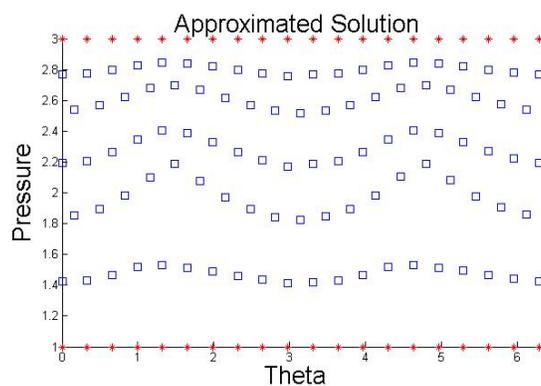


Figure 13: Solution to Laplacian using the 7-point stencil with $y = \sin(x)$ as centerline, constant pressure at boundaries

4 Conclusion

This paper establishes a method in which to determine the solution to Laplace's equation in a periodic domain through the use of numerical methods and conformal mapping. After approximating the curve of the filter to the equation of a circle, a conformal map is applied in order to simplify the domain in which the solution to Laplace's equation is being approximated. An approximate solution is obtained through Taylor approximation about a point in terms of the six surrounding points. The periodicity of the domain allows for the solution, P_i at each of the unknown points to be represented as a linear system, which may then be solved for using the traditional method. Once a solution is obtained in the u, v plane, it may then be mapped back to the original domain, thus providing a solution to the Laplacian in the original, curvilinear domain. The scheme was tested on an annulus, initially with constant pressure, to which it provided an exact solution. The scheme was tested on an annulus for variable pressure as well, with which it displayed second order accuracy. The method was then tested on a domain with a centerline of $y = \sin(x)$ with constant pressures, $p = 1$, $p = 3$. This domain was chosen because of its resemblance to the structure of the lead-over pleat cartridge filter. As a result, the solution to the Laplacian within this domain provides an idea of the pressure distribution within the filter. The solution showed greatest changes in pressure at $\frac{\pi}{2}$ and $-\frac{\pi}{2}$, the locations where the change in curvature of the sine function is greatest. This is consistent with the pressure distribution that one would expect within the filter.

References

- [1] Jacob Bear. *Dynamics of fluids in porous media*. Dover Publications, Inc., 1972.
- [2] James W. Verhey F. Roger Churchill V. Ruel, Brown. *Complex Variables and Applications*. McGraw-Hill Book Company, 1974.
- [3] Pall Corporation. Ultipleat liquid filter elements for aerospace, February 2011.
- [4] Y. Jaffrin, Michel. Hydrodynamic techniques to enhance membrane filtration. (8), 2012.
- [5] Cohen I.M Dowling D.A. (with contributions by Ayyaswamy P.S. and Hu H.H.) Kundu, P.K. *Fluid Mechanics*. Elsevier Inc., 2012.
- [6] Gary S. Logsdon. *Water Filtration Practices; Including Slow Sand Filters and Precoat Filtration*. 2008.
- [7] Kwak Dochan Rosenfeld, Moshe. Multigrid acceleration of a fractional-step solver in generalized curvilinear coordinate systems. *SAO/NASA Astrophysics Data System*, 217:351–367, 1989.
- [8] Marisa Zemsky. Report to pall corporation on a one-dimensional solver for the centerline pressure in a curvilinear filter. (*private communication*), pages 1–9, December 21, 2013.

5 Appendix

5.1 Derivation of 5 Point Stencil

Solving Laplace's equation using finite differences relies on discretizing the spatial domain into a series of grid points. Together these form a mesh whose adjacent points are used to solve Laplace's equation iteratively. Given $\nabla^2 p(x, y) = p_{xx}(x, y) + p_{yy}(x, y) = 0$ on a tranformed region R , Taylor series in the variable x about x_i is used to approximate the second derivative.

$$p(x_i + \Delta x, y_j) = p(x_i, y_j) + p_x(x_i, y_j)\Delta x + \frac{1}{2}p_{xx}(x_i, y_j)\Delta x^2$$

$$p(x_i - \Delta x, y_j) = p(x_i, y_j) - p_x(x_i, y_j)\Delta x + \frac{1}{2}p_{xx}(x_i, y_j)\Delta x^2$$

Which when added together, produce the following sum;

$$p(x_i + \Delta x, y_j) + p(x_i - \Delta x, y_j) = 2p(x_i, y_j) + p_{xx}(x_i, y_j)\Delta x^2$$

Solving for $p_{xx}(x_i, y_j)$, we obtain

$$p_{xx}(x_i, y_j) = \frac{p(x_{i+1}, y_j) - 2p(x_i, y_j) + p(x_{i-1}, y_j)}{\Delta x^2}$$

The additional term, $p_{yy}(x_i, y_j)$ is solved for similarly, using Taylor series in the variable y about y_j , thus producing an interitive method to solve Laplace's equation within the region;

$$\frac{p(x_{i+1}, y_j) - 2p(x_i, y_j) + p(x_{i-1}, y_j)}{\Delta x^2} + \frac{p(x_i, y_{j+1}) - 2p(x_i, y_j) + p(x_i, y_{j-1})}{\Delta y^2} = 0 \quad (10)$$

Looking at the above equation for the Laplacian, we can see that we will need approximations at the points $p(x_{i-1}, y_j), p(x_i, y_j), p(x_{i+1}, y_j), p(x_i, y_{j-1})$, and $p(x_i, y_{j+1})$. That is, at points $p_{i-1,j}, p_{i,j}, p_{i+1,j}, p_{i,j-1}$, and $p_{i,j+1}$.

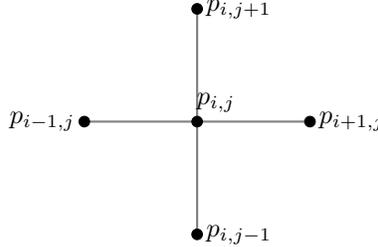


Figure 14: 5 Point Stencil

Therefore, to obtain an approximation to $\nabla^2 p(x_i, y_j) = 0$ at the point $p_{i,j}$, equation 10 is solved for $p(x_i, y_j)$, creating an iterative scheme to solve the Laplacian. Letting $h = \Delta x$ and $k = \Delta y$, 10 becomes

$$\left[1 + \left(\frac{h}{k}\right)^2\right] p_{i,j} - (p_{i+1,j} + p_{i-1,j}) - \left(\frac{h}{k}\right)^2 (p_{i,j+1} + p_{i,j-1}) = 0$$

. That is, for $\lambda = \left(\frac{h}{k}\right)^2$ and $\mu = 2(\lambda + 1)$,

$$p_{i,j} = \frac{(p_{i+1,j} + p_{i-1,j} + \lambda(p_{i,j+1}) + \lambda(p_{i,j-1}) + -h^2 f(x_i, y_j))}{\mu}$$

5.2 Numerical Solution to Toy Problem

To solve Laplace's equation in a rectangular domain using the 5 point finite difference method with defined boundary conditions, a mesh must first be constructed with equal spacing in the x and y variable. The border of the interior mesh will rely on different boundary functions, and the code must be designed to reflect this. The domain is broken into 9 separate equations; one for the top left hand corner of the mesh, the top interior of the mesh, the top right hand corner of the mesh, the right interior of the mesh, the bottom right hand corner of the mesh, the bottom interior of the mesh, the bottom left hand corner of the mesh, the left interior of the mesh, and finally, the interior of the mesh.

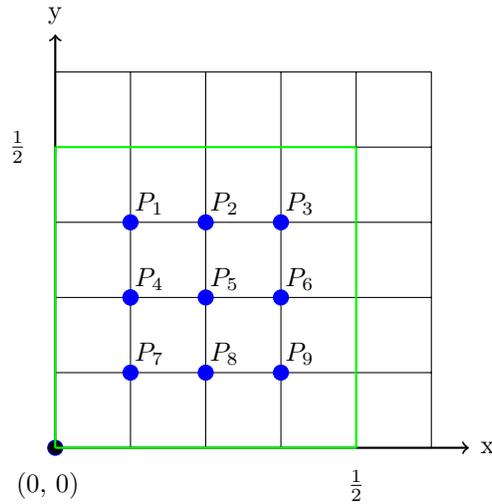


Figure 15: Mesh Using 5 Point Stencil

5.3 Exact Solution to Toy Problem

To test the five point finite difference scheme, an example is introduced which may be solved numerically as well as computed by hand, using the technique of separation of variables. The problem introduced is below;

$$\nabla^2 p = 0 \quad R = \{(x, y) | 0 < x < \frac{1}{2} = L, 0 < y < \frac{1}{2} = K\}$$

$$p(0, y) = f_1(y) = 0 \quad p\left(\frac{1}{2}, y\right) = f_2(y) = 0$$

$$p(x, 0) = g_1(x) = 0 \quad p\left(x, \frac{1}{2}\right) = g_2(x) = \sin(4 \cdot \pi \cdot x)$$

It is assumed the solution will be linear and of the form $p(x, y) = X(x)Y(y)$. This solution is substituted into the original equation, and is then divided by $X(x)Y(y)$. The equation is now separated into $X(x)$ and $Y(y)$ terms which we set equal to a separation constant, $-\lambda$, creating two homogeneous differential equations.

$$X''(x) + \lambda X(x) = 0 \tag{11}$$

$$Y''(y) - \lambda Y(y) = 0 \tag{12}$$

Solving equation 11 yields

$$X_n(x) = \sin\left(\frac{n\pi x}{L}\right) \text{ for } n = 1, 2, \dots \text{ as well as } \lambda = \left(\frac{n\pi x}{L}\right)^2$$

. Solving equation 12 yields the solution

$$Y_n(y) = C_{1n} \sinh\left(\frac{n\pi}{L}y\right) + C_{2n} \cosh\left(\frac{n\pi}{L}y\right)$$

Ensuring that boundary conditions are satisfied, we obtain

$$Y_n(y) = C_{1n} \sinh\left(\frac{n\pi(y-K)}{L}\right) + C_{2n} \cosh\left(\frac{n\pi}{L}y\right)$$

. Placing these two solutions together using the method of superposition, we obtain

$$u(x, y) = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{L}\right) \left(C_{1n} \sinh\frac{n\pi(y-K)}{L} + C_{2n} \sinh\left(\frac{n\pi y}{L}\right) \right)$$

. Each term satisfies the PDE and the two homogeneous boundary conditions. Now, we solve for C_{1n} and C_{2n} so that the remaining non homogeneous boundary conditions are satisfied.

$$\begin{aligned} u(x_1, 0) = g_1(x) &= \sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{L}\right) \left[C_{1n} \sinh\left(\frac{-n\pi K}{L}\right) \right] \\ &= \sum_{n=1}^{\infty} \left(-C_{1n} \sinh\frac{n\pi K}{L} \right) \sin\left(\frac{n\pi x}{L}\right) \\ &= \sum_{n=1}^{\infty} b_{1n} \sin\left(\frac{n\pi x}{L}\right) \end{aligned}$$

Similarly,

$$\begin{aligned} u(x, K) = g_2(x) &= \sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{L}\right) \left[C_{2n} \sinh\left(\frac{2\pi K}{L}\right) \right] \\ &= \sum_{n=1}^{\infty} b_{2n} \sin\left(\frac{n\pi x}{L}\right) \end{aligned}$$

We need the Fourier sine series of g_1 and g_2 .

$$b_{1n} = -C_{1n} \sinh\left(\frac{n\pi K}{L}\right) = -\frac{2}{L} \int_0^L g_1(\bar{x}) \sin\left(\frac{n\pi \bar{x}}{L}\right) d\bar{x}$$

$$b_{2n} = -C_{2n} \sinh\left(\frac{n\pi K}{L}\right) = \frac{2}{L} \int_0^L g_2(\bar{x}) \sin\left(\frac{n\pi \bar{x}}{L}\right) d\bar{x}$$

Which shows that

$$C_{1n} = -\frac{b_{1n}}{\sinh\left(\frac{n\pi K}{L}\right)} = -\frac{2}{L} \operatorname{csch}\left(\frac{n\pi K}{L}\right) \int_0^L g_1(\bar{x}) \sin\left(\frac{n\pi \bar{x}}{L}\right) d\bar{x}$$

$$C_{2n} = -\frac{b_{2n}}{\sinh\left(\frac{n\pi K}{L}\right)} = -\frac{2}{L} \operatorname{csch}\left(\frac{n\pi K}{L}\right) \int_0^L g_2(\bar{x}) \sin\left(\frac{n\pi \bar{x}}{L}\right) d\bar{x}$$

Thus providing the constants for our final solution;

$$u(x, y) = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{L}\right) \left(C_{1n} \sinh\frac{n\pi(y-K)}{L} + C_{2n} \sinh\left(\frac{n\pi y}{L}\right) \right)$$

5.4 Error Analysis

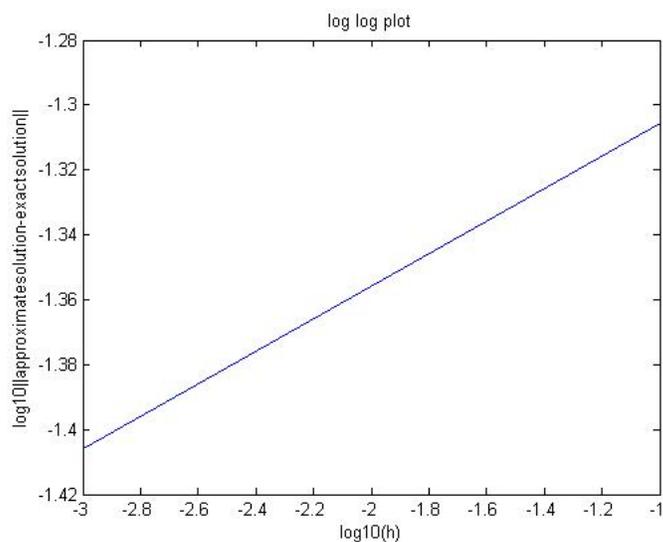
$$\nabla^2 p = 0 \quad R = \{(x, y) \mid 0 < x < \frac{1}{2}, 0 < y < \frac{1}{2}\}$$

$$p(0, y) = 0 \quad p\left(\frac{1}{2}, y\right) = 0$$

$$p(x, 0) = 0 \quad p\left(x, \frac{1}{2}\right) = \sin(4 \cdot \pi \cdot x)$$

Table 1: Approximation of the Solution to Laplace's Equation

Size of Grid	Number of Iterations to Converge	Elapsed Time(seconds)	Relative Error
4x4	19	2.655110	0.049476145050229
499x499	43479	10237.278552	0.039267501163863



Note; n divides the x axis into n equal sections. For $x \in [a, b]$, the step size h is defined by $h = \frac{b-a}{n}$. Similarly, m divides the y axis into m equal sections. For $y \in [c, d]$, the step size k is defined by $k = \frac{d-c}{m}$. All approximate solutions are compared with the exact solution for which $m = n = 500$. ($h = k = \frac{1}{1000}$)

1. The solution to Laplace's equation, with $m = n = 5$.

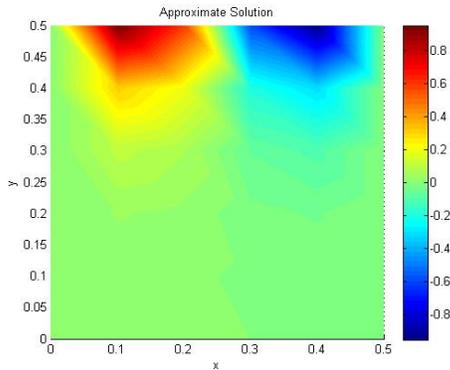


Figure 16: Level Curves of approximated solution to Laplacian using the 5-point stencil, step size 1/10

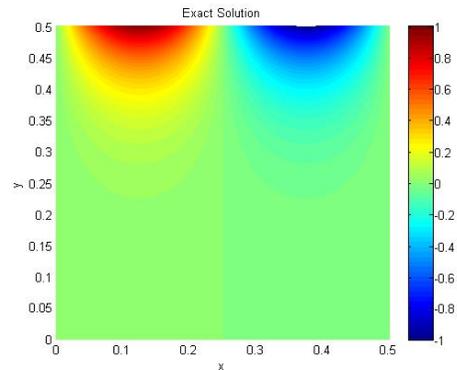


Figure 17: Level curves of exact solution to the Laplacian using, 5-point stencil step size 1/1000

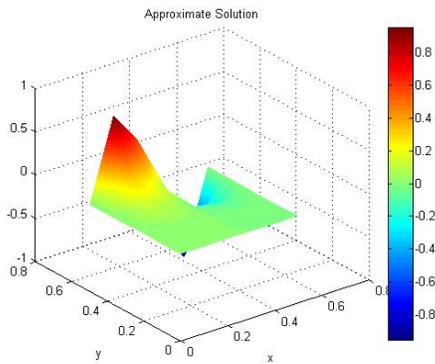


Figure 18: 3D plot of Approximated Solution to Laplacian using the 5-point stencil, step size 1/10

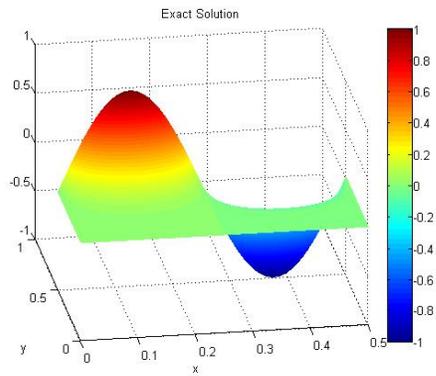


Figure 19: 3D plot of Exact Solution to Laplacian using the 5-point stencil, step size 1/1000

2. The solution to Laplace's equation, with $m = n = 50$.

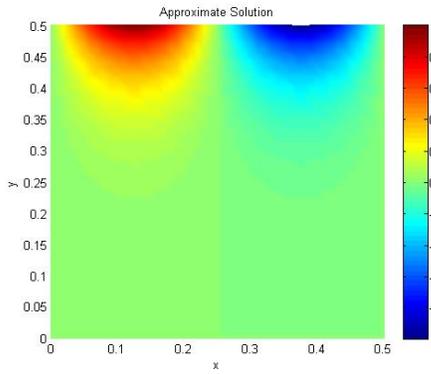


Figure 20: Level curves of approximated solution to the Laplacian using the 5-point stencil, step size $1/100$

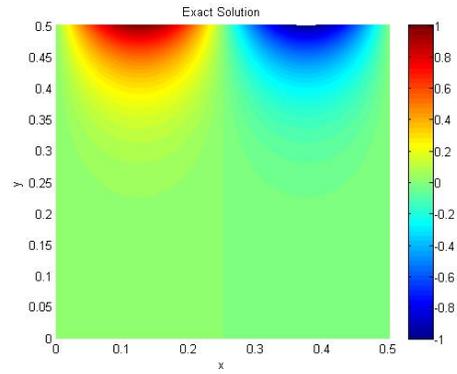


Figure 21: Level curves of exact solution to the Laplacian using 5-point stencil step size $1/1000$

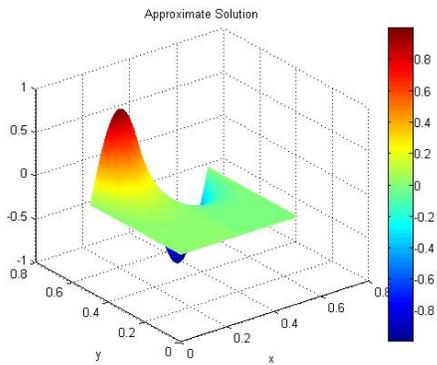


Figure 22: 3D plot of Approximated Solution to Laplacian using the 5-point stencil, step size $1/100$

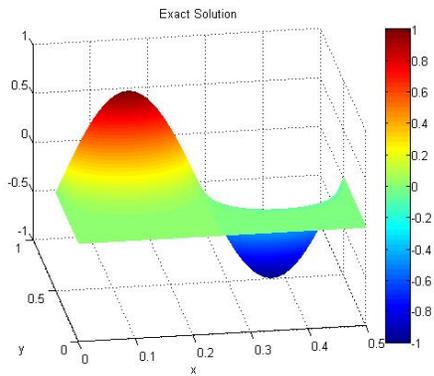


Figure 23: 3D plot of Exact Solution to Laplacian using using the 5-point stencil, step size $1/1000$

3. The solution to Laplace's equation, with $m = n = 500$.

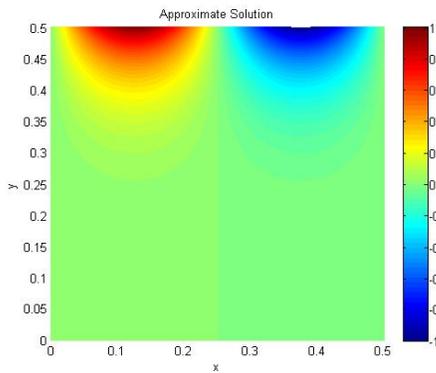


Figure 24: Level curves of approximated solution to the Laplacian using the 5-point stencil, step size $1/1000$

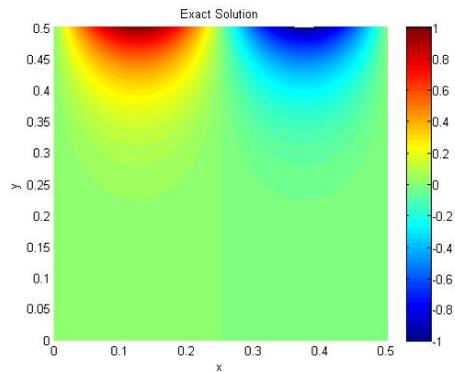


Figure 25: Level curves of exact solution to the Laplacian using, 5-point stencil step size $1/1000$

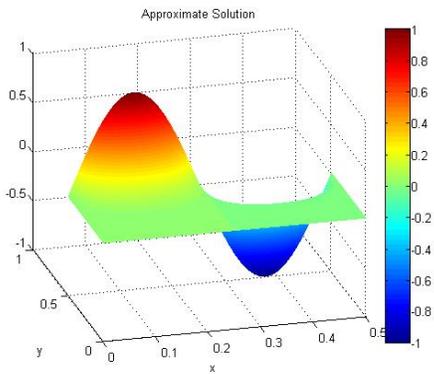


Figure 26: 3D plot of Approximated Solution to Laplacian using the 5-point stencil, step size $1/1000$

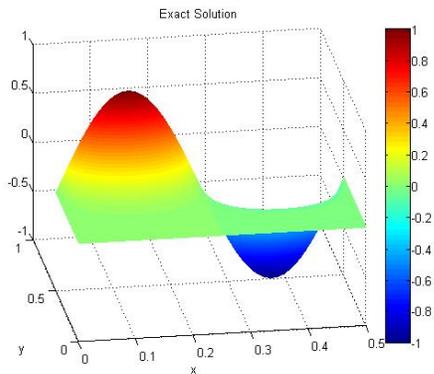


Figure 27: 3D plot of Exact Solution to Laplacian using using the 5-point stencil, step size $1/1000$

5.5 Matlab code for 5-Point Stencil

```

0 % a
.5 % b
0 % c
.5 % d
50 % m
50 % n
60000 % Maximum number of iterations
.000001 %Tolerance

%This script uses the finite difference method to calculate the solution
to Poisson's
%equation,  $\text{div}(\text{grad}(w)) = f$  in a rectangular domain.
clc
clf
clear all
format long
load('init.par')
tic
%% Initializes the dimensions of the domain and the step size of the
domain.
h=(init(2)-init(1))/init(6); % Defines the step size along the x axis
k=(init(4)-init(3))/init(5); % Define the step size along the y axis.
%% Creates the mesh within the domain using the step sizes h and k,
respectively.
for i=1:init(6)-1;
x(i)=(init(1))+i*h; % Moves along the x step sizes of the mesh.
end
for j=1:init(5)-1;
y(j)=init(3)+j*k; % Moves along the y step sizes of the mesh.
end
%% The functions t,l,b,r define the boundary conditions of the rectangle.
The function f is the forcing function for Poisson's equation.
w=zeros(init(5)-1,init(6)-1);
z=zeros(init(5)-1,init(6)-1);
lambda=(h^2)/(k^2); % Define variables for use
of the finite difference method.
mu=2*(1+lambda);
counter=0;

f=@(x,y) 0; % Function f that is on the
right hand side of Poisson's Equation
tf=@(x,y) sin(x*4*pi); % t top function
lf=@(x,y) 0; % left hand function
bf=@(x,y) 0; % bottom function
rf=@(x,y) 0; % right hand function
%% Solves for the w(i,j), the solution to Poisson's equation at each point
within the mesh.

while counter<=init(7); %Sets a maximum number of iterations.

% Calculates the value of w at the top left hand corner of the mesh.
z(1,1)=(-h^2*f(x(1),y(end))+lambda*lf(init(1),y(end))+tf(x(1),init(4))+
lambda*w(1,2)+w(2,1))/mu;

```

```

% Calculates the value of w at the top interior of the mesh.
for i=2:init(6)-2;
    z(1,i)=(-h^2*f(x(i),y(end))+tf(x(i),init(4))+lambda*w(1,i+1)+lambda*w
        (1,i-1)+w(2,i))/mu;
end

% Calculates the value of w at the top right hand corner of the mesh.
z(1,init(6)-1)=(-h^2*f(x(end),y(end))+lambda*rf(init(2),y(end))+tf(x(end),
    init(4))+lambda*w(1,init(6)-2)+w(2,init(6)-1))/mu;

% Calculates the value of w along the right interior border of the mesh.
count1=length(y);
for i=2:init(5)-2;
    count1=count1-1;
    z(i,init(6)-1)=(-h^2*f(x(end),y(count1))+lambda*rf(init(2),y(count1))+
        w(i-1,init(6)-1)+w(i+1,init(6)-1)+lambda*w(i,init(6)-2))/mu;
end

% Calculates the values of w at the bottom right hand corner of the mesh.
z(init(5)-1,init(6)-1)=(-h^2*f(x(end),y(1))+lambda*rf(init(2),y(1))+bf(x(
    end),init(3))+w(init(5)-2,init(6)-1)+lambda*w(init(5)-1,init(6)-2))/mu
;

% Calculates the value of w at the bottom interior of the mesh.
for i=2:init(6)-2;
    z(init(5)-1,i)=(-h^2*f(x(i),y(1))+bf(x(i),init(3))+w(init(5)-2,i)+
        lambda*w(init(5)-1,i+1)+lambda*w(init(5)-1,i-1))/mu;
end

% Calculates the value of w on the bottom left hand corner of the mesh.
z(init(5)-1,1)=(-h^2*f(x(1),y(1))+lambda*lf(init(1),y(1))+bf(x(1),init(3))
    +w(init(5)-2,1)+lambda*w(init(5)-1,2))/mu;

% Calculates the value of w on the interior left border of the mesh.
count2=length(y);
for i=2:init(5)-2;
    count2=count2-1;
    z(i,1)=(-h^2*f(x(1),y(count2))+lambda*lf(init(1),y(count2))+w(i-1,1)+
        lambda*w(i,2)+w(i+1,1))/mu;
end

% Calculates the value of w on the interior of the mesh.
count3=length(y);
for i=2:init(5)-2;
    count3=count3-1;
    for j=2:init(6)-2;
        z(i,j)=(-h^2*f(x(j),y(count3))+w(i-1,j)+lambda*w(i,j+1)+w(i+1,j)+
            lambda*w(i,j-1))/mu;
    end
end

NORM=norm(w-z);
NORM1=max(NORM(:));
w=z;

if NORM<init(8)
    counter

```

```

        break
    end
    counter=counter+1;
end
%% Computes the solution w(i,j) at every point, including the boundary of
the mesh.
mesh=zeros(size(w)+2);
mesh(2:end-1,2:end-1) = w;

for j=2:length(mesh)-1;
    mesh(1,j)=sin(4*pi*x(j-1)); %Uses boundary condition to compute the
value at the top portion of the matrix.
end
%% Calculating the the exact solution on the given rectangular domain.
n=2;
[xx,yy]=meshgrid(init(1):h:init(2),init(4):-k:init(3));
Solution=sin((xx.*2*pi)/init(2)).*(2/init(2)).*exp(((yy-init(4)).*n*pi)/
init(2)).*((1-exp((yy.*-2*n*pi)/init(2)))./(1-exp((-2*n*pi*init(4))/
init(2)))).*(1/4);
%% Plots the Approximate Solution over the domain.
figure(1)
[xxx,yyy]=meshgrid(init(1):h:init(2),init(4):-k:init(3));
surf(xxx,yyy,mesh)
xlabel('x')
ylabel('y')
title('Approximate Solution')
shading interp
colorbar
view(2)
%% Plot the exact solution
figure(2)
surf(xx,yy,Solution)
xlabel('x')
ylabel('y')
title('Exact Solution')
shading interp
colorbar
view(2)
%% Calculates the error difference between the approximate solution and
the exact answer computed above.
approximatesolution=reshape(mesh.',[],1);
exactsolution=reshape(Solution.',[],1);

NORM=norm(abs(approximatesolution-exactsolution),2);
relative_error=NORM/(norm(exactsolution,2))
toc

```

5.6 Derivation 'Scheme 2' of 7-Point Stencil

Similar techniques are applied in order to construct a scheme for points $P_2, P_5, P_3,$ and P_6

$$\begin{aligned} p(u_0 - h_1, v_0 + k_2) &= p(u_0, v_0) - p_u(u_0, v_0)h_1 + p_v(u_0, v_0)k_2 + \\ &\quad \frac{1}{2} [p_{uu}(u_0, v_0)h_1^2 - 2p_{uv}(u_0, v_0)h_1k_2 + p_{vv}(u_0, v_0)k_2^2] \end{aligned} \quad (13)$$

$$\begin{aligned} p(u_0 - h_1, v_0 - k_2) &= p(u_0, v_0) - p_u(u_0, v_0)h_1 - p_v(u_0, v_0)k_2 + \\ &\quad \frac{1}{2} [p_{uu}(u_0, v_0)h_1^2 + 2p_{uv}(u_0, v_0)h_1k_2 + p_{vv}(u_0, v_0)k_2^2] \end{aligned} \quad (14)$$

$$\begin{aligned} p(u_0 + h_2, v_0 + k_2) &= p(u_0, v_0) - p_u(u_0, v_0)h_2 + p_v(u_0, v_0)k_2 + \\ &\quad \frac{1}{2} [p_{uu}(u_0, v_0)h_2^2 + 2p_{uv}(u_0, v_0)h_2k_2 + p_{vv}(u_0, v_0)k_2^2] \end{aligned} \quad (15)$$

$$\begin{aligned} p(u_0 + h_2, v_0 - k_2) &= p(u_0, v_0) - p_u(u_0, v_0)h_2 - p_v(u_0, v_0)k_2 + \\ &\quad \frac{1}{2} [p_{uu}(u_0, v_0)h_2^2 - 2p_{uv}(u_0, v_0)h_2k_2 + p_{vv}(u_0, v_0)k_2^2] \end{aligned} \quad (16)$$

We combine equations (13)-(16) by eliminating terms proportional to p_u and p_v to find

$$\begin{aligned} h_2p(u_0 - h_1, v_0 + k_2) + h_1p(u_0 + h_2, v_0 + k_2) + h_2p(u_0 - h_1, v_0 - k_2) + h_1p(u_0 + h_2, v_0 - k_2) = \\ 2(h_1 + h_2)p(u_0, v_0) + h_2h_1(h_1 + h_2)p_{uu}(u_0, v_0) + (h_2 + h_1)k_2^2p_{vv}(u_0, v_0) \end{aligned}$$

or,

$$h_2P_C + h_1P_E + h_2P_1 + h_1P_4 = 2(h_2 + h_1)P_2 + h_2h_1(h_1 + h_2)P_{uu} \quad (17)$$

For points A and 5 , the pressure is approximated by

$$\begin{aligned} p(u_0 - 2h_1, v_0) &= p(u_0, v_0) - p_u(u_0, v_0)2h_1 + \frac{1}{2}p_{uu}(u_0, v_0)(2h_1)^2 \\ p(u_0 + 2h_2, v_0) &= p(u_0, v_0) + p_u(u_0, v_0)2h_2 + \frac{1}{2}p_{uu}(u_0, v_0)(2h_2)^2 \end{aligned}$$

Which, in terms of our points, is;

$$\begin{aligned} P_A &= P_2 - 2h_1P_{2u} + 2h_1^2P_{uu} \\ P_5 &= P_2 + 2h_2P_{2u} + 2h_2^2P_{uu} \end{aligned}$$

We take $h_2P_A + h_1P_5$ and solve for P_{2uu}

$$P_{uu} = \frac{h_2P_A + h_1P_5 - (h_1 + h_2)P_2}{2(h_2h_1^2 + h_1h_2^2)} \quad (18)$$

Now, returning to (17) where, once again, in order to obtain the desired result, a factor of zero is added, in the form of $(h_1 + h_2)k_2^2P_{uu} - (h_1 + h_2)k_2^2P_{uu}$.

$$(h_1 + h_2)k_2^2 [P_{uu} + P_{vv}] + (h_1 + h_2)h_1h_2P_{2uu} - (h_1 + h_2)k_2^2P_{uu} = h_2P_C + h_1P_E + h_2P_1 + h_1P_4 - 2(h_1 + h_2)P_2$$

The value for P_{uu} expressed in equation 18 is substituted into the equation above, which is then solved for the Laplacian.

$$\begin{aligned} \nabla^2 P &= \left[h_2P_C + h_1P_E + h_2P_1 + h_1P_4 - 2 \left[(h_2 + h_1) - \frac{h_2h_1(h_1 + h_2)^2 - (h_2 + h_1)^2k_2^2}{4(2h_2h_1^2 + h_1h_2^2)} \right] \right] \\ &\quad - \frac{(h_1 + h_2)(h_2h_1 - k_2^2) [h_2P_A + h_1P_5]}{2h_2h_1^2 + h_1h_2^2} \Big] / (h_2 + h_1)k_2^2 \end{aligned} \quad (19)$$

5.7 Application of 7-Point Stencil

We now consider how the stencil may be directly applied to compute the solution. We begin by using the scheme 1 equations, which are given below. The left hand side equations;

$$\begin{aligned}
P_{6,11,16} &= k_{3,2,1}C_{4,3,2} + k_{4,3,2}C_{3,2,1} + k_{3,2,1}P_{4,9,14} + k_{4,3,2}P_{9,14,19} \\
&- 2 \left[(k_{3,2,1} + k_{4,3,2}) - \frac{(k_{3,2,1} + k_{4,3,2})^2(k_{3,2,1}k_{4,3,2} - h_1^2)}{4(k_{3,2,1}k_{4,3,2}^2 + k_{4,3,2}k_{3,2,1}^2)} \right] P_{6,11,16} \\
&- \frac{(k_{3,2,1} + k_{4,3,2})(k_{3,2,1}k_{4,3,2} - h_1^2)(k_{3,2,1}P_{1,6,11} + k_{4,3,2}P_{11,16,21})}{2(k_{3,2,1}k_{4,3,2}^2 + k_{4,3,2}k_{3,2,1}^2)}
\end{aligned}$$

The interior equations,

$$\begin{aligned}
P_{7,12,17} &= k_{3,2,1}P_{9,14,19} + k_{4,3,2}C_{9,14,19} + k_{3,2,1}P_{5,10,15} + k_{4,3,2}P_{10,15,20} \\
&- 2 \left[(k_{3,2,1} + k_{4,3,2}) - \frac{(k_{3,2,1} + k_{4,3,2})^2(k_{3,2,1}k_{4,3,2} - h_2^2)}{4(k_{3,2,1}k_{4,3,2}^2 + k_{4,3,2}k_{3,2,1}^2)} \right] P_{7,12,17} \\
&- \frac{(k_{3,2,1} + k_{4,3,2})(k_{3,2,1}k_{4,3,2} - h_2^2)(k_{3,2,1}P_{2,7,12} + k_{4,3,2}P_{12,17,22})}{2(k_{3,2,1}k_{4,3,2}^2 + k_{4,3,2}k_{3,2,1}^2)}
\end{aligned}$$

The right hand side equations;

$$\begin{aligned}
P_{8,13,18} &= k_{3,2,1}P_{5,10,15} + k_{4,3,2}P_{10,15,20} + k_{3,2,1}D_{4,3,2} + k_{4,3,2}D_{3,2,1} \\
&- 2 \left[(k_{3,2,1} + k_{4,3,2}) - \frac{(k_{3,2,1} + k_{4,3,2})^2(k_{3,2,1}k_{4,3,2} - h_3^2)}{4(k_{3,2,1}k_{4,3,2}^2 + k_{4,3,2}k_{3,2,1}^2)} \right] P_{8,13,18} \\
&- \frac{(k_{3,2,1} + k_{4,3,2})(k_{3,2,1}k_{4,3,2} - h_3^2)(k_{3,2,1}P_{3,8,13} + k_{4,3,2}P_{13,18,23})}{2(k_{3,2,1}k_{4,3,2}^2 + k_{4,3,2}k_{3,2,1}^2)}
\end{aligned}$$

We now consider the scheme 2 equations, left hand equations,

$$\begin{aligned}
P_{4,9,14,19} &= h_2P_{1,6,11,16} + h_1P_{2,7,12,17} + h_2P_{6,11,16,21} + h_1P_{7,12,17,22} \\
&- 2 \left[(h_1 + h_2) - \frac{(h_1 + h_2)^2(h_1h_2 - k_{4,3,2,1}^2)}{4(h_1h_2^2 + h_2h_1^2)} \right] P_{4,9,14,19} \\
&- \frac{(h_1 + h_2)(h_1h_2 - k_{4,3,2,1}^2)(h_1P_{5,10,15,20} + h_2C_{4,3,2,1})}{2(h_1h_2^2 + h_2h_1^2)}
\end{aligned}$$

and the scheme 2 right hand equations,

$$\begin{aligned}
P_{5,10,15,20} &= h_3P_{2,7,12,17} + h_2P_{3,8,13,18} + h_3P_{7,12,17,22} + h_2P_{8,13,18,23} \\
&- 2 \left[(h_2 + h_3) - \frac{(h_2 + h_3)^2(h_2h_3 - k_{4,3,2,1}^2)}{4(h_2h_3^2 + h_3h_2^2)} \right] P_{5,10,15,20} \\
&- \frac{(h_2 + h_3)(h_2h_3 - k_{4,3,2,1}^2)(h_2D_{4,3,2,1} + h_3P_{4,9,14,19})}{2(h_2h_3^2 + h_3h_2^2)}
\end{aligned}$$

Now, all that is left is to take periodicity into account;

$$P_1 \cdot k_4 C_1 + k_1 C_4 + k_4 P_{19} + k_1 P_4 - 2 \left[(k_1 + k_4) - \frac{(k_1 + k_4)^2(k_1 k_4 - h_1^2)}{4(k_1 k_4^2 + k_4 k_1^2)} \right] P_1 + \frac{(k_4 - k_1)(k_1 k_4 - h_1^2)(k_4 P_{16} + k_1 P_6)}{2(k_1 k_4^2 + k_4 k_1^2)}$$

$$\begin{aligned}
P_2.k_4P_{19}+k_1P_4+k_4P_{20}+k_1P_5-2\left[(k_1+k_4)-\frac{(k_1+k_4)^2(k_1k_4-h_2^2)}{4(k_1k_4^2+k_4k_1^2)}\right]P_1-\frac{(k_4+k_1)(k_1k_4-h_1^2)(k_4P_{16}+k_1P_6)}{2(k_1k_4^2+k_4k_1^2)} \\
P_3.k_4P_{20}+k_1P_5+k_4D_1+k_1D_4-2\left[(k_1+k_4)-\frac{(k_1+k_4)^2(k_1k_4-h_3^2)}{4(k_1k_4^2+k_4k_1^2)}\right]P_1-\frac{(k_4+k_1)(k_1k_4-h_3^2)(k_4P_{16}+k_1P_6)}{2(k_1k_4^2+k_4k_1^2)} \\
P_1 = P_{21} \\
P_2 = P_{22} \\
P_3 = P_{23}
\end{aligned}$$

Using the above equations, we place the constants from each equation into matrix A , and the corresponding known boundary points into matrix b .

$$\begin{bmatrix} -2[-] & 0 & 0 & k_1 & \dots & 0 \\ 0 & -2[-] & 0 & k_1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & -2[-] & 0 & 0 & \dots & 0 \end{bmatrix} \times \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ \vdots \\ \vdots \\ P_{23} \end{bmatrix} = \begin{bmatrix} -(k_4C_1+k_1C_4) \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

5.8 Matlab code for computing the radius and theta for the 7-Point Stencil

```

clc
clear all
format long
clf

r=2;
n=3;
a=0;
b=2*pi;
c=pi/10;
theta=[a:c:b];
num_pts=(b-a)/(c);
x=r*cos(theta);
y=r*sin(theta);
p=[x' y'];

width1=.25;
width2=.6;
width3=1;

ro=r;
rmo=ro-width1;
rmi=ro-width2;
ri=ro-width3;

t=[a+(b-a)/(2*(num_pts-1)):(b-a)/(num_pts-1):b-(b-a)/(2*(num_pts-1))];
%Pressure=[(3+cos(t')) ones(length(x)-2,1)];
Pressure=[3*ones(length(x)-2,1) ones(length(x)-2,1)];
m=length(p);
xdiff=p(end,1)-p(1,1);
ydiff=p(end,2)-p(1,2);
Ptnp2=[xdiff+p(2,1) ydiff+p(2,2)];
Pt0m1=[p(end-1,1)-xdiff p(end-1,2)-ydiff];
p=[Pt0m1; p; Ptnp2];
%% Initialize matrices
A1=zeros(); A2=zeros(); A=zeros(); C=zeros(); A_sub=zeros(); C_sub=zeros();
;
center=zeros(); radius0=zeros(); chordlength=zeros(); theta=zeros(); u=
zeros(); v=zeros();
%% Solve for center and radius
for i=1:length(p)-1
    A1(i)=p(i+1,1)-p(i,1);
    A2(i)=p(i+1,2)-p(i,2);
    A=[(A1)' (A2)'];
    C(i)=(1/2)*(p(i+1,1)^2-p(i,1)^2+p(i+1,2)^2-p(i,2)^2);
end
C=C';

for j=1:length(p)-2
    A_sub=A(j:j+1,1:2);
    C_sub=C(j:j+1,1);
    center1=(A_sub)\(C_sub);
    centerx(j,1)=center1(1,1);
    centery(j,1)=center1(2,1);
    radius(j,1)=((p(j+1,1)-center1(1))^2+(p(j+1,2)-center1(2))^2)^(1/2);

```

```

end
center=[centerx centery];
%% Calculating theta
for i=1:length(radius)
    chordlength(i,[1:3])=sqrt((p(i+[0:2],1)-center(i,1)-radius(i)).^2+(p(i
        +[0:2],2)-center(i,2)).^2);
end

for k=1:length(radius)
    theta(k,[1:3])=acos(1-(((chordlength(k,[1:3])).^2)/(2*radius(k).^2)));
end
%% Output values for the center, radius, and theta.
center=[centerx centery];
radius1=[radius radius radius];
radius2=radius1-width1;
radius3=radius1-width2;
radius4=radius1-width3;
%% Conformal map the radius and theta points.
for i=1:length(radius)
    u1(i,[1:3])=log(radius1(i,[1:3]));
    u2(i,[1:3])=log(radius2(i,[1:3]));
    u3(i,[1:3])=log(radius3(i,[1:3]));
    u4(i,[1:3])=log(radius4(i,[1:3]));
end
for i=1:length(theta)
    v1(i,[1:3])=theta(i,[1:3]);
    v2(i,[1:3])=theta(i,[1:3]);
    v3(i,[1:3])=theta(i,[1:3]);
    v4(i,[1:3])=theta(i,[1:3]);
end
%% Radii which are larger than a given tolerance are not conformally
    mapped.
r=find(abs(radius)>1000);
if length(r)>0
    dist(r,2)=sqrt((p(r+2,1)-p(r+1,1)).^2+(p(r+2,2)-p(r+1,2)).^2);
    dist(r,1)=sqrt((p(r+1,1)-p(r,1)).^2+(p(r+1,2)-p(r,2)).^2);
    u1(r,[1:3])=1;
    v1(r,1)=p(r,2);
    v1(r,2)=p(r,2)-dist(r,2);
    u2(r,[1:3])=u1(r,[1:3])-width1;
    v1(r,3)=p(r,2)-dist(r,1)-dist(r,2);
    v2(r,[1:3])=v1(r,[1:3]);
    u3(r,[1:3])=u1(r,[1:3])-width2;
    v3(r,[1:3])=v1(r,[1:3]);
    u4(r,[1:3])=u1(r,[1:3])-width3;
    v4(r,[1:3])=v1(r,[1:3]);
end
%% Solve for the pressure using the finite difference method.
u=[u1(:,1) u2(:,1) u3(:,1) u4(:,1)];
v=[v1];
P=Three_Layer_FD_CS(u(2:end-1,:),v(2:end-1,:),Pressure,m);
%% The approximate solution is plotted, as well as compared with the exact
    solution on the annulus
R=[radius1(:,1) radius2(:,1) radius3(:,1) radius4(:,1)];
Exact_Radius=Map_Centerline_CS(R,theta,u,n);

gbottom= @(s) 1+(s-s);
gtop= @(s) 3+(s-s);

```

```

theta=linspace(a,b, num_pts);

R=[ri; Exact_Radius(1,3); rmi; Exact_Radius(1,2); rmo; Exact_Radius(1,1);
ro];

for k=1:num_pts
r((2*n-1)*(k-1)+(1:n))=R(2*(1:n));
theta2((2*n-1)*(k-1)+(1:n))=theta(k);
end

t=[a+(b-a)/(2*(num_pts-1):(b-a)/(num_pts-1):b-(b-a)/(2*(num_pts-1))];

for jj=1:num_pts-1
theta2((2*n-1)*jj-n+(2:n))=t(jj);
r((2*n-1)*jj-n+(2:n))=R(2*(1:n-1)+1);
end

r=r';
theta2=theta2';

%Exact solution on the annulus from separation of variables
f1=@(s,m) cos(m*s);
f2=@(s,m) ((3+(s-s)).*cos(m*s));
f3=@(s,m) sin(m*s);
f4=@(s,m) ((3+(s-s)).*cos(m*s));

nn=1;
Const_init=[1 log(ri); 1 log(ro)]\[1/(2*pi) * integral(gbottom, 0, 2*pi);
1/(2*pi) * integral(gtop, 0, 2*pi)];
a0=Const_init(1);
b0=Const_init(2);
Const_ab=[ri^nn ri^(-nn); ro^nn ro^(-nn)]\[1/pi * integral(@(s) f1(s,nn)
,0,2*pi); 1/pi * integral(@(s) f2(s,nn),0,2*pi)];
a=Const_ab(1);
b=Const_ab(2);
Const_cd=[ri^nn ri^(-nn); ro^nn ro^(-nn)]\[1/pi * integral(@(s) f3(s,nn)
,0,2*pi); 1/pi * integral(@(s) f4(s,nn),0,2*pi)];
c=Const_cd(1);
d=Const_cd(2);

Exact_Sol=a0+b0*log(r)+(a*r.^nn+b*r.^(-nn)).*(cos(nn*theta2))+(c*r.^nn+d*r
.^(-nn)).*(sin(nn*theta2));

figure(2)
plot(theta2,Exact_Sol,'o')
legend('Exact Solution')
xlabel('Domain')
ylabel('Pressure')
title('Approximated Solution')
hold all
g2=3;
plot(theta2, P,'s',theta,g2,'r*',theta,ones(length(theta),1),'r*')

Rel_Error=norm(P-Exact_Sol,2)/norm(Exact_Sol,2)
Abs_Error=norm(P-Exact_Sol,2);

```

5.9 7-Point Stencil Finite Difference Method

```

function [P]=Three_Layer_FD_CS(u,v,Pressure , numpoints)
%% Initialize all parameters.
m=numpoints-2; %Number of given data points along top curve (horizontal
    rows)
n=3; %Number of Layers (vertical columns)

k1 = abs(v(:,2)-v(:,3));
k1=k1./2;
k2 = abs(v(:,1)-v(:,2));
k2=k2./2;

for i=1:n
    h(:,n-(i-1)) = abs(u(:,i)-u(:,i+1));
    h(:,n-(i-1)) = h(:,n-(i-1))./2;
end

%Initialize left , and right boundary conditions.
C = Pressure(1:end,2);
C=flipud(C);
D = Pressure(1:end,1);
D=flipud(D);
%% Build the system of equations Ax=b
% Building the b matrix , which consists of known boundary values.
b = zeros((m+1)*n+m*(n-1),1);
%Constant terms from each first scheme equation
b((2*n-1)*(1:m-1)+1)=-k1(m-1:-1:1).*C(m-1:-1:1)+k2(m-1:-1:1).*C(m-1:-1:1);
b((2*n-1)*(1:m-1)+n)=-k1(m-1:-1:1).*D(m-1:-1:1)+k2(m-1:-1:1).*D(m-1:-1:1);
%Constant terms from each second scheme equation
b((2*n-1)*(1:m)-n+2) = (h((1:m),1)+h((1:m),2)).*(h((1:m),1).*h((1:m),2)-k2
    (m-1:1).^2).*h((1:m),2).*C(m-1:1)./(2*(h((1:m),1).*h((1:m),2).^2+h
    ((1:m),2).*h((1:m),1).^2));
b((2*n-1)*(1:m)) = (h((1:m),2)+h((1:m),3)).*(h((1:m),2).*h((1:m),3)-k2(m
    :-1:1).^2).*h((1:m),2).*D(m-1:1)./(2*(h((1:m),2).*h((1:m),3).^2+h((1:
    m),3).*h((1:m),2).^2));
%Constant terms from periodicity on equations P1-Pn
b(1) = -(k2(end)*C(1)+k1(1)*C(end));
b(n) = -(k2(end)*D(1)+k1(1)*D(end));

% Building the A matrix , which consists of the constant coefficient terms
    on each unknown.
A = zeros((m+1)*n+m*(n-1));
%Constant terms from the first scheme equation
for i=1:(m-1)
    % Left equations
    A((2*n-1)*(i)+1,(2*n-1)*i-n+2)=k1(m-i);
    A((2*n-1)*(i)+1,(2*n-1)*(i+1)-n+2)=k2(m-i);
    A((2*n-1)*(i)+1,(2*n-1)*(i)+1)=-2*(k1(m-i)+k2(m-i))-(((k1(m-i)+k2(m-i))
        ^2*(k1(m-i)*k2(m-i)-h(i,1)^2))/(4*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-
        i)^2)));
    A((2*n-1)*(i)+1,(2*n-1)*(i-1)+1)=-(((k1(m-i)+k2(m-i))*(k1(m-i)*k2(m-i)
        -h(i,1)^2)*k1(m-i))/(2*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-i)^2)));
    A((2*n-1)*(i)+1,(2*n-1)*(i+1)+1)=-(((k1(m-i)+k2(m-i))*(k1(m-i)*k2(m-i)
        -h(i,1)^2)*k2(m-i))/(2*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-i)^2)));
    %Interior equations
    A((2*n-1)*(i)+2,[(2*n-1)*i-n+2 i*(2*n-1)])=k1(m-i);
    A((2*n-1)*(i)+2, [(2*n-1)*(i+1)-n+2 (2*n-1)*(i+1)])=k2(m-i);

```

```

A((2*n-1)*(i)+2,(2*n-1)*(i)+2)=-2*(k1(m-i)+k2(m-i)-(((k1(m-i)+k2(m-i))
^2*(k1(m-i)*k2(m-i)-h(i,2)^2))/(4*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-
i)^2)))));
A((2*n-1)*(i)+2,(2*n-1)*(i-1)+2)=-(((k1(m-i)+k2(m-i))*(k1(m-i)*k2(m-i)
-h(i,2)^2)*k1(m-i))/(2*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-i)^2)));
A((2*n-1)*(i)+2,(2*n-1)*(i+1)+2)=-(((k1(m-i)+k2(m-i))*(k1(m-i)*k2(m-i)
-h(i,2)^2)*k2(m-i))/(2*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-i)^2)));
%Right equations
A((2*n-1)*(i)+n,(2*n-1)*i)=k1(m-i);
A((2*n-1)*(i)+n,(2*n-1)*(i+1))=k2(m-i);
A((2*n-1)*(i)+n,(2*n-1)*(i)+n)=-2*(k1(m-i)+k2(m-i)-(((k1(m-i)+k2(m-i))
^2*(k1(m-i)*k2(m-i)-h(i,3)^2))/(4*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-
i)^2)))));
A((2*n-1)*(i)+n,(2*n-1)*(i-1)+n)=-(((k1(m-i)+k2(m-i))*(k1(m-i)*k2(m-i)
-h(i,3)^2)*k1(m-i))/(2*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-i)^2)));
A((2*n-1)*(i)+n,(2*n-1)*(i+1)+n)=-(((k1(m-i)+k2(m-i))*(k1(m-i)*k2(m-i)
-h(i,3)^2)*k2(m-i))/(2*(k1(m-i)*k2(m-i)^2+k2(m-i)*k1(m-i)^2)));
end

%Constant terms from the second scheme equation
for i=1:m
A((2*n-1)*i-n+2,[(2*n-1)*(i-1)+1 (2*n-1)*i+1]) = h(i,2);
A((2*n-1)*i-n+2,[(2*n-1)*(i-1)+2 (2*n-1)*i+2]) = h(i,1);
A((2*n-1)*i-n+2,(2*n-1)*i-n+2) = -2*( (h(i,1)+h(i,2))-h(i,1)+h(i,2))
^2*(h(i,1)*h(i,2)-k2(m-(i-1))^2)/(4*((h(i,1)*h(i,2)^2+h(i,2)*h(i,1)
^2)))));
A((2*n-1)*i-n+2,(2*n-1)*i) = -(h(i,1)+h(i,2))*(h(i,1)*h(i,2)-k2(m-(i
-1))^2)*h(i,1)/(2*(h(i,1)*h(i,2)^2+h(i,2)*h(i,1)^2));
A((2*n-1)*i,[(2*n-1)*(i-1)+2 (2*n-1)*i+2]) = h(i,3);
A((2*n-1)*i,[(2*n-1)*(i-1)+n (2*n-1)*i+n]) = h(i,2);
A((2*n-1)*i,(2*n-1)*i) = -2*((h(i,2)+h(i,3))-h(i,2)+h(i,3))^2*(h(i,2)*
h(i,3)-k2(m-(i-1))^2)/(4*(h(i,2)*h(i,3)^2+h(i,3)*h(i,2)^2));
A((2*n-1)*i,(2*n-1)*i-n+2) = -(h(i,2)+h(i,3))*(h(i,2)*h(i,3)-k2(m-(i-1)
)^2)*h(i,3)/(2*(h(i,2)*h(i,3)^2+h(i,3)*h(i,2)^2));
end
%% Terms using periodicity
for i=1:n
A(m*(2*n-1)+i,i)=1;
A(m*(2*n-1)+i,m*(2*n-1)+i)=-1;
end

for i=1:n-1
A(i,n+i) =k1(1);
A(i,(n-1)*(m-1)+n*m+i)=k2(end);
end

for i=1:n-1
A(i+1,n+i) =k1(1);
A(i+1,(n-1)*(m-1)+n*m+i) =k2(end);
end

for i=1:n
A(i,i)=-2*(k1(1)+k2(end)-(((k1(1)+k2(end))^2*(k1(1)*k2(end)-h(1,i)^2))
/(4*(k1(1)*k2(end)^2+k2(end)*k1(1)^2)))));
end
for i=1:n
A(i,(2*n-1)*(m-1)+i)=-(((k2(end)+k1(1))*(k1(1)*k2(end)-h(1,i)^2)*k2(
end))/(2*(k1(1)*k2(end)^2+k2(end)*k1(1)^2)));

```

```

        A(i,2*n-1+i)=-(((k2(end)+k1(1))*(k1(1)*k2(end)-h(1,i)^2)*k1(1))/(2*(k1
            (1)*k2(end)^2+k2(end)*k1(1)^2));
    end
    %% Solving for exact solution
    Sol=A\b;
    P=Sol;
end

```

5.10 Matlab code for Mapping the centerlines of the rectangular domain onto the original domain

```

function [Exact_Radius] = Map_Centerline_CS(R, theta, u, n)

for i=1:n
    h(:,n-(i-1)) = abs(u(:,i)-u(:,i+1));
    h(:,n-(i-1)) = h(:,n-(i-1))./2;
end

u1=log(R(:,1));
u2=log(R(:,2));
u3=log(R(:,3));
u4=log(R(:,4));
%%Maps the points in our rectangular mesh that correspond to the points on
    each boundary between layers (vertical lines).
v1=(theta(:,1)-theta(:,2))./2 + theta(:,1);
x1=exp(u2).*cos(v1);
y1=exp(u2).*sin(v1);
x2=exp(u3).*cos(v1);
y2=exp(u3).*sin(v1);
%%Maps the points in our rectangular mesh that correspond to the points on
    the centerline of each layer (vertical columns)onto the original
    annulus.
u_m1 = u2+h(:,3);
u_m2 = u3+h(:,2);
u_m3 = u4+h(:,1);
v_m = theta(:,2);
x_m1 = exp(u_m1).*cos(v_m);
y_m1 = exp(u_m1).*sin(v_m);
x_m2 = exp(u_m2).*cos(v_m);
y_m2 = exp(u_m2).*sin(v_m);
x_m3 = exp(u_m3).*cos(v_m);
y_m3 = exp(u_m3).*sin(v_m);

figure(1)
plot(x1,y1,'ms',x2,y2,'cs')
plot(x_m1,y_m1,'gs',x_m2,y_m2,'bs',x_m3,y_m3,'rs')
hold all
axis equal

X=[x_m1 x_m2 x_m3 ];
Y=[y_m1 y_m2 y_m3 ];
Exact_Radius=sqrt(X.^2+Y.^2);
end

```