The Tricomi Equation

Jonathan Olson

Spring 2020

Abstract

The Tricomi Equation is a second order linear partial differential equation of mixed elliptic-hyperbolic type. The general solution to the Tricomi's Equation on the rectangle is obtained using separation of variables, which yields a family of solutions in the form of scaled Airy function. The existence and uniqueness of solutions for boundary value problems and initial value problems are investigated and compared to solutions obtained numerically.

1 Introduction

This paper considers solutions to the second order partial differential equation of mixed elliptic-hyperbolic type known as The Tricomi Equation,

$$u_{xx} + xu_{yy} = 0, (x, y) \in \Omega \subset \mathbb{R}^2. \tag{T}$$

Tricomi's Equation is hyperbolic in the half-plane $x < 0$ and elliptic in the half-plane $x > 0$, as seen in Figure 1. Various domains with boundary conditions can be considered, some of which lead to non-existence of a solution. For the case considered here, see Figure 2, we let $\Omega = (a, b) \times (c, d)$ with $a < 0 < b$, $u(a, y) = 0 = u(b, y)$ and $u(x, c) = g_1(x)$ and $u(x, d) = g_2(x)$ where $g_1(x)$ and $g_2(x)$ are given.

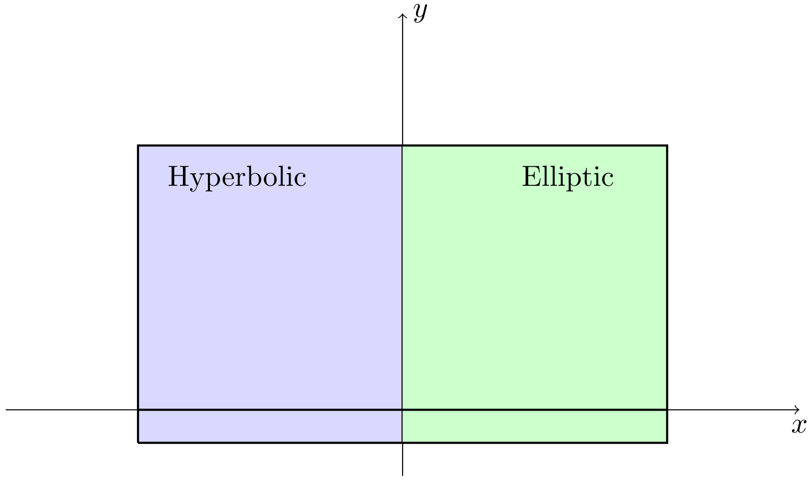

Figure 1: The Tricomi Equation changes classification on either side of the y-axis. The equation is hyperbolic for $x < 0$ and elliptic for $x > 0$.

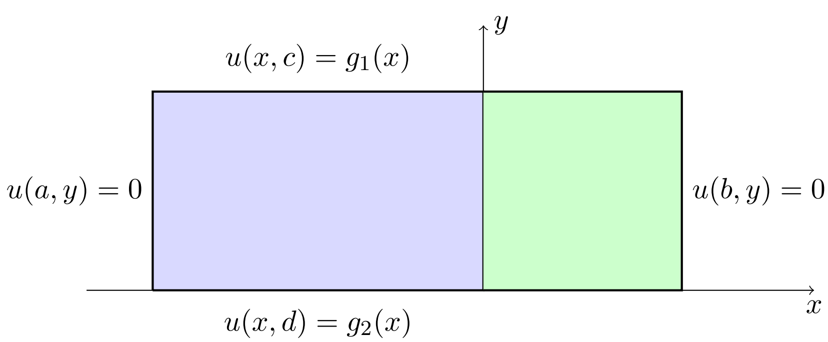

Figure 2: We consider the case with 0-D boundary conditions at $x = a$ and $x = b$ with $a < 0 < b$ and $u(x, c) = g_1(x)$ and $u(x, d) = g_2(x)$, where $g_1(x)$ and $g_2(x)$ are arbitrary

2 Separation Of Variables

With our choice of domain and boundary conditions, exact solution of T can be obtained using separation of variables. For those unfamiliar with the method of separation of variables, I recommend seeing [1] which gives a detailed explanation of the method. Now, if we assume a solution of the form,

$$u(x,y) = \phi(x)h(y) \tag{2}$$

And plug the assumed solution into (T) we obtain, after separating variables, the following system of ODE's,

$$\phi'' - x\lambda\phi = 0 \tag{2.1}$$

$$h'' = -\lambda h \tag{2.2}$$

We consider first a solution for (2.1) by making a change of variables. let $x = \alpha\xi$ with $\alpha = constant$. Which yields,

$$\tilde{\phi}'' - \lambda\alpha^3\xi\tilde{\phi} = 0$$

and set $\lambda\alpha^3 = 1$, to obtain Airy's Equation,

$$\tilde{\phi}'' - \xi\tilde{\phi} = 0 \tag{2.1*}$$

Which implies,

$$\tilde{\phi}(\xi) = \phi\left(\frac{x}{\alpha}\right) = c_1 Ai(\lambda^{1/3}x) + c_2 Bi(\lambda^{1/3}x)$$

If $\lambda < 0$, then (2.2) implies

$$h(y) = sinh(\sqrt{\lambda}y)$$

and for $\lambda > 0$

$$h(y) = sin(\sqrt{\lambda}y)$$

We now enforce boundary conditions on (2.1), namely

$$\phi(a) = 0 = \phi(b)$$

Thus,

$$\phi(a) = c_1 Ai(\lambda^{1/3}a) + c_2 Bi(\lambda^{1/3}a) = 0$$

and

$$\phi(b) = c_1 Ai(\lambda^{1/3}b) + c_2 Bi(\lambda^{1/3}b) = 0$$

implies

$$c_1 Ai(\lambda^{1/3}a) + c_2 Bi(\lambda^{1/3}a) = c_1 Ai(\lambda^{1/3}b) + c_2 Bi(\lambda^{1/3}b)$$

$$c_2 = -c_1 \frac{Ai(\lambda^{1/3}a)}{Bi(\lambda^{1/3}a)} = -c_1 \frac{Ai(\lambda^{1/3}b)}{Bi(\lambda^{1/3}b)}$$

$$c_1(Ai(\lambda^{1/3}a)Bi(\lambda^{1/3}b) - Ai(\lambda^{1/3}b)Bi(\lambda^{1/3}a)) = 0$$

Therefore, either $c_1 = 0$ or

$$Ai(\lambda^{1/3}a)Bi(\lambda^{1/3}b) - Ai(\lambda^{1/3}b)Bi(\lambda^{1/3}a) = 0$$

If $c_1 = 0$, then $c_2 = 0$ and $\phi(x) \equiv 0$. In order to obtain a non-trivial solution, set $c_1 = 1$ and define the function,

$$F(\lambda^{1/3}, a, b) = Ai(\lambda^{1/3}a)Bi(\lambda^{1/3}b) - Ai(\lambda^{1/3}b)Bi(\lambda^{1/3}a) = 0 \tag{2.3}$$

Where the roots of F are the eigenvalues. The roots of Airy's functions only occur for $x < 0$ and can only be determined numerically. Choosing $a = -3$ and $b = 1$ yields,

$$F(\lambda^{1/3}, -3, 1) = Ai(-3\lambda^{1/3})Bi(\lambda^{1/3}) - Ai(\lambda^{1/3})Bi(-3\lambda^{1/3}) = 0$$

Which is only a function of $\lambda^{1/3}$. Listing [1] utilizes a rootfinder and for loop in MatLab to find the roots of and determine the eigenvalues.

```
% Set boundries,
a = -3; b = 1;
crl = linspace(-10,10,1001);

% Define F as a function of lam^1/3
F =@(crl) tanh(airy(a*crl).*airy(2,b*crl) -airy(b*crl).*airy(2,a*crl));
lam = zeros(length(crl),1);

% Solve for roots of F
for i = 2:length(crl)
        if f(crl(i-1))*f(crl(i)) < 0
            L = (crl(i)+crl(i+1))/2;
            lam(i) = fzero(f,L);
        end
end

```

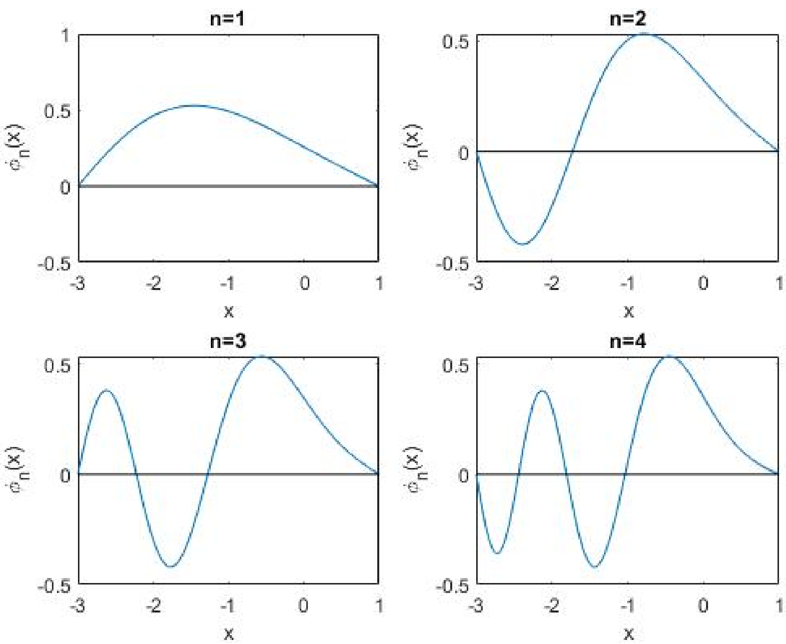

Figure 3: Plots for first four positive eigenfunctions

```
% Eigenvalues
index = find(lam~=0);
crL = lam(index);
Lam = crL.^3;
```

Listing 1: First define the domain of F $\lambda^{1/3} \in (-10, 10)$. If F changes sign from λ_i to λ_{i+1}, then a root of F exists somewhere between λ_i and λ_{i+1}. The `for loop` tests for where F changes sign then takes the midpoint of those values and feeds it into the rootfinder as an initial guess.

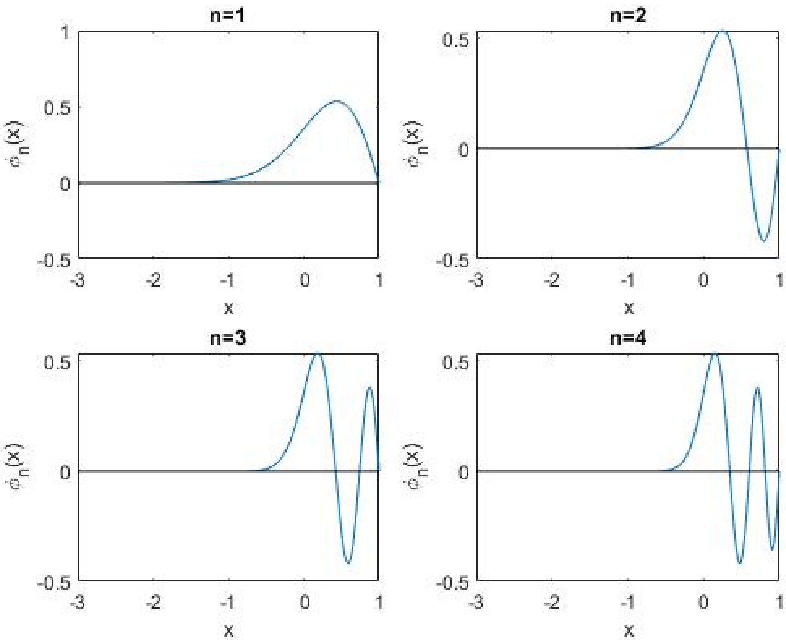

Figure 4: First four negative eigenfunctions

The process found a total of 41 eigenvalues, both positive and negative, on the interval $(-10, 10)$, the first nine of which are listed in Table 1 in increasing absolute value. Plot for the first four positive eigenfunctions are depicted in Figure 3, and Figure 4 shows plots for the first four negative eigenfunctions. Since there are eigenvalues for both $\lambda < 0$ and $\lambda > 0$, we now define a new function $h(\lambda, y)$ such that,

$$h(\lambda, y) = \begin{cases} \frac{sin\left(\sqrt{\lambda}y\right)}{sin\left(\sqrt{\lambda}\right)} & \text{if } \lambda > 0 \\ \\ \frac{sinh\left(\sqrt{\lambda}y\right)}{sinh\left(\sqrt{\lambda}\right)} & \text{if } \lambda < 0 \end{cases} \tag{2.4}$$

Therefore, the general solution to (T) with zero-Dirichlet for $x \in (a, b)$ is given by,

$$u(x, y) = \sum \left(a_k Ai(\lambda_k^{1/3} x) + b_k Bi(\lambda_k^{1/3} x) \right) h_k(\lambda_k, y) \tag{2.5}$$

n	1	2	3	4	5	6	7	8	9
λ_n	.5375	2.5775	6.2548	11.5873	-12.7818	18.5722	27.5027	37.4857	62.9818

Table 1: Both negative and positive eigenvalues of ϕ were found on the interval $(-10, 10)$ using rootfinding procedure. First nine are listed in increasing absolute value.

3 Numerical Solver

The numerical solution to Tricomi's Equation utilizes the block matrix method outlined in [2]. Let N be the number of point-grid points for x, and M be the number of point-grid points for y. Let `xs` and `ys` be the point-grid points for $x \in (-3, 1)$ and $y \in (0, 1)$, respectively. Then let `T1` and `T2` be the $N \times N$ and $M \times M$ point-grid second difference matrices enforcing 0-D boundary conditions with `Im=speye(M)` over the intervals. We can then define the system matrix $L \in M^{NM \times NM}$ for the Tricomi Equation. Listings [2] and [3] build the left and right hand sides of the system, respectively.

```
% Tricomi matrix
Im = speye(M);
xd = spdiags(xs',0,N,N);
T1 = spdiags(kron([1 -2 1],ones(N,1)),-1:1,N,N);
T2 = spdiags(kron([1 -2 1],ones(M,1)),-1:1,M,M);
L = kron(Im,T1)/dx^2 + kron(T2,xd)/dy^2;
```

Listing 2: Builds left-hand side of the system, enforcing 0-D boundary conditions at $x = -3$ and $x = 1$. `xd` is an $N \times N$ sparce diagonal matrix which embeds data for the x variable in Tricomi's equation, which is absent in the standard Laplacian matrix used for Laplaces' equation

```
% Build right-hand side
R = zeros(N,M);
R(:,end) = R(:,end) -xs'.*g1(xs')/dy^2;
R = reshape(R,(N)*(M),1);
```

Listing 3: Builds the right-hand side of the Tricomi system. Boundary data contained in $g(x)$ is moved to the right-hand side of the system, then reshaped to an $NM \times 1$ column vector

After constructing the left and right hand sides, the system is solved using *MatLab's* built-in linear solver '\', then reshaped. Boundary conditions are added back, and the `Meshgrid` command is used to provide x and y-coordinates for surface plots.

```
% Solve system
u1 = L\R;
u1 = [g2(x'),[zeros(1,M); full(reshape(u1,N,M));
   zeros(1,M)], g1(x')];
```

Figures 5,6 and 7 show surface plots for the solution of Tricomi's Equation on the rectangle for three different boundary value problems.

$$1) g_1(x) = \phi_1(x) = Ai\left(\lambda_1^{1/3}x\right) + c_1 Bi\left(\lambda_1^{1/3}x\right)$$

$$2) g_1(x) = \phi_2(x) = Ai\left(\lambda_2^{1/3}x\right) + c_2 Bi\left(\lambda_2^{1/3}x\right)$$

$$3) g_1(x) = \phi_5(x) = Ai\left(\lambda_5^{1/3}x\right) + c_5 Bi\left(\lambda_5^{1/3}x\right)$$

It is clear from the graphics that the solutions obtained using block matrix method are in very close agreement with the exact solutions obtained using separation of variables.

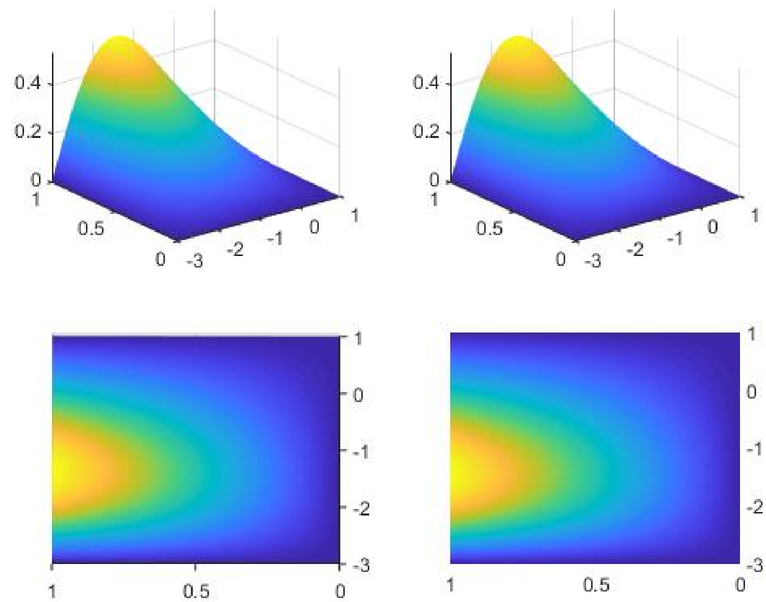

Figure 5: Numerical, (left) and separation of variables, (right) solutions are given side-by-side.

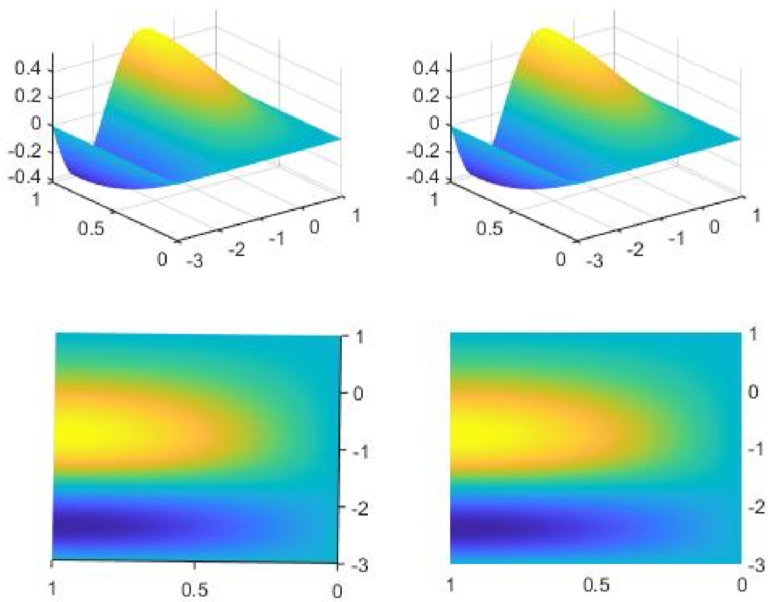

Figure 6: Numerical and separation of variables solution for the Tricomi Equation with 0-D boundary conditions for $x \in (-3, 1)$ and $u(x, 0)$ with $u(x, 1) = \phi_2(x)$

.

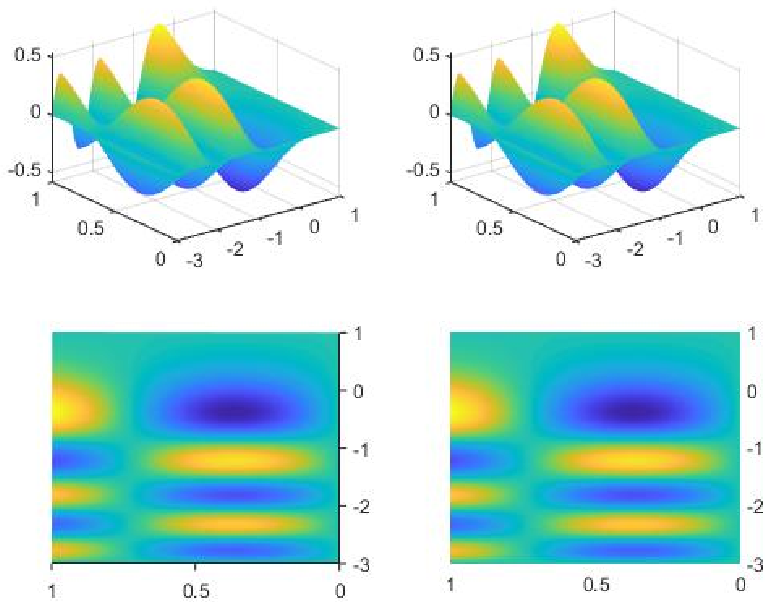

Figure 7: Numerical and separation of variables solution for the Tricomi Equation with 0-D boundary conditions for $x \in (-3, 1)$ and $u(x, 0)$ with $u(x, 1) = \phi_5(x)$.

4 Future Considerations

There are many interesting properties to the Tricomi Equation which have not been discussed here. For example, the issues with existence and uniqueness of solutions. For more detail on the subject, as well as the history of the problem I recommend seeing [3]. Additionally, there is also the consideration of physical applications for the Tricomi Equation. In addition to a wealth of information of Airy functions and their applications, [4] also gives an example of how the Tricomi Equation can be used in certain areas of fluid dynamics.

References

[1] R. Haberman, *Applied partial differential equations.* Person Education Inc, 2013.

[2] J. M, Neuberger, *Difference matrices for ODE and PDE: a matlab companion*. 2019.

[3] A. R. Manwell, “The tricomi equation with applications to the theory of plane transonic flow,” *NASA STI/Recon Technical Report A*, vol. 80, 1979.

[4] V. Olivier and S. Manuel, *Airy functions and applications to physics*. World Scientific Publishing Company, 2010.