METHODS OF INTEGRATION FOR SOLVING THE SOMMERFELD IDENTITY

Wissam Razouki
CSE 3802

INTRODUCTION

The Problem

The Sommerfeld Identity is given in integral form as:

$$S = \int_0^\infty \frac{\alpha J_0(\alpha r)}{\sqrt{\alpha^2 - k^2}} d\alpha \,, \qquad where\ k = \frac{2\pi f}{c}$$

Using two different methods of integration, we must:

1. Evaluate the integral for $f \in [10\ \text{kHz}, 100\ \text{kHz}]$ in steps of 2 kHz for $r = 10000\ m$
2. Compare the results with the exact solution computed from $S = \frac{e^{-jkr}}{r}$
3. Repeat 1 and 2 for $f \in [500\ \text{kHz}, 2000\ \text{kHz}]$ in steps of 10 kHz for $r = 1000\ m$

Methods Used

Reference method:

- The exact solution

Experimental methods:

- Trapezoid method
- Ruffa’s method

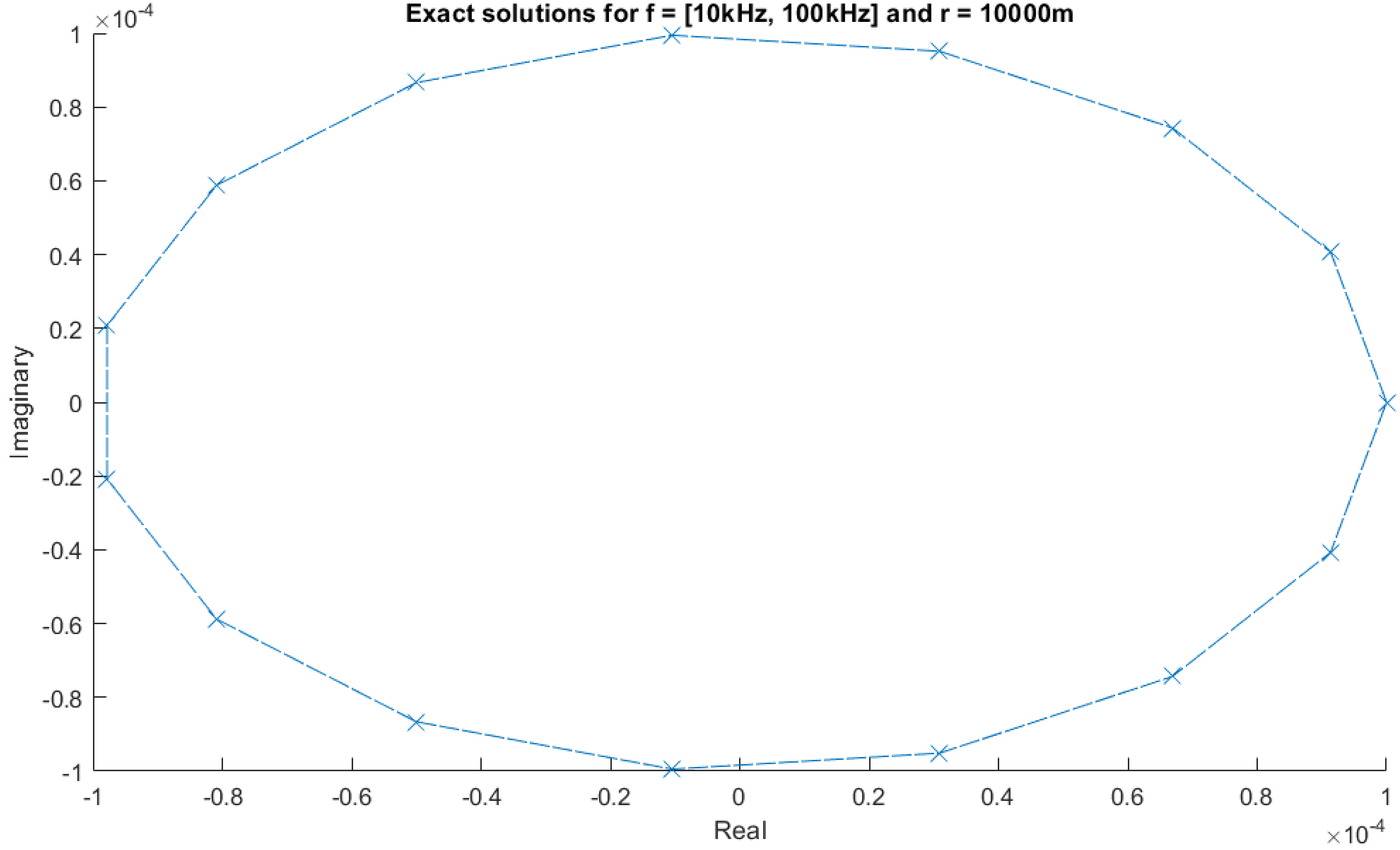
EXACT SOLUTIONS

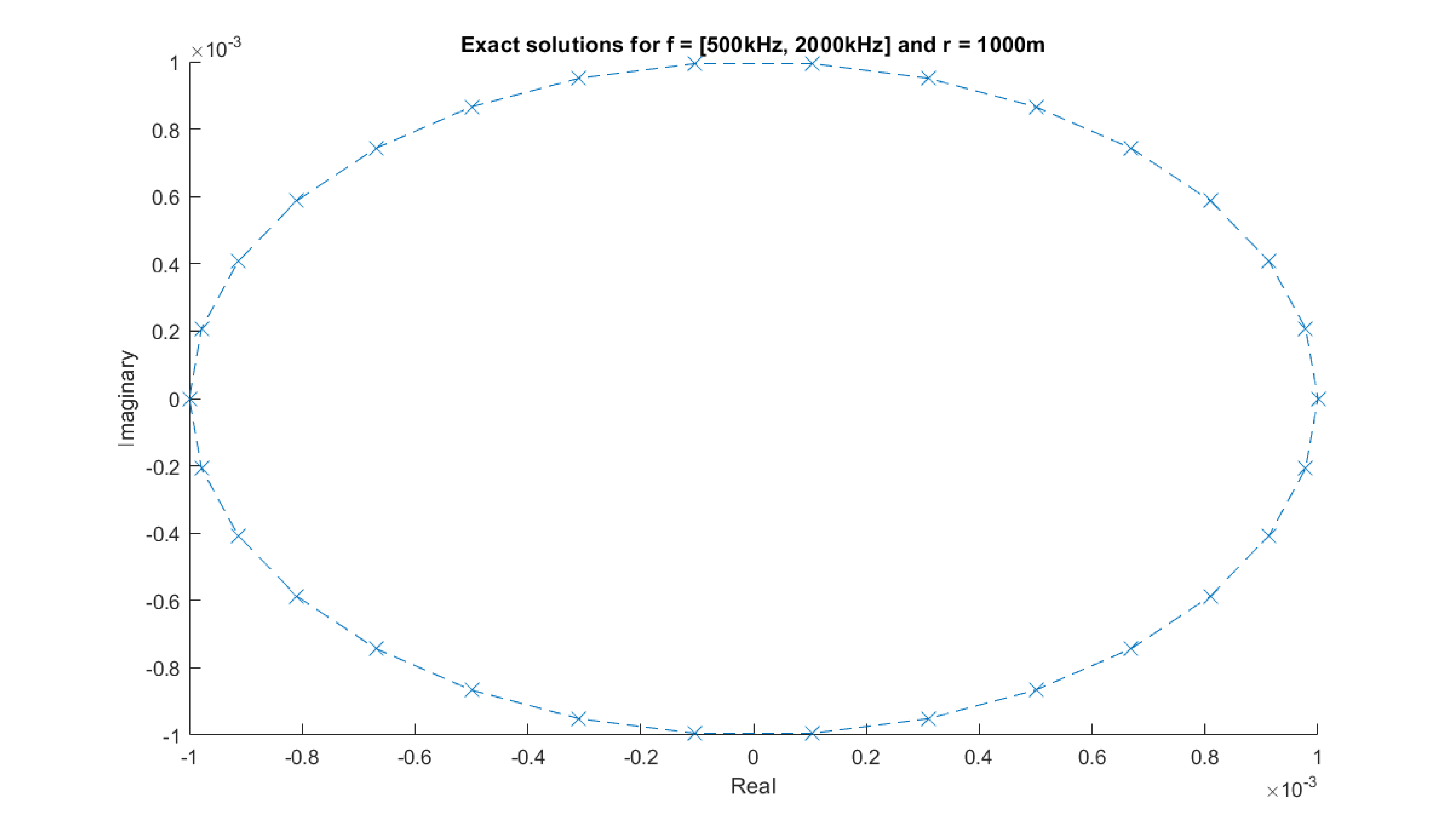
The Exact Solution

Using the formula $S = \frac{e^{-jkr}}{r}$, it is possible to directly solve for the exact solutions to the Sommerfeld problem given the frequency and range.

1. For $f \in [10\text{ kHz}, 100\text{ kHz}]$ in steps of 2 kHz, and $r = 10000\ m$, we expect a total of $\frac{100-10}{2} + 1 = 46$ data points
2. For $f \in [500\text{ kHz}, 2000\text{ kHz}]$ in steps of 10 kHz, and $r = 1000\ m$, we expect a total of $\frac{2000-500}{10} + 1 = 151$ data points

The solutions will be in the complex plane, so it is possible to plot them on a real vs. imaginary graph. The MATLAB graphs are given below.


Exact solutions for f = [10kHz, 100kHz] and r = 10000m
×10^-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Real
×10^-4



Exact solutions for f = [500kHz, 2000kHz] and r = 1000m
×10-3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Real
×10-3

Analysis

Looking at these graphs, we see that the number of points (marked as X's) is severely lower than what was calculated previously (46 and 151).

- This is because, at certain frequencies, the formula produces the same solutions. Thus, multiple X's are overlaid on top of each other
- We expect this to be different for our experimental methods due to error

EXPERIMENTAL METHODS AND SOLUTIONS

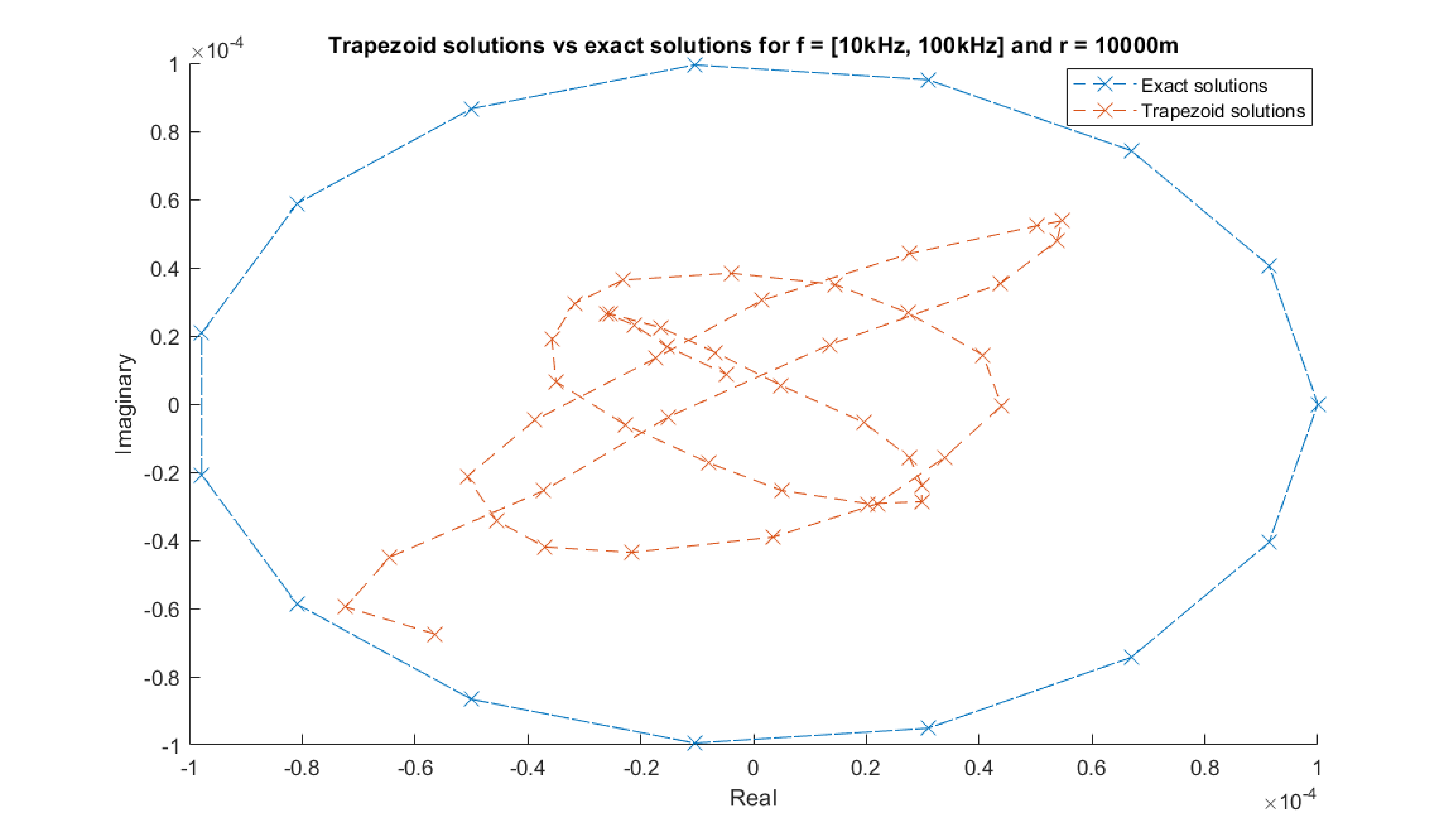
Trapezoid Method

The trapezoid method falls under the Newton-Cotes formulas for integration.

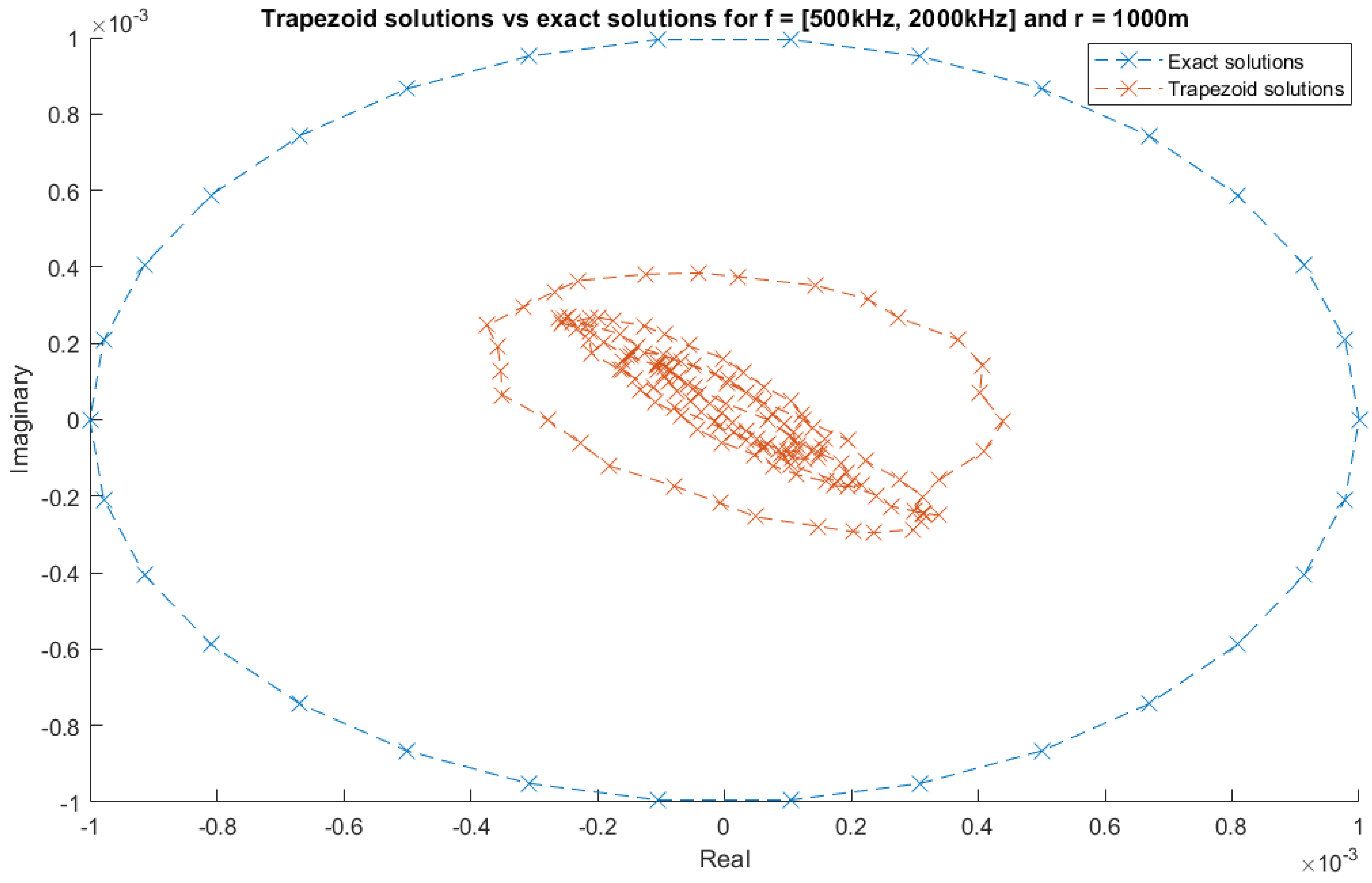
- It uses sets of trapezoids to approximate the area under the curve. Thus, more trapezoids → a better approximation

For this method, I chose an integration limit of $N \in [0, 100]$ and a step size of $n = 0.1$, for a total of 1000 trapezoids.

The resulting solution plots (compared to the exact solutions) are given below (solutions are marked as X's)



Trapezoid solutions vs exact solutions for f = [10kHz, 100kHz] and r = 10000m
×10^-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Real
×10^-4
Exact solutions
Trapezoid solutions


Trapezoid solutions vs exact solutions for f = [500kHz, 2000kHz] and r = 1000m
×10^-3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Real
×10^-3
Exact solutions
Trapezoid solutions

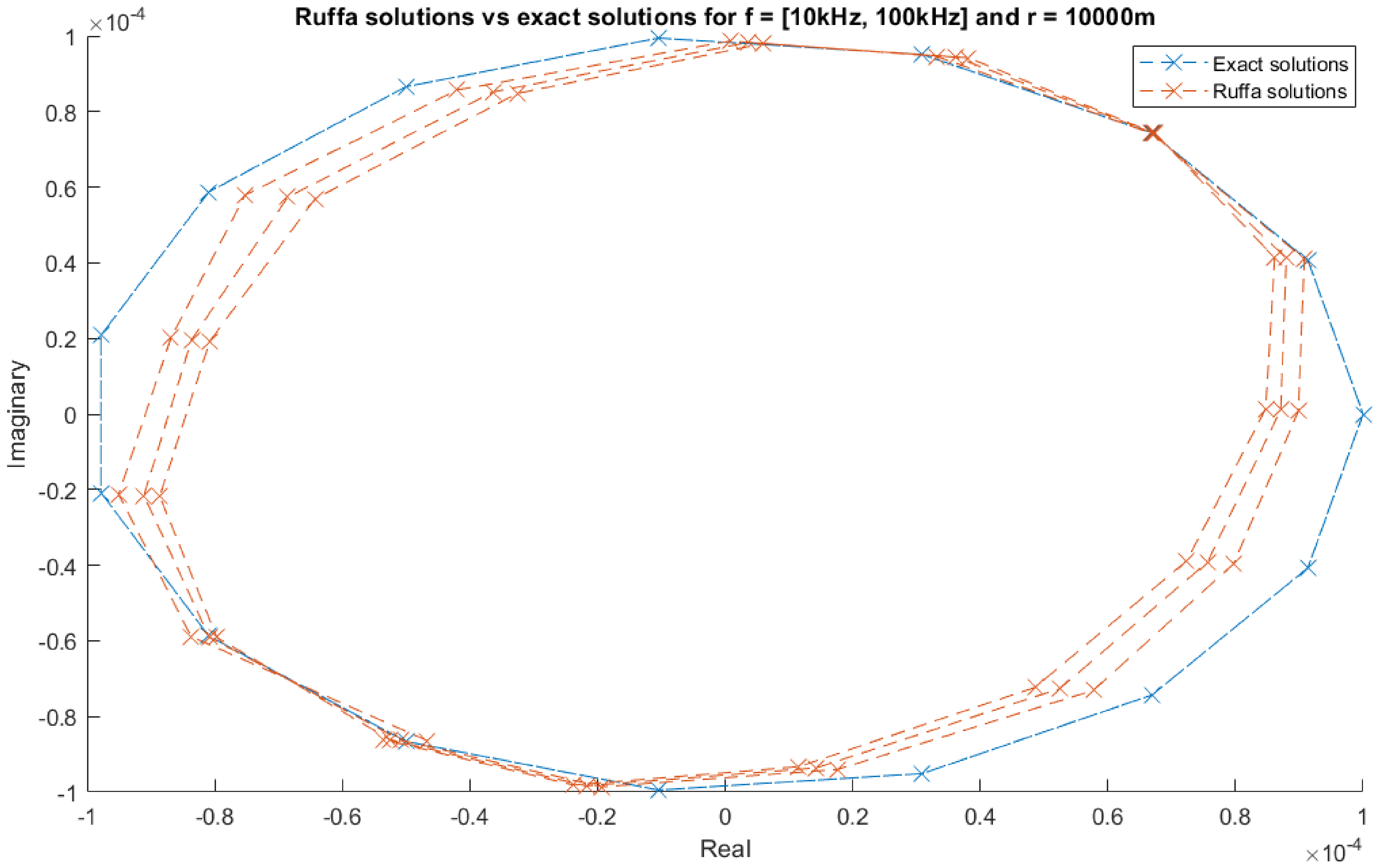
Ruffa’s Method

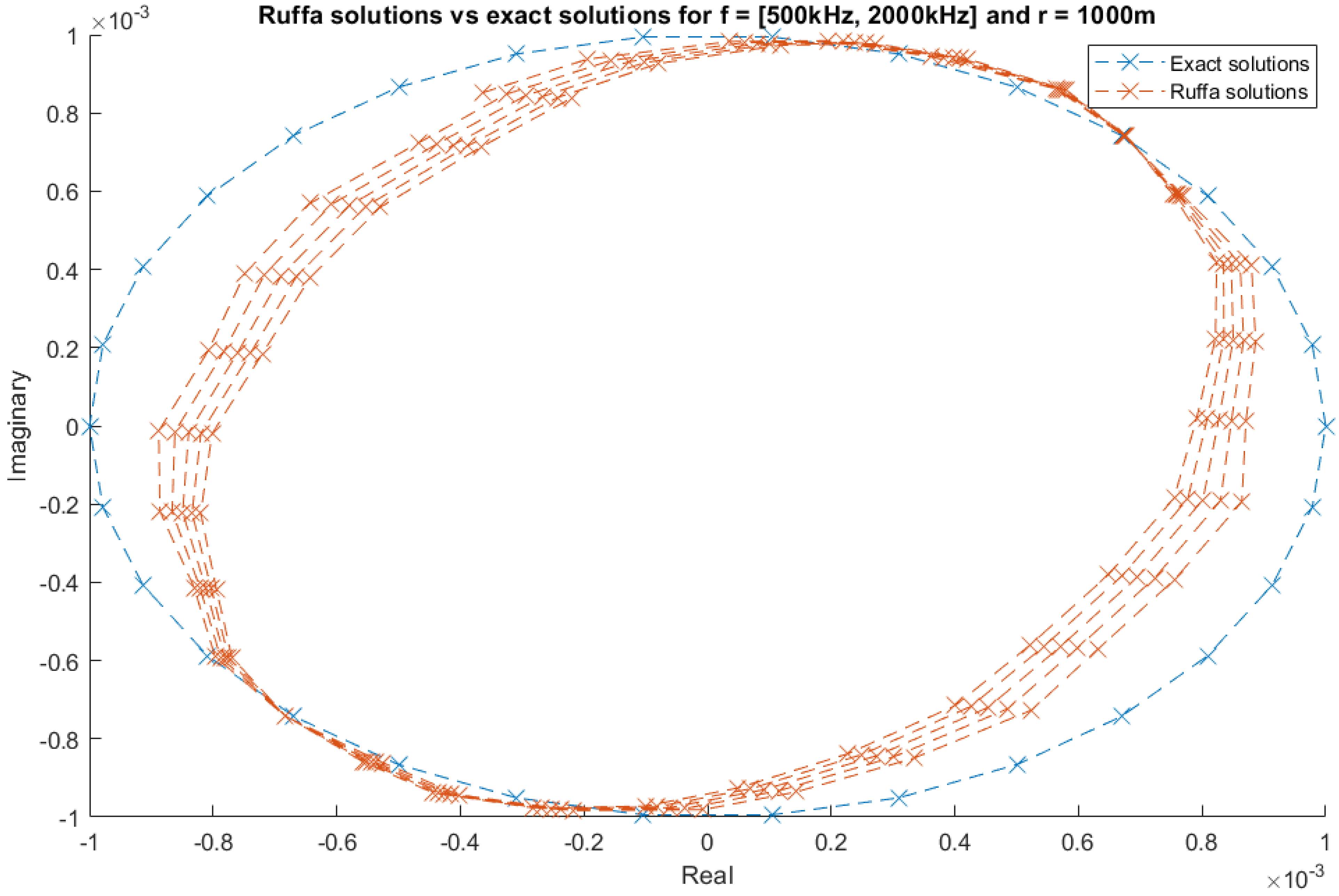
Ruffa’s method is considered a method of exhaustion.

- A summation is built from solving the areas of triangles under the curve
- The triangles are used to approximate the integral. Similarly, more triangles → a better approximation

Since the summation grows exponentially with the number of iterations, I chose a low value of $N = 15$ iterations.

The resulting solution plots (compared to the exact solutions) are given below (solutions are marked as X’s)


Ruffa solutions vs exact solutions for f = [10kHz, 100kHz] and r = 10000m
Exact solutions
Ruffa solutions
Imaginary
Real
×10^-4
×10^-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



Ruffa solutions vs exact solutions for f = [500kHz, 2000kHz] and r = 1000m
×10^-3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Imaginary
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
Real
×10^-3
Exact solutions
Ruffa solutions

Method Verification

In order to test my implementations of the methods above, I first evaluated a definite integral similar in form to the Sommerfeld Identity:

$$\int_{1}^{100} \frac{\sin(x)}{\sqrt{x}}$$

The results are:

- Exact solution using MATLAB’s `integral(…)` function = 0.5468
- Solution using the **trapezoid method** = 0.5421
- Solution using **Ruffa’s method** = 0.5456

From these results, we see that both methods are implemented correctly

Handling Infinities and Singularities

The Sommerfeld Identity integral as given has two problems:

1. **The upper bound goes to infinity**
2. **There is a singularity at $\boldsymbol{\alpha = k}$**

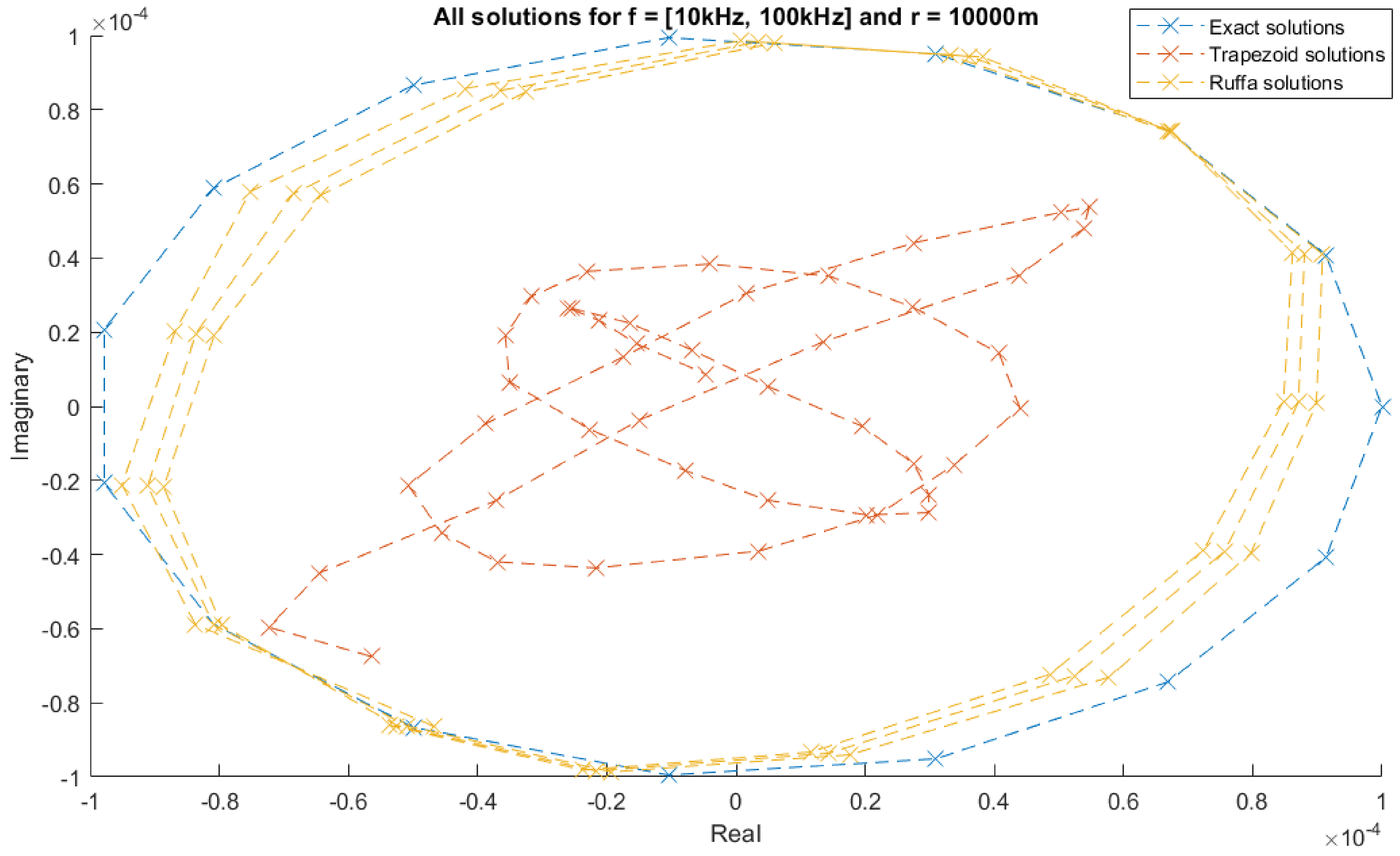
To avoid (1), I chose an upper bound of $100k$ for both methods

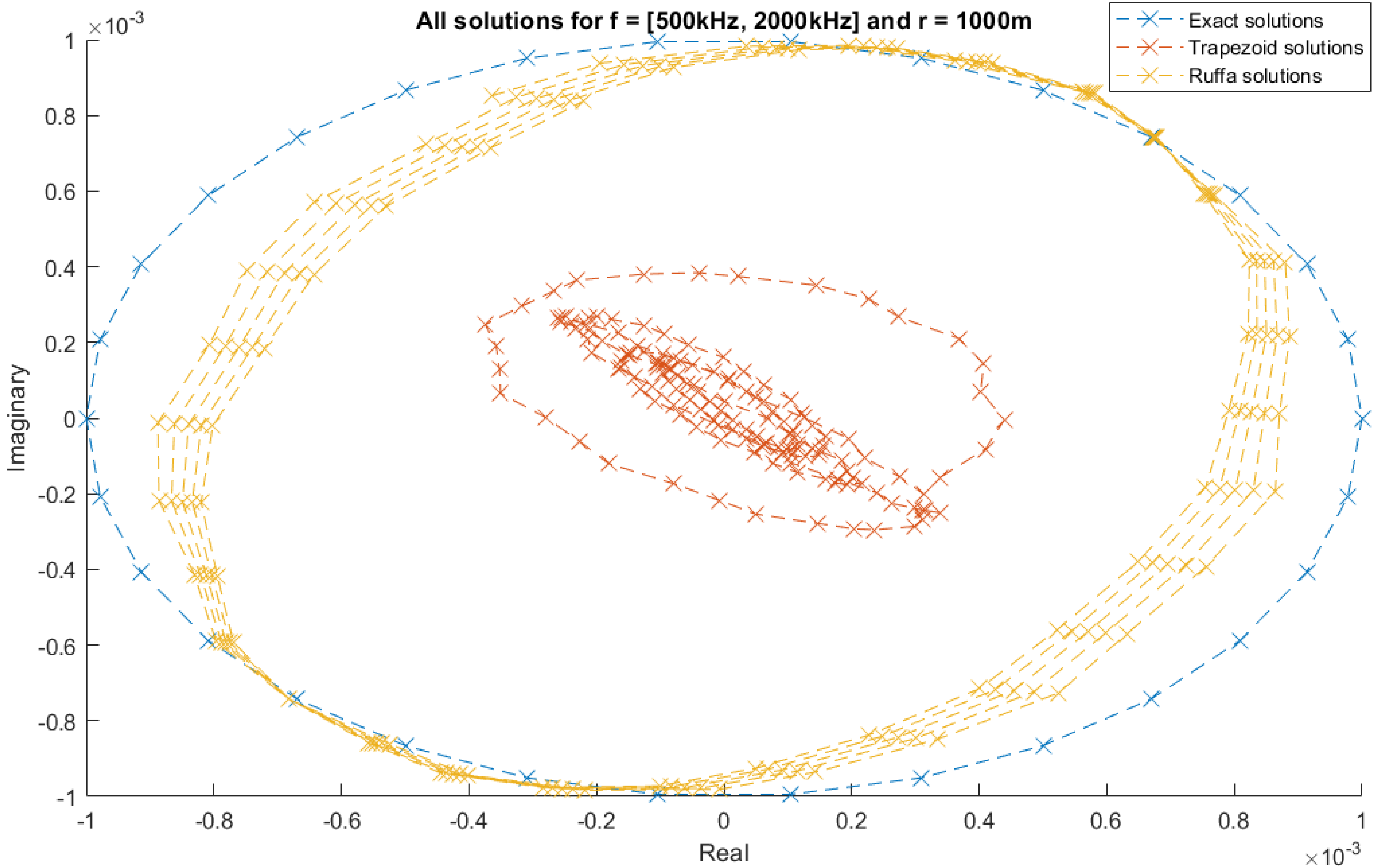
- This causes the solutions to be less accurate, but shortens computation time to a reasonable amount

To avoid (2), I split the integral into two separate integrals (as given in the project description) and I avoid computing them where $\alpha = k$ (but not where $\alpha \approx k$)

- Again, this results in less accuracy, but is a necessary change to avoid division by zero errors

ANALYSIS AND CONCLUSIONS


All solutions for f = [10kHz, 100kHz] and r = 10000m
Exact solutions
Trapezoid solutions
Ruffa solutions
Imaginary
Real
×10^-4
×10^-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


All solutions for f = [500kHz, 2000kHz] and r = 1000m
Exact solutions
Trapezoid solutions
Ruffa solutions
Imaginary
Real
×10^-3
×10^-3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Error Analysis

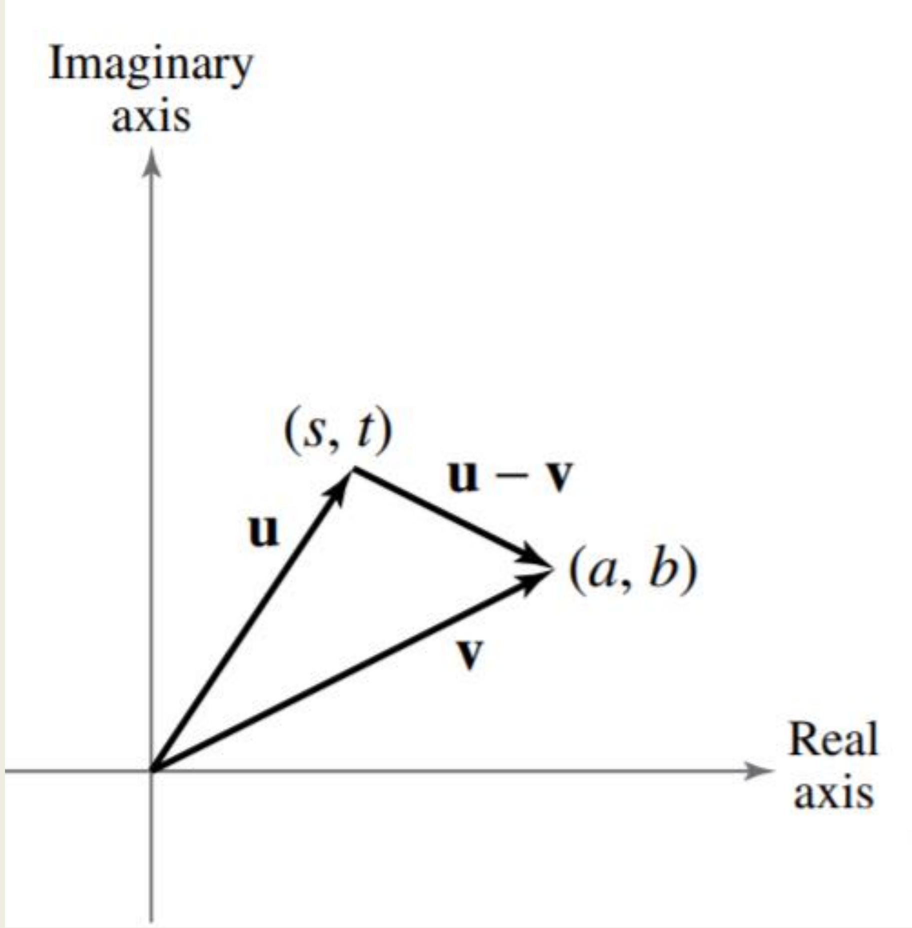
In order to analyze the absolute error of a complex set of values, we can evaluate the magnitude of the distance between them.

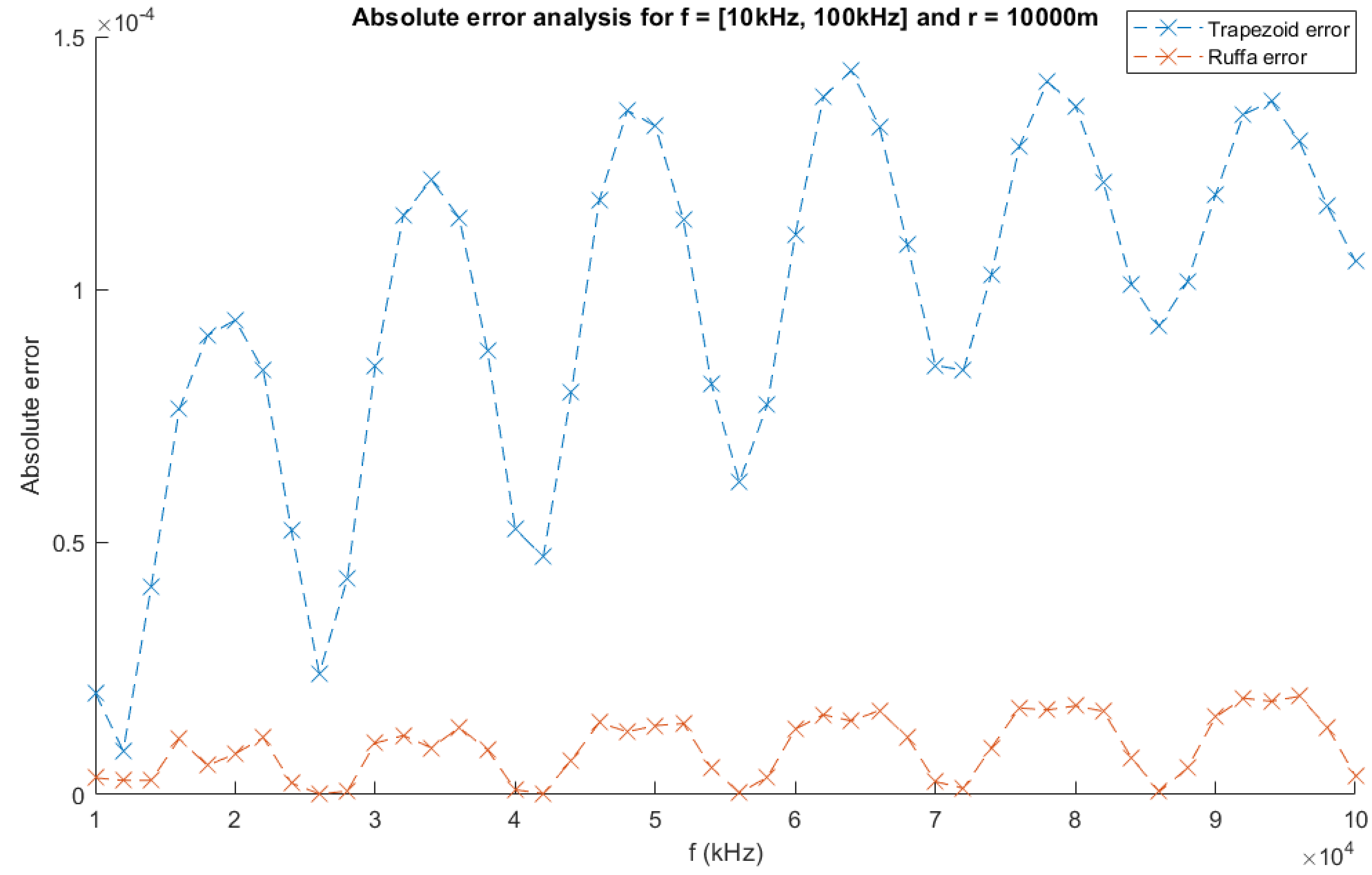
The formula for doing so is:

$$||u - v||$$

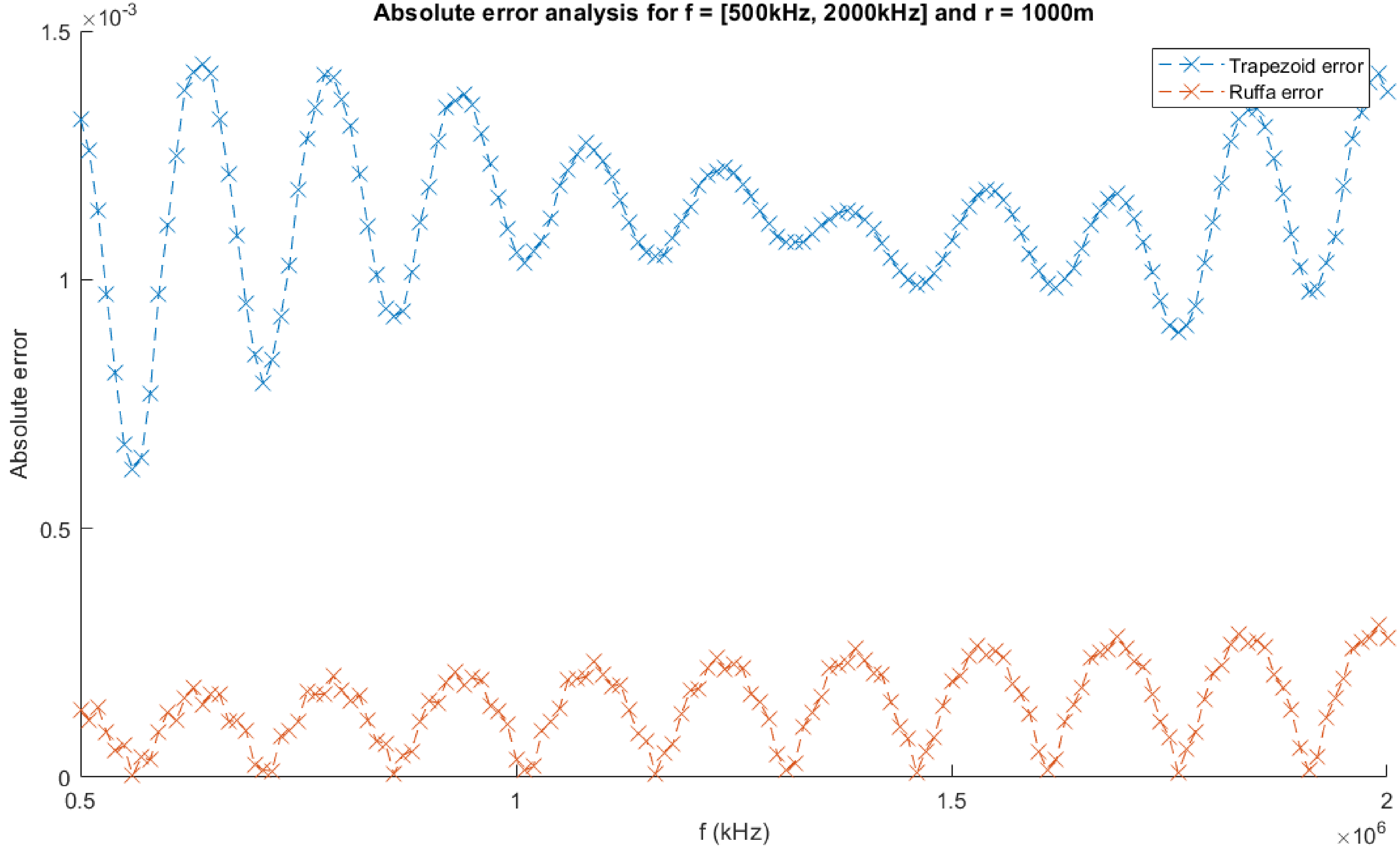
We define u as the **complex experimental** solution, and v as the **complex reference** solution (i.e. the exact solution).

Plotting this error for the given frequency ranges results in the following graphs





Absolute error analysis for f = [10kHz, 100kHz] and r = 10000m
Trapezoid error
Ruffa error
×10^-4
1.5
1
0.5
0
Absolute error
1
2
3
4
5
6
7
8
9
10
f (kHz)
×10^4


Absolute error analysis for f = [500kHz, 2000kHz] and r = 1000m
×10^-3
1.5
1
0.5
0
Absolute error
Trapezoid error
Ruffa error
0.5
1
1.5
2
f (kHz)
×10^6

Conclusions and Remarks

From the data we gathered, it is obvious that **Ruffa's method** worked much better than the trapezoid method. This can be attributed to a few things:

- More iterations were run using Ruffa's method
- The trapezoid method did not handle complex values well
- There was no error term associated with the trapezoid formula. This was done to make computation less heavy
- Loss of significance due to very small values

Regardless, neither methods had any major code-breaking issues, and the waveforms for the absolute error seem consistent for both methods.