A Software Simulation of the Proof of Roth's Theorem on Arithmetic Progressions

Geo Lee
Inquiry Hub Secondary School
BC, Canada
gogeolee@gmail.com

***Abstract*—This paper presents a software simulation of the proof of Roth's theorem on arithmetic progressions. By the theorem, a sufficiently large subset of the natural numbers with positive upper density must contain a 3-term arithmetic progression. The software simulation this paper presents follows the proof via discrete Fourier transform on the indicator function of the subset. This paper presents the general structure of the simulation of the proof in four main modules: preliminary setup, claiming the existence of 3-term AP under a certain condition, claiming the existence of a higher density subset under a certain condition, and a postliminary step to convert an AP from the finite field to positive integers. The steps and complications of implementing the proof of Roth's theorem in a software simulation are also discussed.**

***Key Terms*—Roth's theorem on arithmetic progressions, software simulation, discrete Fourier transform, finite field**

I. Introduction

This project was created as part of a group term project presentation for a mathematics course on Ramsey Theory at Simon Fraser University in Fall 2021 taught by Dr. Veselin Jungic [*]. The goal was to help others understand the initially abstract concepts of the proof of Roth's theorem on arithmetic progressions with a concrete example, displaying the overall structure of the proof to eventually find a 3-term arithmetic progression from a random set of natural numbers. The Fourier-analytic methods Roth used in his proof has become one of the central techniques of additive combinatorics, and has laid the groundwork for further profounding theorems from Gowers, Green, and Tao.

II. Background

The original statement from Roth for his theorem on arithmetic progressions is as follows:

For any $\varepsilon > 0$, there exists $N = N(\varepsilon)$ such that for any $n \geq N$, if $A \subseteq [1, n]$ with $|A| > \varepsilon n$, then A contains a 3-term arithmetic progression. [1]

This paper follows a proof based on a density increment argument via discrete Fourier transform on the indicator function of the set A [2]. Overall, the proof uses iteration to increase relative set density, eventually leading to the existence of a 3-AP to avoid a contradiction of density exceeding one. This section focuses on the key points of the proof. For details, the reader should see [2].

A. Preliminary Setup

Let $A \subseteq [1, n]$ with $|A| = \delta n$ where $\delta > 0$. Define

$$B = A \cap \left[\frac{n}{3}, \frac{2n}{3}\right] \tag{1}$$

The proof requires that

$$|B| \geq \frac{|A|}{5} \tag{2}$$

If condition (2) is not met, then at least one of $A \cap \left[0, \frac{n}{3}\right]$ or $A \cap \left[\frac{2n}{3}, n\right]$ must contain at least $\frac{2}{5}|A|$ elements, and consequently has to have a relative density of at least $\frac{6}{5}\delta > \delta$. Replacing A with this third, this process is repeated from (1). With a big enough n, a subset density greater than 1 will be obtained if (2) is never satisfied, which is a contradiction. This means condition (2) must eventually be met with some subset of A.

Under the condition

$$x, z \in \left[\frac{n}{3}, \frac{2n}{3}\right] \quad \text{and} \quad y \in [1, n], \tag{3}$$

the goal of the proof is to compute the number of possible solutions to

$$x + y \equiv 2z \pmod{n} \tag{4}$$

in $\mathbb{Z}_n$, which is identical to the number of solutions to

$$x - z = z - y \tag{5}$$

Any solution to (5) means x, z, and y are equal distances apart, thereby forming a 3-term AP in $\mathbb{Z}^+$.

$$M = |\{(x, y, z) \in B \times A \times B : x + y \equiv 2z \pmod{n}\}| \tag{6}$$

Under condition (3), the theorem can be proven by showing the number of solutions M of the congruence (4) may be greater than zero.

Using the discrete Fourier transform, the number of solutions to the congruence becomes a sum of two terms.

$$M = \frac{|A|\,|B|^2}{n} + n^2 \sum_{\substack{j\in\mathbb{Z}_n \\ j\neq 0}} \hat{B}(j)\hat{A}(j)\hat{B}(-2j) \tag{7}$$

where $\hat{A}(j)$ and $\hat{B}(j)$ are the discrete Fourier transform of the indicator functions of the sets A and B respectively.

Since the first term in (7) is always positive, the key step is to now show that the bound of the second term is always less than the first term in order to prove M is always greater than zero.

$$E = n^2 \sum_{\substack{j\in z_n \\ j\neq 0}} \hat{B}(j)\hat{A}(j)\hat{B}(-2j) \tag{8}$$

B. Claim 1

The bound of $|E|$ in (8) can be shown to be less than the half of the first term in (7) given there is no Fourier coefficient greater than a threshold value

$$\left|\hat{A}(j)\right| \leq \varepsilon \text{ with } \varepsilon \leq \frac{\delta^2}{10} \text{ for all } j \in \mathbb{Z}_n\backslash\{0\} \tag{9}$$

where δ is the density of set A in $\mathbb{Z}_n$. With condition (2) and (9), it is found that

$$|E| \leq \frac{1}{2n}\,|A|\,|B|^2 \tag{10}$$

With this bound on $|E|$, the count of solutions to the congruence (4) is

$$\frac{1}{2n}\,|A|\,|B|^2 = \frac{\delta^3}{50}n^2$$

The count of non-trivial solutions to the congruence is greater than zero with sufficiently large n

$$\frac{\delta^3}{50}n^2 - \delta n > 0 \tag{11}$$

This finishes the proof of a 3-term arithmetic progression given the conditions stated above.

C. Claim 2

If there exists a non-trivial Fourier coefficient greater than the threshold value $\varepsilon > 0$, an arithmetic progression P_n, of which intersection with A results in a higher set density than that of A can be found.

$$\frac{|A \cap P_n|}{|P_n|} > \delta + \frac{\varepsilon}{8} \tag{12}$$

This procedure cannot be repeated an indefinite amount of times as the density would become greater than 1, meaning the condition in (9) should eventually be satisfied.

Such an arithmetic progression P_n can be constructed from a translation of the zero centered arithmetic progression Q_n by a shift value that results in a large intersection of A and P_n, where $P_n = Q_n(a-j)$. This is done by the following test

$$g(a) = \sum_{j\in\mathbb{Z}_n} C(j)Q_n(a-j) \tag{13}$$

$$g(a) > \frac{|Q_n|}{4}\varepsilon \tag{14}$$

where $C(j) = A(j) - \delta$; $A(j)$ is the indicator function of set A.

The maximum common distance and number of terms in Q_n are defined to help a later transition from $\mathbb{Z}_n$ to $\mathbb{Z}^+$ happen easily.

$$|Q_n| = 2\left\lfloor \frac{\sqrt{n}}{16} \right\rfloor + 3 \tag{15}$$

and

$$d < \sqrt{n}, \tag{16}$$

where d is the common distance of Q_n. At this point, $A \cap P_n$ is taken as the next A, and P_n is taken as the next $[1, n]$.

D. Postliminary Step

The only remaining issue is the case when P_n wraps around in $\mathbb{Z}_n$, leading to two separate arithmetic progressions in $\mathbb{Z}^+$. These two arithmetic progressions are proven to not overlap, as the limits set on $|Q_n|$ and d have been chosen to guarantee it by (15) and (16). The arithmetic progression with the greater intersection with A is taken and labeled as the new P_n as it still raises the density of the set.

III. Software Simulation of the Proof

Below is the pseudocode for the most important functions. 'Roth' is the main function of the simulation; it calls the four main modules in the same structure as the proof. The key-test (9) always follows the preliminary setup, and depending upon the result either code for claim 1 or claim 2 is executed. In claim 2, the postliminary step may follow when needed. This flow of logic is repeated until a 3-term arithmetic progression is found in the given set A_n. 'FindWorkRange' is the preliminary setup for the following logic, 'FindFrequencySpike' is the key test, and 'FindP' finds an arithmetic progression with a good intersection with A_n.

```
// Un = set of all integers from 0 to n-1
// An = subset of Un with positive density
def Roth (Un, An):
   while True do
      Un, An = FindWorkRange(Un, An) // preliminary setup
      k = FindFrequencySpike(Un, An) // key test
      if k == 0
         // Claim 1, 3-AP exists in A
         return 3-AP
      else
         // Claim 2
         // AP  Pn  which makes a good intersection with A
         Pn = FindP(Un, An, k)
         if Pn is split into P1 and P2
            // postliminary step
            Pn = P1 or P2 that makes bigger intersection
                 with An
         end if
         Un = Pn
         An = An intersection Pn
      end if
```

```
    end while

def FindWorkRange(Un, An):
    while True
        M = An intersection mid third of Un
        L = An intersection left third of Un
        R = An intersection right third of Un
        if len(M) >= len(An) / 5
            return Un, An
        else
            if len(L) > len(R)
                Un = left third of Un
                An = L
            else
                Un = right third of Un
                An = R


def FindFrequencySpike(Un, An):
    d = density of set An in Un
    t = (d^2) / 10
    ind_A = indicator function of An
    A_hat = DiscreteFourierTransform(ind_A) / length of Un
    for k in [1, n - 1]
        f = A_hat[k]
        if |f| > t
            return k
    return 0

def FindP(Un, An, k):
    dA = density of set An in Un
    for each valid common distance d by (16)
        qn = AP in Un with d and length by (15)
        t = (len(qn) / 4) * ((dA^2)/10) // threshold
        for a in [0, len(Un) - 1]
            if g(a) > t
                pn = qn translated by a
                return pn
    return []
```

IV. Discussion

As stated previously, the goal of this project was to help those interested in learning how the proof of Roth's theorem on arithmetic progressions works. To do so, the structure of the simulation follows the proof as closely as possible, which has led to several complications.

The biggest issue stemmed from the fact that the proof requires a massive set for the logic to work properly. Each iteration of claim 2 in the proof cuts down the set size drastically by (15). One iteration of an initial set size of a million results in a new set size of only orders of 100. A trillion can only iterate twice before reaching similarly small numbers. Working with such large sets slows the simulation down to the point where it is not runnable. To stay faithful to the proof of the theorem, the simulation needs numbers much greater than what an ordinary computer can handle.

In order to work with smaller set sizes that the original proof does not consider, a hybrid approach was developed. If the size of the set is larger than a threshold value(1000 in the simulation), the code follows the proof. If the size of the set gets smaller than the threshold value, the simulation manually finds a shift value for the arithmetic progression P_n with the greatest intersection with A. This is not part of the proof, as any shift value that makes the density greater than some threshold should be accepted by (14).

There are times when the simulation is unable to find a 3-term arithmetic progression in the set, due to the limits on the elements on Q_n from the theorem by (15) and (16). These limits were set in order to avoid an overlap from occurring while converting the arithmetic progression from $\mathbb{Z}_n$ to $\mathbb{Z}^+$.

The simulation works with a randomly generated set A with uniform distribution for a given length(1 million) and density(0.1). This uniform distribution means the preliminary setup of the proof is effectively bypassed at the moment by (1) and (2). Due to the set being generated with a uniform distribution, increasing the size of the set effectively decreased all Fourier coefficients below the threshold (9) which lead immediately to claim 1.

As the main logic of this proof lies in claim 2, the goal for the simulation was to iterate over claim 2 as many times as possible to display. As previously stated, working with sets too small caused issues with the logic of the proof, while sets that were too large caused issues with the physical limitations of a computer.

The proof of Roth's theorem [2] does not find, but only proves the existence of a 3-term arithmetic progression. This means that the proof requires a set large enough to get close to the contradiction to finish. Trying to copy the proof of a mathematics theorem which can reach unimaginable complexities, with a physical finite computer may be something that is fundamentally challenging.

V. Conclusion

This paper presents a software simulation of the proof of Roth's theorem originally created in order to help those interested to quickly grasp the concepts of this important theorem. The simulation is by no means perfect, but creating such a simulation uncovers a layer of depth to the proof that cannot be easily appreciated otherwise.

Acknowledgement

I would like to express my gratitude to Dr. Veselin Jungic for his valuable and constructive suggestions during the planning and development of this project. His willingness to give his time so generously has been very much appreciated.

References

[1] Roth, K. (1953). "On certain sets of integers". Journal of the London Mathematical Society.

[2] Robertson, A. (2021). Fundamentals of Ramsey Theory (1st ed., pp. 65 - 78). Milton: CRC Press LLC.

Simulation

Google Colab Link

Presentation

Video Link