Application of Linear Equation Systems to Analyze Counter Push Strategies in Clash Royale

Eduard Daniel Ariajaya - 13524129[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13524129@itb.ac.id, [2]eduard.ariajaya@gmail.com

***Abstract*— Clash Royale is a card-based real-time strategy game that requires players to make efficient strategic decisions when facing enemy attacks. One strategy that can be used is the counter push, which is a strategy to defend and then continue with a counterattack using the remaining surviving units. The counter push strategy requires a combination of units capable of withstanding enemy attacks with limitations in elixir, time, and damage output. In this paper, the counter push strategy is modeled using a system of linear equations, taking these variables into account. The solution of the system of linear equations is used to analyze the success conditions of a counter push strategy in several attack scenarios. The results of the analysis show that the system of linear equations can be used as a simple mathematical approach to understand the limitations and potential of the counter push strategy in the context of digital strategy games.**

***Keywords*—Clash Royale, Systems of Linear Equations, Counter Push, Game Strategy Analysis.**

I. Introduction

Clash Royale is a real-time multiplayer strategy game that uses cards with different powers, such as troops, spells, and buildings. In this game, players are required to collect various cards obtained from chests and level up their cards. Players must build a battle deck consisting of 8 cards using the cards they have collected. The deck that has been created will be used in arena matches with the goal of destroying the opponent's King or Princess tower within a specified time limit. Each match requires players to make quick decisions regarding card selection and placement, as well as when to attack and defend. In Clash Royale, there are several arenas with different trophy requirements, so players are required to win matches and earn trophies to advance to the next arena level. A player's success in winning matches depends on their ability to develop effective strategies in various battle situations.

One strategy frequently used in this game is the counter push, which involves defending against an opponent's attack followed by a counterattack using surviving troops. This strategy requires players to determine a card combination that can effectively halt an opponent's attack and create opportunities for counterattacks with surviving troops. With precise elixir calculations and card placement, the counter push can make resource use more efficient and balance defense and offense.

The counter-push problem in Clash Royale is very similar to problems that can be modeled using a system of linear equations. Total unit damage, defense time against attacks, and elixir limitations can be represented as variables and constraints in a system of linear equations. With this model, counter-push strategies can be quantitatively analyzed to determine the conditions under which a given unit combination will succeed or fail against an opponent's attack.

The purpose of this paper is to analyze counter-push strategies in Clash Royale using a system of linear equations approach. It is hoped that this approach can serve as a relevant example of the application of linear algebra and geometry to digital games.

II. Theoretical Foundation

A. Clash Royale and Counter Push Mechanic

Clash Royale is a card-based multiplayer real-time strategy game where the goal is to destroy the opponent's towers. Players use cards to summon troops, spells, or buildings, each with different abilities in terms of damage, durability, and role in the match. Matches are dynamic, requiring players to make quick, strategic, and effective decisions under constantly changing conditions.

A counter push is a defensive strategy followed by a counterattack using the remaining units after the defense. This strategy aims to stop the opponent's attack and create counter-pressure without having to start from scratch. The success of a counter push depends heavily on the balance between defensive capabilities and counter-attack potential.

Figure II-1. Clash Royale Basic Battle

https://clashroyale.fandom.com/wiki/Basics_of_Battle

B. Systems of Linear Equations

An Systems of Linear Equations with m equations and n variables $x_1, x_2, \ldots, x_n$ has the form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \vdots \qquad \vdots \qquad\qquad \vdots \quad &\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned}$$

Figure II-2. General form of a system of linear equations
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-03-Sistem-Persamaan-Linier-2025.pdf

System of Linear Equations in Matrix Form:

$$\begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \\ \vdots \qquad \vdots \qquad\qquad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Figure II-3. System of linear equations in matrix form
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-03-Sistem-Persamaan-Linier-2025.pdf

System of Linear Equations in Augmented Matrix Form:

$$\begin{aligned} x_1 + 3x_2 - 6x_3 &= 9 \\ 2x_1 - 6x_2 + 4x_3 &= 7 \\ 5x_1 + 2x_2 - 5x_3 &= -2 \end{aligned}$$

to:

$$\left[\begin{array}{ccc|c} 1 & 3 & -6 & 9 \\ 2 & -6 & 4 & 7 \\ 5 & 2 & -5 & -2 \end{array}\right]$$

Figure II-4. System of linear equations in matrix form
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-03-Sistem-Persamaan-Linier-2025.pdf

Possible System of Linear Equations Solutions:

a) has a unique (single) solution

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

Figure II-5. system of linear equations with unique solution
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2025.pdf

Solution: $x_1 = 1, x_2 = 0, x_3 = -1$

b) has many (infinitely many) solutions

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Figure II-6. system of linear equations with many solution
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2025.pdf

c) has no solution at all

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Figure II-7. system of linear equations with no solution
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2025.pdf

C. Elementary Row Operations (EOPs)

Elementary Row Operations (EOPs) are basic operations used to manipulate matrices without changing the solution set of a system of linear equations represented. EOPs are the primary basis of the Gaussian elimination method for simplifying systems of linear equations into a more easily solved form.

Three elementary row operations on augmented matrices:

1. Multiply a row by a nonzero constant.
2. Interchange two rows.
3. Add a row to a multiple of another row. [1]

The solution to a system of linear equations is obtained by applying OBE to the augmented matrix until a row echelon matrix is formed for the Gaussian elimination method and a reduced row echelon matrix for the Gaussian elimination method. [1]

D. Gaussian Elimination Method

One common method used to solve systems of linear equations is the Gaussian elimination method. This method works by converting the system of linear equations into matrix form and performing elementary row operations to obtain a simpler form, allowing the solution to the system to be systematically determined [1].

The Gaussian elimination method was chosen because it is straightforward, easy to implement, and suitable for analyzing systems of linear equations with a relatively small number of variables.

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \overset{R1/2}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \underset{R3+2R1}{\overset{R2-4R1}{\sim}} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{bmatrix}$$

$$\overset{R2/(-2)}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 6 & -2 & 6 \end{bmatrix} \overset{R3-6R2}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & -5 & -15 \end{bmatrix} \overset{R3/(-5)}{\sim} \begin{bmatrix} 1 & 3/2 & -1/2 & 5/2 \\ 0 & 1 & 1/2 & 7/2 \\ 0 & 0 & 1 & 3 \end{bmatrix}$$

Figure II-8. elementary row operations
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-03-Sistem-Persamaan-Linier-2025.pdf

E. Back Substitution in Systems of Linear Equations

Back substitution is a fundamental step in solving a system of linear equations after applying Gaussian elimination. After the expanded matrix is converted to row echelon form, the values of the variables are obtained by sequentially substituting the known values into the top row of the system.

This process allows the solution to the system to be determined in a structured and systematic manner. Back substitution is particularly effective for systems with a small number of variables, as it provides a clear interpretation of the relationships between the variables without the need for further matrix operations.

In the context of this study, back substitution is used to calculate the contribution of each unit involved in a counterattack strategy. The obtained values are then interpreted as relative contributions rather than exact unit quantities, allowing the system of linear equations to serve as an analytical tool for evaluating the feasibility of the strategy.

(iii) $x_3 = 3$
(ii) $x_2 + 1/2x_3 = 7/2 \rightarrow x_2 = 7/2 - 1/2(3) = 2$
(i) $x_1 + 3/2x_2 - 1/2x_3 = 5/2 \rightarrow x_1 = 5/2 - 3/2(2) - 1/2(3) = 1$

Figure II-9. back substitution for augmented matrix in Figure II-8
https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-03-Sistem-Persamaan-Linier-2025.pdf

The solution is obtained: $x_1 = 1, x_2 = 2, x_3 = 3$

III. METHOD

This chapter describes the methodology used to analyze counter-push strategies in Clash Royale using a system of linear equations. The method focuses on identifying specific battle scenarios, defining variables and parameters, constructing a mathematical model, solving the system using Gaussian elimination, and interpreting the results within the context of the game.

A. Identify Counter Push Scenarios

The opponent initiates an attack using a combination of a *Giant* as a tank unit and a *Wizard* as a damage-dealing support unit. It is assumed that the *Wizard* has crossed the bridge and is behind the *Giant*.

To counter push this attack, the player deploys a *Giant Skeleton* to target the Wizard and a *Mini P.E.K.K.A.* to target the Giant. This placement separates the defensive roles and allows the defending units to switch to counter push after the enemy attack is neutralized. This analysis is limited to a single defense-counterattack phase and does not consider the entire match duration.

Figure III-1. clash royale battle interaction phase 1

In the subsequent counter push phase, it is assumed that the Wizard has been eliminated while the Giant Skeleton remains active and encounters an additional defensive unit, the Ice Wizard. This secondary interaction is analyzed to evaluate whether the counterattack can continue under simplified conditions.

Figure III-2. clash royale battle interaction phase 2

B. Determination of Variables and Parameters

Based on the defined scenarios, the variables are:

- x: contribution of Giant Skeleton in eliminating the Wizard
- y: contribution of Mini P.E.K.K.A. in eliminating the Giant
- z: contribution of Giant Skeleton in eliminating the Ice Wizard

Several parameters are defined to represent the minimum conditions required for a successful counter push:

- HP_{wizard}: health points of the Wizard
- HP_{giant}: health points of the Giant
- HP_{IW}: health points of the Ice Wizard
- d_{GS}: damage contribution of Giant Skeleton
- d_{MP}: damage contribution of Mini P.E.K.K.A.

C. Model Assumptions and Limitations

The mathematical model used in this study is based on several simplifying assumptions. First, the interactions between units are assumed to be linear and additive, allowing the counter push scenario to be represented using systems of linear equations. Complex in-game dynamics such as precise unit positioning, attack timing, and area-of-effect interactions are not modeled explicitly.

Second, the variables in the system represent relative contributions rather than discrete unit counts. This abstraction enables analytical tractability but does not reflect the exact numerical behavior of units in real gameplay. Additionally, the slowing effect of certain units, such as the Ice Wizard, is not included in the model, as the analysis focuses on feasibility rather than temporal combat dynamics.

Despite these limitations, the model remains effective as a conceptual tool for understanding the conditions under which a counter push strategy may succeed or fail. The simplified assumptions allow linear algebra concepts to be applied clearly and consistently, making the approach suitable for educational and analytical purposes rather than detailed gameplay simulation.

D. Modelling Using Systems of Linear Equations

1) Initial Defense Phase
During the initial defense phase, the Giant Skeleton targets the Wizard while the Mini P.E.K.K.A. targets the Giant. The system of linear equations is expressed as:

$$d_{GS} \cdot x = HP_{wizard}$$
$$d_{MP} \cdot y = HP_{giant}$$

2) Subsequent Counter Push Phase
After the Wizard has been eliminated, the Giant Skeleton is assumed to engage an Ice Wizard deployed by the opponent. This interaction is modeled using a single linear equation:

$$d_{GS} \cdot z = HP_{IW}$$

E. Gaussian Elimination Implementation

In this paper the Gaussian elimination method is implemented using the Python programming language. The systems of linear equations are represented as augmented matrices, and the elimination process is performed programmatically. This implementation reduces manual computation errors and enables the evaluation of multiple counter push scenarios by modifying parameter values.

1) System Representation
The initial step in the implementation is to represent the counter push scenario in the form of a system of linear equations. The analyzed scenario involves an opponent attacking with Giant and Wizard, where Giant Skeleton is assigned to eliminate the Wizard and Mini P.E.K.K.A. is assigned to eliminate the Giant. This interaction is modeled using system of linear equations that has been made.

The system is then converted into an augmented matrix form to facilitate the application of the Gaussian elimination method. This matrix representation serves as the primary input for the elimination process and reflects the minimum conditions required to successfully defend against the opponent's attack and preserve units for a counter push.

```python
d_GS = 250
d_MP = 400
HP_wizard = 800
HP_giant = 3000
HP_ice_wizard = 700

# Matriks augmentasi
matrix_defense = [
    [d_GS, 0, HP_wizard],
    [0, d_MP, HP_giant]
]
matrix_ice_wizard = [
    [d_GS, HP_ice_wizard]
]
```

Figure III-3. augmented matrix form model

```python
def print_matrix(matrix, step=""):
    if step:
        print(f"\n{step}")
    for row in matrix:
        print([round(x, 2) if isinstance(x, float) else x for x in row])
```

Figure III-4. to show augmented matrix form

2) Gaussian Elimination Algorithm
The solution of the system of linear equations is obtained using the Gaussian elimination method. The algorithm operates by applying elementary row operations to transform the augmented matrix into row-echelon form. Once the matrix reaches this form, the values of the variables are determined through back substitution.

To support the analytical process and reduce manual computation errors, the Gaussian elimination procedure is implemented using the Python programming language. The program takes the augmented matrix as input, performs forward elimination and back substitution, and outputs the solution of the system. The implementation of the algorithm is also provided in the Appendix.

```python
def gaussian_elimination(matrix):
    n = len(matrix)
    print_matrix(matrix, "Matriks awal:")
    # Forward elimination
    for i in range(n):
        # Pivoting jika elemen diagonal nol
        if matrix[i][i] == 0:
            for j in range(i + 1, n):
                if matrix[j][i] != 0:
                    matrix[i], matrix[j] = matrix[j], matrix[i]
                    break

        pivot = matrix[i][i]
        if pivot != 0:
            for k in range(i, len(matrix[i])):
                matrix[i][k] /= pivot
            print_matrix(matrix, f"Setelah normalisasi baris {i+1}")

        for j in range(i + 1, n):
            factor = matrix[j][i]
            if factor != 0:
                for k in range(i, len(matrix[j])):
                    matrix[j][k] -= factor * matrix[i][k]
    # Back substitution
    solution = [0] * n
    for i in range(n - 1, -1, -1):
        solution[i] = matrix[i][-1]
        for j in range(i + 1, n):
            solution[i] -= matrix[i][j] * solution[j]
    return solution
```

Figure III-5. gaussian elimination in python

3) Result

The result of the Gaussian elimination implementation is the solution of the system of linear equations that represents the counter push strategy. The output of the program consists of the values of variables x and y, which indicate the relative contribution of Giant Skeleton and Mini P.E.K.K.A. in eliminating the Wizard and the Giant, respectively.

A solution with positive and realistic values indicates that the selected unit placement is sufficient to stop the opponent's attack and allows the surviving units to initiate a counter push. Conversely, the absence of a valid solution or the presence of unrealistic values indicates that the counter push strategy is not feasible under the modeled conditions.

These results demonstrate that the Gaussian elimination method can be effectively applied to analyze counter push strategies in Clash Royale using systems of linear equations, providing a quantitative basis for evaluating the feasibility of unit placement decisions in gameplay scenarios.

```
Matriks awal:
[250, 0, 800]
[0, 400, 3000]

Setelah normalisasi baris 1
[1.0, 0.0, 3.2]
[0, 400, 3000]

Setelah normalisasi baris 2
[1.0, 0.0, 3.2]
[0, 1.0, 7.5]
Kontribusi Giant Skeleton (x): 3.2
Kontribusi Mini P.E.K.K.A. (y): 7.5

Matriks awal:
[250, 700]

Setelah normalisasi baris 1
[1.0, 2.8]
Kontribusi Giant Skeleton terhadap Ice Wizard (z): 2.8
```

Figure III-6. solution result

F. Discussion of Modeling Choices

The modeling approach used in this study involves several deliberate design choices to ensure consistency with linear algebra concepts while maintaining relevance to gameplay scenarios. The decision to represent unit interactions using systems of linear equations allows the counter push strategy to be analyzed through feasibility rather than detailed simulation.

Separating unit roles, such as assigning Giant Skeleton to target the Wizard and Mini P.E.K.K.A. to target the Giant, simplifies the system and avoids overlapping contributions that could introduce ambiguity in interpretation. This separation reflects typical gameplay strategies and ensures that each equation represents a clear and distinct condition.

Furthermore, the choice to apply Gaussian elimination instead of more advanced optimization or simulation techniques aligns with the educational objective of the study. By focusing on systems of linear equations, the analysis remains accessible and emphasizes conceptual understanding over computational complexity. These modeling choices support the use of linear algebra as an analytical tool for strategic decision-making in digital games.

IV. CONCLUSION

This paper analyzes counter push strategies in Clash Royale using a system of linear equations as a mathematical modeling approach. The counter push scenario is represented by defining unit contributions, health constraints, and damage parameters, which are then formulated into linear equations. Gaussian elimination is applied to solve the resulting systems and determine whether a given unit placement can successfully defend against an opponent's attack while allowing surviving units to continue with a counterattack.

The analysis demonstrates that systems of linear equations can be used as a simple yet effective tool to evaluate the feasibility of counter push strategies. By interpreting the existence and nature of solutions, players can gain insight into the limitations and potential of specific unit combinations without relying on full gameplay simulations. The implementation using Python further supports the analytical process by ensuring consistency and reducing manual computation errors.

Although the model applies several simplifying assumptions and does not capture all in-game dynamics such as precise timing or positional interactions, it remains effective as a conceptual framework for strategy analysis. Overall, this study shows that linear algebra, particularly systems of linear equations and Gaussian elimination, can be meaningfully applied to analyze strategic decision-making in digital strategy games such as Clash Royale.

Future research may extend this study by incorporating additional gameplay factors into the mathematical model. Elements such as unit positioning, attack timing, and area-of-effect interactions could be explored using more advanced mathematical frameworks, including non-linear models or time-based analysis.

Additionally, the system of linear equations approach could be applied to a wider range of counter push scenarios involving different unit combinations. Comparing multiple strategies under varying assumptions may provide deeper insights into strategic decision-making processes within the game. These extensions would allow the analytical framework presented in this paper to be expanded beyond feasibility analysis toward more detailed strategic evaluation.

V. Appendix

The complete program implementation code can be seen at the following link: https://github.com/Minaqu28/SPL-ClashRoyale

VI. Acknowledgment

I would like to express my deepest gratitude to Arrival Dwi Sentosa, S.Kom., MT., as a lecturer of IF2123 Linear Algebra and Geometry, for his invaluable guidance throughout this semester, and to Dr. Ir. Rinaldi Munir, M.T., for his valuable resources during the lectures. Their dedication and expertise have played a significant role in deepening my understanding of systems of linear equations and other concepts of linear algebra and geometry, as well as the applications of linear algebra and geometry themselves in real life, which helped me in completing this paper.

References

[1] R. Munir, Sistem Persamaan Linier, STEI ITB, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-03-Sistem-Persamaan-Linier-2025.pdf (Accessed on: December 24, 2025).

[2] R. Munir, Tiga Kemungkinan Solusi Sistem Persamaan Linier, STEI ITB, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-04-Tiga-Kemungkinan-Solusi-SPL-2025.pdf (Accessed on: December 24, 2025).

[3] D. C. Lay, *Linear Algebra and Its Applications*, 5th ed. Boston, MA, USA: Pearson, 2016.

[4] H. Anton and C. Rorres, Elementary Linear Algebra, 11th ed. Hoboken, NJ, USA: Wiley, 2014.

[5] *Basics of Battle*, Clash Royale Wiki, Fandom. [Online]. Available: https://clashroyale.fandom.com/wiki/Basics_of_Battle

STATEMENT

I hereby declare that the paper I wrote is my own writing, not an adaptation or translation of someone else's paper, and is not plagiarized.

Bandung, 24 Desember 2025

Eduard Daniel Ariajaya
13524129