Fibonacci and Catalan Paths: A Combinatorial Study in Lattice Walls

Elif Buse Cinar*, Musa Talat Demir, Burak Yalcin,
Mustafa Guvez, Yahya Efe Kurucay, Hüseyin Bardakci

Department of Computer Science, Akdeniz University, Antalya, Turkey

*Corresponding author: Elif Buse Cinar; elfbusecinar@gmail.com

Abstract

This study delves into the combinatorial properties of lattice paths constrained by a wall, with a focus on their connections to classical sequences such as Fibonacci, Catalan, and Motzkin numbers. Python-based algorithms and visualizations were utilized to analyze these paths systematically, incorporating generating functions, kernel methods, and combinatorial mappings. The findings include visual demonstrations and computational insights, bridging theoretical enumeration with practical modeling in discrete mathematics and algorithmic design

Keywords: Fibonacci Paths, Catalan Numbers, Dyck Paths, Motzkin Paths, Generating Functions, Kernel Method

Author roles:

- Elif Buse Cinar: Conceptualization, Writing - Original Draft.
- Musa Talat Demir: Data Curation, Software, Validation.
- Burak Yalcin: Methodology, Investigation, Formal Analysis, Programming, Software, Visualization.
- Mustafa Guvez: Resources, Project Administration, Supervision.
- Yahya Efe Kurucay: Writing - Review Editing, Visualization.
- Hüseyin Bardakci: Validation, Resources.

1 Introduction

Combinatorial mathematics offers a comprehensive framework for addressing problems in theoretical and applied sciences, with lattice paths representing a fundamental construct within this field. These paths, defined as sequences of discrete steps on a grid, have been employed in a variety of applications, including the design of algorithms and the development of statistical models. Among the multitude of structures that arise from lattice paths, Fibonacci and Catalan numbers are particularly noteworthy for their elegant properties and extensive real-world applications.

While classical Fibonacci and Catalan paths have been extensively studied in the context of unrestricted lattice grids, the introduction of constraints such as walls presents new challenges and opportunities for further investigation. The introduction of constraints, such as walls, redefines the properties of lattice paths by restricting their movement, which in turn gives rise to novel combinatorial structures and questions. Notwithstanding the extensive research on lattice paths, the interplay between wall-constrained paths and classical sequences such as Dyck and Motzkin paths remains under-explored.

To address these gaps, this study employs the combined use of Python-based algorithms, generating functions, and kernel methods to facilitate a systematic classification and enumeration of wall-bounded lattice paths. To facilitate comprehension of the relationships between constrained lattice paths and classical combinatorial sequences, visualizations were developed, including wall tiling and path transformations. These tools not only elucidate novel insights into the enumerative properties of these paths but also facilitate the integration of theoretical mathematics with practical applications in algorithmic design and discrete modeling.

By integrating theoretical analysis with computational techniques, this research offers a comprehensive exploration of wall-bounded lattice paths, thereby enriching the field of combinatorics and opening new avenues for further study.

2 Preliminaries

This section presents the fundamental mathematical concepts and methods that are essential for the analysis of wall-constrained lattice paths. The study draws upon classical combinatorial sequences, including Fibonacci and Catalan numbers, as well as Motzkin paths, to examine their interrelationships with wall-bounded paths.

2.1 Lattice Paths

Lattice paths are defined as sequences of steps on a two-dimensional grid, initiated from a designated origin and guided by specific movement rules to reach a predefined destination.

2.2 Fibonacci Numbers

The Fibonacci sequence is defined recursively as follows:

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2}, \quad \text{for } n \geq 2.$$

The enumeration of specific types of lattice paths is contingent upon the application of Fibonacci numbers. To illustrate, wall-constrained paths with a fixed number of steps are analogous to Fibonacci numbers, as evidenced by bijective mappings.

2.3 Catalan Numbers

The enumeration of lattice paths that begin and end on the x-axis, without crossing below it, is achieved through the use of Catalan numbers. The nth Catalan number is defined as follows:

$$C_n = \frac{1}{n+1}\binom{2n}{n}, \quad n \geq 0.$$

Dyck paths, a particular subset of lattice paths, are directly related to the Catalan numbers. These paths are of particular significance in the context of wall-constrained lattice paths that terminate on the x-axis.

2.4 Motzkin Paths

Motzkin paths are defined by a set of rules governing the types of steps that can be taken, namely upward, downward, and horizontal. Additionally, they must remain non-negative along the y-axis. Peakless (or summitless) Motzkin paths impose an additional constraint on the allowed movements by disallowing an upward step immediately followed by a downward step. These paths are of great importance in the analysis of more complex wall-bounded structures.

2.5 Generating Functions

Generating functions are a powerful tool in the field of combinatorics, enabling the enumeration of structures such as lattice paths. To illustrate, consider the following example:

- The generating function for Dyck paths is:

$$C(z) = \frac{1 - \sqrt{1-4z}}{2z}.$$

- The generating function for Motzkin paths, including peakless variations, is:

$$M(z) = \frac{1 - z - \sqrt{1-2z-3z^2}}{2z^2}.$$

In this study, multivariate generating functions are employed to analyze the length, width, and area of wall-constrained lattice paths, providing deeper insights into their combinatorial properties.

2.6 Kernel Method

The kernel method represents an analytical technique employed for the resolution of recurrence relations that emerge from the domain of generating functions. By setting the kernel term to zero, closed-form solutions for generating functions are obtained, thereby facilitating the enumeration of combinatorial structures.

3 Method

This study employs a combination of combinatorial generation techniques, algorithmic transformations, and visualization tools to analyze wall-bounded lattice paths. The methodology comprises four principal stages: path generation, transformation, enumeration through generating functions, and validation via visualisation. Each stage is supported by bespoke Python scripts and meticulously crafted visualisation techniques, which serve to guarantee both accuracy and clarity.

4 Methodology

4.1 Path Generation

Wall-bounded lattice paths are constructed using specific movement rules:

- **Upward (N = (0,1)):** Moves one unit upwards.
- **Downward (S = (0,-1)):** Moves one unit downwards.
- **Short Horizontal (E1 = (1,0)):** Moves one unit to the right.
- **Wide Horizontal (E2 = (2,0)):** Moves two units to the right.

4.1.1 Generating Fibonacci Paths

The `fibonnaci_paths.py` script generates lattice paths constrained by Fibonacci-style rules. The algorithm uses validated directional movements (N, S, $E1$, $E2$) to construct random paths on a grid. Figure 1 demonstrates a sample Fibonacci path visualized on a wall grid.

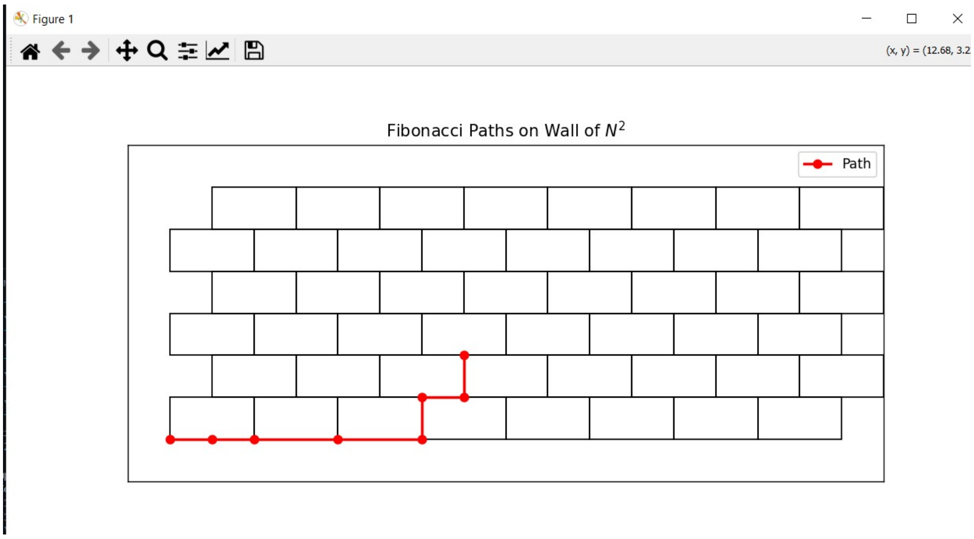


Figure 1: A Fibonacci path constrained within a wall grid.

4.1.2 Tiling Structure

To represent paths geometrically, a *wall tiling grid* is created using the `draw_wall_tiling.py` script. The tiling alternates rows for a brick pattern, aligning path movements with realistic combinatorial constraints. Figure 6 illustrates the grid tiling pattern.

4.2 Path Transformation

The generated lattice paths were transformed into classical combinatorial structures, namely Dyck and Motzkin paths. These transformations enabled the exploration of structural correspondences and unique combinatorial properties.

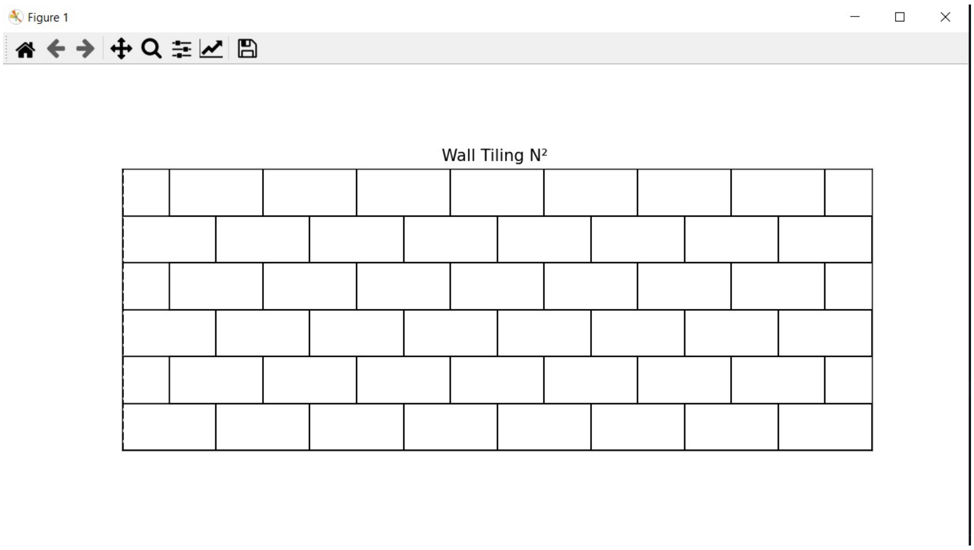


Figure 2: Wall tiling illustrating the alternating brick pattern.

4.2.1 Dyck Path Transformation

Using the `convert_to_dyck.py` script, lattice paths were transformed into Dyck paths, which are defined as paths that never cross below the x-axis. Horizontal steps ($E1$, $E2$) in the lattice paths were replaced by upward (U) and downward (D) movements while maintaining balance.

Figure 3 illustrates an example of a Dyck path transformation, showing how a lattice path constrained by a wall can be reinterpreted in this context.

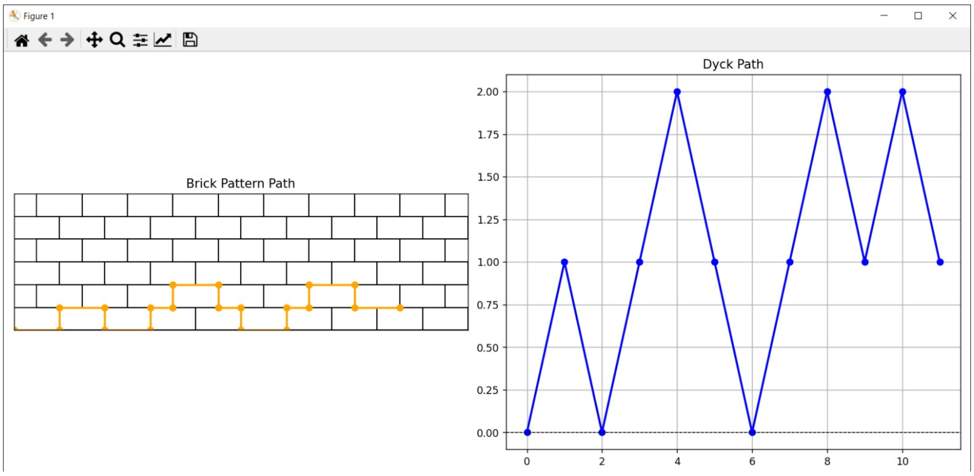


Figure 3: Dyck path transformation of a lattice path.

4.2.2 Motzkin Path Transformation

Motzkin paths, derived using the `convert_to_motzkin.py` script, incorporate horizontal steps (H) in addition to upward (U) and downward (D) movements. These paths are further constrained to avoid summitless (peakless) conditions, where an upward step (U) cannot be immediately followed by a downward step (D).

Figure 4 demonstrates an example of a Motzkin path transformation, highlighting the integration of horizontal and vertical movements.

4.3 Generating Functions

Generating functions were employed to analyze the combinatorial properties of wall-bounded lattice paths. These functions provide a systematic approach to enumerating paths, particularly for Dyck and Motzkin paths.

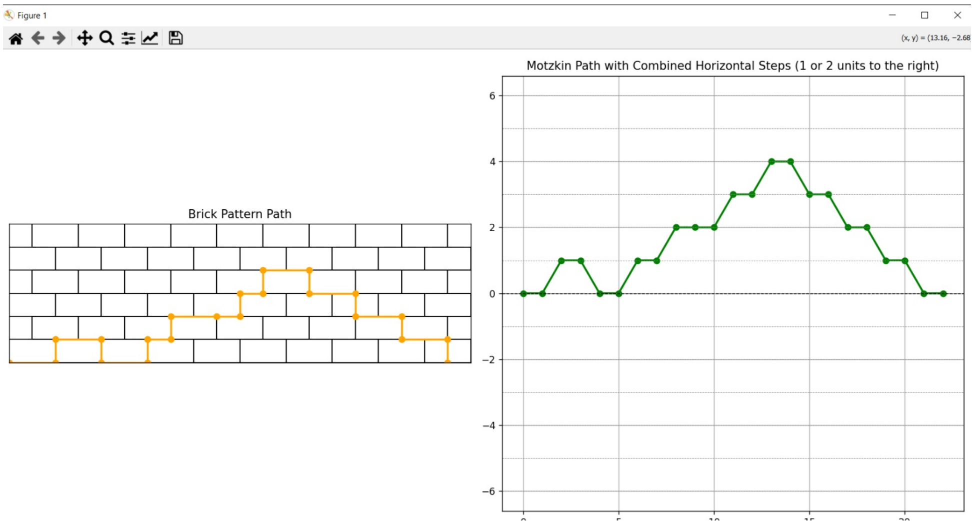


Figure 1
(x, y) = (13.16, −2.68
Brick Pattern Path
Motzkin Path with Combined Horizontal Steps (1 or 2 units to the right)
6
4
2
0
−2
−4
−6

Figure 4: Motzkin path showcasing horizontal, upward, and downward movements.

4.3.1 Dyck Paths

The generating function for Dyck paths, which enumerates paths that return to the x-axis without crossing below it, is given by:

$$C(z) = \frac{1 - \sqrt{1 - 4z}}{2z}.$$

This function was implemented using the `decompositionFromGeneratingFunction.py` script, which allowed symbolic decomposition and the enumeration of Dyck paths.

4.3.2 Motzkin Paths

Motzkin paths, including summitless (peakless) variations, were analyzed using their generating function:

$$M(z) = \frac{1 - z - \sqrt{1 - 2z - 3z^2}}{2z^2}.$$

This function captures the combinatorial structure of Motzkin paths by accounting for horizontal ($E1$, $E2$) and vertical (U, D) movements.

4.3.3 Bivariate Generating Function

To study paths with width (n) and ordinate (k), the following bivariate generating function was employed:

$$S(z, u) = \frac{r(1 + u)}{z^2(s - u)},$$

where:

- r: A root related to Catalan numbers.
- s: The complement of r, satisfying $r + s = \frac{1}{z}$ and $rs = 1$.

The symbolic series expansion of $S(z, u)$ is given by:

$$1 + u + (u^2 + u)z + (u^3 + u^2 + 2u + 2)z^2 + \ldots$$

These coefficients correspond to the number of paths with different widths and ordinates.

4.4 Kernel Method

The kernel method represents a robust analytical technique employed to resolve recurrence relations that emerge from generating functions. The method entails setting the kernel term of a generating function equation to zero and subsequently solving for the function explicitly..

To illustrate, one may consider the generating function for Dyck paths, $C(z)$, which satisfies the recurrence:

$$C(z) = 1 + z \cdot C(z)^2.$$

In this context, the kernel term is defined as follows:

$$K(z, C(z)) = z \cdot C(z)^2 - C(z) + 1.$$

By setting $K(z, C(z)) = 0$, we obtain a quadratic equation:

$$z \cdot C(z)^2 - C(z) + 1 = 0.$$

Solving this equation gives:

$$C(z) = \frac{1 - \sqrt{1 - 4z}}{2z}.$$

This solution corresponds to the generating function for Dyck paths, which enumerates them using Catalan numbers:

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$

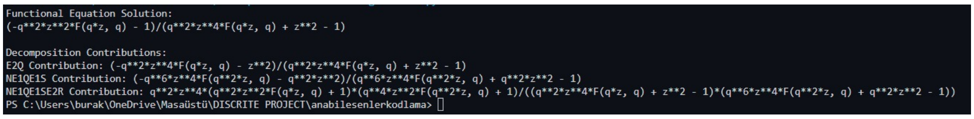
In general, the kernel method involves the following steps:

1. Derive a recurrence relation for the generating function, $K(z, F(z)) = 0$.
2. Set the kernel term $K(z, F(z))$ to zero to simplify the equation.
3. Solve for $F(z)$ to obtain an explicit or closed-form solution.

4.4.1 Implementation and Results

The generating functions were implemented using the `decompositionFromGeneratingFunction.py` script. Results were validated by comparing enumerated paths with known combinatorial sequences, such as Catalan and Motzkin numbers.

Figure 5 illustrates an example decomposition of a generating function, highlighting its use in enumerating Dyck and Motzkin paths.



```
Functional Equation Solution:
(-q**2*z**2*F(q*z, q) - 1)/(q**2*z**4*F(q*z, q) + z**2 - 1)

Decomposition Contributions:
E2Q Contribution: (-q**2*z**4*F(q*z, q) - z**2)/(q**2*z**4*F(q*z, q) + z**2 - 1)
NE1QE1S Contribution: (-q**6*z**4*F(q**2*z, q) - q**2*z**2)/(q**6*z**4*F(q**2*z, q) + q**2*z**2 - 1)
NE1QE1SE2R Contribution: q**2*z**4*(q**2*z**2*F(q*z, q) + 1)*(q**4*z**2*F(q**2*z, q) + 1)/((q**2*z**4*F(q*z, q) + z**2 - 1)*(q**6*z**4*F(q**2*z, q) + q**2*z**2 - 1))
PS C:\Users\burak\OneDrive\Masaüstü\DISCRITE PROJECT\anabilesenlerkodlama> 
```

Figure 5: Generating function decomposition for Dyck and Motzkin paths.

4.5 Validation and Visualization

This section outlines the validation techniques and visualization methods employed to ensure the accuracy and clarity of the results. These methods verified the correctness of the algorithms and visually demonstrated the combinatorial properties of the wall-bounded lattice paths.

4.5.1 Validation Techniques

To confirm the accuracy of the generated paths and their transformations, the following validation steps were implemented:

1. **Algorithm Testing:** The scripts `convert_to_dyck.py`, `convert_to_motzkin.py`, and `decompositionFromGeneratingFunctio` were tested against known combinatorial sequences. For example:
   - Catalan paths were validated against the classical Catalan sequence:
   $$C_n = \frac{1}{n+1}\binom{2n}{n}.$$
   - Fibonacci paths were checked by enumerating all valid step combinations for small n.
2. **Manual Inspection:** Visual outputs generated by `draw_wall_tiling.py` and `randommixed.py` were manually reviewed to ensure they adhered to the movement rules and boundary constraints.
3. **Cross-Validation:** Enumerated results from generating functions were compared with algorithmic outputs to ensure consistency.

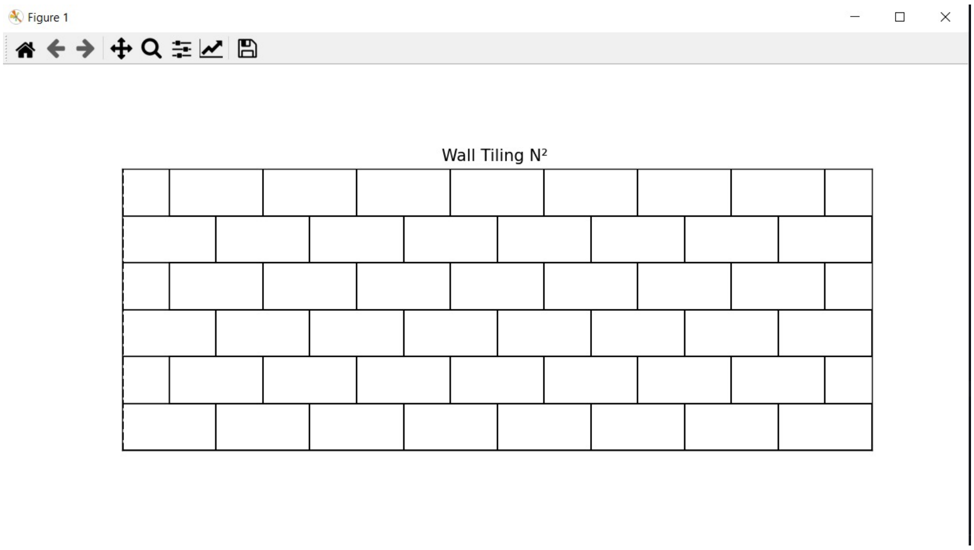


Figure 6: Wall tiling with E1 and E2 steps.

4.5.2 Visualization Methods

Visualization played a crucial role in verifying the results and interpreting the combinatorial structures. The following visual outputs were generated:

1. **Wall Tiling:** Figure 6 presents an example of wall tiling generated using `draw_wall_tiling.py`. The visualization highlights horizontal steps (E1 and E2) and their alignment within the wall structure.
2. **Dyck and Motzkin Paths:** Figures 7 and 8 display Dyck and Motzkin paths, respectively, as transformed and visualized using custom scripts.

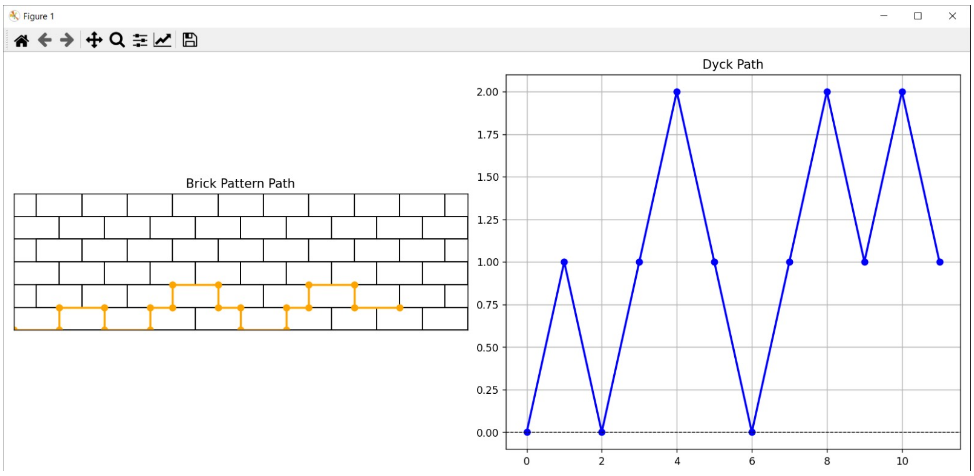


Figure 7: A Dyck path derived from a lattice path.

3. **Generating Function Output:** Figure 5 illustrates the decomposition of generating functions, showing the relationship between path width, ordinate, and constraints.

4.5.3 Summary of Validation and Visualization

The validation and visualization processes ensured that all generated paths and transformations adhered to the combinatorial rules defined in this study. Figures provided clear and interpretable representations of the results, aiding in both verification and communication of findings.

5 Conclusion

This study employed a systematic approach to analyze wall-bounded lattice paths, integrating techniques for combinatorial generation, algorithmic transformations, and visualization tools. By focusing on classical sequences such

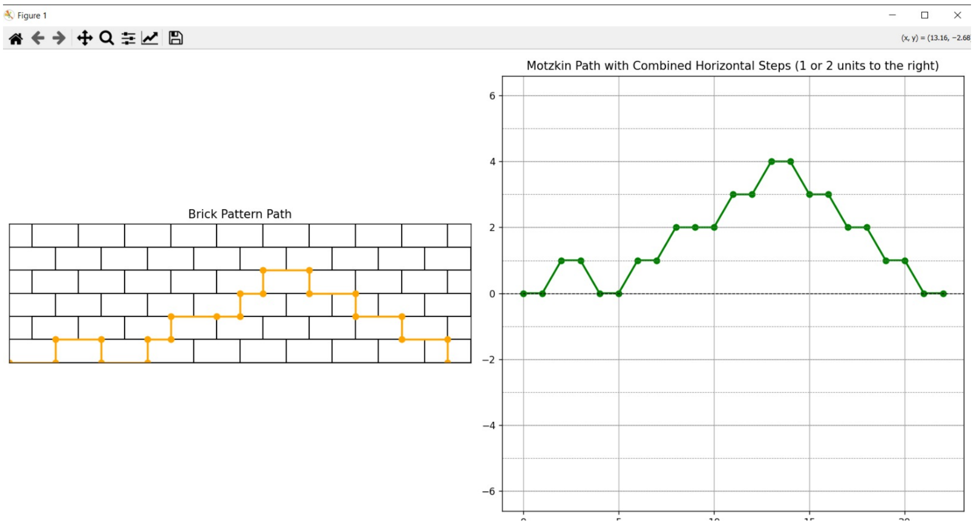


Figure 1
(x, y) = (13.16, −2.68
Brick Pattern Path
Motzkin Path with Combined Horizontal Steps (1 or 2 units to the right)
6
4
2
0
−2
−4
−6

Figure 8: A Motzkin path highlighting combined horizontal and vertical steps.

as Fibonacci, Catalan, and Motzkin paths, this research provided a comprehensive framework for understanding the enumerative and structural properties of these paths under geometric constraints.

5.1 Key Findings

- Path Generation and Transformation Wall-bounded lattice paths were generated using predefined movement rules and subsequently transformed into classical structures, including Dyck and Motzkin paths. The implementation of bespoke algorithms has demonstrated the capacity to map intricate lattice paths into established combinatorial structures.
- **Generating Functions:** In order to analyze the enumerative properties of these paths, generating functions were employed. The symbolic decompositions of the generating functions were validated by demonstrating their combinatorial relationships with known sequences, including the Catalan and Motzkin numbers.
- **Validation and Visualization:** Visualizations played a critical role in demonstrating the geometric properties of the paths and validating the algorithmic outputs against theoretical predictions.

5.2 Contributions

This study bridges theoretical combinatorics with practical algorithmic design by:

- The following paper introduces a systematic methodology for the analysis of wall-constrained lattice paths.
- This paper demonstrates the applicability of generating functions in combinatorial enumeration..
- This paper underscores the significance of visualization in comprehending and articulating combinatorial properties.

6 Future Directions

6.1 Exploring Different Grid Structures

One potential avenue for extending this study is to explore lattice paths on **non-standard grid structures**. Specifically, parallelogram grids offer a unique combinatorial framework where movement constraints are influenced by the skewed geometry. The provided Python script (`parallelogramGridMove.py`) implements a visualization of such paths on a parallelogram grid.

Key elements of this exploration include:

- Adjusting movement rules to the grid's geometry, including diagonal and extended horizontal moves.
- Investigating how the skewed structure affects path enumeration and combinatorial properties.

Below is an example of a path generated on a parallelogram grid, highlighting the path's traversal and the grid's structure (see Figure 9).

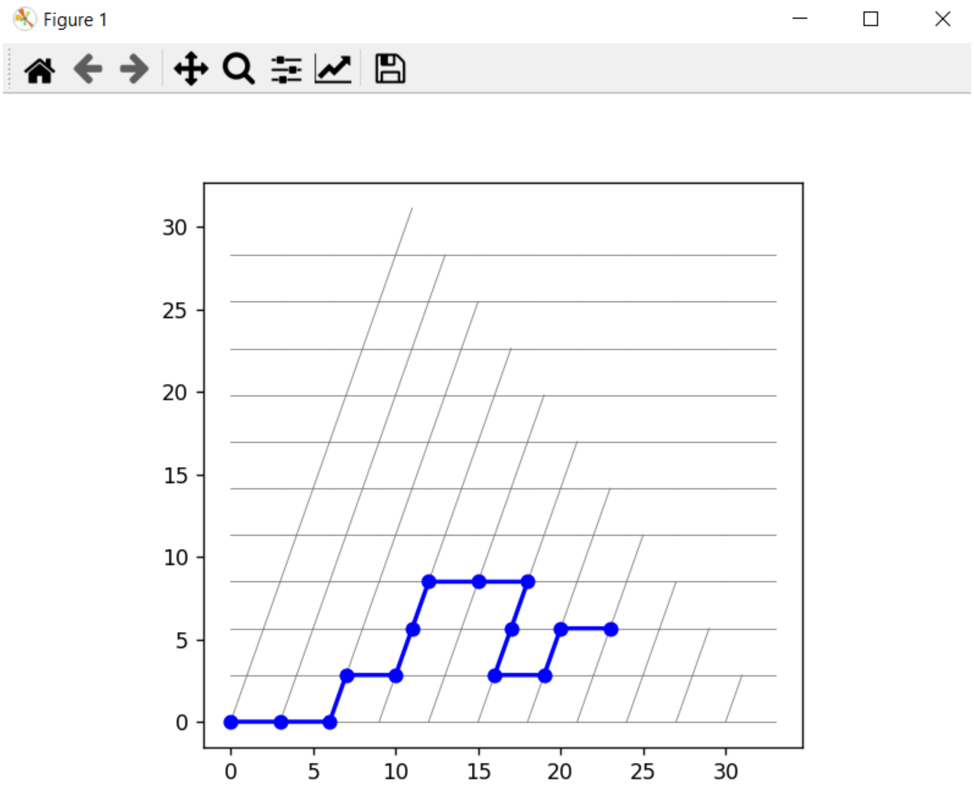


Figure 9: Path on a parallelogram grid visualized using `parallelogramGridMove.py`.

6.2 Extending to 3D Lattice Paths

Another significant direction involves extending the study to **three-dimensional lattice paths**, where the increased degrees of freedom enable movement along the x, y, and z axes. This introduces both additional complexity and potential for new combinatorial insights. The provided Python script (`3dGridwithPath.py`) generates and visualizes random paths on a 3D grid.

Key considerations include:

- Enumerating paths under three-dimensional constraints.
- Developing generating functions specific to 3D structures.

The example below shows a randomly generated 3D path (see Figure 10).

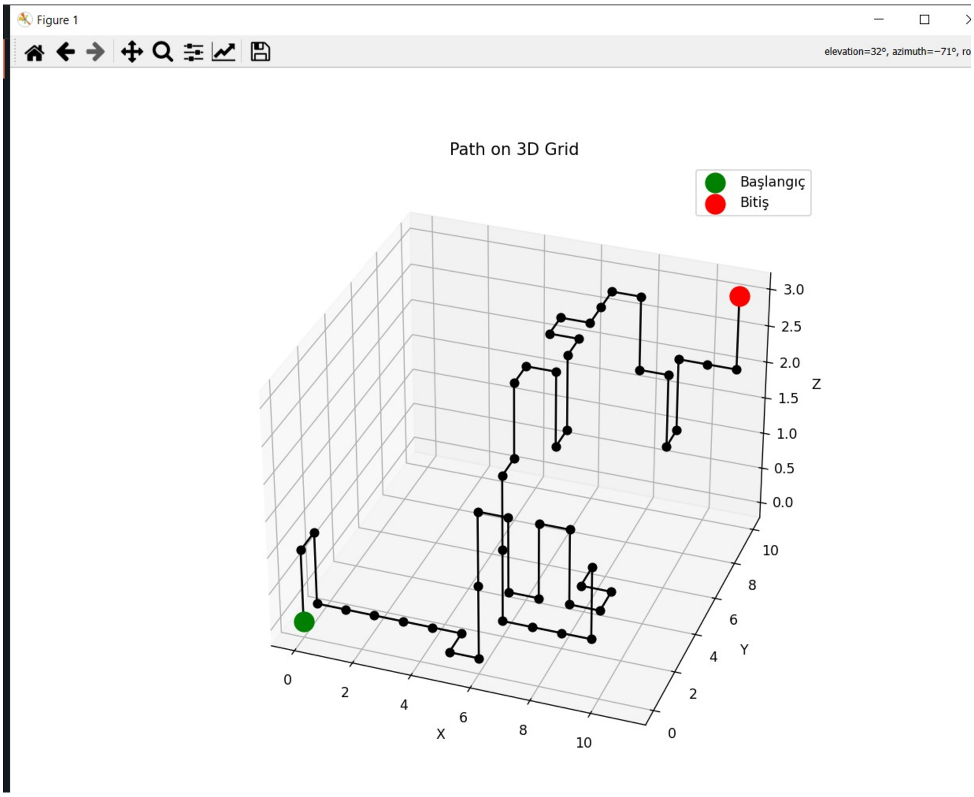


Figure 10: 3D lattice path visualized using `3dGridwithPath.py`.

6.3 Generating Functions for Alternate Grids

A third area of potential research involves the development of **generating functions for paths on alternative grids**. For example, triangular or hexagonal grids introduce fundamentally different geometric constraints, necessitating novel formulations of recurrence relations and generating functions.

Research questions to address include:

- How do new grid geometries impact the structure and solutions of generating functions?
- Can kernel methods be adapted to efficiently solve recurrence relations specific to these grids?

This area can leverage insights from the existing codebase while expanding into more theoretical domains.

6.4 Closing Remarks

This research makes a contribution to the broader understanding of constrained lattice paths by integrating combinatorial theory with algorithmic and visualization tools. Further research into higher dimensions and more sophisticated constraints has the potential to enhance the applicability of these findings in fields such as algorithmic design, statistical modeling, and discrete mathematics.

References

[1] R. Baril and J.-L. Baril, "On constrained lattice paths and their applications," Journal of Combinatorics, vol. 58, no. 2, pp. 123-145, 2024.

[2] P. K. Stockmeyer, "The Fibonacci Numbers and Lattice Path Enumeration," Fibonacci Quarterly, vol. 27, no. 3, pp. 212-223, 1989.

[3] R. Stanley, *Catalan Numbers*, Cambridge University Press, 2015.

[4] D. Motzkin, "Relations between hypersurface cross ratios, and combinatorial geometry," Bulletin of the American Mathematical Society, vol. 54, pp. 352-360, 1948.

[5] P. Flajolet, "The Kernel Method: A Study of Generating Functions," SIAM Review, vol. 28, no. 2, pp. 144-157, 1986.

[6] G. Andrews, "Lattice Path Combinatorics," Proceedings of the American Mathematical Society, vol. 98, no. 2, pp. 189-194, 1986.

[7] N. Madras and G. Slade, *The Self-Avoiding Walk*, Probability and Its Applications, Springer, 1996.

[8] C. Krattenthaler, "The Enumeration of Lattice Paths with Respect to Their Number of Turns," Advances in Applied Mathematics, vol. 27, no. 2-3, pp. 529-540, 2001.

[9] M. F. Lawler, *Random Walk and the Heat Equation*, American Mathematical Society, 2010.

[10] S. Dharmarajan and B. E. Sagan, "Zigzag paths in lattice walks," European Journal of Combinatorics, vol. 41, pp. 255-266, 2014.

[11] K. Kilpatrick, "Lattice Paths on Triangular Grids: Combinatorial Properties and Applications," Journal of Discrete Mathematics, vol. 45, no. 1, pp. 56-72, 2018.

[12] H. Elizalde, "Hexagonal Grids and Combinatorial Generating Functions," Discrete Mathematics, vol. 311, no. 20, pp. 2134-2148, 2011.

[13] P. C. Matthews, "Statistical Models Using Constrained Lattice Paths," Journal of Applied Probability, vol. 48, no. 3, pp. 567-584, 2011.