Spring 5-8-2020

Lattice Paths in Diagonals and Dimensions

Freya Bennett
Coastal Carolina University, fkbennett@coastal.edu

Lattice Paths in Diagonals and Dimensions

By

Freya Bennett

Applied Mathematics

Submitted in Partial Fulfillment of the
Requirements for the Degree of Bachelor of Science
In the HTC Honors College at
Coastal Carolina University

Spring 2020

Louis E. Keiner
Director of Honors
HTC Honors College

Mark Bly
Assistant Professor of Mathematics
Department of Mathematics and Statistics
Gupta College of Science

Lattice Path's in Diagonals and Dimensions

By: Freya Bennett

Mentor: Dr. Mark Bly

December 2019

Abstract

The Lattice Paths of Combinatorics have been used in many applications, normally under the guise of a different name, due to its versatility in surface variety and specificity of answer. The Lattice Path's of game development, in finding paths around barriers in mazes, is called Path Finder with the A$*$ algorithms as its method of solving.

1 Introduction

Lattice Path's in the $\mathbb{Z}^2$ space can be sorted into two types, the $L(x, y)$ type and the $Q(x, y)$ type, referring to the number of ways an object could move from point $(0, 0)$ to point (x, y). The $L(x, y)$ type counts the number of paths an object could take between $(0, 0) \to (x, y)$, in which every path is made up of movements in only the discrete unit Right or the discrete unit Up. Then the $Q(x, y)$ type counts the number of paths an object could take from $(0, 0)$ to (x, y) while only moving Up, Right, and diagonally Up/Right a single unit at a time. The Lattice Path method of $L(x, y)$ is well known [2] along with $Q(x, y)$ [3] which form the basis of research on all other types of Lattice Path's for applications.

Unlike the $\mathbb{Z}^2$ space, the Lattice Path's in the $\mathbb{Z}^3$ dimensional space can be sorted into five types: $L(x, y, z)$, $Q_{xy}(x, y, z)$, $Q_{yz}(x, y, z)$, $Q_{xz}(x, y, z)$, and $Q_{xyz}(x, y, z)$. In which x is the number of units Right, y is the number of units Up, and z is the number of units Deep representing the Depth. The allowed diagonals for Q are based on Q_{xy} which are the Up/Right diagonals, Q_{yz} being the Up/Deep diagonals, Q_{xz} being the Right/Deep diagonals, and Q_{xyz} are all types of diagonals. Remaining in the discrete, "integer unit" measurements when counting distance.

2 Lattice Path's of Dimension 2

The 2^{nd} dimensional surface can be expressed in Lattice Path's, although this paper will explain in the discrete, using $\mathbb{Z}^2$, instead of the continuous, of $\mathbb{R}^3$, measure of real life. The corresponding graphs are two dimensional, such as Figure 1.

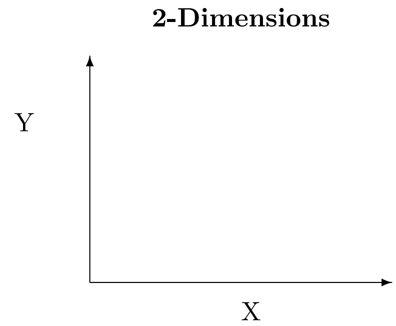


Figure 1: The Basic Visual Second Dimensional Layout

However, beyond the simple context of counting the number of paths, there are multiple formulas for reaching the answer. There is the visual graph method such as in Figure 2 and Figure 3, the Recursive Relation formulas, the Combinatorial formula, and Closed Form Solutions. Each formula and method of counting comes with a different set of properties and proper uses. In Figure 2, this graph is visually better suited for physically overlaying on top of a "grid-like" pattern, such as the roads and city blocks in Cities.

2.1 The L(x, y) Type Lattice Path

The $L(x, y)$ type counts the number of ways an object could move from $(0, 0)$ to (x, y), in which every move is only made as a discrete unit Up or Right. The visual method of expressing the $L(x, y)$ Lattice Path type is the Graph below, where the x and y dimensions can be extended into infinity, or the window of necessity. This graph has an extent of $X : [0, 4], Y : [0, 4]$ for convenience.

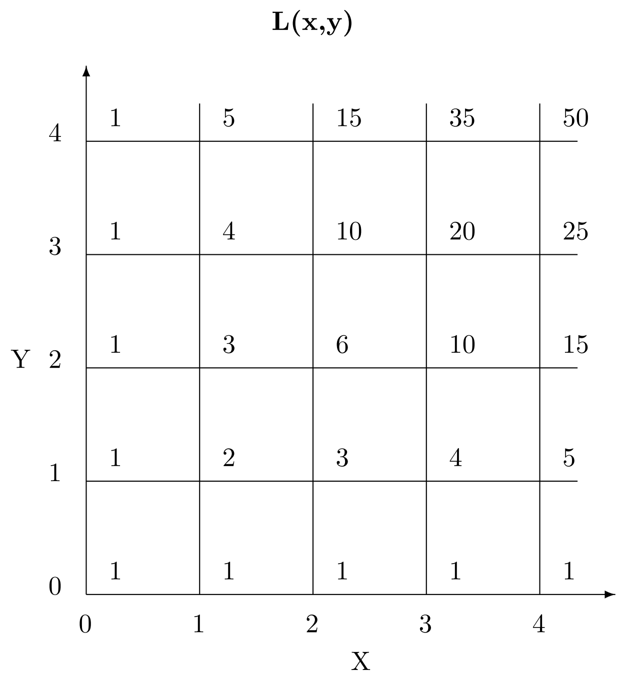


Figure 2: Outputs on the L(x,y) Graph

In the graph of $L(x, y)$ in Figure 2, the 6 at point $(2, 2)$ represents $L(2, 2) = 6$. Thus, every point on the graph is labeled with the number of different lattice paths that can reach said point. The numbers follow the pattern of a sideways Pascal's Triangle where the row (say 3) of the Pascal's Triangle is analogous to the diagonal through the points $(0, 3)$ and $(3, 0)$. We can also use such a graph to continue with the pattern, the number of paths to the point $(2, 3) = 10$ is made from adding the numbers at point $(1, 3) = 4$ and the point $(2, 2) = 6$ together. This pattern can extend the graph and forms the basis for the recursive relation below.

2.1.1 Recursive Relation

In defining how many paths are available to take, we can often express this in terms of how many paths that we could take before, represented as the different "last move"'s that could have been made. A relationship that takes this form is called a Recursive Relation. For $L(x, y)$ the form is:

$$L(x, y) = L(x - 1, y) + L(x, y - 1) \tag{1}$$

Where $L(x - 1, y)$ counts the number of different paths in which the last move was a unit Right to (x, y), and $L(x, y - 1)$ counts the number of paths in which the last move to (x, y) was an Up. Such as how $L(2, 3) = L(1, 3) + L(2, 2)$ from the example before.

2.1.2 Combinatorial Formula and Proof

In Combinatorics, there is a common "n choose r" type of formula that typically expresses the combinations of r number of objects arranged among n number of objects [2]. For the L(x,y) Lattice path, the formula can be written as:

$$L(x,y) = \binom{x+y}{x}$$

In enumerating the different ways for which exactly x number of units Right can be chosen among $x+y$ number of moves. The number of moves for each and every path to (x,y) is $x+y$ as all x units Right and all y units Up must occur for the distance from $(0,0)$ to (x,y) to be achieved. This follows the Pascal's Triangle and Binomial setting in which each row of the Pascal's Triangle is represented by a diagonal in which the x and y of each point adds to the row number. The Pascal's Triangle also contains the same recursive formula as this Lattice Path (from page 26 of [2]). The Binomial formula contains the "Choose" formula to find the coefficients for each variable in the binomial [2].

2.2 The Q(x,y) Type Lattice Path

The $Q(x,y)$ type counts the number of ways an object could move from $(0,0)$ to (x,y) while only moving units Up, Right, and diagonally Up/Right. However, the diagonal unit is still counted discretely despite being a distance of $\sqrt{2}$ and not 1 as it is a single movement from one discrete point to another adjacent discrete point. The values of $Q(x,y)$ can be placed on a first quadrant graph in the same (x,y) locations of the graph as occur in the system of $Q(x,y)$. From a visual perspective, it might be hard to image a system where movement in the Right, Up, and diagonal Up/Right is allowed and not any reverse movement. In real world cases, a current, such as electric or water currents may be involved [1].

A graphical representation of how each of the lattice paths fit, in which at every point (x,y) the corresponding $Q(x,y)$ value is placed, we have Figure 3 below.

In Figure 3 where at point $(3,2)$ the lattice path count with diagonals is $Q(3,2) = 25$. Much as how the $L(x,y)$ graph followed the Pascal's Triangle in the Binomial setting, the $Q(x,y)$ graph is also the triangle of the Delannoy array [3]. Both graph's can be used to continue the pattern and find more points; the pattern for the $Q(x,y)$ graph being seen in $Q(2,3) = 25$ which is also equal to $Q(1,3)+Q(2,2)+Q(1,2) = 7+13+5 = 25$. The pattern is then refined in the recursive relation below.

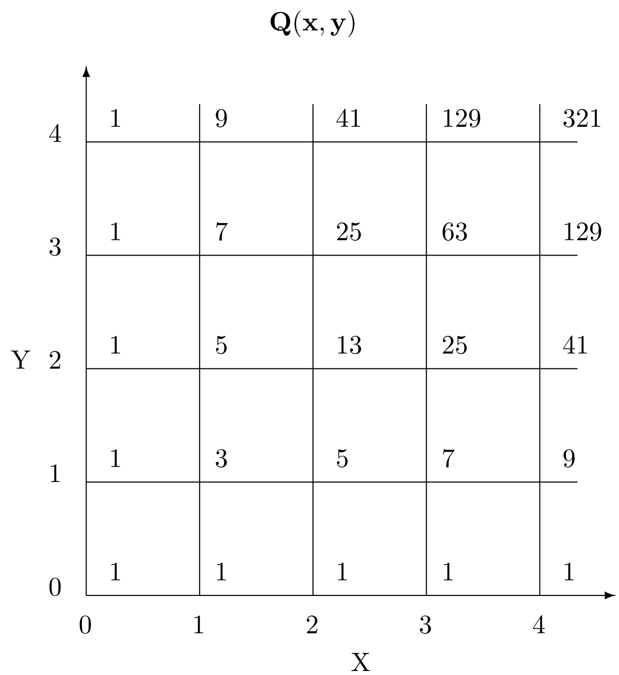


Figure 3: Outputs on the Q(x,y) Graph

2.2.1 Recursive Relation

When expressing the number of paths for a point in terms of the number of paths for the points before it, the recursive relation is used. The recursive relation for $Q(x, y)$ has the form:

$$Q(x, y) = Q(x, y-1) + Q(x-1, y) + Q(x-1, y-1) \tag{2}$$

Where $Q(x-1, y)$ counts the number of different paths in which the last move was a Right unit to (x, y); $Q(x, y-1)$ counts the number of paths in which the last move to (x, y) was an Up unit; then $Q(x-1, y-1)$ counts the number of paths in which the last move to (x, y) was a Diagonal. This makes sense in the context of the the graphical pattern.

2.2.2 Closed Form Solutions

The closed form solution uses the recursive relations to find ”row wise” solutions for $Q(x, y)$ in terms of a formula. The proof for the closed form of $Q(x, y)$ [Eq. 2] can be found algebraically, via $Q(x, 1)$, $Q(x, 2)$ and

$Q(x,3)$ assuming all $Q(x,0)=1$ for row 0. The row 1, $Q(x,1)$ is :

$$\begin{aligned}
Q(x,1) &= Q(x,0)+Q(x-1,0)+Q(x-1,1)\\
&= 1+1+Q(x-1,1)\\
&= 2+Q(x-1,1)\\
&= \sum_{i=0}^{x}(2)+1\\
&= 2x+1
\end{aligned}$$

Then, the form for row 2, $Q(x,2)$ can be found similarly:

$$\begin{aligned}
Q(x,2) &= Q(x,1)+Q(x-1,1)+Q(x-1,2)\\
&= 2x+1+2(x-1)+1+Q(x-1,2)\\
&= 4x+Q(x-1,2)\\
&= 4*\sum_{i=0}^{x}(i)+1\\
&= 2x^2+2x+1
\end{aligned}$$

The form of row 3, $Q(x,3)$ can also be found via:

$$\begin{aligned}
Q(x,3) &= Q(x,2)+Q(x-1,2)+Q(x-1,3)\\
&= 2x^2+2x+1+2(x-1)^2+2(x-1)+1+Q(x-1,1)\\
&= 4x^2+2+Q(x-1,1)\\
&= \sum_{i=0}^{x}(4i^2+2)+1\\
&= \frac{4}{3}x^3+2x^2+\frac{8}{3}x+1
\end{aligned}$$

Thus, for each closed form solution, the number of ways to get to any particular point on row 1 can be found from $Q(x,1)=2x+1$. Similarly, each point on the 2^{nd} row is found on the closed form of $Q(x,2)$, and each point on the 3^{rd} row is found from the closed form of $Q(x,3)$. Noting that the closed form for $Q(x,3)$ includes fractions even though it will always turn out an integer for an integer input, as all points of $Q(x,y)$ have been made from adding integers together. Since no matter how many integers are added their sum will always yield an integer, the fraction form for the closed form solution of row 3 will also always yield an integer answer.

2.2.3 Combinatorial Formula and Proof

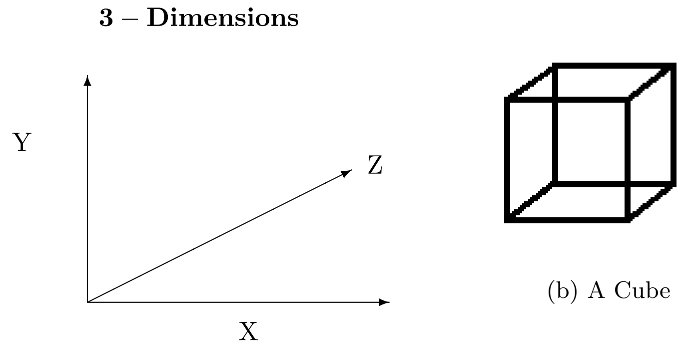
For the $Q(x, y)$ Lattice path, we can use the Summation Combinatorial formula to extend each combinatorial case to cover all possible number of diagonals.

$$Q(x, y) = \sum_{d=0}^{Min\{x,y\}} \begin{pmatrix} x + y - d \\ d, n - d, m - d \end{pmatrix}$$

The combinatorial reasoning is that with each diagonal counted in d's, there can be at minimum no diagonals so $d = 0$ and at max the diagonals can only go as far as the minimum of the distances Up (y) and Right (x). Each of these diagonals will only change the actual unit number distance by one, as instead of both an Up and a Right (2 movements) there is a single diagonal (1 movement). Thus, for every number of diagonals d, the number of total movements is represented by $x + y - d$ for every d number of diagonals. From those total number of movements, the number of ways to rearrange the placement of the Ups, Right, and Diagonals follow a combination. This combination where the number of Ups is $y - d$ as every diagonal takes away from the number of only Up, the number of Rights is $x - d$ as each diagonal takes away a possible Right movement as well, and the number of diagonals is d. The combinations are then added as there is a number of ways to get to (x, y) using 1 diagonal, which are all different from using 2 diagonals, etc. Thus, the total number of different combinations using all of the different possible number of diagonals arranged in every possible way using x Rights and y Ups follows to that formula above.

3 Lattice Path's of Dimension 3

The 3^{rd} dimension can be expressed in continuous Lattice Path's, but this paper focuses on the discrete form; formulas and dimensions rely on $\mathbb{Z}^3$, instead of the continuous $\mathbb{R}^3$ measure of real life. The graphs that this is overlaid on is seen as three dimensional and extends only from $(0, 0, 0)$ to the positive direction (∞, ∞, ∞), such as Figure 4.



(a) The Axis in 3-D

(b) A Cube

Figure 4: The Basic Visual 3rd Dimensional Layout

3.1 The L(x, y, z) Type Lattice Path

The Lattice path of the 3^{rd} dimension, specifically $\mathbb{Z}^3$, where diagonals are not available units is referred to as $L(x, y, z)$ following the convention that L gives the lattice paths without diagonals.

3.1.1 Recursive Relation

This type can also be recursively defined as the number of lattice paths where the last unit movement occurred as either a Right, Up, or Deep. Type $L(x, y, z)$:

$$L(x, y, z) = L(x-1, y, z) + L(x, y-1, z) + L(x, y, z-1) \tag{3}$$

Where the total number of lattice paths is broken into the number of different paths given the last unit was a: Right $L(x-1, y, z)$, Up $L(x, y-1, z)$, or Deep $L(x, y, z-1)$ to reach (x, y, z).

3.1.2 Combinatorial Formula and Proof

The clear combinatorial formula for $L(x, y, z)$, similar in look to the combinatorial formula for $L(x, y)$, for any discrete distance of x units Right, y units Up, and z units Deep is:

$$L(x, y, z) = \begin{pmatrix} x+y+z \\ x, y, z \end{pmatrix}$$

This occurs as the total number of units (n) that must be crossed between $(0, 0, 0)$ and (x, y, z) is a distance of $x + y + z$ units. The number of different ways that the x, y, and z units can be individually rearranged in order following the combinatorial formula of "n choose x and y and z." Looking individually in terms of types of movements, the x types can be rearranged in the string n units long, then the y types can be rearranged in the remaining string $n-x$ units long, with the z types being all of the remaining string units. Formally, we can see:

$$\begin{pmatrix} n \\ x, y, z \end{pmatrix} = \begin{pmatrix} n \\ x \end{pmatrix} * \begin{pmatrix} n-x \\ y \end{pmatrix} * \begin{pmatrix} n-x-y \\ z \end{pmatrix}$$

Factoring back in that $n = x + y + z$, we can break each of these choose formula down into a simple binomial coefficient. The first portion starts out as binomial coefficient, but in relating to it's own direction type as well as the other unit types we can clearly see this merger of three Binomials is the Trinomial case of the combination formula.

$$\begin{pmatrix} n \\ x \end{pmatrix} = \begin{pmatrix} x+y+z \\ x \end{pmatrix}$$

The second choose formula represents the left over $n - x = y + z$ units of distance that the y units can rearrange into. In this, the number ways to choose y units out of a total distance $y + z$ units is the same number of ways to choose y units from the remaining $n - x$ units.

$$\begin{pmatrix} n-x \\ y \end{pmatrix} = \begin{pmatrix} y+z \\ y \end{pmatrix}$$

In the last combination, the number of ways z units can be chosen out of $n - x - y$ leftover units (or the number of units left after all other types of units were removed) is the same as the number of ways z units can be chosen from $n - x - y = (x + y + z) - x - y = z$ units. If all the number of units left to be chosen is exactly the units z that have yet to be placed, then there is exactly one way to place z units in z number of slots left. The last formula becomes,

$$\begin{pmatrix} n-x-y \\ z \end{pmatrix} = \begin{pmatrix} (x+y+z)-x-y \\ z \end{pmatrix} = \begin{pmatrix} z \\ z \end{pmatrix}$$

Which is the mentality of having $x + y + z$ units and placing x units then y units then z units in an order.

3.2 The $\mathbf{Q}_{xy}(x, y, z)$ Type Lattice Path

With lattice Path's in 3 dimensions, we can specify that only the xy diagonal exists, such is the form $Q_{xy}(x, y, z)$ in this paper. This type of lattice path uses much of the type $L(x, y, z)$ as its base along with the xy diagonals as its range of possibilities.

3.2.1 Recursive Relation

The Lattice path of the 3^{rd} dimension, specifically $\mathbb{Z}^3$, where diagonals are available units is referred to as $Q(x, y, z)$ following the convention that Q gives the lattice paths possible diagonals. Further, the Q_{xy} is denoting the only allowed diagonal on the three dimensional path, the Up/Right diagonal. This type is also defined recursively as the number of lattice paths where the last unit movement occurred as either a Right, Up, Deep, or diagonally Up/Right. Type $Q_{xy}(x, y, z)$:

$$Q_{x,y}(x, y, z) = Q(x, y - 1, z) + Q(x - 1, y, z) + Q(x, y, z - 1) + Q(x - 1, y - 1, z) \tag{4}$$

This sections off the final count recursively into the counts of the last move (in order of appearance) being an Up, Right, Deep, or a diagonally Up/Right.

3.2.2 Combinatorial Formula and Proof

Using the formula for $L(x, y, z)$ as a base, and then adding every combination where the number of xy diagonals is $0, 1, 2, 3, \cdots, Min(x, y)$. We can see that $Q_{xy}(x, y, z)$ takes the form;

$$Q_{xy}(x, y, z) = \sum_{d_{xy}=0}^{Min(x,y)} \binom{x + y + z - d_{xy}}{x - d_{xy}, y - d_{xy}, z, d_{xy}}$$

Where the number of diagonals xy (noted as d_{xy}) must stop at the minimum of the two directions x and y. The number of diagonals, although taking an available unit away from both x and y results in replacing 2 units with 1 diagonal unit, so from every total number of units of movement only 1 unit is removed per diagonal.

3.3 The $\mathbf{Q}_{xz}(x, y, z)$ Type Lattice Path

This is where we are allowing for the xz diagonal only, no other diagonals, among the available types of unit movements.

3.3.1 Recursive Relation

The Lattice path of the 3^{rd} dimension, specifically $\mathbb{Z}^3$, where diagonals are available units is referred to as $Q(x, y, z)$ following the convention that Q gives the lattice paths possible diagonals. Further, the Q_{xy} is denoting the only allowed diagonal on the three dimensional path, the Up/Right diagonal. This type is also defined recursively as the number of lattice paths where the last unit movement occurred as either a Right, Up, Deep, or diagonally Up/Right. Type $Q_{xz}(x,y,z)$:

$$Q_{xz}(x,y,z) = Q(x,y-1,z) + Q(x-1,y,z) + Q(x,y,z-1) + Q(x-1,y,z-1) \tag{5}$$

This sections off the final count recursively into the counts of the last move (in order of appearance) being an Up, Right, Deep, or a diagonally Right/Deep.

3.3.2 Combinatorial Formula and Proof

Using the formula for $Q_{xz}(x,y,z)$ as a reference; however, instead of the combinations for the xy diagonals, the combinations for the xz diagonals will be is $0,1,2,3,\cdots,Min(x,z)$. We can see that $Q_{xz}(x,y,z)$ takes the form;

$$Q_{xz}(x,y,z) = \sum_{d_{xz}=0}^{Min(x,z)} \binom{x+y+z-d_{xz}}{x-d_{xz}, y, z-d_{xz}, d_{xz}}$$

The number of possible Lattice Path's, given the possible directions are x, y, z and xz, is the sequence of all combinations with every successive xz diagonal added together. First, each combination represents a string of movements of all possible orders; a string of length n where $n = x+y+z$ will

3.4 The $\mathbf{Q}_{yz}(x,y,z)$ Type Lattice Path

This is the Lattice Path, in $\mathbb{Z}^3$, that allows for diagonals along the yz or Up/Deep variety only with the base x, y, and z units.

3.4.1 Recursive Relation

Again, we can express the number of Lattice Path's to any (x,y,z) point in this specific Q_{yz} type recursively. Type $Q_{yz}(x,y,z)$:

$$Q_{yz}(x,y,z) = Q(x,y-1,z) + Q(x-1,y,z) + Q(x,y,z-1) + Q(x,y-1,z-1) \tag{6}$$

This sections off the final count recursively into the counts of the last move (in order of appearance) being an Up, Right, Deep, or a diagonally Up/Deep.

3.4.2 Combinatorial Formula and Proof

Using the formula for $L(x, y, z)$ as a base, and then adding every combination where the number of yz diagonals is $0, 1, 2, 3, \cdots, Min(y, z)$. We can see that $Q_{yz}(x, y, z)$ takes the form;

$$Q_{yz}(x,y,z) = \sum_{d_{yz}=0}^{Min(y,z)} \binom{x+y+z-d_{yz}}{x, y-d_{yz}, z-d_{yz}, d_{yz}}$$

In explanation, this is much the same to the form of both Q_{xy} and Q_{xz} only instead of the xy diagonals or the xz diagonals, the yz diagonals are used instead and take away 1 unit from both the y and z directions. All else follows similarly to those prior arguments.

3.5 The $\mathbf{Q}_{xyz}(x, y, z)$ Type Lattice Path

This time, the Lattice Path type is given allowance to all the different diagonals xy, yz, xz, and xyz along with the unit movements x, y, and z directions.

3.5.1 Recursive Relation

The type $Q_{xyz}(x, y, z)$ involving all diagonals and all unit directions:

$$\begin{aligned} Q_{xyz}(x,y,z) =& Q(x,y-1,z) + Q(x-1,y,z) + Q(x,y,z-1) + Q(x-1,y-1,z) \\ &+ Q(x,y-1,z-1) + Q(x-1,y,z-1) + Q(x-1,y-1,z-1) \end{aligned}$$

This sections off the final count recursively into the counts of the last move (in order of appearance) being an Up, Right, Deep, diagonally Up/Right, diagonally Up/Deep, diagonally Right/Deep, and diagonally Up/Right/Deep.

3.5.2 Combinatorial Formula and Proof

The form for $Q_{xyz}(x, y, z)$ is a quadruple summation on a septanomial coefficient.

For the $Q_{xyz}(x, y, z)$ Lattice path, we can write our formula $Q_{xyz}(x, y, z)$:

$$= \sum_{D=0}^{A} \sum_{d_{xy}=0}^{B} \sum_{d_{yz}=0}^{C} \sum_{d_{xz}=0}^{E} \binom{x+y+z-2D-d_{xy}-d_{yz}-d_{xz}}{x-D-d_{xy}-d_{xz}, y-D-d_{xy}-d_{yz}, z-D-d_{xz}-d_{yz}, D, d_{xy}, d_{yz}, d_{xz}}$$

Where setting $A = Min\{x, y, z\}$, $B = Min\{x - D, y - D\}$, $C = Min\{y - D - d_{xy}, z - D\}$, and $E = Min\{x - D - d_{xy}, z - D - d_{yz}\}$ gives the end values for each of their respective summations.

Where this describes the total number of ways to get from point $(0,0,0)$ to point (x,y,z) using any diagonal form via an (d_{xy}) Up/Right, (d_{xz}) Right/Deep, (d_{yz}) Up/Deep, and (D) Right/Up/Deep. Each summation takes into account the used x, y, and z's from the previous diagonals and summations so that if one dimension (like x) has already been used up then the next diagonal that needs it (such as d_{xz}) will never get past 0 diagonals. This then sums all the possible rearrangements of the diagonals in all 4 directions based on the available x, y, and z units of movements.

4 Other Versions

As Albert Einstein said, "The Fourth Dimension is Time." So too will we note our (x,y,z,t) point with $t =$ time, in a special $L(x,y,z,t)$ Lattice Path.

This is a spacial lattice path moving in Right, Up, Deep, and Time distances. $L(x,y,z,t) = \begin{pmatrix} x+y+z+t \\ x,y,z,t \end{pmatrix}$

This would follow the prior form used extensively in both the $L(x,y)$ and the $L(x,y,z)$ versions in the $\mathbb{Z}^2$ and $\mathbb{Z}^3$ respectively. The proof of this follows much the same reasoning as with $L(x,y,z)$, only instead of a Trinomial combination from the 3 dimensions, this would be a Quadronomial combination due to the arrangement of 4 different dimensions. The use of this is limited in that the accurate knowledge of the length of time allowed for a situation is not always feasible. This idea of multi-variable Lattice Path type is used, even with diagonals included in modern research [1]. However, there are the cases in which we want to specify a different combination of possible directions. Other diagonal specifications might be that only the xy and yz are allowed among the unit directions.

4.1 Two Diagonals and Unit directions

Staying in the $\mathbb{Z}^3$, perhaps we wish to specify certain diagonals to be allowed, such as the case where a video game character can jump to the right or further away from the camera screen to a platform higher up. This Lattice Path would specify in the xy, yz directions, $Q_{xy,yz}$ being the useful notation. In this case the formula could be written recursively,

$$Q_{xy,yz}(x,y,z) = Q_{xy,yz}(x-1,y,z)+Q_{xy,yz}(x,y-1,z)+Q_{xy,yz}(x,y,z-1)+Q_{xy,yz}(x-1,y-1,z)+Q_{xy,yz}(x,y-1,z-1)$$

In which the last movements in the xy and yz diagonals were counted along with the individual unit directions x, y, and z. Then, the full formula can also be seen:

$$Q_{xy,yz}(x,y,z) = \sum_{d_{xy}=0}^{Min(x,y)} \sum_{d_{yz}=0}^{Min(y-d_{xy},z)} \binom{x+y+z-d_{yz}-d_{xy}}{x-d_{xy}, y-d_{yz}-d_{xy}, z-d_{yz}, d_{yz}, d_{xy}}$$

Where each diagonal and unit direction have their own slot in the computation and the unit directions are only decreased by the amount of diagonals in that direction. The summations serve as counters, much like in computer coding where a small portion of code is to repeatedly run only a certain number of times before the rest of the program may finish. The first summation, like a first while loop, specifies that starting at zero diagonals in the xy direction the computation becomes:

$$\sum_{d_{yz}=0}^{Min(y-0,z)} \binom{x+y+z-d_{yz}-0}{x-0, y-d_{yz}-0, z-d_{yz}, d_{yz}, 0}$$

As each of the zeros occur where the counter d_{xy} exists for the number of xy diagonals. As the number of xy diagonals increase, so does the the available y units in the $Min(y,z)$ and the computation. This reflects the fact that for every xy diagonal done, the number of y units available for yz diagonals have decreased. The second summation also counts all the different possible path's given each new count of yz diagonals as d_{yz} and given whatever number of xy diagonals have been done. At some point, $y - d_{xy} = 0$ and the number of yz diagonals will be forced to be zero due to the xy diagonals exceeding the limits to what the y direction can handle. Thus we can rewrite this same formula for any two diagonal directions desired and the corresponding three unit directions by simply changing the Min's variables, and changing the directions the diagonals are subtracted from.

4.2 One Diagonal and Unit Directions

Similarly to the prior Q_{xy}, Q_{yz}, and Q_{xz} forms, we can also write another Lattice Path specification with only the D_{xyz} diagonal used.

This Q_{xyz} form can be seen as:

$$Q_{xyz} = \sum_{D_{xyz}=0} \binom{x+y+z-2D_{xyz}}{x-D_{xyz}, y-D_{xyz}, z-D_{xyz}, D_{xyz}}$$

In which the diagonal takes D_{xyz} number of units away from each of x, y, and z units in the arrangement, but the only takes two units from the total distance to move as for every 3 units taken there is 1 returned meaning a net change of -2 units.

This xyz diagonal in contrast to the diagonals with 2 dimensions before, operates in the visually apparent "interior" whereas the xy, yz, and xz diagonals appear on the "faces" of a 3 dimensional object such as the cube in Figure 4.

4.3 One Diagonal and Two Unit Directions

Suppose that not all dimensions were used in a path, or even that a Lattice Path needed to be specified such that only a diagonal was used in that direction. This could take multiple forms, but an easy example is if the particular unused unit dimension was y. Let's call this form, $J_{xyz,y}(x, y, z)$ in that the D_{xyz} diagonal is used in the 3^{rd} dimension and the unit excluded is y. Too make this work, at minimum, the number of diagonals must be equal to the number of y units it must replace, else the form would be useless. To have that number of diagonals, both the z and x dimensions must be either greater than or equal to in number of units to the number of y units otherwise the D_{xyz} diagonal could not form to its full length. This form:

$$J_{xyz,y}(x, y, z) = \begin{pmatrix} x + y + z - 2D_{xyz} \\ x - D_{xyz}, y - D_{xyz}, z - D_{xyz}, D_{xyz} \end{pmatrix}$$

This does not include a summation for the diagonals because the diagonals must occupy all of the units of y and not exceed, meaning $D - xyz = y$ for this case. For any other versions of this, the diagonal must include the unit type its replacing, it must be equal to the total length of the replaced unit, and it will follow whatever form the diagonal came from only without a summation as there is only 1 specific number of diagonals allowed in this.

4.4 Gaming Applications

In the Game development world, it might be necessary to program certain enemies to take the shortest possible path to the player character all while avoiding the walls and obstacles. Other times, simulations attempt to program paths in which objects (such as water points) move in relation to the shortest possible distance while refraining from actually moving through objects. Different movement types can be applied to non-player characters/ objects with the specifications given such as Seek, Flee, Follow, Separate, and Wander etc. These games are set on a discrete (whole number line) map. The Algorithms for such paths form the discipline of Path Finding, the most popular medium of Path Finding is A$*$.

4.4.1 2D Gaming

Let Pac-Man be a game example where the "enemies" "seek" out the player character Pac-Man. In this, the red ghost is always moving in the shortest path to where Pac-Man is located, meaning the program the red ghost has to follow a continuous loop occurring every time Pac-Man changes position. This loop is the Path Finding loop where a small code is run dozens of times a second to find the shortest path.

Notice the difference in the goal, Path Finding looks for the shortest distance path or the path which follows criteria the programmer wants to implicate, whereas the Lattice Path problems above look for the number of different possible paths. The looped code might have certain provisions, such as adding weight to every movement and enormous weight to movements that involve unwanted locations etc, then the code will pick the path with the smallest weight. This code could tell the larger program what the shortest distance is, what direction the shortest path is etc. It doesn't tell how many paths were considered nor how many paths fit all of the criteria. To visualize:

$$\begin{vmatrix} X,X,X,X,X,X \\ X,O,O,O,E,X \\ X,O,O,X,O,X \\ X,S,O,O,O,X \\ X,X,X,X,X,X \end{vmatrix}$$

Figure 5: A Possible Simple Maze

Start at S, then move to E across only the O's. In Path Finding, the shortest path distance can be found via the criteria that only unit movements (either an up or a right of one unit) can be made. The weights that would be added would be a 1 for every movement to a O and a 100 (or something large) for every movement to an X. By adding the weights for each path, then specifying to only accept the smallest path total weight, there would be a shortest total distance of 5. It would not tell how many paths were that distance unless a counter was added for it, but it would show the shortest distance possible.

Lattice Path's would obviously point out the shortest distance to be 5, as all path's that it would even count would have a distance of 5. A $L(3,2)$ would give the total number of paths (distance = 5), but not the number of paths without moving across the X. To consider the number of paths following the "no-moving-over-X" criteria, consider locations where no paths would cross the X. Locations such as: $(0,1)$, $(0,2)$, $(1,1)$, $(1,2)$, $(1,0)$, and $(2,0)$ would not have a path that crosses the X but locations $(1,2)$ and $(2,0)$ are the furthest from the origin (the S location). There is only one path to $(3,2)$ after passing through points either $(1,2)$ or $(2,0)$. Thus $L(3,2) = L(1,2) + L(2,0)$, which means there are 4 different paths that are possible.

With a different maze this would get more difficult.

Thus, these different approaches solve different problems. In context, the Path Finding method involves a long code to continuously check all different paths in real time. This is cumbersome as it has to check in the negative directions (left and down in this case) as well as positive directions for every movement, and repeating itself if the path did not reach the goal point at the end. The Lattice Path's tell the number of possible paths, but only at a particular instant in time.

5 Conclusion

Lattice Path's are versatile in the different constructions of paths they could consider. From a flat 2-dimensional surface to the 3-d and 4-d surfaces that exist, they can denote different structures where diagonal paths are allowed to the cases of being only able to move in the diagonal along an axis. The mentality behind them is used in programming movement on surfaces and decision making for code to follow. The Combinatorial Lattice Path's are simply yet another tool used in applications and deep math.

References

[1] Singh, Y., Sharma, S., Sutton, R., Hatton, D., Khan, A, *USV's in the Marine Environment* (2018). A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. Ocean Engineering, 169, 187–201. emphhttps://doi.org/10.1016/j.oceaneng.2018.09.016

[2] C. G. Wagner, *Basic Combinutorics*, (2014) Book. Knoxvile, TN: CreateSpace Independent Publishing Platform.

[3] Eric W. Weisstein, *Delannoy Number* From *MathWorld*–A Wolfram Web Resource. *http://mathworld.wolfram.com/DelannoyNumber.html*

[4] 19 Authors, *A* Search Algorithm* From *RosettaCode.org*–Powered by Semantic MediaWiki, a GNU Free Document Licence 1.2. *https://rosettacode.org/wiki/A*_search_algorithm*