Knowledge and Decision Systems with Applications

www.kadsa.org
eISSN: 3079-6318

Generalizing Pascal's Triangle through Mathematical Modeling: Meta Triangles and Cross-Domain Applications in Energy, Education, and Systems Analysis

Wajdi Mohamed Ratemi [1,*]

[1] Professor Emeritus Nuclear Engineering Department, Faculty of Engineering, University of Tripoli, Tripoli, Libya

ARTICLE INFO

Article history:
Received 10 February 2026
Received in revised form 20 April 2026
Accepted 2 May 2026
Available online 4 May 2026

Keywords:

Pascal's triangle; generalized Pascal triangles; binomial expansion; multinomial expansion; nuclear reactor kinetics; combinatorics; probability distribution functions; genome's tagging; fault tree analysis; computer graphics; knowledge and decision systems.

ABSTRACT

This paper introduces an innovative mathematical construct known as the Guelph Expansion, which enables the generalization of Pascal's Triangle into a family of Meta Triangles. Two efficient algorithms, the Embedded Pascal Triangles (EPTs) method and the Staircase Horizontal–Vertical (SHV) method, are developed to systematically generate these structures while enhancing numerical efficiency and scalability. In addition, the paper utilizes the Guelph expansion to formulate a novel Inhour Polynomial and a corresponding Coefficient-Based Model (CBM) for nuclear reactor kinetics, contributing to robust, and reliable reactor kinetics modeling. Beyond energy applications, the proposed framework demonstrates broad interdisciplinary relevance, including computational approaches for surface generation in computer graphics, genome tagging and DNA sequencing, fault tree analysis, and probabilistic modeling. By unifying diverse applications within a single mathematical tool, the study supports knowledge integration, and decision making across scientific disciplines.

1. Introduction

Pascal's Triangle [1] is a famous combinatoric structure which represents the coefficients of a binomial expansion. Though mathematical formularization of multinomial expansion is available, the generalization of the structure of Pascal's triangle to higher dimensions remains a subject of theoretical interest. Simultaneously, nuclear reactor kinetics, defined by multigroup point reactor kinetics [2], presents a challenge due to the wide range in system time constants. This research addresses both challenges via the introduction of a unified algebraic framework.

This paper introduces a mathematical tool named as the Guelph expansion [3-5] which provides the basis for developing a generalization of Pascal's triangle to Meta triangles with the development of two algorithms to achieve that, namely; the Embedded Pascal Triangles (EPTs) method [5,6] and the Staircase Horizontal Vertical (SHV) method [7] as well as, a development of the Inhour polynomial [3] and the Coefficients Based Model (CBM) [8].

* *Corresponding author.*
E-mail address: wmratemi@gmail.com
w.ratemi@uot.edu.ly
https://doi.org/10.59543/5r7mgj94

An interesting identity of the Guelph expansion formula is that it provides a unified algebraic method for determining polynomial roots from coefficients and coefficients from roots [9], offering a robust alternative to Vieta's formulas [10]. Furthermore, with this identity, one can relate the polynomial coefficients, and roots to the quotients of polynomial derivatives and the factorials as presented by the Tylor expansion formula [11]. In this paper, as well, this identity has been utilized to introduce Tripoli Polynomials [12] which can generate the elements of the Waterloo matrix which are used as a feed stage of the SHV-method algorithm in generating the Meta Triangles to be discussed later.

The Broader contribution of this work can be extended to other disciplines such as combinatorics [6], DNA sequencing and genomics [6, 13], Fault Tree Analysis [14], and computer graphics (Bezier curves), as to be presented in this paper.

Figure 1 presents the modular structure of the paper.

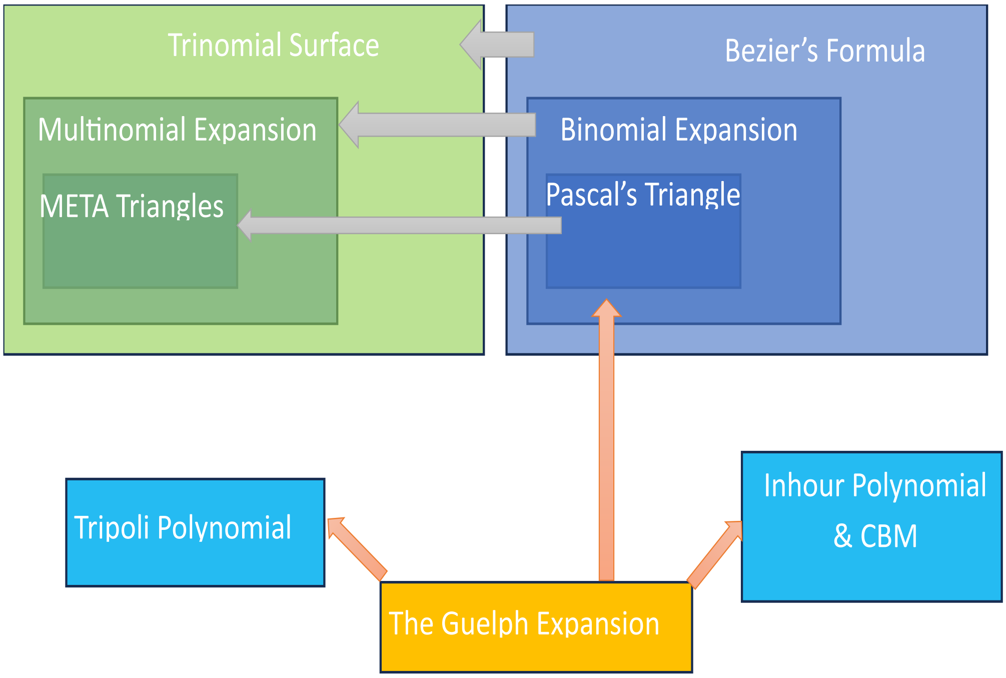

Fig.1. The Paper's Modular Structure

The figure demonstrates that the Guelph expansion generates Tripoli Polynomials and the Inhour polynomial which produces the CBM. Furthermore, The Guelph expansion yields the binomial expansion formula where its binomial coefficient represents Pascal's Triangle. With the dot product of Bezier's control points with the summand of the binomial expansion, Bezier's curve formula is produced. The Structure to the left presents the corresponding generalizations resulting consecutively; the Meta Triangles, the multinomial expansion, and the trinomial surface formula.

The remainder of this paper is organized as follows. Section 2 establishes the necessary Mathematical Preliminaries and Notation. It introduces the Guelph expansion and its dual identity, along with its relation to Vieta's formula and Taylor expansion. Furthermore, it sets the background for developing the Inhour polynomial and Tripoli polynomials. Section 3 develops the Mathematical Expansion Theory, detailing the Guelph Expansion, the construction of the Generalized Pascal's Triangles, and the EPTs and SHV methods. Section 4 introduces the Nuclear Reactor Kinetics (the

Inhour polynomial and the CBM), and presenting a stability comparison. Section 5 explores Other Applications of the Meta Triangles and Multinomial Expansion to probability analysis, DNA sequencing (Tagging of genomes), and extension of Bezier's formula. Finally, Section 6 provides the Conclusion.

This study addresses the extension of Pascal's triangle structure which represents the coefficients of binomial expansion into planar structures representing multinomial coefficients of multinomial expansion. The author devised two algorithms to achieve that, namely the EPTs, and the SHV methods. The significance of this study is to avoid multinomial coefficients calculation using factorial formulas. Applications can be in combinatorics, probability theory, number theory, genetics analysis, and others. In this study, the author introduces innovative mathematical construct known as the Guelph expansion which enables the generalization of Pascal's triangle into a family of Meta triangles. This devised tool was also used to introduce a new representation of the characteristic equation of nuclear reactor kinetics named as the inhour polynomial. This in turn introduces what is called the Coefficients Based Model, CBM, as a model of nuclear reactor kinetics. The analytical solution of the CBM representation overcomes the stiffness problem associated with the numerical solution of the classical representation. Hence, enhancing the modeling of safety studies of nuclear reactors. The paper with its diversity in structure serves different sectors such as education, energy, health, and others.

2. Mathematical Preliminaries and Notation

2.1. Mathematical Background

2.1.1. The Guelph Expansion

To determine the result of expanding the following equation:

$$\prod_{i=1}^{n}(\omega+\lambda_i)=(\omega+\lambda_1)(\omega+\lambda_2)(\omega+\lambda_3)\ldots\ldots..(\omega+\lambda_n)\ , \tag{1}$$

the expression in Eq. (1) can be expanded by direct term multiplication and grouping terms. It is presented in a compact form by Eq. (2) (a detailed derivation can be found in [3]):

$$\prod_{i=1}^{n}(\omega+\lambda_i)=\sum_{k=0}^{n}\omega^{n-k}\sum_{\binom{n}{k}}\lambda\ldots^{k}\ldots\lambda \tag{2}$$

where $\sum_{\binom{n}{k}}\lambda\ldots^{k}\ldots\lambda$ denotes the sum of the products of each and every possible combination of k elements of the set $\lambda_1,\ldots,\lambda_n$.

The number of such combinations is the binomial coefficient:

$$\binom{n}{k}=\frac{n!}{k!(n-k)!} \tag{3}$$

This devised tool was later named The Guelph Expansion [4].

2.1.2. Binomial expansion and Pascal's triangle

Substituting λ for λi (where λ is a single-valued parameter, not distinct λi) in the Guelph Expansion formula, Eq. (2) yields:

$$\prod_{i=1}^{n}(\omega+\lambda_i)=(\omega+\lambda)^n=\sum_{k=0}^{n}\omega^{n-k}\sum_{\binom{n}{k}}\lambda\dots^{k}\dots\lambda=\sum_{k=0}^{n}\omega^{n-k}\binom{n}{k}\lambda^k \quad (4)$$

Rewriting Eq. (4) using the variables x and y yields the form of the binomial expansion:

$$(x+y)^n=\sum_{k=0}^{n}\binom{n}{k}x^{n-k}\,y^k\,, \quad (5)$$

The binomial coefficient $\binom{n}{k}$ (for n ≥ 0 and 0 ≤ k ≤ n) generates what is known in the Western world as Pascal's triangle, in China as Yang Hui's triangle, and in Iran as Khayyam's triangle [15, 16].

Table 1 presents the Elements of Pascal's triangle.

Table 1
Pascal's Triangle

n / k	0	1	2	3	4	…..
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
…..						

In combinatorics study Kollar discussed the Janjić–Petković Inset Counting Function which for special case reduces to binomial coefficients [17].

2.1.3. Binomial probability distribution function

The summand in Eq. (5) represents the binomial probability distribution function:

$$Bi_{pdf}=\binom{n}{k}x^{n-k}y^{k} \quad (6)$$

Given that I is the number of variables, and $x=y=\frac{1}{I}$, then

$$Bi_{pdf}=\frac{\binom{n}{k}}{2^n} \quad (7)$$

2.1.4. A unified method for polynomial root and coefficient determination

A close examination to the expression for the Guelph expansion (Eq. 2) and a polynomial of the form $\sum_{k=0}^{n}A_{n-k}\omega^{n-k}=0$, with roots at $\omega=-\alpha_i$ reveals that:

$$\sum_{k=0}^{n}\omega^{n-k}\sum_{\binom{n}{k}}\alpha\dots^{k}\dots\alpha=\sum_{k=0}^{n}A_{n-k}\omega^{n-k}\,, \quad (8)$$

with

$$A_{n-k}=\sum_{\binom{n}{k}}\alpha\dots^{k}\dots\alpha\quad,k=0,1,2,\dots n \quad (9)$$

Eq. (9) provides the expressions for the polynomial's coefficients as the sum of related combinations, *k*, of its roots ($\omega = -\alpha_i$). This is a two-way calculation: generating the coefficients from the roots, or solving the resulting system of nonlinear equations to determine the roots from the coefficients [6, 9].

Example: If the roots are -1, -2, then n=2, and k=0, 1, 2

$A_{2-0} = \sum_{\binom{2}{0}=1} \alpha \ldots^{0} \ldots \alpha \;= 1$, $A_{2-1} = \sum_{\binom{2}{1}=2} \alpha \ldots^{1} \ldots \alpha \;= +1 + 2 = +3,$

$A_{2-2} = \sum_{\binom{2}{2}=1} \alpha \ldots^{2} \ldots \alpha \;= (+1)(+2) = +2.$

The coefficients, then are:

$A_2 = 1,\ A_1 = 3,\ A_0 = 2$, and the polynomial has the form: $\omega^2 + 3\omega + 2 = 0$.

On the other hand, knowing the coefficients: $A_2 = 1, A_1 = 3, A_0 = 2$, one needs to solve system of 2-equations: $3 = -\alpha 1 - \alpha 2,\ and\ 2 = (-\alpha 1)(-\alpha 2)$, solving these equations yields $\alpha 1 = -1,$ $and\ \alpha 2 = -2$.

It is observed that when the roots are consecutive negative integers—such as (-1), (-1, -2), (-1, -2, -3), and so on—one can generate polynomials. When these polynomials are normalized by dividing the entire polynomial by the coefficient $A_{0=} \sum_{\binom{m}{m}} \alpha \ldots^{m} \ldots \alpha = m!$ (the constant term), they yield the following formula:

$$T_m(n) = \sum_{k=0}^{m} n^{m-k} \left(\frac{\sum_{\binom{m}{k}} \alpha \ldots^{k} \ldots \alpha}{m!} \right) \tag{10}$$

These polynomials are referred to as Tripoli polynomials [12]. They are considered as the generators for the columns of the Waterloo matrix , which are subsequently used in generating the Meta triangles (to be discussed later). One can easily use Eq. (10) to find the first few Tripoli polynomials:

- The first-degree Tripoli polynomial (corresponding to one root: $\alpha 1$ = -1, with m=1 and k∈ {0, 1}) is $T_1(n) = n + 1$.
- The second-degree Tripoli polynomial (corresponding to two roots: $\alpha 1$ = -1, $\alpha 2$ = -2, with m=2 and k∈ {0, 1, 2}) is $T_2(n) = \frac{n^2}{2} + \frac{3}{2}n + 1$.
- The third-degree Tripoli polynomial (corresponding to two roots: $\alpha 1$ = -1, $\alpha 2$ = -2, $\alpha 3$ = -3, with m=3 and k∈ {0, 1, 2, 3}) is $T_3(n) = \frac{n^3}{6} + n^2 + \frac{11}{6}n + 1$.

2.1.5. Conceptual ties with other classical mathematical theorems

It is worth noting that the Guelph Expansion has conceptual ties with both Vieta's formulas [10] (which relate polynomial coefficients to the sums and products of its roots) and the Taylor expansion formula [11], which calculates the coefficients via quotients of function derivatives and factorials. Table 2 presents a summary of these ties.

Table 2
Conceptual Ties Between the Guelph Expansion, Vieta's Formulas, and the Taylor Expansion

Vieta's formulas (Coefficients ↔ Roots)	Taylor Expansion	Guelph Expansion
$\sum_{1\le i_1<i_2<\cdots<i_k\le n} r_{i1}\, r_{i1} \ldots \ldots r_{ik} = (-1)^k \frac{a_{n-k}}{a_n}$	$\sum_{k=0}^{n} \omega^{n-k} \frac{f^{(n-k)}(0)}{(n-k)!}$	$\sum_{k=0}^{n} \omega^{n-k} A_{n-k} = \sum_{k=0}^{n} \omega^{n-k} \sum_{\binom{n}{k}} \alpha \ldots^{k} \ldots \alpha$

2.1.6. Bezier's curves connection

When examining Bezier's curves formula [18], one can connect it to binomial expansion formula by simply making the dot product of the summand of the binomial expansion with the control points, P_k, suggested by Bezier. In *x, y* notation, and using parameter t for *y*, and *1-t* for *x*, the Bezier's formula which generates Bezier's curves is given by:

$$B(t) = \sum_{k=0}^{n} x^{n-k} \binom{n}{k} y^k . P_k = \sum_{k=0}^{n} (1-t)^{n-k} \binom{n}{k} t^k . P_k \quad , \quad t \in [0,1] \tag{11}$$

2.1.7. The Inhour polynomial

The classical inhour equation for n groups of delayed neutrons is given by:

$$\rho = \Lambda\omega + \sum_{i=1}^{n} \frac{\beta_i \omega}{\omega+\lambda_i} \quad , \tag{12}$$

where, Λ represents the neutron generation time, ρ is the reactivity, β_i is the ith delayed neutron fraction, λ_i is the ith group precursors decay constant, and ω is the roots of the equation, representing the reciprocal of system time constants. Expanding Eq. (12) yields:

$$(\Lambda\omega - \rho)\prod_{i=1}^{n}(\omega+\lambda_i) + \beta_1 \prod_{i\ne 1}^{n}(\omega+\lambda_i) + \beta_2 \prod_{i\ne 2}^{n}(\omega+\lambda_i) + \cdots \beta_n \prod_{i\ne n}^{n}(\omega+\lambda_i) = 0 \tag{13}$$

Analysis of Eq. (13) shows that the π-terms can be expanded using the Guelph Expansion, Eq. (2). Subsequent mathematical manipulation of this expansion yields a new compact form of the inhour polynomial (detailed derivation is provided in [3]). The Inhour polynomial is given by:

$$F(\omega) = \sum_{m=0}^{n+1} A_m \omega^m = 0 \, , \tag{14}$$

where,

$$A_m = \Lambda \sum_{\binom{n}{k}} (\lambda \ldots^{k} \ldots \lambda) + \sum_{i=1}^{n} \beta_i \sum_{\binom{n-1}{k-1}} (\lambda \ldots^{k-1} \ldots \lambda)_{\ne i} - \rho \sum_{\binom{n}{k-1}} (\lambda \ldots^{k-1} \ldots \lambda) \, , \tag{15}$$

and m=n+1→ 0, k=0 →n+1, with one-to-one correspondence.

Eq. (14) demonstrates that the coefficients of the inhour polynomial given by Eq. (15) are determined by the parameters of the point reactor kinetics (Λ, β's, λ's, and ρ). Equivalently, Eq. (15) can be represented as [3]:

$$A_m = \Lambda a_m + b_m + \rho c_m \quad , \tag{16}$$

where, a_m, b_m, c_m are the universal abc-values for specific type of reactor fuel.

3. Mathematical Expansion Theory

3.1 Generalized Pascal's Triangles and their Relation to Multinomial Coefficients

In extending the binomial expansion, $(x+y)^n$, into a multinomial expansion, $(x+y+z+w+r+s+t+\cdots..)^n$ [5, 6], with the x_i – notation, one can presents the expansion as:

$$\left(\sum_{i=1}^{I} x_i\right)^n = \sum_{\substack{k \\ k' \\ k'' \\ \vdots \\ k^{(I-2)}}} \binom{n}{k}\binom{k}{k'}\binom{k'}{k''}\cdots\binom{k^{(I-3)}}{k^{(I-2)}} x_1^{n-k} x_2^{k-k'} x_3^{k'-k''} \ldots x_{I-1}^{k^{(I-3)}-k^{(I-2)}} x_I^{k^{(I-2)}} \quad , \tag{17}$$

where the variables $k, k', k'', k''', \ldots, k^{(I-2)}$ are subject to the nested constraints on the values $k \in \{0, \ldots, n\}$ and $k^{(i)} \in \{0, \ldots, k^{(i-1)}\}$ for $i = 1,, 2, \ldots, I-2$ (where $k^{(0)} \equiv k$). The detailed derivation is provided in [6].

Eq. (17) can equally be presented as:

$$\left(\sum_{i=1}^{I} x_i\right)^n = \sum_{\substack{k \\ k' \\ k'' \\ \vdots \\ k^{(I-2)}}} \underbrace{\begin{pmatrix} n \\ k \\ k' \\ k" \\ . \\ . \\ . \\ k^{(I-2)} \end{pmatrix}}_{\substack{k=0,\ldots,n \\ k'=0,..,k \\ k"=0,\ldots,k' \\ k^{(I-2)}=0,\ldots k^{(I-3)}}} x_1^{n-k} x_2^{k-k'} x_3^{k'-k''} \ldots x_{I-1}^{k^{(I-3)}-k^{(I-2)}} x_I^{k^{(I-2)}} \quad , \tag{18}$$

where the multinomial coefficients are given by:

$$\underbrace{\begin{pmatrix} n \\ k \\ k' \\ k" \\ . \\ . \\ . \\ k^{(I-2)} \end{pmatrix}}_{\substack{k=0,\ldots,n \\ k'=0,..,k \\ k"=0,\ldots,k' \\ k^{(I-2)}=0,\ldots k^{(I-3)}}} = \frac{n!}{(n-k)!(k-k')!\ldots.(k^{(I-3)}-k^{(I-2)})!k^{(I-2)}!} \tag{19}$$

These coefficients represent the elements of the Generalized Pascal's Triangles (GPTs). For convenience and in line with the present work, the author newly suggests the name "Meta Triangles" for this structure. The summand of Eq. (18) represents the multinomial probability distribution function.

3.2 Algorithmic Generation of Generalized Pascal's Triangles

Direct computation of the multinomial coefficients often becomes computationally prohibitive due to the rapid growth of factorials involved in their calculation. Therefore, building upon the mathematical framework established in Section 3.1, this section presents two algorithmic methods for generating Generalized Pascal's Triangles (GPTs), or Meta Triangles; the Embedded Pascal's Triangles (EPTs) method and the Staircase-Horizontal-Vertical (SHV) method.

3.2.1 The Embedded Pascal's Triangles (EPTs) method

Such elements of Pascal's triangles can be generated using the Embedded Pascal Triangles, EPTs, method [5, 6]. Table 3 presents the EPTs- method for generating generalized Pascal triangles (Meta Triangles). The method is simply developed through a three- step algorithm:

1. Vertical Setup: The classical Pascal's triangle is laid out vertically in patches n=0, 1, 2, 3,…etc (this forms the first row of each patch)
2. Horizontal Insertion: For the next order Meta Triangles, the row elements of the classical Pascal's triangle are laid out horizontally beneath the above cell in sub-cells corresponding for each patch (1st row of each expansion).
3. Multiplication and Result: Each element in the sub-cell is multiplied by the element of the above cell, resulting the new elements for the following order Meta Triangle (2nd row of each expansion).

The algorithm is repeated iteratively to produce higher order Meta Triangles.

An algorithm, has been developed to generate the elements of the Generalized Pascal's triangles, as well as, expressing the full expansion of the multinomials in its most general form presented by Eq. (17) [6], or equivalently by Eq. (18).

3.2.2 The Staircase Horizontal Vertical (SHV) method

The Staircase Horizontal Vertical (SHV) method is an alternative method for generating different orders of Meta Triangles (GPTs) [7]. The method follows a three-step algorithm:

1- Staircase Structure: A staircase is set up with different step-size depending on the order of Meta Triangle to be generated. The step size follows the elements of a selected column of the Waterloo matrix [19]. To generate order 2 Meta triangle (Pascal's triangle) one initiates staircase of unit step W1(n) = {1 1 1 1 1…}
2- Seed Stage: Lay out ones under each step
3- Feed Stage: Use these laid out elements as multipliers to the columns of the Waterloo matrix WI(n), I= {1 2 3 4 ….}, and n= {0 1 2 3 ….}.

This then generates order2 Meta triangle. To generate order 3 Meta triangle, one repeats the algorithm by structuring the next staircase with step-size corresponding to the elements of W2(n)={1 2 3 4 ….}, then use the previous generated elements as a seed under these steps followed by the feed stage where those elements are multiplied by the columns of the waterloo matrix WI(n), I={ 1 2 3 4 ….}, and n={0 1 2 3 ….}

Table 3

EPTs method for generating the elements of GPTs [19]

Expansion	Horizontally Embedded Pascal’s Triangles (1st row of each expansion) and Polynomial Coefficients (2nd row of each expansion)																																		
$(x+y)^0$	1																																		
$(x+y+z)^0$	1																																		
$(x+y+z+w)^0$	1																																		
$(x+y)^1$	1	1																																	
$(x+y+z)^1$	1	1	1																																
$(x+y+z+w)^1$	1	1	1	1																															
$(x+y)^2$	1	2			1																														
$(x+y+z)^2$	1	1	1		1	2		1																											
	1	2	2		1	2		1																											
$(x+y+z+w)^2$	1	1	1	1	1	1	1	1	2	1																									
	1	2	2	2	1	2	2	1	2	1																									
$(x+y)^3$	1	3			3						1																								
$(x+y+z)^3$	1	1	1		1	2		1			1	3		3			1																		
	1	3	3		3	6		3			1	3		3			1																		
$(x+y+z+w)^3$	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	3	3	1															
	1	3	3	3	3	6	6	3	6	3	1	3	3	3	6	3	1	3	3	1															
$(x+y)^4$	1	4			6						4										1														
$(x+y+z)^4$	1	1	1		1	2		1			1	3		3			1				1	4		6			4				1				
	1	4	4		6	12		6			4	12		12			4				1	4		6			4				1				
$(x+y+z+w)^4$	1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1	1	3	3	1	1	1	1	1	2	1	1	3	3	1	1	4	6	4	1
	1	4	4	4	6	12	12	6	12	6	4	12	12	12	24	12	4	12	12	4	1	4	4	6	12	6	4	12	12	4	1	4	6	4	1

Table 4 presents the Waterloo Matrix (shaded area). Furthermore, the elements of the Waterloo Matrix can be equally generated using the binomial coefficient $\binom{n+k}{k} = \frac{(n+k)!}{n!k!}$, or the Tripoli polynomials $T_m(n) = W_I(n)$, $m = I - 1, k = I - 1$.

For example, referring to Table 4 for I =4 (k =3) and n =4:

- The matrix element is read as **35** (Bolded)
- Using the binomial coefficient, it is $\binom{7}{3} = 35$
- Using the Tripoli polynomial, the same element is found as:

  $T_m(n) = T_3(4) = \frac{n^3}{6} + n^2 + \frac{11}{6}n + 1$ =35

Table 4
The Waterloo matrix [19]

I	1	2	3	4	5	6	....
n / k	0	1	2	3	4	5	.....
0	1	1	1	1	1	1	
1	1	2	3	4	5	6	
2	1	3	6	10	15	21	
3	1	4	10	20	35	56	
4	1	5	15	**35**	70	126	
.....							

Figure 2 presents the detailed method for generating different orders of Generalized Pascal Triangles (GPTs), also called Meta Triangles.

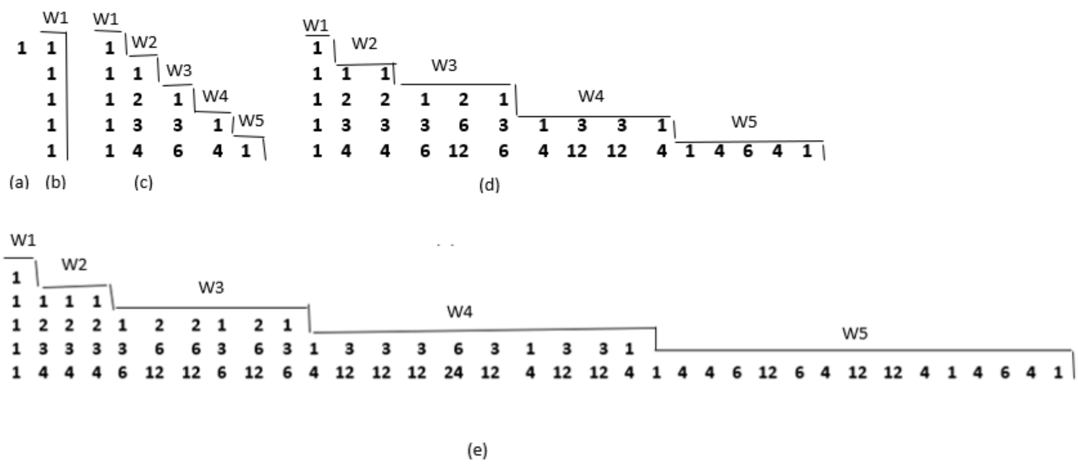

Fig.2. The Staircase Horizontal Vertical (SHV) method for generating different orders of Pascal's triangles for: (a) zeronomials $(0)^n$, (b) monomials $(x)^n$, (c) binomials $(x+y)^n$, (d) trinomials $(x+y+z)^n$ [7], and (e) quadrinomials $(x+y+z+w)^n$

Table 5 presents the calculation of the coefficients from a trinomial expansion using Eq. (19). These coefficients correspond to the elements of the fourth row (n=3, since n starts counting from 0) of the third-order Meta Triangle shown above in Figure 2(d). Here, M represents the number of coefficients in that row, which corresponds to $W_I(n) = W_3(3) = 10$.

Table 5
Generating elements of the nth row of the 3rd order Meta Triangle using the formula

n	3									
k	0	1		2			3			
k'	0	0	1	0	1	2	0	1	2	3
$\begin{pmatrix}3\\k\\k'\end{pmatrix}$	$\begin{pmatrix}3\\0\\0\end{pmatrix}$	$\begin{pmatrix}3\\1\\0\end{pmatrix}$	$\begin{pmatrix}3\\1\\1\end{pmatrix}$	$\begin{pmatrix}3\\2\\0\end{pmatrix}$	$\begin{pmatrix}3\\2\\1\end{pmatrix}$	$\begin{pmatrix}3\\2\\2\end{pmatrix}$	$\begin{pmatrix}3\\3\\0\end{pmatrix}$	$\begin{pmatrix}3\\3\\1\end{pmatrix}$	$\begin{pmatrix}3\\3\\2\end{pmatrix}$	$\begin{pmatrix}3\\3\\3\end{pmatrix}$
Coefficients M=W_3(3) =10	1	3	3	3	6	3	1	3	3	1

$$\begin{pmatrix}3\\k\\k'\end{pmatrix} = \frac{3!}{(3-k)!\,(k-k')!\,k'!}$$

The algorithms of the EPTs, and the SHV methods are simple methods designed to be suitable to achieve sophisticated computational results, a concept which has been addressed by Sergei in his book "From Counting to Computing" [20].

4. Nuclear Reactor Kinetics: The Inhour Polynomial and the Coefficient Based Model, CBM

This section applies the developed Guelph Expansion to Nuclear Reactor Kinetics. One utilizes the expansion to deduce the coefficients of a new Inhour polynomial, thereby formalizing the Coefficient Based Model (CBM) representing the point reactor kinetics.

4.1 The Universal abc-Values and the A_m Coefficients of the Inhour Polynomial

As demonstrated in Subsection 2.1.6, the coefficients of the Inhour polynomial can be calculated directly from the reactor system parameters. Table 6 presents the derived universal abc-values for U-235 fueled reactors, with precursors decay constants $\lambda 1$=0.0124, $\lambda 2$=0.0305, $\lambda 3$= 0.111, $\lambda 4$=0.301, $\lambda 5$=1.14, and $\lambda 6$=3.01. The group delayed neutron fractions are $\beta 1$=0.000215, $\beta 2$=0.001424, $\beta 3$=0.001274, $\beta 4$=0.002568, $\beta 5$=0.000748, $\beta 6$=0.000273, with total delayed neutron fraction as β=0.0065 [21]:

Table 6

The universal abc-values for U235-fueled reactors (Eq. (15), Eq. (16)) [8]

a0	0	0
a1	$\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6$	4.3359e-05
a2	$\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 + \lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_6 + \lambda_1 \lambda_2 \lambda_3 \lambda_5 \lambda_6 + \lambda_1 \lambda_2 \lambda_4 \lambda_5 \lambda_6 + \lambda_1 \lambda_3 \lambda_4 \lambda_5 \lambda_6 + \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6$	0.0055
a3	$\lambda_1 \lambda_2 \lambda_3 \lambda_4 + \lambda_1 \lambda_2 \lambda_3 \lambda_5 + \lambda_1 \lambda_2 \lambda_3 \lambda_6 + \lambda_1 \lambda_2 \lambda_4 \lambda_5 + \lambda_1 \lambda_2 \lambda_4 \lambda_6 + \lambda_1 \lambda_2 \lambda_5 \lambda_6 + \lambda_1 \lambda_3 \lambda_4 \lambda_5 + \lambda_1 \lambda_3 \lambda_4 \lambda_6 + \lambda_1 \lambda_3 \lambda_5 \lambda_6 + \lambda_1 \lambda_4 \lambda_5 \lambda_6 + \lambda_2 \lambda_3 \lambda_4 \lambda_5 + \lambda_2 \lambda_3 \lambda_4 \lambda_6 + \lambda_2 \lambda_3 \lambda_5 \lambda_6 + \lambda_2 \lambda_4 \lambda_5 \lambda_6 + \lambda_3 \lambda_4 \lambda_5 \lambda_6$	0.1832
a4	$\lambda_1 \lambda_2 \lambda_3 + \lambda_1 \lambda_2 \lambda_4 + \lambda_1 \lambda_2 \lambda_5 + \lambda_1 \lambda_2 \lambda_6 + \lambda_1 \lambda_3 \lambda_4 + \lambda_1 \lambda_3 \lambda_5 + \lambda_1 \lambda_3 \lambda_6 + \lambda_1 \lambda_4 \lambda_5 + \lambda_1 \lambda_4 \lambda_6 + \lambda_1 \lambda_5 \lambda_6 + \lambda_2 \lambda_3 \lambda_4 + \lambda_2 \lambda_3 \lambda_5 + \lambda_2 \lambda_3 \lambda_6 + \lambda_2 \lambda_4 \lambda_5 + \lambda_2 \lambda_4 \lambda_6 + \lambda_2 \lambda_5 \lambda_6 + \lambda_3 \lambda_4 \lambda_5 + \lambda_3 \lambda_4 \lambda_6 + \lambda_3 \lambda_5 \lambda_6 + \lambda_4 \lambda_5 \lambda_6$	1.7761
a5	$\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_1 \lambda_4 + \lambda_1 \lambda_5 + \lambda_1 \lambda_6 + \lambda_2 \lambda_3 + \lambda_2 \lambda_4 + \lambda_2 \lambda_5 + \lambda_2 \lambda_6 + \lambda_3 \lambda_4 + \lambda_3 \lambda_5 + \lambda_3 \lambda_6 + \lambda_4 \lambda_5 + \lambda_4 \lambda_6 + \lambda_5 \lambda_6$	5.3707
a6	$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6$	4.6049
a7	1	1

b0	0	0
b1	$\beta_1(\lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6) + \beta_2(\lambda_1 \lambda_3 \lambda_4 \lambda_5 \lambda_6) + \beta_3(\lambda_1 \lambda_2 \lambda_4 \lambda_5 \lambda_6) + \beta_4(\lambda_1 \lambda_2 \lambda_3 \lambda_5 \lambda_6) + \beta_5(\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_6) + \beta_6(\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5)$	3.6761e-06
b2	$\beta_1(\lambda_2 \lambda_3 \lambda_4 \lambda_5 + \lambda_2 \lambda_3 \lambda_4 \lambda_6 + \lambda_2 \lambda_3 \lambda_5 \lambda_6 + \lambda_2 \lambda_4 \lambda_5 \lambda_6 + \lambda_3 \lambda_4 \lambda_5 \lambda_6) + \beta_2(\lambda_1 \lambda_3 \lambda_4 \lambda_5 + \lambda_1 \lambda_3 \lambda_4 \lambda_6 + \lambda_1 \lambda_3 \lambda_5 \lambda_6 + \lambda_1 \lambda_4 \lambda_5 \lambda_6 + \lambda_3 \lambda_4 \lambda_5 \lambda_6) + \beta_3(\lambda_1 \lambda_2 \lambda_4 \lambda_5 + \lambda_1 \lambda_2 \lambda_4 \lambda_6 + \lambda_1 \lambda_2 \lambda_5 \lambda_6 + \lambda_1 \lambda_4 \lambda_5 \lambda_6 + \lambda_2 \lambda_4 \lambda_5 \lambda_6) + \beta_4(\lambda_1 \lambda_2 \lambda_3 \lambda_5 + \lambda_1 \lambda_2 \lambda_3 \lambda_6 + \lambda_1 \lambda_2 \lambda_5 \lambda_6 + \lambda_1 \lambda_3 \lambda_5 \lambda_6 + \lambda_2 \lambda_3 \lambda_5 \lambda_6) + \beta_5(\lambda_1 \lambda_2 \lambda_3 \lambda_4 + \lambda_1 \lambda_2 \lambda_3 \lambda_6 + \lambda_1 \lambda_2 \lambda_4 \lambda_6 + \lambda_1 \lambda_3 \lambda_4 \lambda_6 + \lambda_2 \lambda_3 \lambda_4 \lambda_6) + \beta_6(\lambda_1 \lambda_2 \lambda_3 \lambda_4 + \lambda_1 \lambda_2 \lambda_3 \lambda_5 + \lambda_1 \lambda_2 \lambda_4 \lambda_5 + \lambda_1 \lambda_3 \lambda_4 \lambda_5 + \lambda_2 \lambda_3 \lambda_4 \lambda_5)$	3.3403e-04
b3	$\beta_1(\lambda_2 \lambda_3 \lambda_4 + \lambda_2 \lambda_3 \lambda_5 + \lambda_2 \lambda_3 \lambda_6 + \lambda_2 \lambda_4 \lambda_5 + \lambda_2 \lambda_4 \lambda_6 + \lambda_2 \lambda_5 \lambda_6 + \lambda_3 \lambda_4 \lambda_5 + \lambda_3 \lambda_4 \lambda_6 + \lambda_3 \lambda_5 \lambda_6 + \lambda_4 \lambda_5 \lambda_6) + \beta_2(\lambda_1 \lambda_3 \lambda_4 + \lambda_1 \lambda_3 \lambda_5 + \lambda_1 \lambda_3 \lambda_6 + \lambda_1 \lambda_4 \lambda_5 + \lambda_1 \lambda_4 \lambda_6 + \lambda_1 \lambda_5 \lambda_6 + \lambda_3 \lambda_4 \lambda_5 + \lambda_3 \lambda_4 \lambda_6 + \lambda_3 \lambda_5 \lambda_6 + \lambda_4 \lambda_5 \lambda_6) + \beta_3(\lambda_1 \lambda_2 \lambda_4 + \lambda_1 \lambda_2 \lambda_5 + \lambda_1 \lambda_2 \lambda_6 + \lambda_1 \lambda_4 \lambda_5 + \lambda_1 \lambda_4 \lambda_6 + \lambda_1 \lambda_5 \lambda_6 + \lambda_2 \lambda_4 \lambda_5 + \lambda_2 \lambda_4 \lambda_6 + \lambda_2 \lambda_5 \lambda_6 + \lambda_4 \lambda_5 \lambda_6) + \beta_4(\lambda_1 \lambda_2 \lambda_3 + \lambda_1 \lambda_2 \lambda_5 + \lambda_1 \lambda_2 \lambda_6 + \lambda_1 \lambda_3 \lambda_5 + \lambda_1 \lambda_3 \lambda_6 + \lambda_1 \lambda_5 \lambda_6 + \lambda_2 \lambda_3 \lambda_5 + \lambda_2 \lambda_3 \lambda_6 + \lambda_2 \lambda_5 \lambda_6 + \lambda_3 \lambda_5 \lambda_6) + \beta_5(\lambda_1 \lambda_2 \lambda_3 + \lambda_1 \lambda_2 \lambda_4 + \lambda_1 \lambda_2 \lambda_6 + \lambda_1 \lambda_3 \lambda_4 + \lambda_1 \lambda_3 \lambda_6 + \lambda_1 \lambda_4 \lambda_6 + \lambda_2 \lambda_3 \lambda_4 + \lambda_2 \lambda_3 \lambda_6 + \lambda_2 \lambda_4 \lambda_6 + \lambda_3 \lambda_4 \lambda_6) + \beta_6(\lambda_1 \lambda_2 \lambda_3 + \lambda_1 \lambda_2 \lambda_4 + \lambda_1 \lambda_2 \lambda_5 + \lambda_1 \lambda_3 \lambda_4 + \lambda_1 \lambda_3 \lambda_5 + \lambda_1 \lambda_4 \lambda_5 + \lambda_2 \lambda_3 \lambda_4 + \lambda_2 \lambda_3 \lambda_5 + \lambda_2 \lambda_4 \lambda_5 + \lambda_3 \lambda_4 \lambda_5)$	0.0058
b4	$\beta_1(\lambda_2 \lambda_3 + \lambda_2 \lambda_4 + \lambda_2 \lambda_5 + \lambda_2 \lambda_6 + \lambda_3 \lambda_4 + \lambda_3 \lambda_5 + \lambda_3 \lambda_6 + \lambda_4 \lambda_5 + \lambda_4 \lambda_6 + \lambda_5 \lambda_6) + \beta_2(\lambda_1 \lambda_3 + \lambda_1 \lambda_4 + \lambda_1 \lambda_5 + \lambda_1 \lambda_6 + \lambda_3 \lambda_4 + \lambda_3 \lambda_5 + \lambda_3 \lambda_6 + \lambda_4 \lambda_5 + \lambda_4 \lambda_6 + \lambda_5 \lambda_6) + \beta_3(\lambda_1 \lambda_2 + \lambda_1 \lambda_4 + \lambda_1 \lambda_5 + \lambda_1 \lambda_6 + \lambda_2 \lambda_4 + \lambda_2 \lambda_5 + \lambda_2 \lambda_6 + \lambda_4 \lambda_5 + \lambda_4 \lambda_6 + \lambda_5 \lambda_6) + \beta_4(\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_1 \lambda_5 + \lambda_1 \lambda_6 + \lambda_2 \lambda_3 + \lambda_2 \lambda_5 + \lambda_2 \lambda_6 + \lambda_3 \lambda_5 + \lambda_3 \lambda_6 + \lambda_5 \lambda_6) + \beta_5(\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_1 \lambda_4 + \lambda_1 \lambda_6 + \lambda_2 \lambda_3 + \lambda_2 \lambda_4 + \lambda_2 \lambda_6 + \lambda_3 \lambda_4 + \lambda_3 \lambda_6 + \lambda_4 \lambda_6) + \beta_6(\lambda_1 \lambda_2 + \lambda_1 \lambda_3 + \lambda_1 \lambda_4 + \lambda_1 \lambda_5 + \lambda_2 \lambda_3 + \lambda_2 \lambda_4 + \lambda_2 \lambda_5 + \lambda_3 \lambda_4 + \lambda_3 \lambda_5 + \lambda_4 \lambda_5)$	0.0265
b5	$\beta_1(\lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6) + \beta_2(\lambda_1 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6) + \beta_3(\lambda_1 + \lambda_2 + \lambda_4 + \lambda_5 + \lambda_6) + \beta_4(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5 + \lambda_6) + \beta_5(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_6) + \beta_6(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5)$	0.0273
b6	$\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6$	0.0065
b7	0	0

c0	c1	c2	c3	c4	c5	c6	c7
-a1	-a2	-a3	-a4	-a5	-a6	-a7	0
-.000043359	-0.0055	-0.1832	-1.7761	-5.3707	-4.6049	-1	0

The coefficients Am were calculated using Eq. (14) (subsection 2.1.7) with the parameters $\Lambda = 0.0001$ and ρ =0.00065, and the universal abc-values presented in Table 6. The resulting coefficients *Am* are presented in Table 7.

Table 7
The A_m coefficients

A0	A1	A2	A3	A4	A5	A6	A7
-2.8183e-8	1.0544e-7	0.0002155	0.0047	0.0232	0.0248	0.0063	0.0001

The roots of Eq. (14) are presented in Table 8.

Table 8
The solution of the Inhour Polynomial

Value	1	2	3	4	5	6	7
ω	0.0101	-0.0138	-0.0623	-0.1882	-1.0127	-2.8804	-58.9575

Figure 3 provides a qualitative graphical representation, while Figure 4 offers a quantitative representation of the Inhour polynomial for six groups of delayed neutrons.

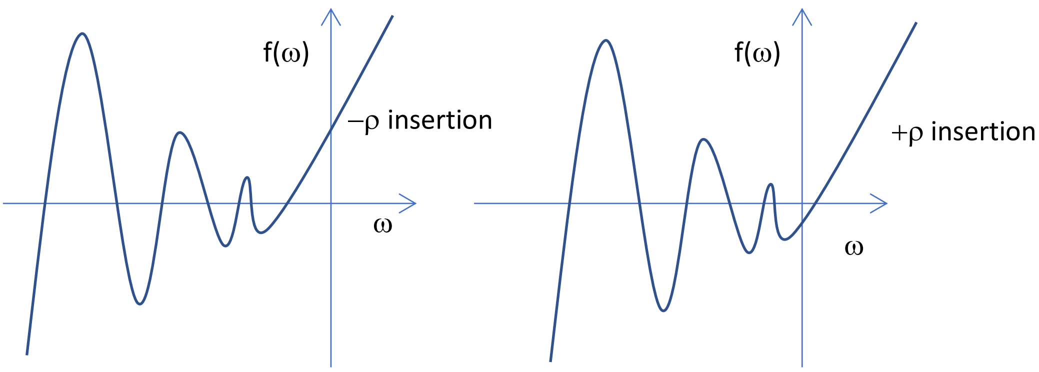

Fig.3. Qualitative behavior for the inhour polynomial [3]

4.2 The CBM- Reactor Equation

Eq. (14) from Section 2.1.7 represents the characteristic equation of point reactor kinetics. Consequently, the corresponding nuclear reactor model can be directly derived in the following compact form:

$$\sum_{m=0}^{n+1} A_m \frac{d^m P}{dt^m} = 0 \tag{20}$$

In this model, P represents the nuclear reactor power and t is the time. The coefficients, A_m, are the Inhour polynomial coefficients; they are defined by Eq. (15), detailed in Table 6, and presented as numerical values in Table 7 for a positive step reactivity insertion. This structure is referred to as the Coefficient-Based Model (CBM), which is a single m^{th}- order differential equation. The CBM replaces the m-coupled first-order differential equations that represent the standard point reactor kinetics model (for n groups of delayed neutrons) [8].

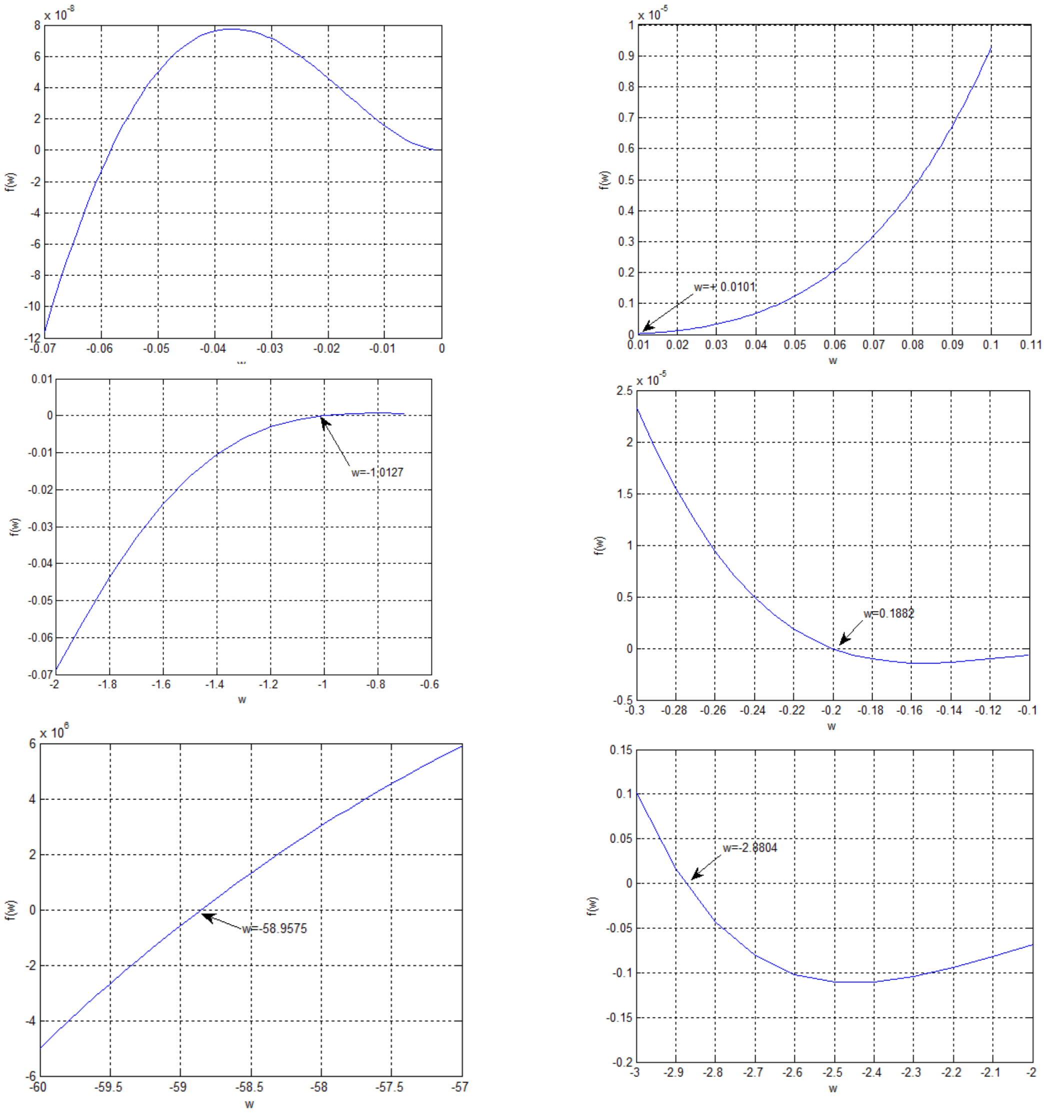

Fig.4. Quantitative behavior for the inhour polynomial equation:
$f(\omega) = .0001\omega^7 + .0063\omega^6 + .0248\omega^5 + .0232\omega^4 + .0047\omega^3 + .0002\omega^2$. ($\rho = .1\$, \Lambda = .0001\ sec$)

4.3 Calculation of the Inhour Polynomial Coefficients (A_m): Two Methods

The Inhour polynomial coefficients (A_m) can be determined using two equivalent methods. First, they can be calculated directly from the fundamental reactor kinetics parameters, as formally presented in Table 6. Second, these same coefficients can be calculated by applying the Guelph Expansion formula, which relates the roots of the function to its coefficients, a technique detailed in subsection 2.1.4 above.

To calculate the coefficients using the roots of the inhour polynomial, we normalize the Am coefficients to make the highest degree coefficient equal one by Dividing Eq. (15) by Λ which then normalizes to:

$$\widetilde{A_m} = \sum_{\binom{n}{k}} (\lambda \ldots^{k} \ldots \lambda) + \frac{1}{\Lambda} \sum_{i=1}^{n} \beta_i \sum_{\binom{n-1}{k-1}} (\lambda \ldots^{k-1} \ldots \lambda)_{\neq i} - \frac{\rho}{\Lambda} \sum_{\binom{n}{k-1}} (\lambda \ldots^{k-1} \ldots \lambda) \quad (21)$$

or one can do it equivalently:

$$\widetilde{A_m} = a_m + \frac{b_m}{\Lambda} + \frac{\rho}{\Lambda} c_m \quad (22)$$

Referring to Eq. (9), these normalized coefficients can also be calculated directly from the roots of the polynomial as:

$$\widetilde{A_{M-k}} = \sum_{\binom{M}{k}} \alpha \ldots^{k} \ldots \alpha \qquad , k = 0,1,2,3, \ldots M \tag{23}$$

Table 9 presents the algebraic formulas for calculating the coefficients using the roots.

Table 9

Explicit combinations of roots formulae for each coefficient for M=n+1=7

M	k	# of terms $\binom{M}{k}$	$A_{M-k} = \sum_{\binom{M}{k}} \alpha \ldots^{k} \ldots \alpha$
7	0	1	–
7	1	7	Σα's = α1+ α2+ α3+ α4+ α5+ α6+ α7
7	2	21	Σαα's= α1*α2+ α1 *α3+ α1* α4+ α1 *α5+α1 *α6+ α1 *α7+α2 *α3+α2*α4+α2* α5+α2* α6+ α2 *α7+α3* α4+α3* α5+α3 *α6+ α3* α7+α4* α5+ α4* α6 + α4* α7+α5* α6 + α5* α7+ α6* α7
7	3	35	Σααα's = α1*α2*α3+ α1 *α2*α4+ α1* α2*α5+ α1* α2*α6+ α1* α2*α7+ α1*α3* α4+ α1 *α3*α5+ α1* α3*α6+ α1* α3*α7+ α1* α4*α5+ α1*α4*α6+ α1 * α4*α7+ α1* α5*α6+ α1* α5*α7+ α1* α6*α7+α2 *α3*α4+ α2 *α3*α5+ α2 *α3*α6+ α2*α3*α7+α2*α4*α5+ α2*α4*α6+ α2*α4*α7+α2* α5*α6 + α2* α5*α7+ α2* α6*α7+α3 *α4*α5+ α3 *α4*α6+ α3 *α4*α7+ α3*α5 *α6+α3*α5*α7+ α3*α6*α7+α4*α5*α6+α4* α5*α7+α4*α6*α7+ α5*α6 *α7
7	4	35	Σαααα's= α1*α2*α3*α4+α1*α2*α3*α5+α1*α2*α3*α6+α1*α2*α3*α7+ α1*α2*α 4*α5+α1*α2*α4*α6+ α1*α2*α4*α7+α1*α2*α5*α6+ α1*α2*α5*α7+α1*α2*α6*α7+α1*α3*α 4*α5+α1*α3*α4*α6+ α1*α3*α4*α7+α1*α3*α5*α6+α1*α3*α5*α7+α1*α3*α6*α7+α1*α4*α5 *α6+α1*α4*α5*α7+ α1*α4*α6*α7+ α1*α5*α6*α7+α2*α3*α4*α5+α2*α3*α4*α6+α2*α3*α 4*α7+α2*α3* α5*α6+α2*α3*α5*α7+ α2*α3*α6*α7+ α2*α4*α5*α6+α 2*α4*α5*α7+ α2*α4*α6*α7+ α2*α5*α6*α7+ α3*α4*α5*α6+α3*α4*α5 *α7+ α3*α4*α6*α7+ α3*α5*α6*α7+ α4*α5*α6*α7
7	5	21	Σααααα's= α2*α3*α4*α5*α6+α2*α3*α4*α5*α7+α2*α3*α4*α6*α7+α2*α3*α5*α6 *α7+α2*α4*α5*α6*α7 +α3*α4*α5*α6*α7+ α1*α3*α4* α5*α6+α1*α3 *α4*α5*α7+α1*α3*α4*α6*α7 +α1*α3*α5*α6*α7 +α1*α4*α5*α6*α7 +α1*α2*α4*α5*α6 +α1*α2*α4*α5*α7+α1*α2*α4*α6*α7 +α1*α2*α5* α6*α7 + α1*α2*α3*α5*α6+α1*α2*α3*α5*α7 +α1*α2*α3*α6*α7 + α1* α2*α3*α4*α6 +α1*α2*α3*α4*α7+ α1*α2*α3*α4*α5
7	6	7	Σαααααα's= α1*α2*α3*α4*α5*α6+ α1*α2*α3*α4*α5*α7+ α1*α2*α3*α4* α6*α7+ α1*α2*α3* α5*α6*α7+ α1*α2* α4*α5*α6*α7+ α1* α3*α4*α5*α6*α7+ α2*α3*α4*α5*α6*α7
7	7	1	Σααααααα's= α1*α2*α3*α4*α5*α6*α7

Tables 10 and 11 present the calculated Coefficients using the parameters of the nuclear reactor kinetics, and using the roots (inverse of reactor time constants, τ) , respectively. The Tables reveal the equivalence of the calculated coefficients using the two methods.

Furthermore, examining Eq. (22) and Eq. (23), allows one to deduce a relationship given as:

$$\widetilde{A_{M-k}} = a_m + \frac{b_m}{\Lambda} + \frac{\rho}{\Lambda} c_m = \sum_{\binom{M}{k}} \alpha \ldots^{k} \ldots \alpha \qquad , k = 0,1,2,3, \ldots M \;, \quad m = M, M-1, \ldots 0 \tag{24}$$

So, knowing the system time constants, τ, one can calculate explicitly the system reactivity, ρ, for a certain reactor type (Λ), and a certain type of reactor fuel (universal abc-values). Table 12 demonstrate the calculation of the reactivity, ρ, using Eq. (24).

Table 10 The normalized coefficients, $\widetilde{A_m}$ (using nuclear reactor parameters)

$\widetilde{A_0}$	$\widetilde{A_1}$	$\widetilde{A_2}$	$\widetilde{A_3}$	$\widetilde{A_4}$	$\widetilde{A_5}$	$\widetilde{A_6}$	$\widetilde{A_7}$
−2.8183e − 4	0.0011	2.1550	46.6385	231.8665	248.4389	63.1049	1

Table 11 The Inhour coefficients (using the roots)

A0	A1	A2	A3	A4	A5	A6	A7
−2.8114e − 4	0.0010	2.1558	46.6386	231.8666	248.4389	63.1043	1

Table 12 Reactivity calculations using universal abc values and the coefficients: $\rho=(A_m * \Lambda - a_m * \Lambda - b_m)/c_m$

m	a^	b^	c^	Coefficients (Table 7)	Coefficients (Table 6)	ρ (roots)	ρ (parameters)
7	1	0	0	1	1	-	-
6	4.6049	.0065	-1	63.1043	63.1049	.00065	.00065
5	5.3707	.0273	-4.6049	248.4389	248.4389	.00065	.00065
4	1.7761	0.0265	-5.3707	231.8666	231.8665	.00065	.00065
3	0.1832	0.0058	-1.7761	46.6386	46.6385	.00065	.00065
2	0.0055	3.3403e-4	-0.1832	2.1558	2.155	.00064956	.00065
1	4.3359e-5	3.6761e-6	-0.0055	.001	.0011	.00065099	.00064917
0	0	0	-4.3359e-5	-2.8114e-4	-2.8183e-4	.0006484	.00064999

a^,b^,c^ data from Table 2

4.4 Analytical CBM versus Numerical Classical Reactor Model (Stability Study)

This subsection presents the models for a single group of delayed neutrons. We first revisit the foundations of neutron kinetics by presenting the classical point reactor model (defined by two coupled first-order equations) and its corresponding transcendental Inhour equation. Critically, these classical concepts are then reintroduced and formalized within the context of the Inhour polynomial and its associated single second-order differential equation, the CBM equation.

The classical model is given as:

$$\frac{\mathrm{dP}}{dt} = \frac{\rho-\beta}{\Lambda}\,\mathrm{P} + \lambda c \,, \qquad \mathrm{P}(0) = \mathrm{P}_0 \qquad , \tag{25}$$

$$\frac{\mathrm{d}c}{dt} = \frac{\beta}{\Lambda}\,\mathrm{P} - \lambda c, \qquad c(0) = \frac{\beta}{\Lambda\lambda}\,\mathrm{P}_0 \qquad , \tag{26}$$

The system state is defined by the following variables: P, c, t, which represent the power, the precursor concentration, and time, respectively.

The corresponding Inhour equation for the case of one group of delayed neutrons is given by:

$$\rho = \Lambda\omega + \frac{\beta\omega}{\omega+\lambda} \tag{27}$$

The derived Inhour polynomial is then presented as:

$$F(\omega) = \sum_{m=0}^{2} A_m \omega^m = 0 \tag{28}$$

The CBM is:

$$\sum_{m=0}^{2} A_m \frac{d^m \mathrm{P}}{dt^m} = 0 \quad . \tag{29}$$

Table 13 presents the coefficient values of the CBM.

Table 13
Coefficient Values of the CBM

m	a_m	b_m	c_m	$A_m = \Lambda a_m + b_m + c_m \rho$
2	1	0	0	Λ
1	λ	β	-1	$\Lambda\lambda + \beta - \rho$
0	0	0	$-\lambda$	$-\lambda\rho$

The analytical solution of Eq. (29) is given by:

$$\mathrm{P}(t) = C_1 e^{\omega_1 t} + C_2 e^{\omega_2 t} = \frac{\mathrm{P}_0}{(\omega_1-\omega_2)}\left(\left(\frac{\rho}{\Lambda} - \omega_2\right) e^{\omega_1 t} + \left(\omega_1 - \frac{\rho}{\Lambda}\right) e^{\omega_2 t}\right) \tag{30}$$

Figure 5 presents the reactor power response for a positive reactivity insertion of 0.00065 by comparing the analytical CBM solution with the numerical solution of the classical model, which was obtained using the Euler algorithm. Good agreement is observed between both models when the time step is set to 0.01 seconds.

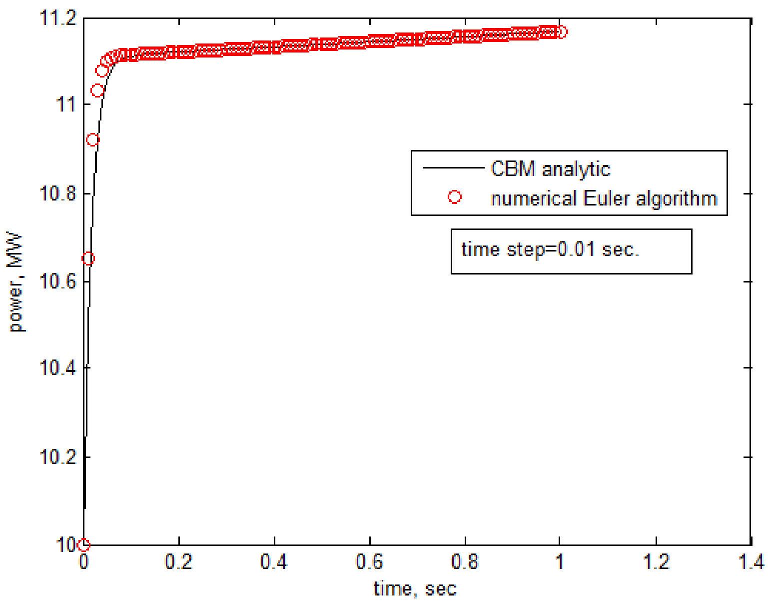

Fig.5. Power response for the two model with a time step of 0.01 seconds

In contrast, Figure 6 and Figure 7 display the CBM and classical model solutions for an increased time step of 0.1 seconds. These figures clearly demonstrate that as the solution time step is increased, the CBM solution remains stable, while the classical model fails due to numerical instability. This makes the CBM model faster contributing positively to safety analysis of reactor transients in nuclear reactors.

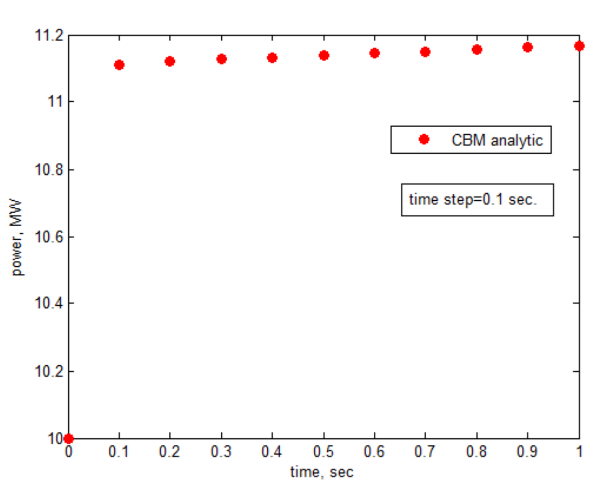

Fig.6. Power response for the CBM with a time step of 0.1 seconds

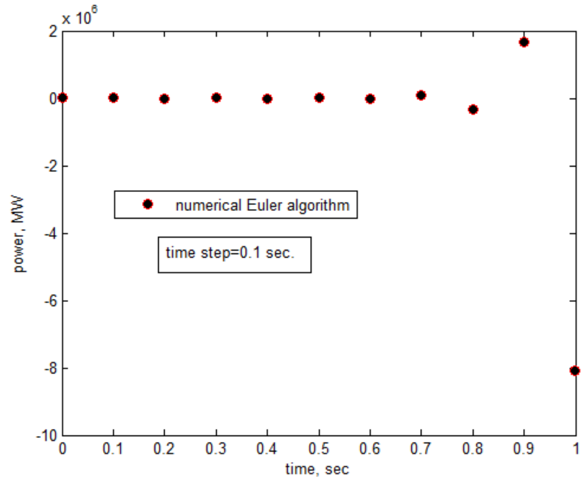

Fig.7. Power response for the classical model with a time step of 0.1 seconds (Unstable)

Researchers study the stiffness problem in solving numerically the point reactor kinetics alone or coupled to thermohydraulic. A generalized analytical solution to the point reactor kinetics is provided by Oyama *et al*. [22].

5. Some Other Applications

5.1 Coin and Dice Tossing

A typical application of Pascal's triangle is the calculation of coin-tossing probabilities. These probabilities can be determined either from the elements of the classical Pascal's triangle (which represents the 2nd order of Meta Triangles) or directly from the binomial probability distribution function, Eq. (6), and Eq. (7). This is done by replacing the variables x and y with H (Head) and T (Tail), respectively, where n represents the number of tosses [23].

If a 4-sided die is used, one can apply the quadrinomial expansion$(x + y + z + w)^n$ and the associated 4th order Meta Triangle to analyze the corresponding probabilities. An example for n=4 is presented in Table 14, showing the generated 4th order Meta Triangle elements along with their associated probabilities. This methodology can be easily extended to standard dice tossing by using the six-variable multinomial expansion $(f1 + f2 + f3 + f4 + f5 + f6)^n$. Here, *f1* to *f6* represent the faces of the die (from one to six), and n represents the number of tossing.

5.2 DNA Sequencing and Tagging of Genomes

If one replaces the four variables with the four DNA nucleotides (A, T, C, and G) and considers a strand length of n, then the probabilities presented in Table 14 represent the associated probabilities of the 4-DNA strand sequencing [6]. For instance, the probability for the sequence {A A A A} is 0.00390625 (read from the 1st row of the table), and the probability for the sequence {A T C G} is 0.09375 (read from the 15th row).

In another study, the Ratemi [13] explored the tagging of the complete DNA sequence of the Hepatitis B Virus (HBV) genome using the novel tagger $T(n, k, k', k'')$.

Table 14
Detailed tabular coding of the expansion of $(x+y+z+w)^4$ and the probabilities corresponding to the coefficients [6]

n	k	k'	k''	T_{nk}	$T_{kk'}$	$T_{k'k''}$	Coefficients $T_i = T_{nk}\, T_{kk'}\, T_{k'k''}$	Lexicographic monomials: $x^{n-k}\, y^{k-k'}\, z^{k'-k''}\, w^{k''}$	Probabilities $P_i = T_i/\Sigma_i T_i = T_i/4^n$
4	0	0	0	1	1	1	1	x^4	0.00390625
	1	0	0	4	1	1	4	$x^3\, y$	0.015625
		1	0	4	1	1	4	$x^3\, z$	0.015625
			1	4	1	1	4	$x^3\, w$	0.015625
	2	0	0	6	1	1	6	$x^2\, y^2$	0.0234375
		1	0	6	2	1	12	$x^2\, y\, z$	0.046875
			1	6	2	1	12	$x^2\, y\, w$	0.046875
		2	0	6	1	1	6	$x^2\, z^2$	0.0234375
			1	6	1	2	12	$x^2\, z\, w$	0.046875
			2	6	1	1	6	$x^2\, w^2$	0.0234375
	3	0	0	4	1	1	4	$x\, y^3$	0.015625
		1	0	4	3	1	12	$x\, y^2\, z$	0.046875
			1	4	3	1	12	$x\, y^2\, w$	0.046875
		2	0	4	3	1	12	$x\, y\, z^2$	0.046875
			1	4	3	2	24	$x\, y\, z\, w$	0.09375
			2	4	3	1	12	$x\, y\, w^2$	0.046875
		3	0	4	1	1	4	$x\, z^3$	0.015625
			1	4	1	3	12	$x\, z^2\, w$	0.046875
			2	4	1	3	12	$x\, zw^2$	0.046875
			3	4	1	1	4	xw^3	0.015625
	4	0	0	1	1	1	1	y^4	0.00390625
		1	0	1	4	1	4	y^3z	0.015625
			1	1	4	1	4	y^3w	0.015625
		2	0	1	6	1	6	$y^2\, z^2$	0.0234375
			1	1	6	2	12	$y^2\, z\, w$	0.046875
			2	1	6	1	6	$y^2\, w^2$	0.0234375
		3	0	1	4	1	4	$y\, z^3$	0.015625
			1	1	4	3	12	$y\, z^2\, w$	0.046875
			2	1	4	3	12	$y\, z\, w^2$	0.046875
			3	1	4	1	4	$y\, w^3$	0.015625
		4	0	1	1	1	1	z^4	0.00390625
			1	1	1	4	4	$z^3\, w$	0.015625
			2	1	1	6	6	$z^2\, w^2$	0.0234375
			3	1	1	4	4	$z\, w^3$	0.015625
			4	1	1	1	1	w^4	0.00390625

The convergence of natural language processing (NLP) and genomics has given rise to a new class of transformer-based models genome large language models (Gene-LLMs) capable of interpreting the language of life at an unprecedented scale and resolution [24]. Multinomial expansion and the provided EPTs, and SHV algorithms can contribute in these types of studies related to genomics and DNA sequencing.

5.3 Identifying Minimal Cut Sets

The author also employed the EPTs method for identifying the minimal cut sets (a total of 82) for a fault tree analysis of a system experiencing a vacuum loss fault in two identical loops [14].

The system was modeled with a multinomial expansion having 18 variables, which represent the basic fault events. Since a combination of two events (one from each loop) is sufficient to cause a system halt, the multinomial is raised to the power of two. That is, one must expand $(x_1 + x_2 + x_3 + \cdots + x_{18})^2$.

5.4 Bezier's Curves and Computer Graphics

Referencing Eq. (11) (Section 2.1.6), one can generate Bezier's curves by incorporating the control points into the binomial expansion formula. This approach represents a generalization of Bezier's curves. By selecting the degree n, one can generate different curves:

1. Linear Bezier Curve (Two Control Points)

For a basic two control points case, P1 = (x1, y1) and P2= (x2, t2), using n=1, k=0, 1:

$$\mathrm{B(t)} = \sum_{\mathrm{k}=0}^{1} \binom{1}{k} (1-t)^{1-\mathrm{k}}\, t^k\, P_{k+1} = \binom{1}{0}\,(1-t)P_1 + \binom{1}{1} t\, P_2 = (1-t)P_1 + t\, P_2 \tag{31}$$

Selecting the parameter $t \in [0,1]$ results a straight-line segment.

2. Quadratic Bezier Curve (Three Control Points)

For three control points (P1, P2, and P3), using n=2 and k=0, 1, 2:

$$\mathrm{B(t)} = \sum_{\mathrm{k}=0}^{2} \binom{2}{k} (1-t)^{2-\mathrm{k}}\, t^k\, P_{k+1} = \binom{2}{0}\,(1-t)^2 P_1 + \binom{2}{1}(1-t)t\, P_2 + \binom{2}{2} t^2 P_3 = (1-t)^2 P_1 + 2(1-t)t\, P_2 + t^2 P_3 \tag{32}$$

This relationship generates a quadratic curve.

In a similar manner, for 4 control points, one gets the following Bezier's curve:

$$\mathrm{B(t)} = (1-t)^3 P_1 + 3(1-t)^2 t\, P_2 + 3(1-t)t^2 P_3 + t^3\, P_4 \tag{33}$$

One notice that the number of points equals to the number of the expanded terms.

A quadratic Bezier's curve (Eq. 32) is generated using three control points. For example, selecting the points P1 = (1, 2), P2= (4, 6), and P3 = (7, 3), the curve is traced by letting the parameter t varies continuously from 0 to 1. The selection of t can be continuous, step-wise, or randomly selected within the [0, 1] interval (see Figure 8).

Alternatively, the same curve can be generated equivalently using De Casteljau's algorithm [25]. This recursive geometric method involves:

1. Generating two segments connecting the control points (P1-P2 and P2-P3).
2. Selecting a value of t ($0 \leq t \leq 1$) and locating that fractional point on each respective segment.
3. Connecting these two new intermediate points and locating a point on this connecting segment using the same value of t. This point lies on the generated Bezier's curve.

Repeating this process for various values of t traces the full curve. The curvature of the resulting curve is controlled by manipulating point, P2 in this quadratic case.

Eq. (33), which utilizes four control points, is used to generate a cubic Bezier's curve. Bezier's curves are fundamental tools widely employed in computer graphics and font generation.

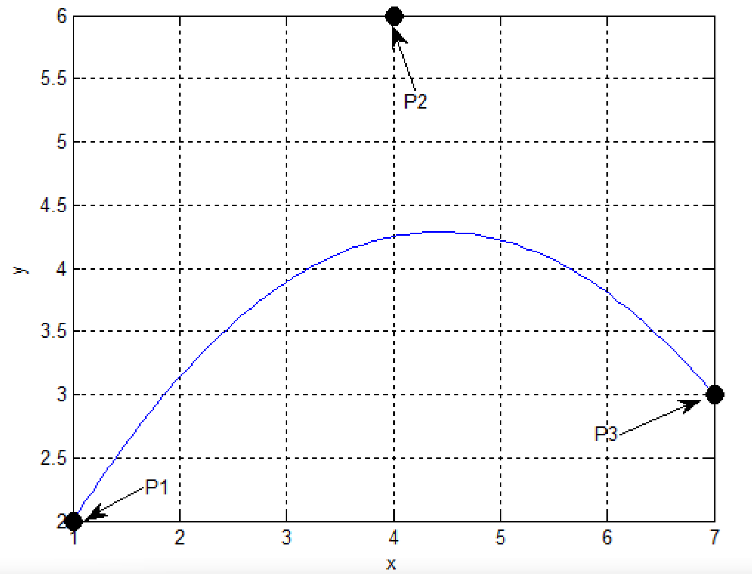

Fig.8. Generating quadratic Beziers curve (three control points)

Furthermore, it is suggested here to extend this method to multinomial expansion. For example, the extension to the trinomial case results in a generalized trinomial Composer formula:

$$C_{tri} = \sum_{\substack{k=0....n \\ k'=0.....k}} \begin{pmatrix} n \\ k \\ k' \end{pmatrix} x^{n-k}\, y^{k-k'} z^{k'} . P_m \qquad m = 1,2,..M = W_I(n) \tag{34}$$

If we specifically select the basic points for the trinomial case as three control points: P1= (x1, y1, z1), P2= (x2, y2, z2), and P3= (x3, y3, z3), with the parameters n=1, k=0…n, k'=0…k, the Composer formula will be:

$$C_{tri}(t) = xP_1 + yP_2 + zP_3 = t_xP_1 + t_yP_2 + t_zP_3 \tag{35}$$

where t_x is selected randomly from 0 to 1, t_y is selected randomly from 0 to 1 provided that $t_x + t_y \leq 1$. Then $t_z = 1 - t_x - t_y$. It is noted that the trinomial coefficients for this case (n=1) correspond to the second row of the Order 3 Meta triangle (Fig.(2d)), namely {1 1 1}. For n=2, Table 4 indicates that $W_I(n)$=$W_3(2)$ = 6 corresponding to the number of coefficients of the expansion. These coefficients, according to the third row of the Order 3 Meta triangle (Fig.(2d)), are {1 2 2 1 2 1}.

The trinomial Composer is then:

$$C_{tri} = \sum_{\substack{k=0....2 \\ k'=0.....k}} \begin{pmatrix} 2 \\ k \\ k' \end{pmatrix} x^{2-k}\, y^{k-k'} z^{k'} . P_m \qquad m = 1,2,..M = W_3(2) = 6 \tag{36}$$

Table 15 presents the calculated parameters.

Table 15
The calculated parameter for the trinomial Composer

n	2					
k	0	1		2		
K'	0	0	1	0	1	2
$\begin{pmatrix}2\\k\\k'\end{pmatrix}=\frac{2!}{(2-k)!\,(k-k')!\,k'!}$	$\begin{pmatrix}2\\0\\0\end{pmatrix}$	$\begin{pmatrix}2\\1\\0\end{pmatrix}$	$\begin{pmatrix}2\\1\\1\end{pmatrix}$	$\begin{pmatrix}2\\2\\0\end{pmatrix}$	$\begin{pmatrix}2\\2\\1\end{pmatrix}$	$\begin{pmatrix}2\\2\\2\end{pmatrix}$
	1	2	2	1	2	1
$x^{2-k}\, y^{k-k'} z^{k'}$	x^2	$x\,y$	x z	y^2	yz	z^2
t-parameter	${t_x}^2$	$t_x t_y$	$t_x t_z$	${t_y}^2$	$t_y t_z$	${t_z}^2$
Control points, P_m	P_1	P_2	P_3	P_4	P_5	P_6

Substituting the values of Table 15 into Eq. (36) yields:

$$\mathrm{C}_{tri} = x^2 P_1 + 2x\,yP_2 + 2x\,zP_3 + y^2 P_4 + \mathrm{yz}P_5 + z^2 P_6 \tag{37}$$

In terms of the t-parameters, one gets:

$$\mathrm{C}_{tri}(\mathrm{t}) = {t_x}^2 P_1 + 2t_x t_y P_2 + 2t_x t_z P_3 + {t_y}^2 P_4 + t_y t_z P_5 + {t_z}^2 P_6 \tag{38}$$

The trinomial Composer generates the shapes illustrated in Figures 9 and 10. This includes a plane triangular surface when n=1 (Figure 9) and a curved surface shape when n=2 (Figure 10).

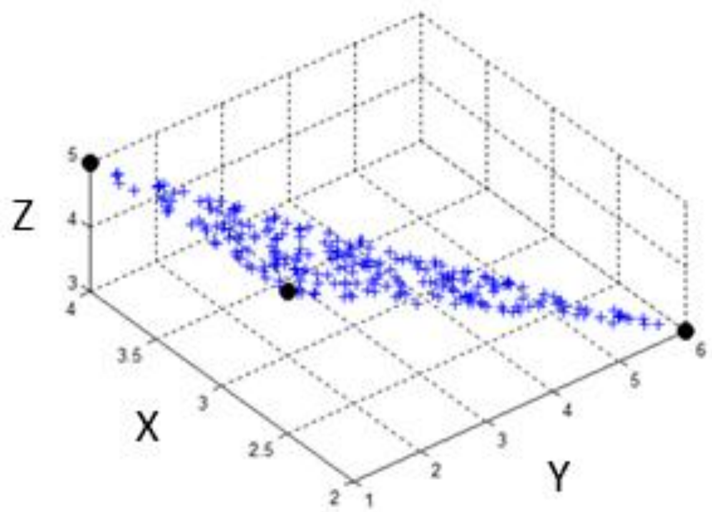

Fig. 9. Triangular surface, n=1, P1, P2, P3 Eq. (35)

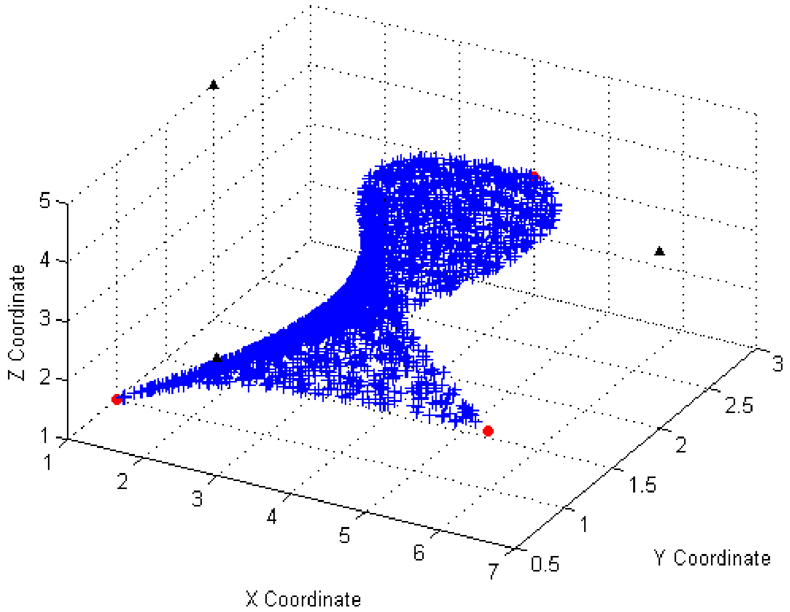

Fig. 10. Curved surface, n=2, P1, P2, P3, P4, P5, P6**,** Eq. (38)

6. Conclusion

The Guelph Expansion is a developed tool for calculating the roots of a polynomial knowing its coefficients and vice versa. It resembles the Vieta's formulas in its objectives with the advantage of stating explicitly the number of combined roots through the binomial coefficient. Its application to reactor kinetics yields a new Inhour polynomial leading to the Coefficients Based Model (CBM) for the point reactor kinetics.

A special case of the Guelph expansion reproduces the binomial expansion formula, with the binomial coefficient representing the structure of Pascal's triangle. It is readily possible to extend the expansion to multinomial expansion. The generated multinomial expansion, through its multinomial coefficients, represents the Generalized Pascal's Triangles (GPTs) or the META Triangles. This work introduces the structuring of these META triangles by presenting two algorithms to generate them; the EPTs and SHV methods. Furthermore, it is noted that Bezier's curves formula can be related to the dot product of the summand of binomial expansion and the selected control points. This principle was extended in the paper to introduce a modified trinomial expansion dotted with selected control points resulting in surface curves. The paper highlights some applications including combinatorics, computer graphics, DNA sequencing, genomic studies, fault tree analysis, and probability analysis (e.g., coin and dice tossing).

With the nature of the dimensionality of the multinomial expansions, and the META Triangles, relevant research could be pursued in Big Data Analysis, Number Theory, Encryption, Probability Theory, DNA-Large Language Models: gene-LLMs, and Quantum Computing. In future studies, recent adopted AI tools can be used [26-30].

Funding

This research received no external funding.

Conflicts of Interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was not funded by any grant.

References

[1] Edwards, A.W.F. (2002). Pascal's Arithmetical Triangle: The Story of a Mathematical Idea. The John Hopkins University Press.

[2] Duderstadt, J.J., & Hamilton, L.J. (1976). Nuclear Reactor Analysis. John Wiley& Sons. p.244.

[3] Ratemi, W.M., & Eshabo, A.E. (2009). New Form of the Inhour Equation and its Universal ABC-Values for Different Reactor Types . Annals of Nuclear Energy, 25(6): 377-386. https://doi.org/10.1016/S0306-4549(97)00066-2

[4] Ratemi, W.M., & Abdulla, H. (2009). The Guelph Expansion: a Mathematical Expansion for Polynomial Expansion. in: Abstracts of International Conference on Mathematical Sciences. Istanbul. p. 381.

[5] Ratemi, W.M. (2010). Embedded Pascal's Triangles for Polynomial Expansion. in: Abstract of the 19th International Workshop on Matrices and Statistics. Shanghai Finance University, Shanghai, China. 5-8 June, 2010, pp. 45-46.

[6] Provost, S.B., & Ratemi, W.M. (2011). Polynomial Expansions via Embedded Pascal's Triangles. Acta et Commentationes Universitatis Tartuensis de Mathematica, 15(1): 45-60. https://doi.org/10.12697/ACUTM.2011.15.04

[7] Ratemi, W.M. (2016). A Generalization of Planar Pascal's Triangle to Polynomial Expansion and Connection with Sierpinski Patterns. International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering, 10(2):15-20. https://doi.org/10.5281/zenodo.1338732

[8] Ratemi, W.M. (2010). Reactor Kinetics Revisited: A Coefficient Based Model (CBM). Editors: Khan, Syed, and Chohan. In Proceedings of International Conference on Power Generation Systems& Renewable Energy Technologies (Pgsret). Publisher: University of engineering and Technology, International Islamic University (IIU). pp.147-162.

[9] Ratemi, W.M. (2010). R-transforms for Polynomial Expansion and Root Extraction. Abstract-International Conference on Analysis and Applications (ICAA2010). Sultan Qaboos University, Muscat, Oman. p. 52.

[10] Beckmann, Petr. (1971). A History of pi. 2nd Edition. The Golem Press.

[11] Apostol, T. M. (1967). Calculus, Vol. 1: One-Variable Calculus, with an Introduction to Linear Algebra. 2nd. John Wiley & Sons.

[12] Ratemi, W.M. (2015). The Tripoli Polynomial and its Accompanied Differential Equations. in: Abstracts of International Congress on Industrial and Applied Mathematics, ICIAM. pp.219-220.

[13] Ratemi, W.M. (2014). Tagging of Genomes using Embedded Pascal Triangles Expansion. Universal Journal of Mathematics and Mathematical Sciences, 5(2): 121-132. https://www.pphmj.com/abstract/8309.htm

[14] Ratemi, W.M. (2013). Embedded Pascal Triangles and Its Application for Minimal Cut Sets of Fault Tree Analysis. Middle-East J. Sci. Res. 13 (Mathematical Applications in Engineering): 90-96. https://idosi.org/mejsr/mejsr13(mae)13/14.pdf

[15] Green, T.M. (1968). Recurrent Sequences and Pascal's Triangle, Math. Mag, 41(1): 13-21.

[16] Kisak, P.F. (2015). A History of Mathematics. CreateSpace Independent Publishing Platform. p.50.

[17] Kollar, M. (2025). The Janjić–Petković Inset Counting Function: Riordan Array Properties and a Thermodynamic Application. Mathematics, 13,3007. https://doi.org/10.3390/math13183007

[18] Shawesh, F. (2025). Architectural Geometry Reimagined: The Role of Bezier and Parametric Curves in Modern Design Workflow. Libyan J. Eng. Research, 40: 17-26.

[19] Ratemi, W.M., & Basir, O. (2014). Waterloo Numbers and Their Relation to Pascal Triangle and Polygons. The Aligarh Bulletin of Mathematics, 33(1-2): 39-53. http://tabmaths.com/issues/ABM_33_2014.pdf

[20] Sergei Abrammovich. (2025). From Counting to Computing: Ideas for Mathematics Education in Information Age. Emerald publishing Limited. https://doi.org/10.1108/978-1-83708-898-0

[21] Lamarsh, J.R., & Baratta, A.J. (2001). Introduction to Nuclear Engineering. 3rd ed. Prentice Hall.

[22] Oyamaa, A. L. Y. V., da Silvab, A. C, Gonçalvezb, A. C., Martinezb, A. S. (2025). Generalized Point Rêctor Kinetics. Braz. J. Radiat. Sci., Rio de Janeiro. 13(1). pp:1-17. https://doi.org/10.15392/2319-0612.2025.2773

[23] Hollos, S., & Hollos, J.R. (2019). Coin Tossing: The Hydrogen Atom of Probability. Abrazol Publishing. https://abrazol.com/books/cointoss2/

[24] Consens, A., et al. (2025). Gene-LLMs: A comprehensive survey of transformer-based genomic. *Frontiers in Genetics*, *16*. https://doi.org/10.3389/fgene.2025.1634882

[25] De Casteljau's algorithm. https://en.wikipedia.org/wiki/De_Casteljau%27s_algorithm

[26] Bouraima, M. B., & Badi, I. (2026). Advancing Environmental Sustainability through Artificial Intelligence: A Fuzzy SWOT–LOGSTA-Based Strategic Analysis. Knowledge and Decision Systems with Applications, 2, 422-435.

[27] Mushtaq, Y., Ali, W., Ghani, U., Khan, R. U., & Adak, A. K. (2025). Advancing aviation safety and sustainable infrastructure: High-accuracy detection and classification of foreign object debris using deep learning models. International Journal of Sustainable Development Goals, 1, 82-98.

[28] Kayıran, H. F. (2026). Thermo-Elastic Analysis of an Axisymmetric Layered Cylinder under Constant Thermal Loading with Artificial Intelligence-Based Validation. Intelligent Systems Research and Applications Journal, 2, 76-89.

[29] Farag, M. I. H. (2026). Artificial Intelligence and Strategic Decision-Making for Sustainable Governance. Knowledge and Decision Systems with Applications, 2, 447-468.

[30] Bouraima, M. B. (2026). Unlocking Artificial Intelligence for Sustainable Energy Transition: A Fuzzy MCDM Assessment of Economic and Environmental Barriers. International Journal of Sustainable Development Goals, 2, 448-460.

Appendix

List of symbols

The symbols and notation used throughout this paper are defined below for clarity and ease of reference.

Symbol	Description
Λ	Neutron generation time
ρ	Reactivity
β_i	The ith delayed neutron fraction
λ_i	The ith group precursors decay constant
ω	The roots of the inhour equation representing the reciprocal of system time constants
$\sum_{T_{nk}} \lambda \dots^{k} \dots \lambda$	The sum of the products of each and every possible combination of k elements of the set $\lambda_1, \dots, \lambda_n$
n	Number of groups of delayed neutrons
$\binom{n}{k}$	Binomial coefficient
$F(\omega)$	The inhour polynomial as a function of the variable ω
A_m	Coefficients of the inhour polynomial
a_m	Coefficient of Λ in A_m
b_m	Constant coefficient in A_m
c_m	Coefficient of r in Am
P	Nuclear reactor power
c	Precursors concentration (fission products producing delayed neutrons)
t	time
$\widetilde{A_m}$	Normalized coefficients of the inhour polynomial Am/Λ
a_i	Roots of a polynomial
P_0	Initial reactor power
b	Total delayed neutron fraction
l	Decay constant of group of delayed neutrons
t	Time constant (reactor period= 1/a)
$T_m(n)$	Tripoli Polynomials of degree m
$f^{(n-k)}$	*(n-k)*th derivative of the function *f*
r	Roots of polynomial
I	Number of variables in the multinomial expansion
$W_I(n)$	Element of the waterloo matrix for column I and row n
H	Head
T	Tail
Bi_{pdf}	Binomial probability distribution function
A, T, C, G	Four DNA nucleotides
B(t)	Bezier's curve
$C_{tri}(t)$	Tri-Composer
M	Total number of control points
P_m	m-th control point
t_x	t-parameter for x -variable
t_y	t-parameter for y -variable
t_z	t-parameter for z -variable