Hacking the Colebrook-White equation: The Tolentino-*Z* method via algebraic collapse and its bit-to-bit equivalence in 64-bit architectures

San L. TOLENTINO*,[1,2]

*Corresponding author
*,[1]Department of Mechanical Engineering,
Universidad Nacional Experimental Politécnica "AJS" (UNEXPO),
Bolívar, Venezuela,
stolentino@unexpo.edu.ve
[2]Group of Mathematical Modeling and Numerical Simulation (GMMNS),
Faculty of Oil, Natural Gas and Petrochemical Engineering,
Universidad Nacional de Ingeniería (UNI), Lima, Perú

DOI: 10.13111/2066-8201.2026.18.2.15

Received: 25 March 2026/ Accepted: 26 April 2026/ Published: June 2026

Abstract: *The resolution of the Colebrook-White equation has historically been a stability challenge in Computational Fluid Dynamics (CFD) due to its transcendental and implicit nature. This work presents the Tolentino-Z method (T_z), an explicit sequential resolution architecture based on an algebraic collapse that eliminates the need for conventional iterative schemes. Unlike iterative methods that depend on stopping criteria, the T_z method is structured as a fractional sequential equation with a closed two-stage cycle (Z_1, Z_2), composed of three concatenated stages: generation of calibrated explicit logarithmic seeds S_k; execution of the Z-refinement operator (Z-Engine), which is an explicit rational operator derived from the algebraic collapse of the Newton-Raphson iteration; and direct recovery of the friction factor f. The parameters b = 2.25 and n = 0.89 of the base seed were calibrated using a systematic grid search in Python 3.12. The results demonstrate that the Z_2 configuration with S_2 seed, over the range of relative roughness from 0.0 to 0.05 and Reynolds number from 4×10^3 to 1×10^8, achieves a maximum relative error of 2.74×10^{-17}, attaining bit-to-bit equivalence under the IEEE 754 standard. This precision exhausts the 52-bit mantissa in 64-bit architectures, guaranteeing absolute stability and asymptotic convergence over the entire domain of Reynolds number and relative roughness. The Tolentino-Z method constitutes a deterministic, explicit alternative for high-reliability engineering applications, digital twins, and industrial software with high energy efficiency.*

Key Words: *algebraic collapse, asymptotic convergence, Colebrook-White, computational fluid dynamics, digital twins, friction factor, IEEE 754 standard, Tolentino-Z method, Z-Engine*

1. INTRODUCTION

The calculation of the friction factor in pipes is a fundamental problem in fluid mechanics, with applications ranging from the design of water supply networks and oil pipelines to ventilation systems and industrial processes. The semi-empirical Colebrook-White equation [1], [2] constitutes the standard for estimating the friction factor in turbulent flow conditions, for both incompressible and compressible flow [3], [4]:

$$\frac{1}{\sqrt{f}} = -2log\left(\frac{\epsilon/d}{3.7} + \frac{2.51}{Re\sqrt{f}}\right) \quad (1)$$

where f is the friction factor; R_e is the Reynolds number [3], [5]; ϵ/d is the relative roughness, with ϵ being the average height of the roughness of the pipe inner wall and d the internal pipe diameter. Equation (1) is implicit for f, thus requiring iterative solution [6]. From a mathematical engineering perspective, this implicit nature demands analytical transformations, which has motivated decades of approximate explicit equations to avoid the computational burden and instability of such iterations.

As an alternative to the implicit Colebrook-White equation (1), Moody [7] proposed a diagram composed of families of plotted curves to obtain the friction factor by direct observation; this family of curves is known as the Moody chart

Among the most widespread explicit approximations are the Swamee-Jain [8] and Haaland [9] equations, with typical errors on the order of 2%. Subsequently, dozens of correlations have been published, many of them compiled and evaluated by Niazkar and Talebbyedokhti [10], which achieve relative errors below 0.1% over wide operating ranges. Reference is also made to other relevant works that have repeatedly evaluated the friction factor [11], [12], [13], [14], [15], [16].

In recent years, approaches based on Taylor series [17], Lagrange's theorem [18], algebraic and recursive adjustments [19], [20], decision tree models [21], genetic algorithms [22], artificial intelligence techniques with neural networks [23], [24], and symbolic regression based on genetic programming [25] have been explored. Likewise, complex network simulation through sectionalizing methods based on continuity and momentum residuals [26] and step-by-step algorithms coupled with multiphase flow models [27] have been investigated. However, none of these alternatives completely eliminates the need for iterations or provides a precision that exhausts the representation capacity of double-precision floating-point arithmetic (IEEE 754) [28], [29], which is standard in most computational simulation environments.

This work presents the Tolentino-Z method (T_z), a deterministic algorithm that solves the Colebrook-White equation through an algebraic collapse. In this context, the term "hacking" is defined as an unconventional analytical strategy that restructures a classical mathematical formulation to overcome its inherent computational limitations, enabling an explicit solution. Unlike traditional iterative methods, which depend on an initial seed and a stopping criterion, the proposed methodology structures the calculation into three concatenated phases: the definition of a calibrated seed S_k based on logarithmic nesting; the application of a closed operator called the Z-refinement operator (Z-Engine) that compresses the Newton-Raphson iteration into an algebraic expression; and the final recovery of the friction factor f.

This paper is organized as follows: Section 2 describes in detail the methodology, including the construction of the seeds, the Z-refinement operator, and the procedure for calculating the friction factor. Section 3 presents the comparative results and numerical validation. Finally, Section 4 presents the conclusions and the scope of the proposed method.

2. MATERIALS AND METHODS

To solve the implicit nature of the Colebrook-White equation (1), this work proposes an explicit sequential resolution architecture based on a stability algorithm called the Tolentino-Z method (T_z). The mathematical model (Recurrent Asymptotic Transformation Model

(RATM)) proposed for the calculation of the friction factor is defined as a fractional sequential equation. The proposed T_z method has a defined structure, which decomposes the problem into three critical phases: the seed S_k, the Z-refinement operator, and the friction factor f, designed to annihilate the numerical residue asymptotically.

2.1 Calibrated seeds to guarantee numerical convergence

To ensure convergence in the iterations, different seeds S_k are formulated, defined with different orders based on logarithmic nesting, following a recursion pattern similar to that reported in [20]. The base seed is established as $S_0 = a_3 + b \cdot a_2^n$ with empirical coefficients $b = 2.25$ and $n = 0.89$. The parameters b and n were calibrated in Python 3.12 using a systematic grid search with 100 levels for each parameter in the ranges $b \in [2.20, 2.30]$ y $n \in [0.87, 0.91]$, minimizing the absolute error of S_2 with respect to the reference value Z_2 for the critical case of Reynolds number 4×10^3 and relative roughness $\epsilon/d = 0.0$. It should be noted that the seed parameters a_2, a_3 and a_4 are defined in Section 2.2, Steps 1 and 3.

The seed in its general form is expressed as:

$$S_k = a_3 + a_4 ln(S_{k-1}) \tag{2}$$

where $k = 1, 2, 3, 4$ indicates the number of nested logarithms, which can be increased for $k > 4$. The coupling of the seed occurs by definition $Z_0 = S_k$. In particular, four seeds have been established:

Seed S_1: $Z_0 = a_3 + a_4 ln(a_3 + b \cdot a_2^n)$
Seed S_2: $Z_0 = a_3 + a_4 ln(a_3 + a_4 ln(a_3 + b \cdot a_2^n))$
Seed S_3: $Z_0 = a_3 + a_4 ln\big(a_3 + a_4 ln\big(a_3 + a_4 ln(a_3 + b \cdot a_2^n)\big)\big)$
Seed S_4: $Z_0 = a_3 + a_4 ln\big(a_3 + a_4 ln\big(a_3 + a_4 ln(a_3 + a_4 ln(a_3 + b \cdot a_2^n))\big)\big)$

2.2 Construction of the Z-refinement operator

Unlike conventional Newton-Raphson iterations, Tolentino-Z introduces a domain transformation that accounts for the linearization of the Newton-Raphson method, with the purpose of generating an algebraic collapse for which it was designed.

The procedure is presented below:

Step 1: Formulation of the original Colebrook-White equation.
Equation (1) is rearranged into a compact form, taking into account the logarithm transformation $log(y) = ln(y)/ln(10)$, which yields:

$$x = a_1 ln(a_3 + a_2 x) \tag{3}$$

The auxiliary variable x is related to the friction factor f by $x = 1/\sqrt{f}$.
The canonical constants are expressed as:
Logarithmic constant: $a_1 = -2/ln(10)$.
Reynolds scale factor: $a_2 = 2.51/Re$.
Roughness factor: $a_3 = (\epsilon/d)/3.7$.

Step 2: Auxiliary transformation (fundamental change of variable).
A new variable Z that absorbs the non-linearity of the equation is defined by:

$$Z = a_3 + a_2 x \tag{4}$$

It should be noted that the change of variable Z achieves the mapping needed to shift the instability of the implicit equation to a domain of absolute convergence.

Solving for x from equation (4):

$$x = \frac{Z - a_3}{a_2} \tag{5}$$

Step 3: Substitution and canonical form.
Substituting equations (4) and (5) into equation (3) gives:

$$\frac{Z - a_3}{a_2} = a_1 ln(Z) \tag{6}$$

Multiplying both sides by a_2:

$$Z - a_3 = a_1 a_2 ln(Z) \tag{7}$$

Define the composite parameter $a_4 = a_1 a_2$. Then the canonical form of the Colebrook-White equation is obtained:

$$Z = a_3 + a_4 ln(Z) \tag{8}$$

This expression reveals the essential structure of the equation: an implicit relation between Z and its natural logarithm, with two parameters a_3 (function of relative roughness) and a_4 (function of Reynolds number).

Step 4: Root-finding formulation and application of Newton-Raphson. Newton-Raphson establishes the iterative scheme:

$$Z_{i+1} = Z_i - \frac{f(Z_i)}{f'(Z_i)} \tag{9}$$

To solve equation (8), for $Z = Z_i$, we define the function:

$$f(Z_i) = a_3 + a_4 ln(Z_i) - Z_i \tag{10}$$

whose zeros correspond to the desired solutions. The derivative of $f(Z_i)$ is:

$$f'(Z_i) = \frac{a_4}{Z_i} - 1 \tag{11}$$

Substituting equations (10) and (11) into (9) yields:

$$Z_{i+1} = Z_i - \frac{a_3 + a_4 ln(Z_i) - Z_i}{\frac{a_4}{Z_i} - 1} \tag{12}$$

Noting that $\frac{a_4}{Z_i} - 1 = \frac{a_4 - Z_i}{Z_i}$, we have:

$$Z_{i+1} = Z_i - \frac{Z_i(a_3 + a_4 ln(Z_i) - Z_i)}{a_4 - Z_i} \tag{13}$$

Step 5: Algebraic collapse to the Z-refinement operator.
Expressing as a single quotient:

$$Z_{i+1} = \frac{Z_i(a_4 - Z_i) - Z_i(a_3 + a_4 ln(Z_i) - Z_i)}{a_4 - Z_i} \tag{14}$$

Expanding the numerator:

$$Z_{i+1} = \frac{a_4 Z_i - Z_i^2 - a_3 Z_i - a_4 Z_i ln(Z_i) + Z_i^2}{a_4 - Z_i} \tag{15}$$

Canceling Z_i^2:

$$Z_{i+1} = \frac{a_4 Z_i - a_3 Z_i - a_4 Z_i ln(Z_i)}{a_4 - Z_i} \tag{16}$$

Factoring Z_i:

$$Z_{i+1} = Z_i \cdot \frac{a_4 - a_3 - a_4 ln(Z_i)}{a_4 - Z_i} \tag{17}$$

Rearranging the numerator:

$$Z_{i+1} = Z_i \cdot \frac{a_4\big(1 - ln(Z_i)\big) - a_3}{a_4 - Z_i} \tag{18}$$

Step 6: Z-refinement operator (Z-Engine).
The algebraic collapse results in the Z-Engine, the core computational operator of the methodology:

$$Z_{i+1} = Z_i \left[\frac{a_4\big(1 - ln(Z_i)\big) - a_3}{a_4 - Z_i}\right] \tag{19}$$

where $i = 0,1,2 \ldots$ and $Z_0 = S_k$ is the chosen seed. This architecture collapses the Newton-Raphson iteration into a single rational expression, eliminating iterative overhead and facilitating stability analysis. Furthermore, calibrated seeding prevents singularities in the denominator $(a_4 - Z_i)$, guaranteeing the absolute stability across physical domain.

2.3 Reconditioning of the Colebrook-White equation to solve for the friction factor

Equation (4) is substituted into equation (3). Setting $Z = Z_{i+1}$, the following expression is obtained:

$$x = a_1 ln(Z_{i+1}) \tag{20}$$

Substituting the auxiliary variable $x = 1/\sqrt{f}$ (defined in Step 1) into equation (20) and solving for f gives the direct recovery of the friction factor:

$$f = [a_1 ln(Z_{i+1})]^{-2} \tag{21}$$

This final stage for computing the friction factor also acts as a convergence filter. Therefore, since it is coupled with equation (19), it facilitates its implementation for any case in the refinement stage of Z_{i+1}, for the entire Colebrook-White domain under turbulent flow.

2.4 Procedure sequence for calculating the friction factor

Based on the Tolentino-Z method, the calculation procedure to obtain the final friction factor is established according to the following structure:

1: *Seed* S_k: The type of seed $Z_0 = S_k$ (equation (2)) is chosen according to its logarithmic nesting. This selection is based on the conditions required for the initial positioning of numerical accuracy.

2: Z-refinement operator (Z-Engine): The selected seed $Z_0 = S_k$ is substituted into the equation (19). For Z_1 as a function of Z_0 we have the first refinement stage. The second refinement stage is Z_2 as a function of Z_1, and the third is Z_3 as a function of Z_2. This can be scaled to increase Z_{i+1} to accelerate the reduction of numerical error.

As a validation, for the scenario of Z_2 as a function of S_2, bit-to-bit equivalence was achieved under the IEEE 754 standard [28], [29] for 64-bit architectures. At this refinement level, the Z-refinement operator is expressed operationally as:

$$Z_2 = Z_1\left[\frac{a_4\big(1 - ln(Z_1)\big) - a_3}{a_4 - Z_1}\right]; \quad Z_1 = S_2\left[\frac{a_4\big(1 - ln(S_2)\big) - a_3}{a_4 - S_2}\right]$$

3: *Friction factor* f: Equation (21) is used to calculate the friction factor. For example, for the particular case where Z_2 is known, the final friction factor is obtained as:

$$f = [a_1 ln(Z_2)]^{-2}$$

2.5 Parameter ranges considered in the turbulent flow regime

Relative roughness ϵ/d: The relative roughness range was set from smooth pipe $\epsilon/d = 0.0$ to fully rough pipe $\epsilon/d = 0.05$. Six cases were evaluated: $\epsilon/d = 0.0$, $\epsilon/d = 0.00001$, $\epsilon/d = 0{,}0001$, $\epsilon/d = 0.001$, $\epsilon/d = 0.01$ and $\epsilon/d = 0.05$.

Reynolds number R_e: The Reynolds number range was set from $4\text{x}10^3$ to $1\text{x}10^8$.

Relative error of the friction factor f. The relative error and the maximum relative error of the friction factor were evaluated over the entire domain of the Colebrook-White equation for the relative roughness range $\epsilon/d = 0.0$ to $\epsilon/d = 0.05$ and for the Reynolds number range $4\text{x}10^3$ to $1\text{x}10^8$, for the refined stages Z_1, Z_2 and Z_3 as functions of the seeds S_1, S_2, S_3 and S_4, respectively. The Z-refinement operator is implemented as a deterministic refinement scheme (typically $Z_1 - Z_3$), in which the Newton-Raphson iteration is collapsed into an explicit algebraic sequence, avoiding open iterative loops dependent on convergence criteria.

The Newton-Raphson method was used as a control parameter for the friction factor to compare the results yielded by the Z-refinement operator. For the calculations, the Python 3.12 library "decimal" was used to achieve 50 decimal places.

The computational equipment used had the following characteristics: Dell CPU, Optiplex 7010 model, i5, 32 GB RAM.

3. RESULTS AND DISCUSSIONS

This section presents a comparative analysis of the friction factor for three refinement stages of the Z-refinement operator based on four defined seeds, for which the corresponding graphs and tables are shown.

3.1 Curve trajectories for three refinement stages of the Z-refinement operator

For the case of seed S_1, Fig. 1 illustrates the behavior of the curve trajectories (logarithmically spaced), where for relative roughness $\epsilon/d = 0.0$ (Fig. 1a) the relative errors are largest at the Reynolds number reference $1\text{x}10^8$.

As relative roughness increases, the errors slightly decrease at the Reynolds number reference $4\text{x}10^3$, while at the reference $1\text{x}10^8$ the errors drop dramatically. The curves have

pronounced valleys at the beginning ($\epsilon/d = 0.0$); as relative roughness increases, the curves become smooth and tend to be straight with negative slopes ($\epsilon/d = 0.05$).

The smallest drops in relative error occur in the pronounced valleys and they decrease even more as the refinement stages of the Z-refinement operator increase, as observed for relative roughness cases $\epsilon/d = 0.0$, $\epsilon/d = 0.00001$, $\epsilon/d = 0.0001$ at Reynolds numbers near $1x10^4$ and $1x10^6$, respectively.

Similar error drops occur in Figs. 2, 3 and 4, and it is observed that seeds S_2, S_3 and S_4 have a significant effect when applied to refinement stages Z_1, Z_2 and Z_3, shifting the curves in a stepwise manner.

As a reference, in Figs. 1, 2, 3 and 4 a dashed red line is shown; this line corresponds to the approximate relative error limit of $1.11x10^{-16}$ of the IEEE 754 standard [28], [29] for bit-to-bit equivalence, achieving 17 decimal places. Therefore, any curve or segment of the friction factor relative error that crosses the $1.11x10^{-16}$ boundary attains values greater than 17 decimal places.

The bounded limit for the graphical representation of the friction factor relative error is up to $1x10^{-50}$, which is compared with the Newton-Raphson benchmark at 50 decimal places, to ensure numerical accuracy.

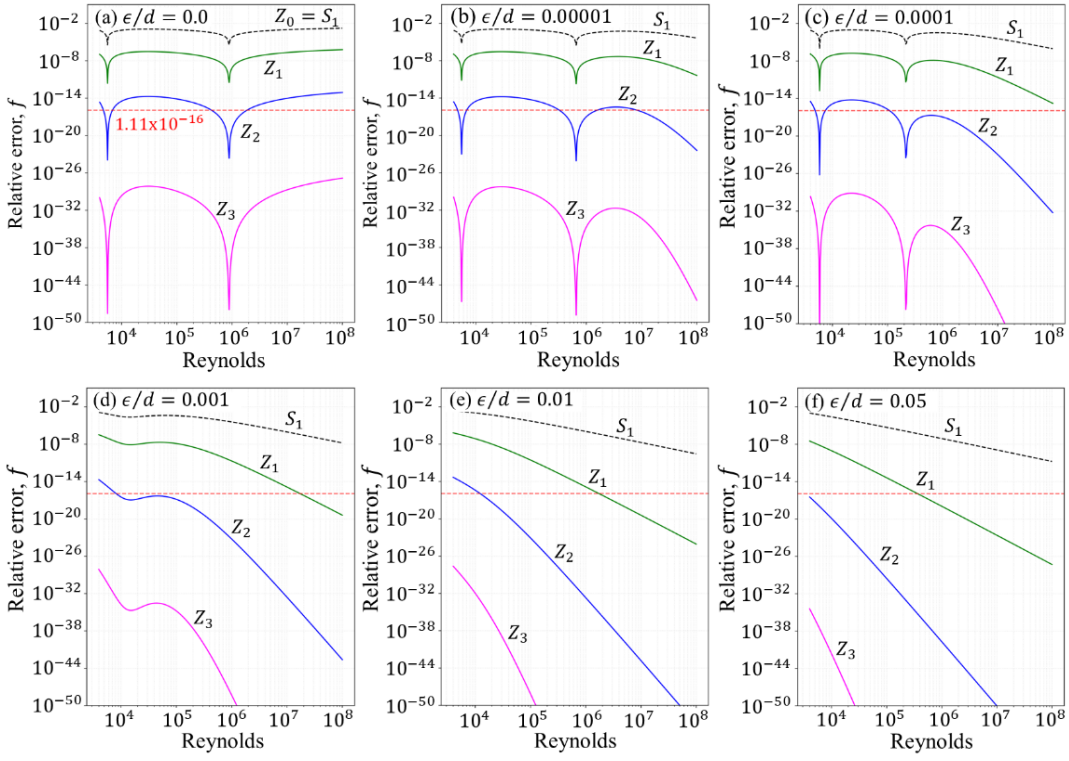

Fig. 1 – Seed S_1: Relative error of the friction factor f for refinement stages Z_1, Z_2 and Z_3. Range of relative roughness from 0.0 to 0.05 and Reynolds number from $4x10^3$ to $1x10^8$

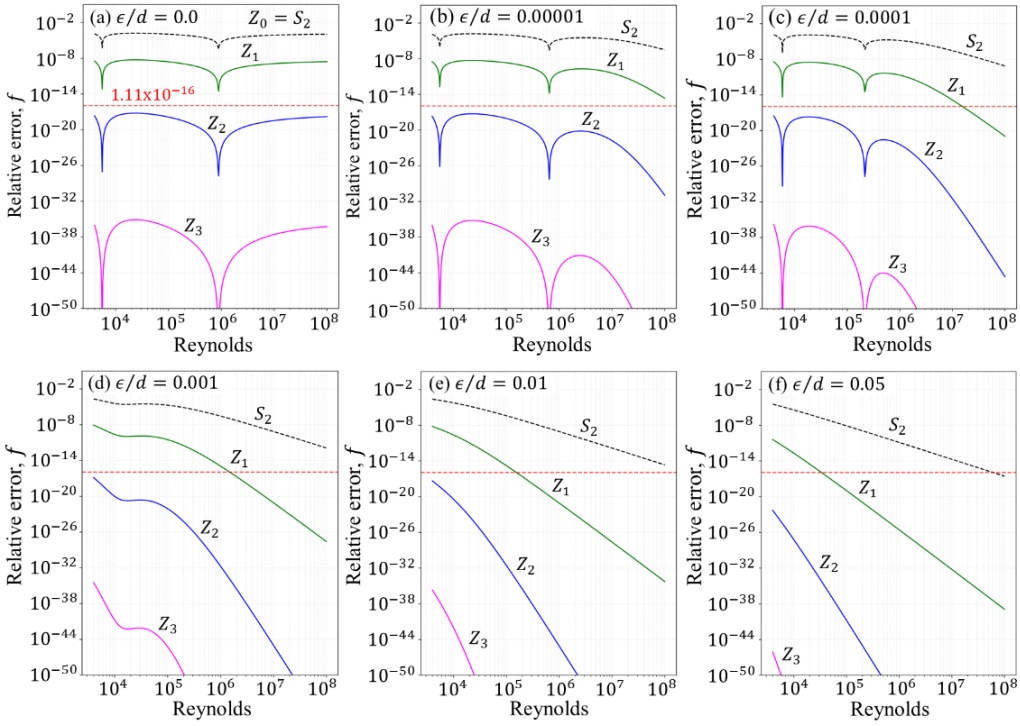

Fig. 2 – Seed S_2: Relative error of the friction factor f for refinement stages Z_1, Z_2 and Z_3. Range of relative roughness from 0.0 to 0.05 and Reynolds number from $4x10^3$ to $1x10^8$

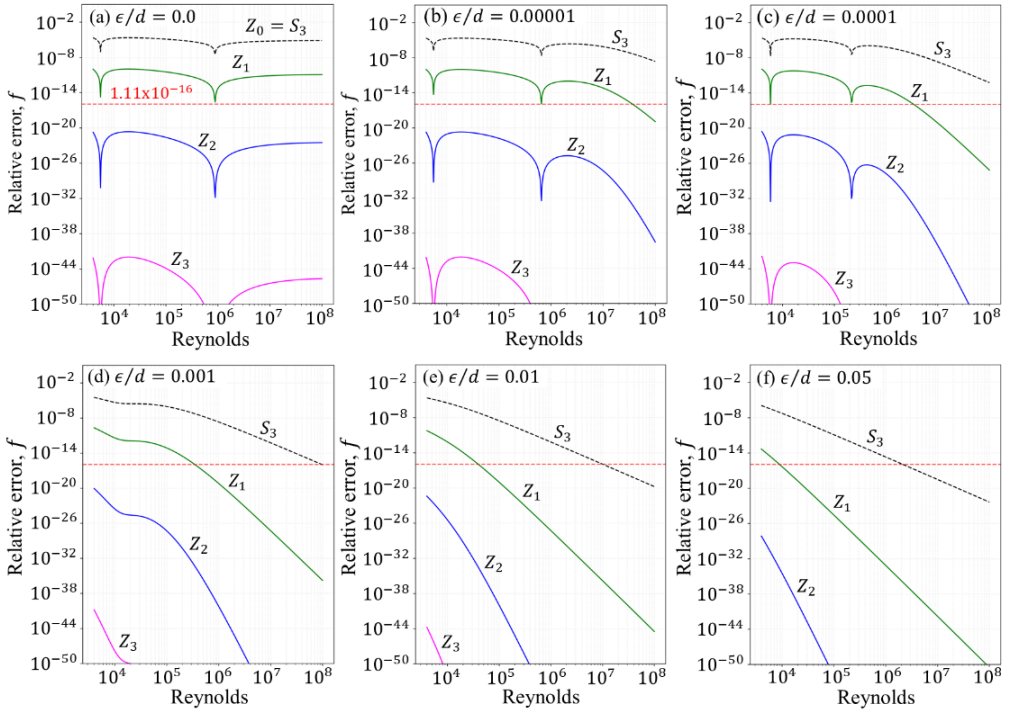

Fig. 3 – Seed S_3: Relative error of the friction factor f for refinement stages Z_1, Z_2 and Z_3. Range of relative roughness from 0.0 to 0.05 and Reynolds number from $4x10^3$ to $1x10^8$

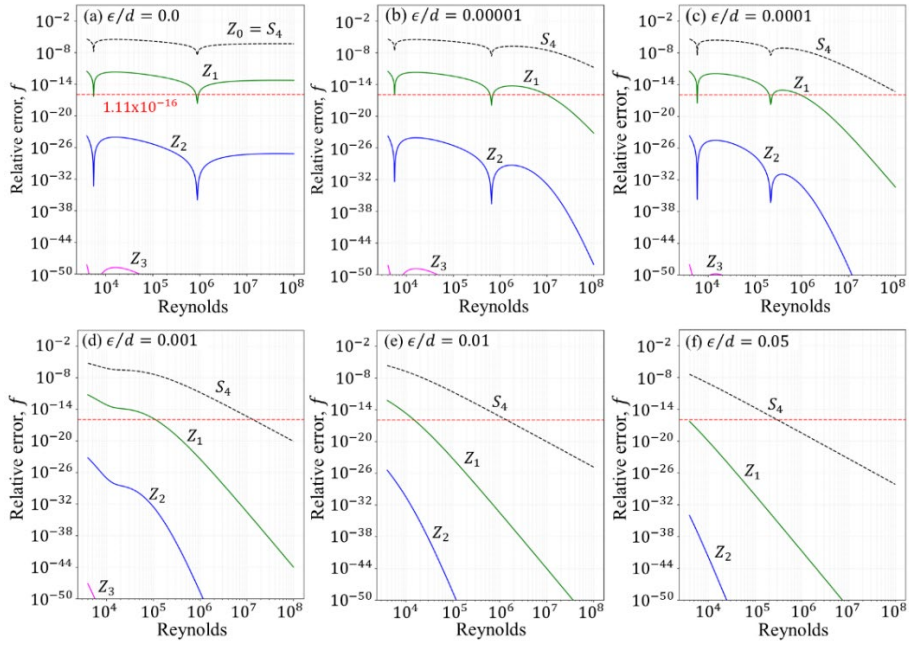

Fig. 4 – Seed S_4: Relative error of the friction factor f for refinement stages Z_1, Z_2 and Z_3. Range of relative roughness from 0.0 to 0.05 and Reynolds number from 4x10^3 to 1x10^8

3.2 Comparative analysis of the friction factor for three stages of the Z-refinement operator

The behavior of the curve trajectories for the refinement stages as a function of the seeds is illustrated in Fig. 5. The vertical axis shows the maximum relative error of the friction factor. In the initial stages, for Z_0 and Z_1, the maximum relative errors are of the order of 1x10^{-1} to 1x10^{-12} with ULP (Units in the Last Place) distances [28], [29] of the order of 1x10^{13} to 1x10^4, indicating that machine precision has not yet been reached. However, starting from configuration Z_2 with S_2, , the error drops drastically below 1x10^{-16} and the ULP distance reduces to values ≤ 1, demonstrating convergence to the theoretical approximate limit of 1.11x10^{-16} of the IEEE 754 double-precision standard (52-bit mantissa) [29]. Notably, ULP is defined as the distance between two consecutive representable numbers in floating-point arithmetic [29]; a value ≤ 1 indicates that the numerical error is smaller than the machine resolution, thus achieving bit-to-bit equivalence under the IEEE 754 standard [28], [29].

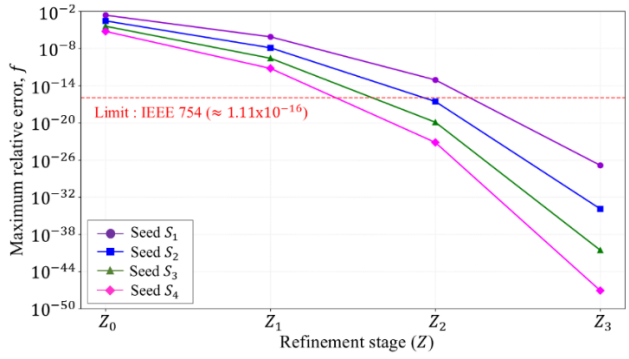

Fig. 5 – Maximum local relative error of the friction factor f. Range of relative roughness from 0.0 to 0.05 and Reynolds number 4x10^3 to 1x10^8

The maximum relative errors, reported to 50 decimal places, are summarized in Table 1 for the refinement stages Z_2 and Z_3 as functions of seeds S_1, S_2, S_3 and S_4. For stage Z_2 with seed S_2, the error is 2.74x10^{-17}, which is smaller than both the rounding error bound of 1.11x10^{-16} for (IEEE 754) and the size of 1 ULP (at 1.0) of 2.22x10^{-16} [29]. Therefore, configuration Z_2 with S_2 is identified as the convergence optimum, achieving deterministic bit-to-bit equivalence.

Table 1 – Maximum relative error for different stages of the Z-refinement operator as a function of different seeds

Z	S	Maximum relative error (50 decimal places)	Maximum ULP distance
Z_2	S_1	8.36x10^{-14}	541
Z_2	**S_2**	**2.74x10^{-17}**	**1**
Z_2	S_3	1.24x10^{-20}	0
Z_2	S_4	6.71x10^{-24}	0
Z_3	S_1	1.37x10^{-27}	0
Z_3	S_2	1.29x10^{-34}	0
Z_3	S_3	2.81x10^{-41}	0
Z_3	S_4	8.46x10^{-48}	0

Table 2 presents a comparison of friction factor results f_Z obtained by the Tolentino-Z method against f_N obtained by the Newton-Raphson method, for 50 decimal places; for comparative purposes under the IEEE 754 standard, the values were truncated to 17 decimal places for presentation.

The obtained maximum percentage relative errors correspond to local points of relative roughness and Reynolds number. It is evident that the numerical errors of the friction factor decrease as the logarithmic nesting in the seed increases. The critical zone where the largest errors occur is at Reynolds number 4x10^3. For Z_3 with S_1 and smooth pipe ($\epsilon/d = 0$), the error appears at Reynolds number 1x10^8.

For stage Z_2 with seed S_1 (Table 2 and Fig. 5), the maximum percentage relative error for the friction factor at 17 decimal places is 8.36x10^{-12}%, a value that is satisfactory for engineering applications that do not require numerical precision with 17 decimal places for the friction factor.

The optimal point presenting a relative percentage error of 0.0% is configuration Z_2 with S_2 (Table 2 and Fig. 5), which yields bit-to-bit equivalence under the IEEE 754 standard for 64-bit architectures.

For extremely rigorous numerical precision requirements in the friction factor that exceed 17 decimal places, configuration Z_2 with S_3 is satisfactory, as it guarantees numerical convergence when using the "decimal" library in Python or other code.

Likewise, increasing the nesting level to S_4 would be satisfactory for specific user requirements. However, under the IEEE 754 standard, it exceeds the demands for 64-bit architectures because it exhausts the representation capacity of double-precision computational architecture.

For refinement stage Z_3 and seeds S_1, S_2, S_3 and S_4 (Tables 1 and 2, Fig. 5), the data have been recorded to understand the behavior of the convergence curve trajectories, which shows that the Z-refinement operator is scalable, with the error of the friction factor tending toward zero.

Table 2 – Maximum percentage relative errors of the friction factor f_Z and f_N (values shown with 17 decimal places)

Z	S	ϵ/d	Re	f_Z (Tolentino-Z)	f_N (Newton)	Error %
Z_0	S_1	8.24x10^{-3}	4x10^3	0.04747214296782232	0.04758252572594279	2.32x10^{-1}
Z_0	S_2	4.19x10^{-3}	4x10^3	0.04397779036975399	0.04396551279430830	2.79x10^{-2}
Z_0	S_3	2.67x10^{-3}	4x10^3	0.04253323279976310	0.04253486380634844	3.83x10^{-3}
Z_0	S_4	2.13x10^{-3}	4x10^3	0.04201730124592102	0.04201706376584958	5.65x10^{-4}
Z_1	S_1	5.25x10^{-3}	4x10^3	0.04493726666280266	0.04493730016847840	7.46x10^{-5}
Z_1	S_2	3.34x10^{-3}	4x10^3	0.04317535899971726	0.04317535954958154	1.27x10^{-6}
Z_1	S_3	2.13x10^{-3}	4x10^3	0.04201706375491186	0.04201706376584958	2.60x10^{-8}
Z_1	S_4	1.70x10^{-3}	4x10^3	0.04159940814199579	0.04159940814224396	5.97x10^{-10}
Z_2	S_1	5.25x10^{-3}	4x10^3	0.04493730016847464	0.04493730016847840	8.36x10^{-12}
Z_2	**S_2**	**3.34x10^{-3}**	**4x10^3**	**0.04317535954958154**	**0.04317535954958154**	**0.0**
Z_2	S_3	2.13x10^{-3}	4x10^3	0.04201706376584958	0.04201706376584958	0.0
Z_2	S_4	1.70x10^{-3}	4x10^3	0.04159940814224396	0.04159940814224396	0.0
Z_3	S_1	0.0	1x10^8	0.00594046635163676	0.00594046635163676	0.0
Z_3	S_2	2.67x10^{-3}	4x10^3	0.04253486380634844	0.04253486380634844	0.0
Z_3	S_3	2.13x10^{-3}	4x10^3	0.04201706376584958	0.04201706376584958	0.0
Z_3	S_4	1.70x10^{-3}	4x10^3	0.04159940814224396	0.04159940814224396	0.0

The proposed mathematical model is a three-step structure that allows a synchronized stage solution, comprising the seed S_k based on equation (2), the Z-refinement operator based on equation (19), and the friction factor f based on equation (21); this structure is a fractional sequential equation of explicit direct calculation.

Unlike iterative methods where the seed is an external agent prone to error, in the Tolentino-Z method, the seed configuration S_k acts as an internal precision modulator. The Z_2 refinement stage with the S_2 seed is found to be optimal, achieving agreement within the limits of double-precision arithmetic (IEEE 754) and exhibiting bit-to-bit equivalence up to the 17th decimal place under high-precision analysis. Furthermore, the Z_2 configuration with S_3 ensures robust performance under extreme conditions.

3.3 Double-precision floating-point arithmetic (IEEE 754)

Double-precision floating-point arithmetic (IEEE 754), natively used by Python, provides sufficient precision for most engineering calculations. Within this framework, the friction factors f_Z, f_N and f_P obtained with the Tolentino-Z method (configuration Z_2 with S_2), the Newton-Raphson iterative scheme, and fixed-point iteration are numerically indistinguishable, as shown in Table 3 for the critical region (Re = 4x10^3, ϵ/d = 0.0 to 0.05). For other Reynolds number, the behavior is analogous.

When results are displayed with 16 decimal places, all three methods produce identical values. Minor differences beyond this level are inherent to the floating-point representation and do not affect engineering calculations, which typically require far fewer decimal places.

Standard spreadsheet software and many engineering tools operate with a precision of 15–16 total digits, which is sufficient for most practical applications. However, for analyses requiring precision beyond double-precision arithmetic (e.g., bit-level agreement or numerical sensitivity studies), the Python decimal module can be used with higher precision. In this work, 50 decimal places were employed. Under this high-precision configuration, the Tolentino- Z

method (configuration Z_2 with S_2) coincides with Newton–Raphson up to the 16th decimal place, confirming deterministic equivalence under the IEEE 754 standard.

Table 3 – Comparison of results for Reynolds number 4x10^3, critical region. (configuration Z_2 with S_2)

ϵ/d	Re	f_Z (Tolentino-Z)	f_N (Newton)	f_P (Fixed-point)
0.0	4x10^3	0.0399070140556349	0.0399070140556349	0.0399070140556349
0.00001	4x10^3	0.0399171668504101	0.0399171668504101	0.0399171668504101
0.0001	4x10^3	0.0400084312335555	0.0400084312335555	0.0400084312335555
0.001	4x10^3	0.0409103898628461	0.0409103898628461	0.0409103898628461
0.01	4x10^3	0.0490822694478997	0.0490822694478997	0.0490822694478997
0.05	4x10^3	0.0769868348892249	0.0769868348892249	0.0769868348892249

3.4 Moody diagram generated with the Tolentino-Z Method

To visually demonstrate the equivalence between the Tolentino-Z method and the conventional Newton-Raphson iterative scheme, the Moody diagram was constructed using both approaches. The friction factor was computed for Reynolds numbers ranging from 4x10^3 to 1x10^8 (logarithmically spaced) and for relative roughness values $\epsilon/d = 0.0, 0.00001, 0.0001, 0.001, 0.01, 0.05$. These ranges cover the typical operating conditions of turbulent pipe flow, from smooth to fully rough regimes.

As a reference, the Newton-Raphson method was implemented with an initial guess provided by the Swamee-Jain explicit equation. A convergence tolerance of 1x10^{-12} was used, and the algorithm consistently converged in fewer than 10 iterations for all points evaluated.

Figure 6 shows the resulting Moody diagram. The solid black lines correspond to the Newton-Raphson reference, while the dashed red lines correspond to the Tolentino-Z method with the configuration Z_2 and S_2 seed. The curves are indistinguishable at the scale of the plot, confirming that the explicit T_z method, which requires no iterative loops or convergence criteria, reproduces the friction factor with the same accuracy as the Newton-Raphson iterative scheme. Consequently, the explicit formulation of the T_z method emulates the behavior of the friction factor curves over the entire Colebrook-White domain for turbulent flow.

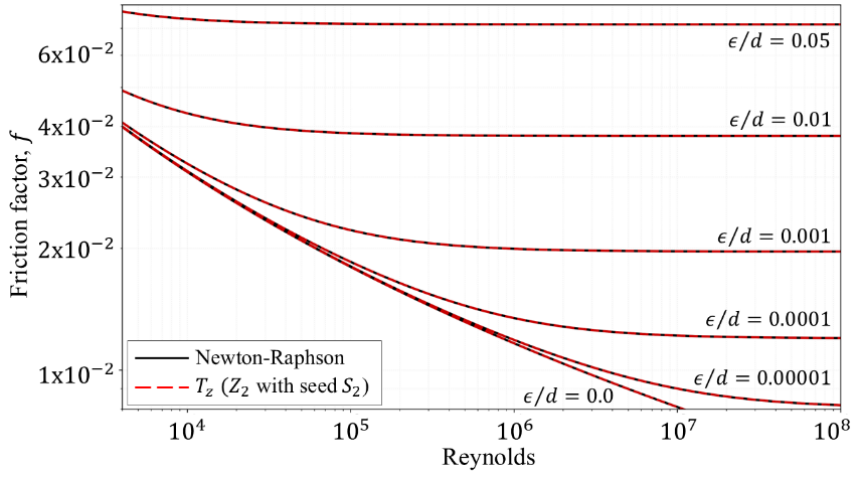

Fig. 6 – Moody diagram: Comparison between the Newton-Raphson method (solid black lines) and the Tolentino-Z method (dashed red lines). Range of relative roughness from 0.0 to 0.05 and Reynolds number 4x10^3 to 1x10^8

3.5 Industrial implications and energy efficiency

The Tolentino-Z method, by achieving results that are numerically indistinguishable within machine precision, effectively eliminates numerical discrepancies in the calculation of head loss, enabling improved optimization of power consumption in compression and pumping systems. The machine-precision friction factor can be directly substituted into the Darcy-Weisbach equation, yielding head losses with the same exactness and eliminating a key source of uncertainty in hydraulic design.

Furthermore, its deterministic nature with a closed two-stage cycle (Z_1, Z_2) ensures absolute stability in high-reliability hydraulic simulators, where the convergence of traditional iterative methods is often compromised by noise from field sensor data. This unprecedented precision translates directly into reduced operating costs and a significant decrease in the carbon footprint associated with large-scale fluid transport.

4. CONCLUSIONS

After the physical-mathematical analysis and numerical validation of the Tolentino-Z method (T_z), the following conclusions are drawn:

Transition from implicit to deterministic: The T_z method successfully transforms the transcendental nature of the Colebrook-White equation into an explicit sequential resolution architecture. By fixing the refinement of the Z-refinement operator (Z-Engine) in a closed two-stage cycle (Z_1, Z_2), the scheme transcends the open iterative process to become a constant-time algorithm, eliminating the uncertainty associated with traditional convergence criteria.

Efficiency of algebraic collapse: The development of the Z-refinement operator through the algebraic collapse of the Newton-Raphson iteration reduces the numerical calculation to its minimal expression. This closed operator, together with the calibrated logarithmic seeds S_k, guarantees stability over the entire operating domain, avoiding singularities and divergences.

Bit-to-bit equivalence (IEEE 754): It is demonstrated that the configuration Z_2 with seed S_2, which involves two applications of the Z-refinement operator, achieves a maximum relative error of 2.74×10^{-17}. This value is below the machine precision limit for 64-bit architectures (1.11×10^{-16}), achieving bit-to-bit equivalence and exhausting the double-precision mantissa representation capacity.

Scalability and extreme precision: The method is intrinsically scalable. By increasing the nesting level of the seed or the number of refinements of the Z-refinement operator, it is possible to achieve precision on the order of 1×10^{-48}, surpassing the requirements of any current high-reliability simulation.

Impact on engineering and CFD: The Tolentino-Z method constitutes a disruptive tool for computational fluid dynamics and digital twins. Its deterministic convergence architecture allows seamless integration into industrial software, eliminating the instability of traditional methods, optimizing energy efficiency, and significantly reducing computational load in large-scale pipe networks.

REFERENCES

[1] C. F. Colebrook and C. M. White, Experiments with fluid friction in roughened pipes, *Proceedings of the Royal Society of London, Series A-Mathematical and Physical Sciences*, vol. **161**, no. 906, pp. 367-381, 1937, https://doi.org/10.1098/rspa.1937.0150

[2] C. F. Colebrook, Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws, *Journal of the Institution of Civil Engineers*, vol. **11**, no. 4, pp. 133–156, 1939, https://doi.org/10.1680/ijoti.1939.13150
[3] F. M. White, *Fluid Mechanics*, McGraw Hill, 2021
[4] Y. A. Cengel and J. M. Cimbala, *Fluid Mechanics: Fundamentals and Applications*, McGraw Hill, 2024
[5] O. Reynolds, An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels, *Philosophical Transactions of the royal society of London,* vol. 174, pp. 935-982, 1883, https://doi.org/10.1098/rstl.1883.0029
[6] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, McGraw Hill, 2021
[7] L. F. Moody, Friction factor for pipe flow, *Transaction of ASME*, vol. **66**, no. 8, pp. 671–684, 1944, https://doi.org/10.1115/1.4018140
[8] P. K. Swamee and A. K. Jain, Explicit equations for pipe-flow problems, *Journal of the Hydraulics Division*, vol. **102,** no. 5, pp. 657–664, 1976, https://doi.org/10.1061/JYCEAJ.0004542
[9] S. E. Haaland, Simple and explicit formulas for the friction factor in turbulent pipe flow, *Journal of Fluids Engineering*, vol. **105**, no. 1, pp. 89–90, Mar 1983, https://doi.org/10.1115/1.3240948
[10] M. Niazkar and n. Talebbeydokhti, Comparison of explicit relations for calculating Colebrook friction factor in pipe network analysis using h-based methods, *Iran J Sci Technol Trans Civ Eng,* vol. **44**, pp. 231-249, 2020, https://doi.org/10.1007/s40996-019-00343-2
[11] M. Tchawe, F. N. Ngongang, M. T. Nsiewe, T. Djiako, D. Tcheukam and B. Kenmeugne, Determination of the coefficient of friction in the penstock of a hydro-electric dam: Application of two approaches, *Hydrology Research*, vol. **55**. No. 10, p. 1030, 2024, https://doi.org/0.2166/nh.2024.054
[12] R. T. Minhoni, F. F. Pereira, T. B. da silva, E. R. Castro and J. C. Saad, The performance of explicit formulas for determinig the Darcy-Weisbach friction factor, *Engenharia Agrícola*, vol. **40**, no. 2, pp. 258-265, 2020, http://dx.doi.org/10.1590/1809-4430-Eng.Agric.v40n2p258-265/2020
[13] M. López, D. S. Camenates, N. Delgado and N. Chunga, Explicit pipe friction factor equations: evaluation, classification, and proposal, *Revista Facultad de Ingeniería Universidad de Antioquía*, no. 111, pp. 38-47, 2024, https://doi.org/10.17533/udea.redin.20230928
[14] L. E. Muzzo, G. K. Motoba and L. F. Ribeiro, Uncertainty of pipe flow friction factor equations, *Mechanics Research Communications*, vol. **116**, p. 103764, 2021, https://doi.org/10.1016/j.mechrescom.2021.103764
[15] A. Nasir, A. O. Salau, E. Dribssa and M. Grima, Technical an economic analysis of a pump as a turbine for rural electrification, *International journa of Sustainable energy*, vol. **42**, no. 1, pp. 914-928, 2023, https://doi.org/10.1080/14786451.2023.2244606
[16] V. Mileikovskyi and T. Tkachenko, Precise Explicit Approximations of the Colebrook-White Equation for Engineering Systems, *In: Blikharskyy, Z. (eds) Proceedings of EcoComfort 2020. Lecture Notes in Civil Engineering*, vol. **100**, pp. 303-310, 2021, Springer, Cham. https://doi.org/10.1007/978-3-030-57340-9_37
[17] Z. Hafsi, Accurate explicit analytical solution for Colebrook-White equation, *Mechanics Research Communications*, vol. **111**, 2021, p. 103646, https://doi.org/10.1016/j.mechrescom.2020.103646
[18] A. A. Lamri and S. M. Easa, Computationally efficient and accurate solution for Colebrook equation based on Lagrange theorem, *Journal of Fluids Engineering*, vol. **144**, p. 014504, 2022, https://doi.org/10.1115/1.4051731
[19] G. B. Ferreri, A new approach for explicit approximation of the Colebrook-White formula for pipes flows, *Journal of Hydroinformatics,* vol. **26**, no. 7, p. 1558, 2024*,* https://doi.org/10.2166/hydro.2024.280
[20] S. L. Tolentino and O. González, Correlation for the calculation of turbulent friction in pipes, *Ingenius, Revista de Ciencia y Tecnología,* no. 30, pp. 54-63, 2023, https://doi.org/10.17163/ings.n30.2023.05
[21] M. Niazkar, A., Menapace and M., Righetti, Estimating Colebrook-White Friction Factor Using Tree-Based Machine Learning Models, In: *Concli, F., Maccioni, L., Vidoni, R., Matt, D.T. (eds) Latest Advancements in Mechanical Engineering, ISIEA 2024. Lecture Notes in Networks and Systems*, vol **1124**. Springer, Cham. https://doi.org/10.1007/978-3-031-70462-8_26
[22] C. C. Shiu, C.C., Chung and T. Chiang, Enhancing the EPANET Hydraulic Model through Genetic Algorithm Optimization of Pipe Roughness Coefficients, *Water Resources Management*, vol. **38**, pp. 323–341, 2024. https://doi.org/10.1007/s11269-023-03672-0
[23] V. Srivastava, A. Prakash and A., Rawat, To Predict Frictional Pressure-Drop of Turbulent Flow of Water Through a Uniform Cross-Section Pipe Using an Artificial Neural Network. In: *Tadepalli, T., Narayanamurthy, V. (eds) Recent Advances in Applied Mechanics. Lecture Notes in Mechanical Engineering*, Springer, Singapore, 2022. https://doi.org/10.1007/978-981-16-9539-1_28
[24] M. Cahyono, Hybrid models for solving the colebrook–white equation using artificial neural networks, *Fluids*, vol. **7**, no. 7, pp. 211, 2022. https://doi.org/10.3390/fluids7070211

[25] Y. E. Ünal and Ö. Ertunç, Order of Magnitude analysis and data-based physics-informed symbolic regression for turbulent pipe flow, *arXiv:2602.17082v1 [Physics.flu-dyn],* Feb. 2026

[26] D. Zhang and Y. Yang, Development of a sectionalizing method for simulation of large-scale complicated natural gas pipeline networks, *Journal of pipeline science and engineering*, vol. **4**, no. 4, p. 100209, 2024. https://doi.org/10.1016/j.jpse.2024.100209

[27] Z. Zhang, B. Cao, G. Ruan, D. Cao, Z. Tan and S. Yang, Calculation method of multi-stage gas lift process parameters in deep offshore low-permeability oil and gas fields, *Earth Energy Science*, 2025, https://doi.org/10.1016/j.ees.2025.10.002

[28] *IEEE Standard for Floating-Point Arithmetic*, IEEE Std 754-2019 (revision of IEEE Std 754-2008), 2019. https://doi.org/10.1109/IEEESTD.2019.8766229

[29] S. L. Abrams, W. Cho, C.-Y. Hu, T. Maekawa, N. M. Patrikalakis, E. C. Sherbrooke, and X. Ye, Efficient and reliable methods for rounded-interval arithmetic, *Computer-Aided Design*, vol. **30**, no. 8, pp. 657–665, 1998, https://doi.org/10.1016/S0010-4485(98)00019-2