Transactions of NAMP 24, (2026) 179-188

Transactions of
The Nigerian Association of
Mathematical Physics
Journal homepage: https://nampjournals.org.ng

A MODULAR PYTHON-BASED FRAMEWORK FOR AUTOMATED HYDRAULIC FLOW BALANCING IN LOOPED PIPE NETWORKS USING THE HARDY CROSS METHOD

[1]Moses, G. O., [2]Aboloje, J. C., [3]Otobor, O. D., [4]Iseguan F. E and [5]Audu, O. R

[1]Department of Civil Engineering, University of Benin, Benin City, Edo Sate
[2]Delta State Ministry Works, Asaba, Delta State
[3]Department of Civil Engineering, Benson Idahosa University, Edo State
[4,5]Department of Civil Engineering, Ambrose Alli University, Ekpoma, Edo Sate .

ARTICLE INFO

Article history:
Received xxxxx
Revised xxxxx
Accepted xxxxx
Available online xxxxx

Keywords:
Hydraulic pipe networks,
Hardy Cross method,
Darcy–Weisbach equation,
Hazen–Williams equation

ABSTRACT

The hydraulic analysis of looped pipe networks is a core yet computationally intensive task in civil and water resources engineering. Although the Hardy Cross method is valued for its simplicity, manual application becomes inefficient and error-prone for complex multi-loop systems. This study presents a modular Python-based framework for automated hydraulic flow balancing using an enhanced Hardy Cross algorithm. The model incorporates both Darcy–Weisbach and Hazen–Williams equations, with friction factors computed dynamically via the Swamee–Jain equation, removing the need for predefined resistance coefficients. It systematically handles multi-loop interactions and shared pipe corrections through an iterative convergence scheme with user-defined tolerance. Validation was performed using four benchmark pipe network cases and comparison with established solutions. Results show strong agreement, with coefficients of determination (R^2) above 0.999 and minimal errors. The framework demonstrates improved efficiency, scalability, and flexibility, making it valuable for engineering applications and academic use.

INTRODUCTION

The analysis of flow distribution in looped pipe networks is a fundamental problem in hydraulic engineering, underpinning the design and operation of water distribution systems, gas pipelines, and other fluid transport infrastructures. These systems are governed by nonlinear relationships derived from conservation of mass at nodes and conservation of energy around loops, making their solution computationally demanding [1,2]. The Hardy Cross method, first introduced in 1936, remains one of the earliest and most widely taught iterative techniques for solving flow distribution problems in looped networks. The method applies successive flow corrections to satisfy energy conservation around loops while maintaining nodal continuity [3].

*Corresponding author: MOSES, G. O.
E-mail address: ukattahprecious21@gmail.com
https://doi.org/10.60787/tnamp.v24.686

Despite its simplicity and transparency, the method becomes inefficient and prone to cumulative error as network complexity increases. Recent studies have revisited the Hardy Cross method with computational enhancements. For example, object-oriented and graph-based implementations have been shown to significantly improve convergence efficiency and computational performance when compared to traditional approaches [4]. Similarly, Python-based and numerical implementations have demonstrated that automation can effectively eliminate manual computational errors while maintaining high accuracy in multi-loop systems [5].

In parallel, modern hydraulic solvers based on Newton–Raphson and gradient methods have gained widespread adoption due to their faster convergence and robustness. Comparative studies indicate that while Hardy Cross remains accurate, Newton-based approaches generally achieve faster convergence in complex networks [6]. Furthermore, EPANET and similar tools implement global gradient algorithms that provide efficient and scalable solutions for real-world water distribution systems [7,8]. Despite these advancements, the Hardy Cross method continues to play a significant role in engineering education and preliminary design due to its conceptual clarity and ease of implementation. Recent applications demonstrate its continued relevance in modeling real-world water distribution systems and validating computational hydraulic tools [9,10]. However, existing computational implementations of the Hardy Cross method exhibit several limitations. Many models rely on pre-defined resistance coefficients, are restricted to a single head-loss formulation, or inadequately handle multi-loop interactions and shared pipes. Additionally, validation in many studies is limited to small benchmark problems without comparison to established hydraulic solvers or real-world datasets [5,9].

This study addresses these limitations by developing a modular and extensible Python-based computational framework that integrates both Darcy–Weisbach and Hazen–Williams head-loss formulations, dynamically evaluates friction factors using the Swamee–Jain equation, efficiently handles shared pipes in multi-loop systems, and incorporates structured data input through excel sheets alongside automated convergence control mechanisms. The developed model is validated using benchmark pipe network problems and evaluated using multiple statistical performance metrics. By bridging classical hydraulic theory with modern computational tools, this study contributes to the advancement of efficient, scalable, and user-friendly methods for hydraulic network analysis in both engineering practice and education.

2.0 COMPUTATIONAL FRAMEWORK

The computational model was developed using Python due to its flexibility, readability, and extensive scientific computing libraries. The framework is structured into modular components, including data input, hydraulic computation, iterative solver, and output visualization. Input data are provided through structured spreadsheet formats, enabling user-friendly interaction and scalability for large networks.

2.1 Governing Equations

From a theoretical standpoint, the analysis of hydraulic flow in looped networks is governed by two foundational principles, mass conservation and energy conservation. The principle of mass conservation (also known as the continuity equation) states that the total inflow to a junction must equal the total outflow plus any demand or withdrawal at that junction. Mathematically, this can be expressed as:

$$\sum Q_{in} - \sum Q_{out} = D_j \quad (1)$$

where Q represents pipe flow and Dj is the demand at node j. This ensures that water is neither lost nor created within the network. In this Study, it is assumed that there is no withdrawal or demand at junctions:

$$\sum Q_{in} = \sum Q_{out} \tag{2}$$

The principle of energy conservation is applied around closed loops within the network and dictates that the algebraic sum of head losses (or gains) around any closed path is zero. This can be expressed as:

$$\sum h_f = 0 \tag{3}$$

2.2. The Hazen-Williams Equation: Used predominantly for water distribution systems, the head loss h_L is calculated as:

$$h_f = 10.667 \, . \, \frac{L}{C^{1.852} D^{4.87}} \, . Q^{1.852} \tag{4}$$

2.3. The Darcy-Weisbach Equation: For general fluid flow analysis, the program utilizes:

$$h_f = f \, . \, \frac{L}{D} \, . \, \frac{V^2}{2g} \tag{5}$$

To ensure the program remains fully automated, the friction factor f is not manually entered but is computed using the Swamee-Jain equation, which approximates the Colebrook-White equation for turbulent flow:

$$\boldsymbol{f = 0.25 \left[\log_{10} \left(\frac{\epsilon/D}{3.7} + \frac{5.74}{R_e^{0.9}} \right) \right]^{-2}} \tag{6}$$

2.4. Algorithm Flow Chart for Balancing Flow

The method of balancing flow begins with an initial estimation of flow rates that satisfies the continuity of flow at each junction. It then iteratively adjusts these flows until the continuity of potential is also achieved across every loop in the system. The Hardy Cross algorithm implemented in this study follows an iterative correction approach. Initial flow values are assigned such that continuity is satisfied at all nodes. For each loop, a correction term is computed as:

$$\Delta Q = -\frac{\sum h_f}{\sum \left(\frac{dh_f}{dQ} \right)} \tag{7}$$

This correction is applied to each pipe in the loop, considering flow direction. The process is repeated iteratively until the correction term falls below a predefined tolerance. The flow chat of methodology is presented in Figure 1.

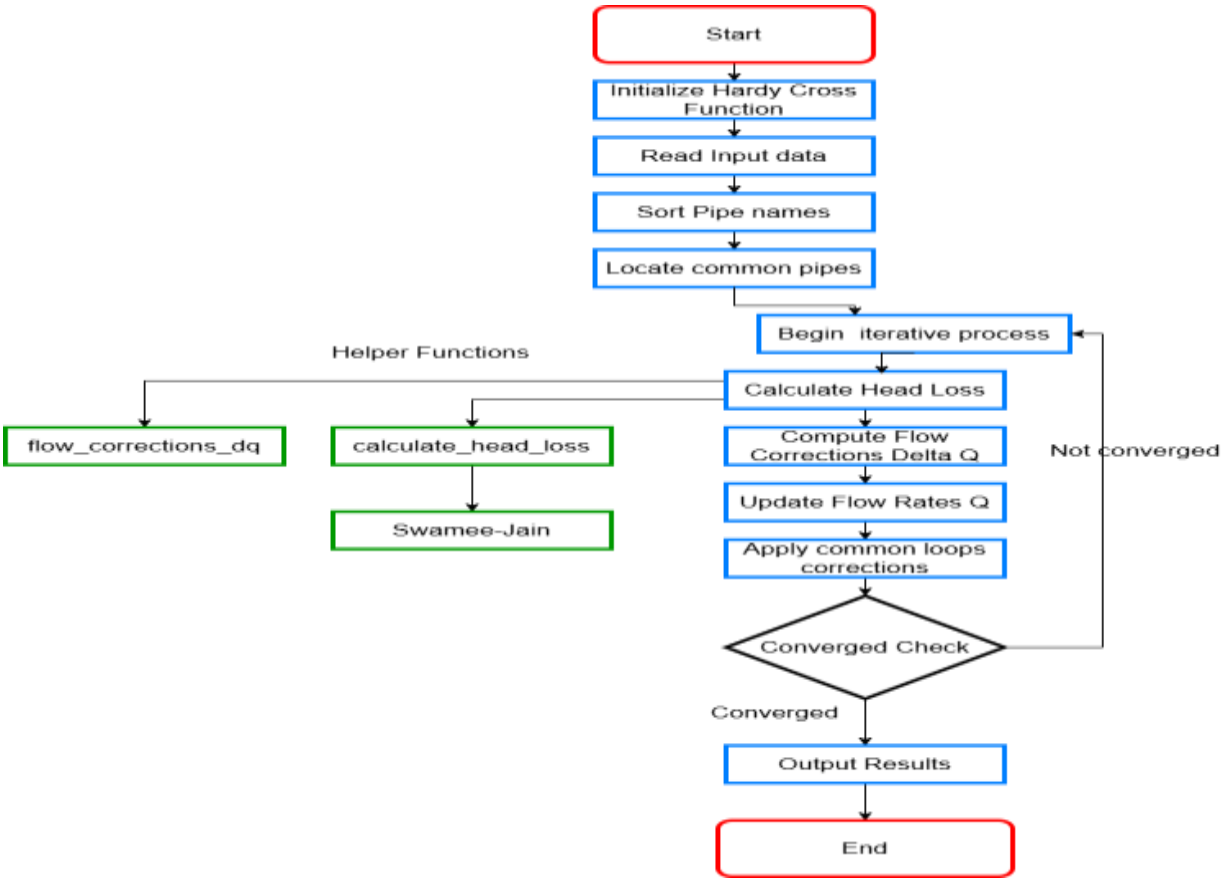

Figure 1: Program Flow Chart.

2.5 Program Validation Methods

Verifying the performance of the model requires conducting statistical analysis. The analysis schemes include; the coefficient of determination (R^2), the root mean square error (RMSE), and the mean bias error (MBE). RMSE measures the observations. The smaller the RMSE, the more precise is the approximation. MBE is a representation of the mean deviation of the predicted values from the respective observations. The smaller the MBE, the more superior is the model performance [11]. The expressions for the aforementioned statistical parameters are:

$$R^2 = 1 - \frac{\sum\left(Q_{ref} - Q_{prog}\right)^2}{\sum\left(Q_{ref} - \bar{Q}_{ref}\right)^2} \quad (8)$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=N}^{N}(Q_{prog} - Q_{ref})^2} \quad (9)$$

$$\text{MBE} = \frac{1}{N}\sum_{k=1}^{N}(Q_{prog} - Q_{ref}) \quad (10)$$

$$\text{MAE} = \frac{1}{N}\sum_{k=1}^{N}|Q_{prog} - Q_{ref}| \quad (11)$$

Where, Q_{prog} represents the flow rate calculated from the program and Q_{ref} represents the reference flow rate [9].

RESULTS AND DISCUSSION

This section presents the results obtained from the computer-programmed implementation of the Hardy Cross method applied to looped pipe networks. The convergence and the accuracy of the flow corrections calculated after several iterations between the various network loops and pipe configurations are analyzed. The findings give a clear assessment on the accuracy and computational ease of the developed Python-based program, which has revealed the reliability of the model in its capacity to achieve hydraulic flow balance and also its comparability with benchmark solutions that are documented in the existing literature.

3.1 Network Configuration Diagrams and Program Outputs

In this section, the physical configurations of the four benchmark networks as shown in (Figures 2, 3 and 4) are presented alongside the direct computational results captured from the Python execution environment. Figure 2, gives the first benchmark model validation set-up. This network constitutes a simple looped network with a few pipes, which allows it to be used to test the algorithm first.

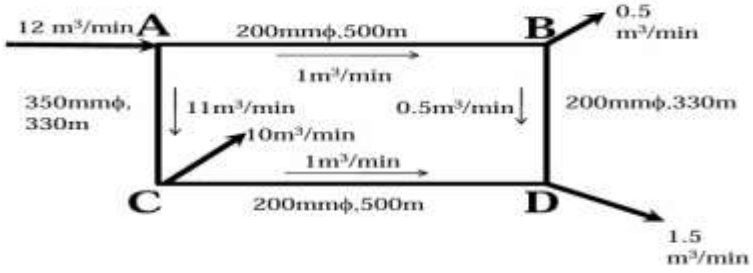

Figure 2: Hydraulic Network-1.

The comparison between reference flow values and program-computed results for Network 1 is presented in Table 1.

Table 1: Comparison of Reference and Program-Computed Pipe final Flow Rates for Network-1

Pipe ID	Reference Flow (m^3 / s)	Program Flow (m^3 / s)
AB	0.0282	0.0276
BD	0.0199	0.0193
AC	0.1718	0.1724
CD	0.0051	0.0057

As shown in Table 1, the computed flow values closely match the reference data, with only negligible deviations observed across all pipes. This indicates that the developed program accurately reproduces expected hydraulic behavior even in its initial validation stage.

The statistical performance of the model for Network 1 is summarized in Table 2.

Table 2: Statistical Metrics of Network-1

Metric	Value	Interpretation
RMSE	0.000	Perfect Agreement between program and reference results
MAE	0.000	Perfect Agreement
MBE	0.000	No Overall bias; program and reference results balance out
R^2	0.999	Very Strong Agreement.

A more complex looped network configuration is presented in Figure 3, consisting of multiple interconnected loops and shared pipes.

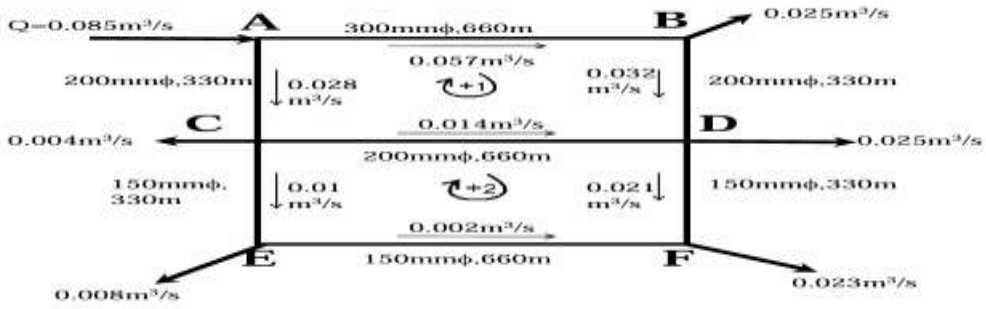

Figure 3: Hydraulic Network-2

The computed and reference flow rates for Network 2 are compared in Table 3.

Table 3: Comparison of Reference and Program-Computed Pipe final Flow Rates for Network-2

Pipe ID	Reference Flow (m^3/s)	Program Flow (m^3/s)
AB	0.05392	0.0524
BD	0.0289	0.0274
AC	0.0311	0.0326
CD	0.0121	0.0135
DF	0.0159	0.0159
CE	0.0150	0.0151
EF	0.0070	0.0071

Table 4: Statistical Metrics of Network-2

Metric	Value	Interpretation
RMSE	0.001	Perfect Agreement between program and reference results
MAE	0.001	Very Strong Agreement
MBE	0.000	No Overall bias; program and reference results balance out
R^2	0.999	Very Strong Agreement.

The third benchmark network, shown in Figure 4, represents a significantly larger and more complex system with multiple nodes and loops.

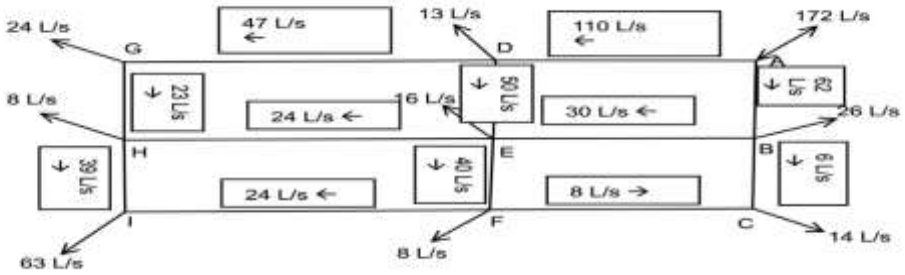

Figure 4: Hydraulic Network-3.

A comparison of computed and reference flow values for Network 3 is presented in Table 5.

Table 5: Comparison of Reference and Program-Computed Pipe final Flow Rates for Network-3

Pipe ID	Reference Flow (m^3/s) $*\times 10^{-3}$	Program Flow (m^3/s) $*\times 10^{-3}$
DG	70.6	70.7
EH	7.8	7.9
GH	46.6	46.7
DE	14.8	14.8
AD	98.3	98.5
AB	73.7	73.5
BE	16.9	16.8
EF	7.8	7.8
FI	16.6	16.5
HI	46.4	46.5
BC	30.8	30.7
CF	16.8	16.7

The statistical performance metrics for Network 3 are summarized in Table 6.

Table 6: Statistical Metrics of Network-3

Metric	Value	Interpretation
RMSE	0.000	Perfect Agreement between program and reference results
MAE	0.000	Perfect Agreement
MBE	0.000	No Overall bias; program and reference results balance out
R^2	0.999	Very Strong Agreement.

The close values of errors and exceptionally high value of the coefficient of determination further confirm the reliability of the program. This evidence shows that the algorithm is still stable and accurate even when used on a large scale.

The final benchmark configuration, shown in Figure 5, incorporates the Darcy–Weisbach formulation with dynamically computed friction factors.

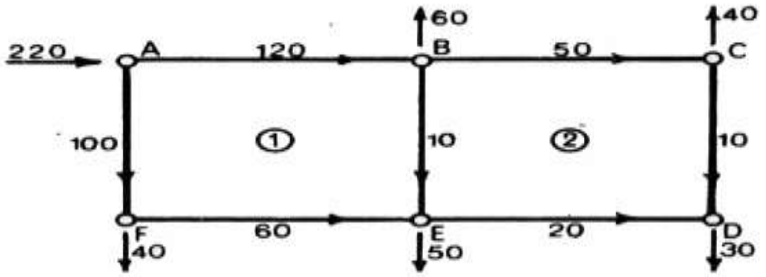

Figure 5: Hydraulic Network-4

The comparison of computed and reference flow values for Network 4 is presented in Table 7

Table 7: Comparison of Reference and Program-Computed Pipe final Flow Rates for Network-4

Pipe ID	Reference Flow (m^3/s) $*\times 10^{-3}$	Program Flow (m^3/s) $*\times 10^{-3}$
AB	131.99	131.30
BE	26.23	25.71
EF	48.01	48.69
AF	88.01	88.69
BC	45.76	45.61
CD	5.76	5.61
DE	24.24	24.39

The statistical evaluation for Network 4 is summarized in Table 8.

Table 8: Statistical Metrics of Network-4

Metric	Value	Interpretation
RMSE	0.001	Perfect Agreement between program and reference results
MAE	0.001	Very Strong Agreement
MBE	0.000	No Overall bias; program and reference results balance out
R^2	0.999	Very Strong Agreement.

The values of RMSE and MAE are very low, albeit marginally greater than the cases before, which proves the accuracy of the model. The large R 2 value means that inclusion of the Darcy Weisbach

equation does not affect the performance of the model, but on the contrary, adds physical realism to the model.

The statistical evaluation of the developed model indicates a very strong agreement between the computed flow values and the benchmark solutions across all network configurations. The coefficient of determination ($R^2 \approx 0.999$) demonstrates an almost perfect correlation, while the near-zero Mean Bias Error (MBE) indicates the absence of systematic overestimation or underestimation in the predicted flows. These results confirm the high level of numerical accuracy and consistency achieved by the implemented algorithm. The application of the Darcy–Weisbach formulation in Network 4 further highlights the robustness of the model. Unlike conventional manual implementations that typically assume constant friction factors, the present model incorporates a dynamic evaluation of the friction factor using the Swamee–Jain equation. This enhances the physical realism of the simulation by accounting for variations in flow conditions. The minor deviations observed in Network 4 (RMSE and MAE ≈ 0.001) can be attributed to differences in convergence tolerance and the approximations inherent in friction factor estimation, which are within acceptable engineering limits. In addition, the model effectively addresses the complexity associated with shared pipes in multi-loop systems. Through its computational structure, corrections from adjacent loops are applied in a consistent and coordinated manner, ensuring accurate flow redistribution. This represents a significant improvement over manual Hardy Cross procedures, where such interactions are often difficult to track and prone to error. Overall, the results demonstrate that the developed framework provides a reliable and efficient tool for hydraulic analysis of looped pipe networks.

CONCLUSION

This study presented the development of a Python-based computational framework for hydraulic flow balancing in looped pipe networks using the Hardy Cross method. The model successfully automates the iterative procedure traditionally performed manually, thereby enhancing computational efficiency and reducing the likelihood of human error. Validation of the model using four benchmark network configurations demonstrated a high level of accuracy, with a coefficient of determination (R^2) of approximately 0.999 and negligible bias, confirming the reliability of the approach. The integration of both Darcy–Weisbach and Hazen–Williams formulations, along with the dynamic computation of friction factors using the Swamee–Jain equation, improves the flexibility and physical relevance of the model. Furthermore, the developed framework effectively handles multi-loop interactions and shared pipe conditions, which are often challenging in manual computations. This makes the tool suitable for both educational purposes and preliminary engineering analysis, where transparency and ease of implementation are essential. Despite these strengths, the model is limited to steady-state conditions and idealized network assumptions. Future work should focus on extending the framework to incorporate real-world complexities such as varying nodal demands, multiple reservoirs, pumping systems, and transient flow conditions. The development of a graphical user interface (GUI) is also recommended to enhance usability for non-programmers. In conclusion, the study demonstrates that classical hydraulic methods such as the Hardy Cross technique can be effectively integrated with modern computational tools to provide accurate, efficient, and accessible solutions for pipe network analysis.

REFERENCES

[1] White, F. M. (2011). Fluid Mechanics (7th ed.). McGraw-Hill.

[2] Round, G. F. (2019). Analysis of flow in pipe networks. Canadian Journal of Civil Engineering, 10(1), 19–28.

[3] Cross, H. (1936). Analysis of flow in networks of conduits or conductors. Bulletin No. 286, University of Illinois Engineering Experiment Station, III

[4] Jha, K., & Mishra, M. K. (2020). Object-oriented integrated algorithms for efficient water pipe network analysis using modified Hardy Cross method. Journal of Computational Design and Engineering, 7(1), 56–64.

[5] Dumka, P., Samaiya, N., Gandhi, S., & Mishra, D. R. (2023). Modelling of Hardy Cross method for pipe networks. SSRG International Journal of Mechanical Engineering, 10(2), 1–8.

[6] Ainullah, M., Lutfullah, S., & Mujeebullah, M. (2024). Comparison of Newton–Raphson and Hardy Cross methods for looped water supply networks. Journal of Natural Science Review, 2(2).

[7] Todini, E., & Rossman, L. A. (2012). Unified framework for water distribution network analysis. Journal of Hydraulic Engineering, 139(5), 511–526.

[8] Sheikh, M. R., & Coulibaly, P. (2024). Review of recent developments in hydrologic forecasting techniques. Water, 16(2), 301.

[9] Obura, D., Kimera, D., & Khaldi, A. (2022). A Hardy Cross approach for hydraulic modelling of water pipe networks. East African Journal of Engineering, 5(1), 28–56.

[10] Silvia, C. S., Safriani, M., Ikhsan, M., & Rauza, U. (2022). Analysis of water distribution systems using the Hardy Cross method. IOP Conference Series: Earth and Environmental Science, 955(1), 012029.

[11] Al-Shamisi, M.H., Assi, A.H. and Hejase, H.A.N., 2011. *Using MATLAB to develop artificial neural network models for predicting global solar radiation in Al Ain City – UAE*. In: InTech.