

Article

Hybrid Models for Solving the Colebrook–White Equation Using Artificial Neural Networks

Muhammad Cahyono 

Faculty of Civil and Environmental Engineering, Bandung Institute of Technology, Jalan Ganesa No. 10, Bandung 40132, Indonesia; cahyono@itb.ac.id

Abstract: Abstract This study proposes hybrid models to solve the Colebrook–White equation by combining explicit equations available in the literature to solve the Colebrook–White equation with an error function. The hybrid model is in the form of $f_H = f_o - e_A$. f_H is the friction factor value f predicted by the hybrid model, f_o is the value of f calculated using several explicit formulas for the Colebrook–White equation, and e_A is the error function determined using the neural network procedures. The hybrid equation consists of a series of hyperbolic tangent functions whose number corresponds to the number of neurons in the hidden layer. The simulation results showed that the hybrid models using five hyperbolic tangent functions could produce reasonable predictions of friction factors, with the maximum absolute relative error (MAXRE) around one tenth, or ten times lower than that produced by the corresponding existing formula. The simplified hybrid models are also given using four and three tangent hyperbolic functions. These simplified models still provide accurate results with MAXRE of less than 0.1%.

Keywords: Colebrook–White equation; friction factor; explicit approximation; artificial neural networks; hybrid model



Citation: Cahyono, M. Hybrid Models for Solving the Colebrook–White Equation Using Artificial Neural Networks. *Fluids* **2022**, *7*, 211. <https://doi.org/10.3390/fluids7070211>

Academic Editor: Markus Klein

Received: 30 May 2022

Accepted: 17 June 2022

Published: 21 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The friction factor f is an important parameter in flow analysis in pipes and open channels because it is related to head or energy loss. The loss of head ΔH in the pipe over a distance L along the flow direction is given by the Darcy Equation (1),

$$\Delta H = f \frac{L}{D} \frac{V^2}{2g} \quad (1)$$

where f is the Darcy–Weisbach friction factor, L is the length of pipe, D is the hydraulic diameter or equivalent pipe diameter, V is the mean velocity, and g is gravity acceleration. An open channel can be considered as full flow in the pipe split in half [1], and the Darcy–Weisbach equation originally developed for pipe flow can be adopted for an open channel [1–4]. Thus, the friction factor in open channels could be related to the flow conditions in the pipe [1–4]. For uniform flow in an open channel, Equation (1) could be converted into Equation (2) by substituting $D = 4R$ [1–6], where $R = A/P =$ hydraulic radius, $A =$ area of pipe cross-section $= \pi D^2/4$, $P = D =$ wet perimeters of pipe, and $S_f = \Delta H/L =$ energy slope. From Equation (2), we obtain Equation (3).

$$S_f = \frac{f V^2}{2g4R} \quad (2)$$

$$V = \sqrt{\frac{8g}{f}} \sqrt{R S_f} \quad (3)$$

The value of f for flow in a pipe depends on relative roughness ϵ/D as a ratio of the sand grain size to the pipe diameter and Reynold numbers, R_e , defined by

$$R_e = \frac{\rho DV}{\mu} \tag{4}$$

$$\text{or } R_e = \frac{\rho 4RV}{\mu} \tag{5}$$

where ρ = fluid density and μ is the dynamic viscosity. The relationship between f to ϵ/D and R_e also depends on the flow conditions, laminar or turbulent. For laminar flow with $R_e < 2100$, the f value can be computed from Equation (6) [1–9]

$$f = \frac{64}{R_e} \tag{6}$$

There are three types of turbulent flows: smooth, transitional, and fully rough [1–9]. They are distinguished according to the dimensionless Reynolds number R_* based on roughness height ϵ and the shear velocity, u_* defined as [1–9]

$$u_* = \sqrt{\tau_b/\rho} = \sqrt{gRS_f} \tag{7}$$

$$R_* = \frac{u_*\epsilon}{\vartheta} \tag{8}$$

where τ_b is boundary shear stress and ϑ is the kinematic viscosity of the fluid. The flow is hydraulically smooth if $R_* < 4$, transitional if $4 < R_* < 60$ for pipe flow, and $4 < R_* < 100$ for channel flow, respectively, and fully rough if $R_* > 60$ for pipe flow and $R_* > 100$ for flow in open channel, respectively [1–9]. When the flow is hydraulically smooth and $R_e < 100,000$, the f value has a relatively small roughness effect. Therefore, the f value is directly related to R_e [1–9] given by the Blasius formula defined by Equation (9).

$$f = \frac{0.316}{R_e^{1/4}} \tag{9}$$

When $R_e > 100,000$ and the boundaries are still hydraulically smooth, the f value is given by the Karman–Prandtl Equation (10) [1,6] as

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{2.51}{R_e\sqrt{f}}\right) \tag{10}$$

For fully roughly turbulent flow, f is more influenced by roughness or dependent on ϵ/D [1,2,6] and f can be computed by Karman–Prandtl ‘s Equation (11) [1–9].

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{\epsilon}{3.7D}\right) \tag{11}$$

For transitional flow, the friction factor f may be estimated from the Colebrook–White formula defined by Equations (12) and (13) [1–10].

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{\epsilon}{3.71D} + \frac{2.51}{R_e\sqrt{f}}\right) \tag{12}$$

$$\text{or } \frac{1}{\sqrt{f}} = -2\log\left(\frac{\epsilon}{14.84R} + \frac{2.51}{R_e\sqrt{f}}\right) \tag{13}$$

Equations (12) and (13) are the two forms of the Colebrook–White equation, one for full flow in pipes and another for partially full in pipes and open channels but these are essentially the same since $D = 4R$ [3].

Equations (12) and (13) are implicit concerning f , which must be determined by trial-and-error method or by using an iterative procedure, such as the Newton–Raphson Method or the Secant Method [11,12]. Several explicit formulas to solve the Colebrook–White Equation (12) have been proposed to avoid the iteration procedure. The first attempt to make the friction factor calculation easier was made by Moody [13] by producing a $f - R_e$ plot based on Equation (12), generally called Moody’s diagram. Moody also proposed an explicit formula for f . After that, several explicit equations for solving the Colebrook–White Equation (12) have been proposed. The study and attempt to find an explicit form of the Colebrook–White equation is an interesting topic, as can be seen from many studies available relating to the approximation of the Colebrook–White equation. The comparison and performance of these equations were examined by some authors, including Brkic [14], Zeghadnia et al. [15], Plascendia et al. [16], and Perez Pupo et al. [17], compared 49 explicit equations, ranging from the old equation of Moody [13] to relatively new and complex equations. Moreover, the study and development of the approximation of the Colebrook–White equation are still going on today. Some of the latest studies on this can be seen in several kinds of literature [18–23].

Another approach to determine the value of f is to use an artificial neural network (ANN), widely applied in engineering and science. The ANN approach was inspired by the structure and workings of biological neural networks. It is used to find the correlation between large, complex, and non-linear independent and dependent variables without knowing the physical process defining the relationship between these parameters [24–26]. Therefore, ANN could be considered a black-box model representing the processes modeled by networks and weights. Its structure consists of the input, hidden, and output layers with several neurons. In the ANN networks, information is transferred from one neuron to another using a specific transform function.

The ANN method starts from the parameters in the input layer, where the neurons receive input from the outside environment. The inputs received are usually independent parameters that describe a problem under study. Their signals are transferred to neurons in the next hidden layer and transmitted through a specific transfer function. The output signal in neurons at the hidden layer is then transferred to the neurons in the output layer that sends it to the user. The application of ANN consists of training, validation, and testing. The training process aims to obtain the weighting coefficient through optimization to produce minimal prediction errors. The ANN model needs to be tested through validation and testing processes. The data for the ANN model generally consists of 70% training, 15% validation, and 15% testing.

The commonly used and simple neural network model is the multi-layered perceptron (MLP), consisting of three main layers: input, hidden, and output. Cahyono [27] proposed an explicit model of ANN-based on the MLP model for the output layer consisting of a single node. The model was applied to develop equations for estimating the setting velocity of natural sediment. The explicit ANN equation consists of a series of hyperbolic tangent functions with the sequence number according to the number of nodes in the hidden layer. Each hyperbolic tangent part contains coefficients whose number is equal to the input parameters. For example, the explicit ANN with two input parameters is defined by Equation (14).

$$Y_{ANN} = \sum_{k=1}^N E_k \tanh(a_k \bar{X}_1 + b_k \bar{X}_2 + c_k) + F \quad (14)$$

where Y_{ANN} is the predicted output value; \bar{X}_1 and \bar{X}_2 are the normalized input values, and E_k and F are coefficients that depend on the weights a_k , b_k , and c_k , where $k = 1, 2, \dots, N$ with N = number of nodes in the hidden layer. These weight coefficients were obtained

through a training process. The MATLAB software can determine the coefficient and the initial value for the optimization to obtain the best coefficient.

This study’s main objective is to develop explicit models of the Colebrook–White equation. Several explicit formulas available in the literature for solving the Colebrook–White equation are used to create the models. The developed models, called hybrid, are built by developing an error equation, which is the formula expressing the difference between the friction factor computed by the Colebrook–White equation and that obtained with the explicit formula. The error equation is derived using the ANN Equation (14). Thus, with this method, the hybrid models can improve the accuracy of the explicit formula. The study considers three hybrid models with differences in the number of neurons in the hidden layer.

The following section discusses the methodology of developing hybrid models; the creation of data points for training, validation, and testing; and the analysis of the simulation results for hybrid models, discussion, and conclusions.

2. Materials and Methods

2.1. Development of Hybrid Formula

Hybrid models are developed using a combination of explicit formulas available in the literature to solve the Colebrook–White equation and an error function. Consider Equation (15) below

$$f_E = f_o + e_0 \tag{15}$$

where f_E and f_o are the friction factor computed by the Colebrook–White Equation (12) and that obtained by the explicit formula, and e_0 is an error that is the difference between f_E and f_o , i.e., $e_0 = f_E - f_o$. For every pair of R_e and ϵ/D , one can compute f_E , f_o , and e_0 . Therefore, there is a correlation between parameters R_e and ϵ/D with e_0 . In this study, an equation to approximate e_0 as a function of R_e and ϵ/D , i.e., $e_A = f(R_e, \epsilon/D)$, is developed by using the ANN model. The main objective of the contribution is to model e_A accurately so that $e_0 \approx e_A$ by minimizing the difference between the e_0 and e_A . Thus, the hybrid model for the solution of the Colebrook–White Equation (12) can be defined by Equation (16) below:

$$f_H = f_o + e_A \tag{16}$$

where f_H is the friction factor computed by the hybrid model. This study applied the ANN method with the multi-layered perceptron (MLP) architecture consisting of three main layers: the input layer, the hidden layer, and the output layer. The input layer consists of two neurons that receive two input parameters, X_1 and X_2 , relating to R_e and ϵ/D . These input parameters are in a logarithmic transformation of R_e and ϵ/D , i.e., $X_1 = \log(R_e)$ and $X_2 = \log(\epsilon/D)$. Choosing these types of input parameters is in line with Sablani et al. [28], showing that the accuracy of the ANN model is better when the input parameters used are logarithmic transformations of R_e and ϵ/D .

The hidden layer has several neurons depending on the ANN model. The output layer has single neurons relating to e_A . However, in this model, the output value is e_A , which is the value e_A multiplied by 10^6 . The e_A value needs to be multiplied by the number 10^6 to avoid significant error calculations dealing with a small value e_A . Thus, applying Equation (14) for $e_A = f(R_e, \epsilon/D)$, the explicit ANN model for the error function $e_A = f(R_e, \epsilon/D)$, can be expressed in a series of hyperbolic tangent functions as stated by Equation (17) below.

$$e_A(R_e, \epsilon/D) = 10^{-6} \left[\left(\sum_{k=1}^N E_k \tanh(a_k \zeta + b_k \eta + c_k) \right) + F \right] \tag{17}$$

where ζ and η are normalized input parameters $X_1 = \log(R_e)$ and $X_2 = \log(\epsilon/D)$, respectively, and $a, b, c, E,$ and F are the coefficients of the function. Substituting Equation (17) into Equation (16), the hybrid model function becomes

$$f_H = f_o + 10^{-6} \left[\left(\sum_{k=1}^N E_k \tanh(a_k \zeta + b_k \eta + c_k) \right) + F \right] \tag{18}$$

The coefficients $a, b, c, E,$ and F depend on the value of the ANN weighting coefficients that are obtained through optimization in the training process. The optimization is carried out by using the following objective function

$$\min E_r = \min \sum_{i=1}^M (e_{0,i} - e_{A,i})^2 \tag{19}$$

where $e_0 = f_E - f_o$ is the error value defined by Equation (15), and e_A = the error value predicted by the ANN model defined by Equation (17).

2.2. Existing Equation Considered

The five explicit formulas reviewed in this study were selected based on their simplicity in mathematical expression and rough accuracy, namely, the maximum absolute relative error value of less than 1.0% in line with the study by Pérez Pupo et al. [17]. The considered explicit formulas are shown below:

1. Formula (1) of Chen [29],

$$f_{o,Chen} = \left[-2 \log \left[\frac{\epsilon}{3.7065D} - \frac{5.0452}{R_e} \log \left(\frac{1}{2.8257} \left(\frac{\epsilon}{D} \right)^{1.1098} + \frac{5.8506}{R_e^{0.8981}} \right) \right] \right]^{-2} \tag{20}$$

2. Formula (2) of Schorle et al. [30],

$$f_{o,S} = \left[-2 \log \left[\frac{\epsilon}{3.7D} - \frac{5.02}{R_e} \log \left(\frac{\epsilon}{3.7D} + \frac{14.5}{R_e} \right) \right] \right]^{-2} \tag{21}$$

3. Formula (3) of Bar and White [31],

$$f_{o,BW} = \left[-2 \log \left\{ \frac{\epsilon}{3.7D} + \frac{4.518 \log \left(\frac{R_e}{7} \right)}{R_e \left[1 + \frac{R_e^{0.52}}{29} \left(\frac{\epsilon}{D} \right)^{0.7} \right]} \right\} \right]^{-2} \tag{22}$$

4. Formula (4) of Sousa et al. [32],

$$f_{o,So} = \left[-2 \log \left[\frac{\epsilon}{3.7D} - \frac{5.16}{R_e} \log \left(\frac{\epsilon}{3.7D} + \frac{5.09}{R_e^{0.87}} \right) \right] \right]^{-2} \tag{23}$$

5. Formula (5) of Ofor and Alabi [33],

$$f_{o,of} = \left[-2 \log \left\{ \frac{\epsilon}{3.71D} - \frac{1.975}{R_e} \left(\ln \left(\left(\frac{\epsilon}{3.93D} \right)^{1.092} + \left(\frac{7.627}{R_e + 395.9} \right) \right) \right) \right\} \right]^{-2} \tag{24}$$

2.3. Statistical Measure

The performance of the hybrid model is assessed using the following error measures:

1. The coefficient of determination, R^2 , of the linear regression line between f_H represents the predicted friction factor by the hybrid model Equation (18) and the desired output friction factor f_E obtained from iteration Equation (12).

2. The mean absolute relative error, MRE , is defined by:

$$MRE = \frac{1}{M} \sum_{i=1}^M \left| \frac{f_{E,i} - f_{H,i}}{f_{E,i}} \right| \times 100 \quad (25)$$

where $|(f_E - f_H)/f_E| = RE$ is the absolute relative error, and M is the amount of data.

3. The maximum absolute relative error, $MAXRE$, is defined by:

$$MAXRE = \text{Max} \left| \frac{f_{E,i} - f_{H,i}}{f_{E,i}} \right| \times 100 \%, \quad i = 1, 2, \dots, M \quad (26)$$

2.4. Data Generation

The data points for the training, validation, and testing of the ANN model are generated by applying an iterative procedure using the Secant method [11,12] on the Colebrook–White Equation (12). The data points consist of a combination of R_e ranging from 2×10^3 to 2×10^9 and ε/D values ranging from 2.5×10^{-7} to 0.05. The first group of data points for the training process comprises 30,351 points resulting from the solution of Equation (12) with system $\log(R_e)(i) \times \log(\varepsilon/D)(j) = 151 \times 201$ data grids with uniform intervals in each $\log(R_e)$ and $\log(\varepsilon/D)$. The second group of data points for validation were from 301 R_e and 301 ε/D values, resulting in 90,601 data points. The validation data points have uniform intervals in R_e and ε/D . The third data group for testing comprises 200,901 data points made from 401 $\log(R_e)$ and 501 $\log(\varepsilon/D)$ values using uniform intervals as in the training data. The total data points obtained for training, validation, and testing are 321,853 points. The number of data points is enlarged for the resulting robust model for the range of R_e and ε/D under consideration. Generally, the development of the ANN model uses a more significant proportion of data for training than validation data. In contrast, the data for validation and testing in this study is greater than the training data to ensure the robustness of the model obtained. Each data point has R_e and ε/D and the associated f value, i.e., $(R_e, \varepsilon/D, f)$. Solving Equation (12) results in the value of f for each pair R_e and ε/D through an iterative process [11,12]. By using the initial value $f = 0.01$, $\varepsilon_1 =$ the error tolerance absolute relative error for f , $\varepsilon_1 < 10^{-8}$; and $\varepsilon_2 =$ the absolute value of implicit Equation (12), $\varepsilon_2 < 10^{-8}$. To get a solution f with these tolerances requires the iteration of Equation (12) between five and eight times, with an average iteration of seven.

3. Results and Discussion

The training process using the nntool facility in MATLAB software produced coefficients a, b, c, E , and F , defined in Equation (18). An Excel program was developed based on the ANN algorithm described using MS excel software to obtain the explicit e_A defined by Equation (17). This Excel program refines the ANN model by further optimizing computations. The optimization with MATLAB uses the Levenberg–Marquardt method, while the excel program uses the nonlinear generalized reduced gradient (GRG) technique available in the solver engine. Further optimization with the Excel program uses the coefficient values of a, b, c, E , and F obtained from simulation with MATLAB version R2022a licence academic ITB as initial values. Furthermore, optimization with the excel program applies different objective functions. The objective function is to minimize E_r defined by Equation (19), and two other objective functions including minimizing MRE defined by Equation (25) and minimizing $MAXRE$ defined by Equation (26). However, based on the simulation results, the MRE and $MAXRE$ values generated from these three objective functions did not significantly differ in the MRE and the $MAXRE$ values. Therefore, this study provides the coefficients obtained using the objective function of minimizing MRE . The resulting hybrid models using five nodes in the hidden layer, called hybrid model 1, are given in Equations (27)–(31).

- a. The formula hybrid model 1 of Chen [29]

$$f_{H,Chen} = \left[-2\log \left[\frac{\epsilon}{3.7065D} - \frac{5.0452}{R_e} \log \left(\frac{1}{2.8257} \left(\frac{\epsilon}{D} \right)^{1.1098} + \frac{5.8506}{R_e^{0.8981}} \right) \right] \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1 \eta + c_1) + E_2 \tanh(a_2\zeta + b_2 \eta + c_2) + E_3 \tanh(a_3\zeta + b_3 \eta + c_3) + E_4 \tanh(a_4\zeta + b_4 \eta + c_4) + E_5 \tanh(a_5\zeta + b_5 \eta + c_5) + F] \tag{27}$$

where $\zeta = \log(R_e)/3 - 2.1$ and $\eta = 0.37729 \log(\epsilon/D) + 1.49089$, $a_1 = 1.24842$, $a_2 = -1.67471$, $a_3 = 3.21556$, $a_4 = -1.46582$, $a_5 = -6.74339$, $b_1 = 1.95053$, $b_2 = -1.97848$, $b_3 = 1.11762$, $b_4 = -1.93413$, $b_5 = -0.21634$, $c_1 = 0.45646$, $c_2 = -0.19816$, $c_3 = 1.09509$, $c_4 = -0.31873$, $c_5 = -6.91059$, $E_1 = 710.654$, $E_2 = -1039.726$, $E_3 = 28.674$, $E_4 = 1773.675$, $E_5 = 378.630$, and $F = 363.173$.

b. The formula for hybrid model 1 of Schorle et al. [30],

$$f_{H,S} = \left[-2\log \left[\frac{\epsilon}{3.7D} - \frac{5.02}{R_e} \log \left(\frac{\epsilon}{3.7D} + \frac{14.5}{R_e} \right) \right] \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1 \eta + c_1) + E_2 \tanh(a_2\zeta + b_2 \eta + c_2) + E_3 \tanh(a_3\zeta + b_3 \eta + c_3) + E_4 \tanh(a_4\zeta + b_4 \eta + c_4) + E_5 \tanh(a_5\zeta + b_5 \eta + c_5) + F] \tag{28}$$

c. The formula for hybrid model of Bar and White [31],

$$f_{H,BW} = \left[-2\log \left\{ \frac{\epsilon}{3.7D} + \frac{4.518 \log(\frac{R_e}{7})}{R_e \left[1 + \frac{R_e^{0.52}}{29} \left(\frac{\epsilon}{D} \right)^{0.7} \right]} \right\} \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1 \eta + c_1) + E_2 \tanh(a_2\zeta + b_2 \eta + c_2) + E_3 \tanh(a_3\zeta + b_3 \eta + c_3) + E_4 \tanh(a_4\zeta + b_4 \eta + c_4) + E_5 \tanh(a_5\zeta + b_5 \eta + c_5) + F] \tag{29}$$

d. The formula for hybrid model 1 of 4 Sousa et al. [32],

$$f_{H,So} = \left[-2\log \left[\frac{\epsilon}{3.7D} - \frac{5.16}{R_e} \log \left(\frac{\epsilon}{3.7D} + \frac{5.09}{R_e^{0.87}} \right) \right] \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1 \eta + c_1) + E_2 \tanh(a_2\zeta + b_2 \eta + c_2) + E_3 \tanh(a_3\zeta + b_3 \eta + c_3) + E_4 \tanh(a_4\zeta + b_4 \eta + c_4) + E_5 \tanh(a_5\zeta + b_5 \eta + c_5) + F] \tag{30}$$

e. The formula for hybrid model 1 of Offor and Alabi [33],

$$f_{H,of} = \left[-2\log \left\{ \frac{\epsilon}{3.71D} - \frac{1.975}{R_e} \left(\ln \left(\left(\frac{\epsilon}{3.3D} \right)^{1.092} + \left(\frac{7.627}{R_e + 395.9} \right) \right) \right) \right\} \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1 \eta + c_1) + E_2 \tanh(a_2\zeta + b_2 \eta + c_2) + E_3 \tanh(a_3\zeta + b_3 \eta + c_3) + E_4 \tanh(a_4\zeta + b_4 \eta + c_4) + E_5 \tanh(a_5\zeta + b_5 \eta + c_5) + F] \tag{31}$$

The coefficients a , b , c , and E and F for hybrid models Equations (28)–(31) are shown in Table A1. In comparison, the models with more nodes in the hidden layer are also considered, including models 2 and 3 with nodes in the hidden layer of 7 and 10, respectively. The equations and coefficients a , b , c , E , and F for hybrid models 2 and 3 are given in Tables S1–S4 in Supplementary Materials. Table 1 shows statistical measures such as MRE and $MAXRE$ for each considered explicit formula and its corresponding hybrid models. The value of R^2 is not displayed because almost all simulation results have an R^2 value close to 1. Furthermore, it was discovered that the hybrid models could significantly increase the existing formula’s accuracy in reducing the $MAXRE$ value.

Table 1. Statistical measures for considered explicit equations and the hybrid models developed in this study.

Explicit Equation		Hybrid Model	MRE (%)	MAXRE (%)
Equation (26) (Chen [29])	Original		0.117	0.689
	Hybrid	Model 1	0.014	0.090
		Model 2	0.010	0.082
		Model 3	0.004	0.060
Equation (27) (Schorle et al. [30])	Original		0.283	1.889
	Hybrid	Model 1	0.055	0.156
		Model 2	0.017	0.050
		Model 3	0.012	0.045
Equation (28) (Barr and White [31])	Original		0.098	0.942
	Hybrid	Model 1	0.039	0.117
		Model 2	0.026	0.100
		Model 3	0.011	0.067
Equation (29) (Sousa et al. [32])	Original		0.088	0.394
	Hybrid	Model 1	0.010	0.035
		Model 2	0.005	0.026
		Model 3	0.002	0.020
Equation (30) (Offor and Alabi [33])	Original		0.017	0.278
	Hybrid	Model 1	0.007	0.043
		Model 2	0.005	0.035
		Model 3	0.003	0.033

The hybrid models produce the *MAXRE* of about six to forty times lower than those provided by existing formulas, especially in a hybrid model 3 with 10 hidden neurons, as seen in Table 1. Thus, the hybrid model 3 gives very accurate results but is less attractive in terms of computational time because it contains 10 hyperbolic tangent functions. From Table 1, the difference between the *MAXRE* values for Sousa et al. and Offor and Alabi models 1 and that for hybrid models 2 and 3 is relatively small. The results of this simulation indicate that using five nodes in the hidden layer for these two models can produce very satisfactory results.

To evaluate the computational efficiency of the hybrid models, the models are applied to compute data points for validating (90,601 data points). Using Microsoft Developer Studio for Fortran Compiler with a processor specification of Intel® Core™ i7-10510U CPU @ 1.80 GHz–2.30 GHz, and installed memory (RAM) 160 GB, the CPU times for generating the data with hybrid models 1 of Chen, Schorle et al., Bar and White, Sousa et al., and Offor and Alabi are 0.069, 0.051, 0.069, 0.052, and 0.052 s, respectively. In contrast, the corresponding hybrid models 3 require 0.101, 0.087, 0.094, 0.088, and 0.089 s, respectively. The simulation results indicate that the average CPU time required to compute validating data by applying model 3 is 0.092 s, and model 2 is 0.058 s. The simulation with the hybrid model 3 requires the CPU time to be around 160% longer than that of the hybrid model 2. The results also indicate that the hybrid model 2 of Sousa et al. and Offor and Alabi require shorter CPU time than the other hybrid models. As a comparison, the CPU times to compute the validating data using the existing formulas of Chen [29], Schorle et al. [30], Bar and White [31], Sousa et al. [32], and Offor and Alabi [33] are 0.042, 0.027, 0.042, 0.031, and 0.032 s, respectively, while the CPU time for iteration of Equation (12) is 1.567 s.

It can be seen in Table A1 that the coefficients E_3 and E_4 for model 1 of Sousa et al. are relatively small, so their effects on the overall function value are minor. These coefficients, E_3 and E_4 , can be eliminated to get a simplified model. By removing the coefficient E_3 and then applying the optimization technique, new coefficients a, b, c, E , and F are determined, resulting in a simplified Sousa et al. model with four tanh functions. Moreover, removing the coefficients E_3 and E_4 and then applying the same approach resulted in a

simplified formula for hybrid model 1 of Sousa et al. with three tanh functions. The hybrid models of Sousa et al. using four and three hyperbolic tangent functions are given by Equations (32) and (33).

$$f_{H,So} = \left[-2\log \left[\frac{\epsilon}{3.7D} - \frac{5.16}{Re} \left(\frac{\epsilon}{3.7D} + \frac{5.09}{Re^{0.87}} \right) \right] \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1\eta + c_1) + E_2 \tanh(a_2\zeta + b_2\eta + c_2) + E_3 \tanh(a_3\zeta + b_3\eta + c_3) + E_4 \tanh(a_4\zeta + b_4\eta + c_4) + F] \tag{32}$$

$$f_{H,So} = \left[-2\log \left[\frac{\epsilon}{3.7D} - \frac{5.16}{Re} \left(\frac{\epsilon}{3.7D} + \frac{5.09}{Re^{0.87}} \right) \right] \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1\eta + c_1) + E_2 \tanh(a_2\zeta + b_2\eta + c_2) + E_3 \tanh(a_3\zeta + b_3\eta + c_3) + F] \tag{33}$$

where $\zeta = \log(R_e)/3 - 2.1$ and $\eta = 0.37729 \log(\epsilon/D) + 1.49089$. Coefficients a, b, c, E , and F are given in Table 2. The values of *MAXRE* and *MRE* for Equation (32) are 0.079% and 0.032%, around one-fifth or five times lower than those of existing Sousa et al.'s formula, while Equation (33) gives the *MAXRE* and *MRE* of 0.107% and 0.045%, respectively, around one-third or three times lower than those of the existing formula. The computation of validating data with Equations (32) and (33) requires CPU times of 0.046 and 0.048 s, respectively.

Table 2. Coefficients of the simplified hybrid model of Sousa et al. [32].

<i>k</i>	Coefficients of Equation (32)				Coefficients of Equation (33)			
	<i>a_k</i>	<i>b_k</i>	<i>c_k</i>	<i>E_k & F</i>	<i>a_k</i>	<i>b_k</i>	<i>c_k</i>	<i>E_k & F</i>
1	−4.18046	3.32259	−7.37097	−141.907	−4.06502	3.38230	−7.47230	−56.828
2	0.13511	2.13256	−3.64888	−1122.593	0.12831	1.71004	−2.75631	−560.767
3	−1.81466	−0.02359	−1.70540	−52.697	11.44103	−0.00528	12.13929	−503.155
4	4.88475	−0.00526	5.87472	−976.426				<i>F</i> = −116.987
5				<i>F</i> = −340.704				

The coefficients E_1 and E_4 of Ofor and Alabi model 1 in Table A1 are relatively small compared to other coefficients. By performing the same procedure to obtain the simplification of Sousa et al. model, simplified hybrid models of Ofor and Alabi using four and three tangent hyperbolic functions are given by Equations (34) and (35).

$$f_{H,of} = \left[-2\log \left\{ \frac{\epsilon}{3.71D} - \frac{1.975}{Re} \left(\ln \left(\left(\frac{\epsilon}{3.93D} \right)^{1.092} + \left(\frac{7.627}{Re+395.9} \right) \right) \right) \right\} \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1\eta + c_1) + E_2 \tanh(a_2\zeta + b_2\eta + c_2) + E_3 \tanh(a_3\zeta + b_3\eta + c_3) + E_4 \tanh(a_4\zeta + b_4\eta + c_4) + F] \tag{34}$$

$$f_{H,of} = \left[-2\log \left\{ \frac{\epsilon}{3.71D} - \frac{1.975}{Re} \left(\ln \left(\left(\frac{\epsilon}{3.93D} \right)^{1.092} + \left(\frac{7.627}{Re+395.9} \right) \right) \right) \right\} \right]^{-2} + 10^{-6} [E_1 \tanh(a_1\zeta + b_1\eta + c_1) + E_2 \tanh(a_2\zeta + b_2\eta + c_2) + E_3 \tanh(a_3\zeta + b_3\eta + c_3) + F] \tag{35}$$

where $\zeta = \log(R_e)/3 - 2.1$ and $\eta = 0.37729 \log(\epsilon/D) + 1.49089$. Coefficients a, b, c, E , and F are given in Table 3. The simplification of the Ofor and Alabi hybrid model gives *MAMXRE* and *MRE* values of 0.039% and 0.009% (around one-seventh or seven times lower than those of the corresponding existing formula) for Equation (34), and 0.068% and 0.013% (one-third or three times lower) for Equation (35). The CPU times for computing data for validation with Equations (34) and (35) are 0.047 and 0.050 s, respectively.

Table 3. Coefficients of the simplified hybrid model of Offor-Alabi [33].

k	Coefficients of Equation (34)				Coefficients of Equation (35)			
	a_k	b_k	c_k	$E_k \text{ \& } F$	a_k	b_k	c_k	$E_k \text{ \& } F$
1	−2.80211	−0.62177	−2.94729	−836.494	−10.32700	−1.82460	−8.70690	16.457
2	10.73230	−2.83803	13.28526	−155.146	12.01430	−3.42070	15.08060	−112.035
3	3.08238	0.66715	3.10158	−769.969	−13.07130	0.08710	−13.08580	−164.880
4	10.06938	−0.20182	10.72137	400.633				$F = -37.108$
5				$F = -310.679$				

It is observed that the simplification of the Sousa et al. [32] and Offor-Alabi [33] hybrid models defined in Equations (32)–(35) still produces very accurate results with a *MAXRE* value of less than 0.1%, much more accurate than most of the more than 49 explicit formulas reviewed by Zeghadnia et al. [15] and Pérez Pupo et al. [17]. Therefore, Equations (32)–(35) are potentially used in practical applications, especially Equations (32) and (33), because they are more straightforward than Equations (34) and (35). However, to obtain more accurate results, the hybrid model 1 with five hyperbolic tangent functions should be used with small-time additional time computation. Figure 1 shows the spatial distribution of the absolute relative error *RE* of the existing formula of Sousa et al. [32] compared with *RE* predicted by its hybrid models with three, four, and five hyperbolic tangent functions. Figure 1 is generated by using MATLAB R2022a.

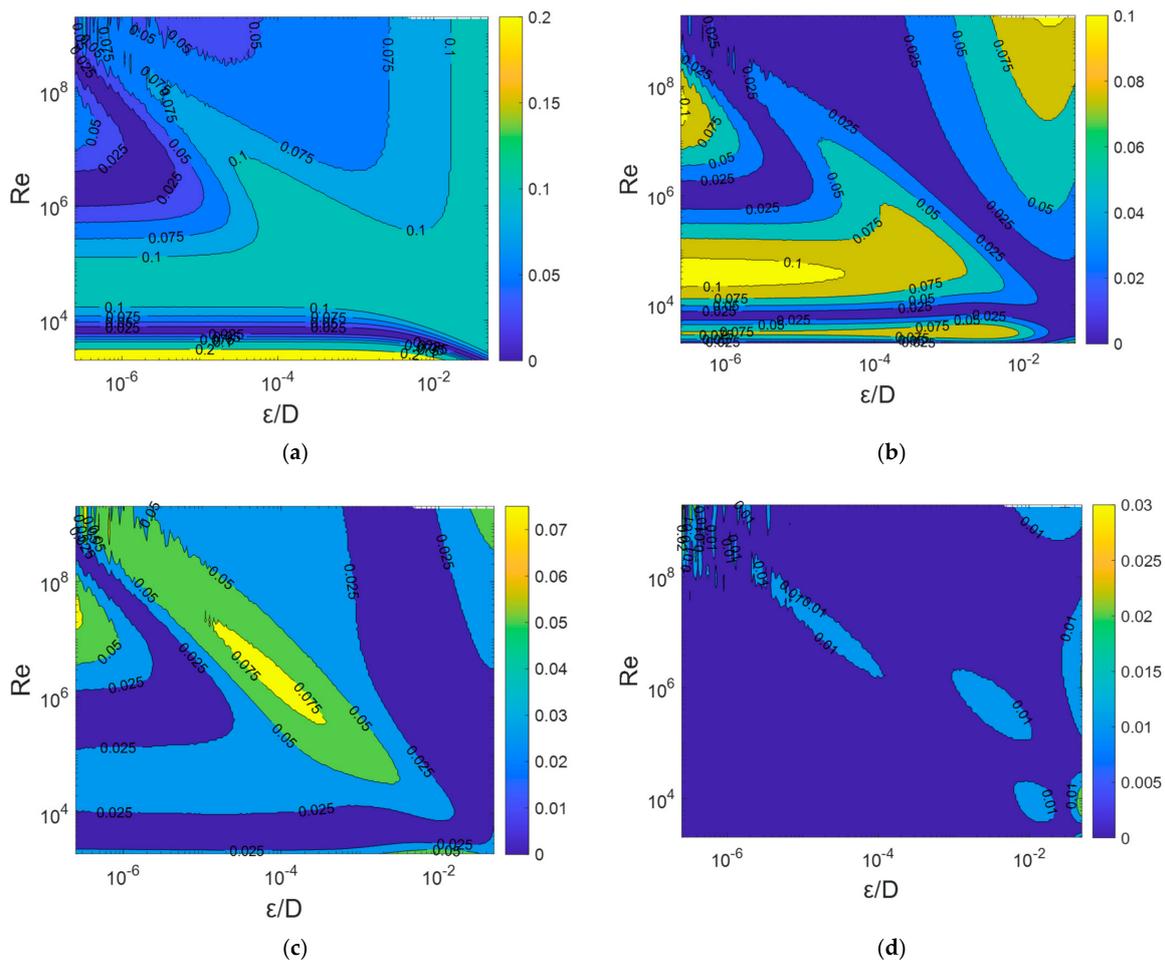


Figure 1. Spatial distribution of absolute relative error *RE* (%) of friction factor *f* for (a) predicted by the original formula of Sousa et al. [32] and (b–d) predicted by the hybrid model of Sousa et al. with three, four, and five hyperbolic tangent functions, respectively.

4. Conclusions

Hybrid models have been developed using a combination of explicit formulas available in the literature to solve the Colebrook–White equation with an error equation, e_A . The error equation e_A estimates e_0 which is the difference between the friction factor computed by the Colebrook–White equation, f_E , and that obtained by the explicit formula, f_o . The existing formulas reviewed were those proposed by Chen [29], Schorle et al. [30], Bar and White [31], Sousa et al. [32], and Offor and Alabi [33]. The hybrid model is $f_H = f_o + e_A$, where f_H is the f value predicted by the hybrid model. This study applied the ANN method with the multi-layered perceptron (MLP) architecture consisting of three main layers—the input layer, the hidden layer, and the output layer—for modeling the error equation e_A . The input layer consists of two input parameters: the logarithmic transformation of the Reynolds number R_e , and the relative roughness ε/D . The output is the e_A value for given R_e and ε/D values. The error equation e_A consists of a series of hyperbolic tangent functions whose number corresponds to the number of neurons in the hidden layer.

Three hybrid models are considered, including models 1, 2, and 3, with the number of hyperbolic tangent functions being 5, 7, and 10. Hybrid model 3 produces excellent results but is less attractive in computational time because it contains 10 hyperbolic tangent functions and requires a CPU time around 160% longer than hybrid model 2. However, the hybrid model 1 using five hyperbolic tangent functions could produce reasonable predictions of friction factors, with the *MAXRE* value one-tenth or ten times lower than that produced by the existing formula.

The equations for simplified hybrid models of Sousa et al. [32] and Offor-Alabi [33] using four and three tangent hyperbolic functions are also given. Simplifying these models still provides very accurate results, with a maximum absolute error *MAXRE* value of less than 0.1%, lower than that produced by most existing explicit formulas available in the literature. Therefore, these equations are potentially used in practical applications. However, for more precise results, hybrid model 1 with five hyperbolic tangent functions should be used with only small-time additional time computation, about 10 to 20% longer than that of the simplified model.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/fluids7070211/s1>. Equations (S1)–(S10): listed the equations and coefficients for all hybrid formulas. Table S1: Coefficient of hybrid model 2 of Chen [29], Schorle et al. [30] and Barr and White [31] given by Equations (S1), (S2) and (S3), respectively; Tabel S2: Coefficient of hybrid model 2 of Sousa et al. [32] and Offor and Alabi [33] given by Equations (S4) and (S5), respectively; Tabel S3: Coefficient of hybrid model 3 of Chen [29], Schorle et al. [30] and Barr and White [31] given by Equations (S6), (S7) and (S8), respectively; Tabel S4: Coefficient of hybrid model 3 of Sousa et al. [32] and Offor and Alabi [33] given by Equations (S9) and (S10), respectively.

Funding: This research was funded by Faculty of Civil and Environmental Engineering, Bandung Institute of Technology, grant number: 453.11/IT1.C06/TA.00/2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

This appendix lists the coefficients for hybrid models (existing explicit formula + explicit ANN model for error function) to be used in Equations (28)–(31) for models 1 of Schorle et al. [30], Barr and White [31], Sousa et al. [32], and Offor and Alabi [33], respectively.

Table A1. Coefficient of Equations (28)–(31) for hybrid models 1 of Schorle et al. [30], Barr and White [31], Sousa et al. [32], and Offor and Alabi [33].

Hybrid Model	Coefficients				
	k	a_k	b_k	c_k	E_k & F
Schorle et al. [30] + ANN (2-5-1), Equation (28)	1	−2.26933	−0.05760	0.74884	28.288
	2	0.93914	−0.66953	2.51097	25,423.641
	3	−1.04822	0.70094	−2.18636	11,387.088
	4	4.90973	0.04289	5.39717	−1765.822
	5	2.66153	1.81116	0.83719	69.487
	6				$F = -12,307.506$
Barr and White [31] + ANN (2-5-1), Equation (29)	1	0.83941	−2.98641	3.79362	3239.372
	2	1.94701	−3.08646	5.81603	9137.427
	3	−0.37782	−5.25154	1.95626	12.447
	4	0.90315	−2.76715	4.22969	−11,058.386
	5	−4.51262	0.18143	−5.95373	4272.718
	6				$F = 2942.393$
Sousa et al. [32] + ANN (2-5-1), Equation (30)	1	−4.14477	3.26576	−7.56141	−203.631
	2	0.13488	2.04836	−3.88041	−1598.224
	3	7.34543	6.94243	2.03779	−5.643
	4	−1.49063	−0.02376	−1.41946	−77.800
	5	4.69215	−0.00526	5.80950	−1382.211
	6				$F = -494.097$
Offor and Alabi [33] + ANN (2-5-1), Equation (31)	1	−6.74589	−3.60800	1.99030	1.199
	2	−2.66078	−0.61588	−2.73123	−457.489
	3	12.23306	−3.42486	15.28631	−134.388
	4	3.09142	0.68423	2.97875	−415.722
	5	12.38692	−0.21233	12.52520	192.577
	6				$F = -98.881$

References

- Subramanya, K. *Flow in Open Channels*, 3rd ed.; Tata McGraw-Hill Publishing Company Ltd.: New Delhi, India, 2009; pp. 86–91.
- Osman, A.A. *Open Channel Hydraulics*; Elsevier: Amsterdam, The Netherlands, 2006; pp. 67–74.
- Knight, D.W.; Hazlewood, C.; Lamb, R.; Samuels, P.G.; Shiono, K. *Practical Channel Hydraulics Roughness, Conveyance and Afflux*, 2nd ed.; CRC Press: Balkema, Leiden, 2018; pp. 29–33.
- White, F.M. *Fluid Mechanics*, 7th ed.; McGraw-Hill: New York, NY, USA, 2011; pp. 365–390.
- Çengel, Y.A.; Cimbala, J.M. *Fluid Mechanics Fundamentals and Applications*, 3rd ed.; McGraw-Hill: New York, NY, USA, 2014; pp. 367–374.
- Henderson, F.M. *Open Channel Flow*; The Macmillan Company: New York, NY, USA, 1966; pp. 90–101.
- Sturm, T.W. *Open Channel Hydraulics*, 2nd ed.; McGraw-Hill: New York, NY, USA, 2010; pp. 114–128.
- French, R.H. *Open-Channel Hydraulics*; McGraw-Hill Book Co.: Singapore, 1985; pp. 111–119.
- Chadwick, A.; Morfett, J.; Borthwick, M. *Hydraulics in Civil and Environmental Engineering*, 5th ed.; CRC Press: Abingdon, UK; Boca Raton, FL, USA, 2013; pp. 126–130.
- Colebrook, C.F. Turbulent Flow in Pipes with Particular Reference to the Transition Between the Smooth and Rough Pipe Laws. *J. Inst. Civ. Eng. Lond.* **1939**, *11*, 133–156. [[CrossRef](#)]
- Chapra, S.C.; Canale, R.P. *Numerical Method for Engineer*; McGraw Hill Education: New York, NY, USA, 2010; pp. 151–161.
- Larock, B.E.; Jeppson, R.W.; Watters, G.Z. *Hydraulics of Pipeline Systems*; CRC Press LLC.: Boca Raton, FL, USA, 2002; pp. 17–22, 148–151.
- Moody, L.F. Friction Factors for Pipe Flow. Transactions of the American Society of Mechanical Engineers. *J. Mater. Sci. Chem. Eng.* **1944**, *66*, 671–681.
- Brkić, D. Review of Explicit Approximations to the Colebrook Relation for Flow Friction. *J. Pet. Sci. Eng.* **2011**, *77*, 34–48. [[CrossRef](#)]
- Zeghadnia, L.; Robert, J.L.; Achour, B. Explicit solutions for turbulent flow friction factor: A review, assessment and approaches classification. *Ain Shams Eng. J.* **2019**, *10*, 243–252. [[CrossRef](#)]
- Plascencia, G.; Díaz-Damacillo, L.; Robles-Agudo, M. On the estimation of the friction factor: A review of recent approaches. *SN Appl. Sci.* **2020**, *2*, 163. [[CrossRef](#)]

17. Pérez Pupo, J.R.; Navarro-Ojeda, M.N.; Pérez-Guerrero, J.N.; Batista-Zaldívar, M.A. On the Explicit Expressions for the Determination of the Friction Factor in Turbulent Regime. *Rev. Mex. Ing. Química* **2020**, *19*, 313–334. [[CrossRef](#)]
18. Qiu, M.; Ostfeld, A. A Head Formulation for the Steady-State Analysis of Water Distribution Systems Using an Explicit and Exact Expression of the Colebrook–White Equation. *Water* **2021**, *13*, 1163. [[CrossRef](#)]
19. Choe, Y.W.; Sim, S.B.; Choo, Y.M. New Equation for Predicting Pipe Friction Coefficients Using the Statistical Based Entropy Concepts. *Entropy* **2021**, *23*, 611. [[CrossRef](#)]
20. Brkić, D.; Pavel Praks, P. Colebrook’s Flow Friction Explicit Approximations Based on Fixed-Point Iterative Cycles and Symbolic Regression. *Computation* **2019**, *7*, 48. [[CrossRef](#)]
21. Pavel Praks, P.; Brkić, D. Review of new flow friction equations: Constructing Colebrook’s explicit correlations accurately. *Rev. Int. Métodos Numér. Cál. Diseño Ing.* **2020**, *36*, 41. [[CrossRef](#)]
22. Brkić, D.; Praks, P. What Can Students Learn While Solving Colebrook’s Flow Friction Equation? *Fluids* **2019**, *4*, 114. [[CrossRef](#)]
23. Brkić, D.; Čojbašić, Ž. Evolutionary Optimization of Colebrook’s Turbulent Flow Friction Approximations. *Fluids* **2017**, *2*, 15. [[CrossRef](#)]
24. Haykin, S. *Neural Networks: A Comprehensive Foundation*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1999.
25. Srinivasulu, S.; Jain, A.A. Comparative analysis of training methods for artificial neural network rainfall-runoff models. *Appl. Soft Comput.* **2006**, *6*, 295–306. [[CrossRef](#)]
26. Huo, Z.; Feng, S.; Kang, S.; Huang, G.; Wang, F.; Guo, P. Integrated neural networks for monthly river flow estimation in arid inland basin of Northwest China. *J. Hydrol.* **2012**, *420*, 159–170. [[CrossRef](#)]
27. Cahyono, M. The Development of Explicit Equations for Estimating Settling Velocity Based on Artificial Neural Networks Procedure. *Hydrology* **2022**, *9*, 98. [[CrossRef](#)]
28. Sablani, S.S.; Shayya, W.H.; Kacimov, A. Explicit calculation of the friction factor in pipeline flow of Bingham plastic fluids: A neural network approach. *Chem. Eng. Sci.* **2003**, *58*, 99–106. [[CrossRef](#)]
29. Chen, N.H. An explicit equation for friction factor in pipe. *Ind. Eng. Chem. Fundam.* **1979**, *18*, 296–297. [[CrossRef](#)]
30. Schorle, B.J.; Churchill, S.W.; Shacham, M. Comments on: “An Explicit Equation for Friction Factor in Pipe”. *Ind. Eng. Chem. Fundam.* **1980**, *19*, 228–230. [[CrossRef](#)]
31. Barr, D.; White, C. Technical note. solutions of the colebrook-white function for resistance to uniform turbulent flow. *Proc. Inst. Civ. Eng.* **1981**, *71*, 529–535. [[CrossRef](#)]
32. Sousa, J.; Da Conceição, M.; Marques, A.S. An explicit solution of the Colebrook-White equation through simulated annealing. *Water Ind. Syst. Model. Optim. Appl.* **1999**, *2*, 347–355.
33. Offor, U.H.; Alabi, S.B. An accurate and computationally efficient explicit friction factor model. *Adv. Chem. Eng. Sci.* **2016**, *6*, 237–245. [[CrossRef](#)]