

Article

What Can Students Learn While Solving Colebrook's Flow Friction Equation?

Dejan Brkić ^{1,2,*}, [†] and Pavel Praks ^{1,*}, [†]

¹ IT4Innovations, VŠB—Technical University of Ostrava, 708 00 Ostrava, Czech Republic

² Research and Development Center "Alfatec", 18000 Niš, Serbia

* Correspondence: dejanbrkic0611@gmail.com (D.B.); pavel.praks@vsb.cz (P.P.)

† Both authors contributed equally to this study.

Received: 12 May 2019; Accepted: 25 June 2019; Published: 27 June 2019



Abstract: Even a relatively simple equation such as Colebrook's offers a lot of possibilities to students to increase their computational skills. The Colebrook's equation is implicit in the flow friction factor and, therefore, it needs to be solved iteratively or using explicit approximations, which need to be developed using different approaches. Various procedures can be used for iterative methods, such as single the fixed-point iterative method, Newton–Raphson, and other types of multi-point iterative methods, iterative methods in a combination with Padé polynomials, special functions such as Lambert W, artificial intelligence such as neural networks, etc. In addition, to develop explicit approximations or to improve their accuracy, regression analysis, genetic algorithms, and curve fitting techniques can be used too. In this learning numerical exercise, a few numerical examples will be shown along with the explanation of the estimated pedagogical impact for university students. Students can see what the difference is between the classical vs. floating-point algebra used in computers.

Keywords: Colebrook equation; Lambert W function; Padé polynomials; iterative methods; explicit approximations; learning; teaching strategies; floating-point computations

1. Introduction

The Colebrook equation for flow friction is an empirical formula developed by C. F. Colebrook in 1939 [1] and it is based on his joint experiment, which was performed with Prof. C. M. White in 1937 [2]. Colebrook was at this time a PhD student of Prof. White at Imperial College in London, UK [3]. The experiment was performed with a set of pipes of which the inner side was covered with sand of different grain sizes, while one was left completely without sand to be smooth. The sand was fixed on the pipe's inner surface using a sort of glue as adhesive material, while air was used as the fluid in this experiment.

The Colebrook equation follows logarithmic law and, at first sight, it seems to be very simple. However, it is given implicitly, with respect to the unknown flow friction factor f in a way that it can be expressed explicitly only in terms of the Lambert W function, with further difficulties in its evaluation [4–6] (it can be rearranged for gas flow [7]). Otherwise, it needs to be solved iteratively [8–10] or explicitly, using approximate formulas specially developed for such a purpose [10–17]. The unknown flow friction factor f in the Colebrook's equation depends on the known Reynolds number Re and the relative roughness of the inner pipe surface $\frac{\varepsilon}{D}$, all three of which are dimensionless, shown in Equation (1):

$$\frac{1}{\sqrt{f}} = -2 \cdot \log_{10} \left(\frac{2.51}{Re} \cdot \frac{1}{\sqrt{f}} + \frac{\frac{\varepsilon}{D}}{3.71} \right) \quad (1)$$

The relative roughness $\frac{\varepsilon}{D}$ can take values from 0 to 0.05, while the Reynolds number Re is from about 2500 to 10^8 [18]. The Colebrook equation is valid for the turbulent condition of flow [19].

In this learning exercise, students can have numerous numerical tasks that can improve their computational skills. The task can start with a simple trial/error approach that can, in a certain way, put the Colebrook equation in balance, while, furthermore, a simple fixed-point iteration method can be introduced in the curriculum [20]. Then, more complex approaches that require derivatives of the Colebrook equation can lead students to the Newton–Raphson iterative methods [21–23] or to the more complex multi-point iterative methods [8,9]. A first simple explicit approximate formula with inner iterative cycles can be introduced using such approaches [24].

Transcendental functions, such as logarithmic and exponential functions, on computers require the execution of additional floating point operations [25–27]. On the contrary, basic mathematical operations (addition, subtraction, multiplication, and division) are very fast on computers, because they are executed directly. Students can learn how to avoid using these time-consuming functions [27–30]. This simplification process can have a large effect on the students' future jobs, regarding the design of complex pipe networks, which involve extensive computations [31–36]. For example, in Section 2.2.1, students can reuse the value of the logarithm, which is computed only in the first iteration, to all subsequent iterations [30]. The less computationally demanded Padé polynomials are used instead of the logarithmic function in all subsequent iterations [30]. The students in that way need to know how to generate the appropriate polynomial expression (using appropriate software tools).

Regarding special functions, the Lambert W function can be introduced in the curriculum [5,6], as the Lambert W function transforms the Colebrook equation in the explicit form [4]. The widely used explicit form has one fast-growing term that makes calculations impossible in computers, due to an overflow error [37,38]. Students should figure out why such a numerical error appears in computers and then understand how such situations can be avoided. For this reason, the next step involves the series expansion of the Lambert W function, and its cognate, the Wright ω function [39]. In this step, students will learn how to make accurate, but at the same time, efficient approximations of the Colebrook equation using the Wright ω function [27–29,40,41]. Last, but not least, students can use raw numerical data that can efficiently simulate the Colebrook equation and, in that way, learn principles of using artificial intelligence in everyday engineering practice [12,42–45].

Finally, keeping in mind that the Moody diagram represents a graphical interpretation of the Colebrook equation, students should make a sketch of the Moody diagram [18,46], which will add an additional value both from the engineering and mathematical points of view (Figure 1 of this article can be used for this purpose).

2. Tasks for Students

2.1. Iterative Methods

For the purpose of this case study, two numerical examples are given: (1) $Re = 2.3 \times 10^5$ and $\frac{\varepsilon}{D} = 10^{-4}$; (2) $Re = 4.6 \times 10^7$ and $\frac{\varepsilon}{D} = 0.037$. Corresponding solutions for the flow friction factor are (1) $f = 0.016050961$, and (2) $f = 0.062427396$, respectively. Positions of these two examples with respect to the hydraulic regimes (laminar, transition to turbulent, and turbulent, which includes smooth, smooth to rough, and fully turbulent rough regime) are shown in Figure 1. Cases that can capture the nonlinearity of the Colebrook equation are: (a) Smooth pipe at very low, moderate, and very high Re , (b) pipe with moderate roughness at very low, moderate, and very high Re , and (c) pipe with high roughness ($\frac{\varepsilon}{D} \sim 0.05$) and again, low, moderate, and high Re . Example 1 is in the zone of moderate values of the Reynolds number Re , where the proposed value of relative roughness, which is small ($\frac{\varepsilon}{D} \sim 10^{-4}$), is large enough to provide a transition from smooth towards the turbulent regime. On the other hand, Example 2 with higher values of Re and $\frac{\varepsilon}{D}$ is in a fully turbulent zone. Using Figure 1, other different examples can be provided for these teaching exercises.

Example 1 is set to be exactly in the zone where the smooth nature of the viscose zone of the inner pipe surface starts to be too thin and when roughness starts to have influence [47,48].

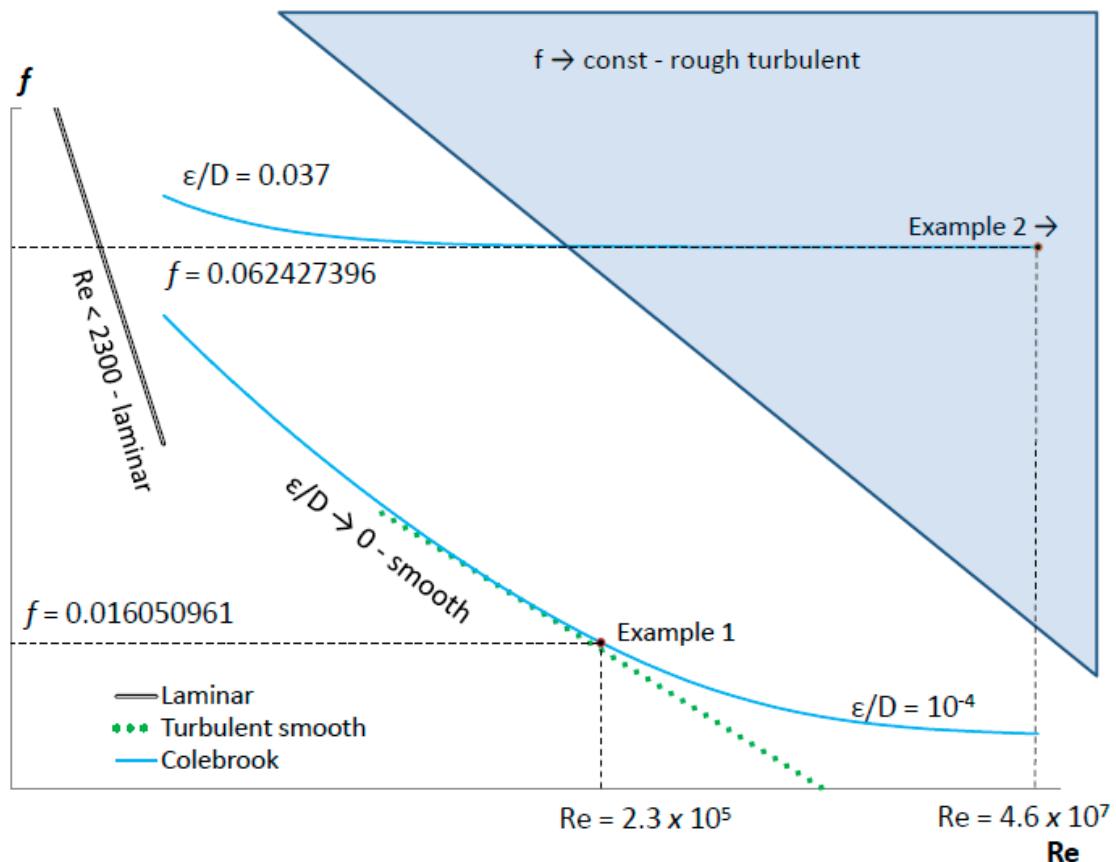


Figure 1. Position of the two presented examples with respect to the hydraulic regimes (laminar, transition to turbulent, and turbulent, which includes smooth, smooth to rough, and fully turbulent rough regime).

From Figure 1, students should see that our numerical Example 1 is set to be in the turbulent zone. More precisely, Example 1 is set at the end of the smooth turbulent regime, where the transition to fully turbulent begins. On the contrary, Example 2 is located in the fully developed rough turbulent zone. Students should understand the physical interpretation of different hydraulic regimes.

For this task, 45 min should be assigned.

2.1.1. Trial/Error Method

To do this task, students should guess somehow the initial value of the friction factor f , then insert it into the Colebrook equation, and finally, calculate a new value of f until an error criterion is fulfilled. Using this iterative procedure, students should show how many iterations will probably be required, as the number of iterations depends on the initial guess and whether such a rudimentary iterative procedure will always lead to the final balanced solution or not. Here will be introduced a concept of convergence, divergence, and oscillations in iterative procedures. Students should understand how significant it is to find the adequate value for the starting point for an iterative procedure [9]. For our Example 1, which uses $Re = 2.3 \times 10^5$ and $\frac{\epsilon}{D} = 10^{-4}$, the starting point needs to be chosen between -2.469 and 1574003 when Microsoft (MS) Excel is used and students should discuss why [10,20]. Subsequently, students should analyze the starting point for Example 2, in which $Re = 4.6 \times 10^7$ and $\frac{\epsilon}{D} = 0.037$. Students should explain consequences of the inappropriately chosen initial starting point. Finally, students should understand that the starting point for -2.5 , for example, will actually be

$\frac{1}{\sqrt{f_0}} = \frac{\frac{\epsilon}{D}}{3.71} - 2.5 = \frac{10^{-4}}{3.71} - 2.5 = -2.499973046$ for our Example 1, in which $\frac{\epsilon}{D} = 10^{-4}$ (the negative starting point does not have physical meaning and it is chosen purely for testing methods and to introduce the concept of “thinking outside of the box”, which could contribute to development of creative thinking in general).

For this task, 45 min should be assigned.

2.1.2. Fixed-Point Iterative Procedure

The fixed-point iterative method is suitable for spreadsheet solvers, such as MS Excel [10,20]. To prepare this task, students should rearrange the Colebrook equation in the suitable form $F(x) = 0$; where $x = \frac{1}{\sqrt{f}}$, i.e., $x + 2 \cdot \log_{10}\left(\frac{2.51}{Re} \cdot x + \frac{\epsilon}{3.71}\right) = 0$. The iterative process starts with a selected starting point x_0 and continues by $x_{i+1} = x_i - F(x_i)$ until a stopping criteria is fulfilled, for example, when the results from two subsequent iterations are almost equal ($x_{i+1} - x_i \sim 0$). From this approach, students will learn that the trial/error method can be used in a more systematic way. This task can improve the ability of students to use spreadsheet solvers, because students should set the solver to allow iterative computations. Moreover, the maximal number of iterations and the stopping criteria represented by the minimal value between two successive iterations should be set in the solver. For example, the MS Excel spreadsheet solver allows a maximal number of iterations of 32,767, and students should figure out why this constraint is set. The answer is that $2^{15} = 32,768$ as this represents the maximum it can be set for 16-bit registers in binary logic because one bit is reserved for the sign (\pm). In that way, students will learn that for human perception, more common is the decimal system, which is used, for example, for banknotes, \$1, \$10, \$100, while, for computers, the binary system is more suitable, which commonly uses 2, 4, 8, 16, 32, 64, etc.

In our case, (1) $Re = 2.3 \times 10^5$ and $\frac{\epsilon}{D} = 10^{-4}$, for $\frac{1}{\sqrt{f_0}} = \frac{\frac{\epsilon}{D}}{3.71} - 2.4 \approx -2.4$ and for $\frac{1}{\sqrt{f_0}} = \frac{\frac{\epsilon}{D}}{3.71} + 1,574,003 \approx 1,574,003$, the number of required iterations for nine decimal accuracy will be 11 in both cases, i.e., the accurate solution will be $\frac{1}{\sqrt{f_{11}}} = 7.8931$. Students should make further tests with the same example, but with different starting points and, in addition, perform a similar test for Example 2.

Using the spreadsheet solver, students should first solve the Colebrook equation iteratively, using only one cell of the Colebrook equation for smooth conditions, when $\frac{\epsilon}{D}$ is zero. Students should realize how important the role of the initial starting point is, because, in this case, software will declare a division by zero error. In other conditions, when $\frac{\epsilon}{D} \neq 0$, the initial starting point will be automatically set as $f_0 = \log_{10}\left(\frac{\frac{\epsilon}{D}}{3.71}\right)$.

Here, students may try to make a sketch of the function, i.e., to make a two-dimensional (2D) diagram of the Colebrook equation. Here, two possibilities should be analyzed: $\frac{1}{\sqrt{f}} = F(Re \cdot \sqrt{f}, \frac{\epsilon}{D})$ as suggested by Rose [46], or $f = F(Re, \frac{\epsilon}{D})$ as suggested by Moody [18] (approach from Figure 1 of this article can be used). In addition, students can analyze whether a simple transformation $x = \frac{1}{\sqrt{f}}$ accelerates solving of the Colebrook equation.

For this task, 45 min should be assigned.

2.1.3. Newton–Raphson Iterative Procedure

In this part, students will learn how to find the first derivative of the function with respect to the variable flow friction factor f , where $x = \frac{1}{\sqrt{f}}$ and $F(x) = x + 2 \log_{10} \left(\frac{2.51}{Re} \cdot x + \frac{\epsilon}{3.71} \right)$, as in Equation (2):

$$\frac{\partial(F(x))}{\partial x} = F'(x) = F' \left(x + 2 \log_{10} \left(\frac{2.51}{Re} \cdot x + \frac{\epsilon}{3.71} \right) \right) = -\frac{1}{2} \cdot \left(\frac{1}{\sqrt{|f_i|}} \right)^3 \cdot \left(1 + 2 \cdot \frac{2.51}{Re \cdot \ln(10) \cdot \left(\frac{2.51}{Re} \cdot \frac{1}{\sqrt{|f_i|}} + \frac{\epsilon}{3.71} \right)} \right), \quad (2)$$

The procedure for finding the first derivative can be done manually, but also for Equation (2), it can be generated using appropriate software, e.g., Matlab by the “diff” command.

The first derivative is used in the Newton–Raphson procedure as $f_{i+1} = f_i - \frac{F(f_i)}{F'(f_i)}$ and students may test whether the Newton–Raphson procedure is faster or not, compared with the simple fixed-point method of the Colebrook equation, where $F(f_{i+1}) = \frac{1}{\sqrt{f}} + 2 \log_{10} \left(\frac{2.51}{Re} \cdot \frac{1}{\sqrt{f_i}} + \frac{\epsilon}{3.71} \right)$.

Students should realize that the previous example with the fixed point method is set for solving $x = \frac{1}{\sqrt{f}}$, while here, the Newton–Raphson method is set for solving $F(x) = 0$. For Example 1, in which $Re = 2.3 \times 10^5$ and $\frac{\epsilon}{D} = 10^{-4}$, we can use the already analyzed starting point $\frac{1}{\sqrt{f_0}} = \frac{\epsilon}{3.71} + 1,574,003 \approx 1,574,003 \rightarrow f_0 = 4.04 \times 10^{-13}$, which will give accurate results. Unfortunately, this iteration process requires 27 iterations, $f_{27} = 0.016050961$, whereas $\frac{1}{\sqrt{f_0}} = \frac{\epsilon}{3.71} - 2.4 \approx -2.4 \rightarrow f_0 = 0.173611$ will report a division by zero error in the fourteenth iteration and the calculation will stop without obtaining the precise solution. Here, students should understand that even a more powerful iterative procedure can fail when the problem is not set in an appropriate way and when the role of a good initial starting point is not fully understood.

The Newton–Raphson method, in which $F'(f_i) = 1$ is assumed, is equivalent to the fixed-point method in the form $f_{i+1} = f_i - F(f_i)$. However, this formulation shows very bad convergence properties. Students should understand why the iterative process $x_{i+1} = \frac{1}{\sqrt{f_{i+1}}} = x_i - F(x_i)$ gives the fast solution, while the iterative process $f_{i+1} = f_i - F(f_i)$ is slow. Students should again analyze coordinates in diagrams from Rose and Moody, and discuss which formulation is better [18].

Here, it should be noted that the function $F(x) = x + 2 \log_{10} \left(\frac{2.51}{Re} \cdot x + \frac{\epsilon}{3.71} \right)$ in form $F \left(\frac{1}{\sqrt{f}} \right) = \left(2 \log_{10} \left(\frac{2.51}{Re} \cdot \frac{1}{\sqrt{f}} + \frac{\epsilon}{3.71} \right) \right)^{-2} + \frac{1}{\sqrt{f}}$ shows very slow convergence when it is used in the iterative process $f_{i+1} = f_i - \frac{F(f_i)}{F'(f_i)}$. Students should discover why. Students should investigate how the initial starting point affects the required number of iterations.

For this task, 90 min should be assigned (including repetition and clarifications from the previous tasks).

2.1.4. Multi-Point Iterative Procedures

The shown Newton–Raphson method belongs to the two-point iterative methods and, in this task, students should find more powerful iterative methods. For example, the proposed methods for this exercise are three-point iterative methods, such as Halley or Schröder or even more powerful multi-point methods [8,9]. Students should figure out whether the number of required iterations to find the accurate solution decreases with the increased complexity of the chosen method. If the number of required iterations does not decrease, students should analyze why. In addition, it needs to be found how important it is to find the adequate initial starting point. In this task, students should find the second derivative of the Colebrook equation, which will increase their computing skills.

In this exercise, students should develop the ability to make a balance between complexity, speed, and required accuracy. Students obtain an experience for choosing the appropriate iterative method.

Here, we will not repeat the Newton–Raphson calculations in the form of Equation (3), which gives fast accurate results, but we will show the Halley method: Equation (4) for which the first derivative $F'(x)$ and the second $F''(x)$, with respect to variable $x = \frac{1}{\sqrt{f_i}}$, are given by Equations (5) and (6), respectively.

$$x_{i+1} = x_i - \frac{F(x_i)}{F'(x_i)}, \quad (3)$$

$$x_{i+1} = x_i - \frac{2 \cdot F(x_i) \cdot F'(x_i)}{2 \cdot F'(x_i)^2 - F(x_i) \cdot F''(x_i)}, \quad (4)$$

$$F'(x_i) = \frac{9287 \cdot \ln(10) \cdot x_i + 1000 \cdot \frac{\epsilon}{D} \cdot Re \cdot \ln(10) + 18574}{\ln(10) \cdot (9287 \cdot x_i + 1000 \cdot \frac{\epsilon}{D} \cdot Re)}, \quad (5)$$

$$F''(x_i) = -\frac{172496738}{\ln(10) \cdot (9287 \cdot x_i + 1000 \cdot \frac{\epsilon}{D} \cdot Re)^2}, \quad (6)$$

For the first derivatives, [8,9,21–23] should be consulted.

For Example 1, in which $Re = 2.3 \times 10^5$ and $\frac{\epsilon}{D} = 10^{-4}$, for initial starting point $x_0 = \frac{1}{\sqrt{f_0}} = \frac{\frac{\epsilon}{D}}{3.71} + 1574003 \approx 1574003$, the final balanced solution will be achieved after 27 iterations; $x_{27} = \frac{1}{\sqrt{f_{27}}} = 7.89313398624169 \rightarrow f_{27} = 0.016050961$. However, for the initial starting point $x_0 = \frac{1}{\sqrt{f_0}} = \frac{\frac{\epsilon}{D}}{3.71} - 2.4 \approx -2.4$, the procedure in MS Excel will report a division by zero error in the third step, so it will be a good opportunity for students to reexamine the importance of the appropriate chosen starting point. Thus, $x_0 = \frac{1}{\sqrt{f_0}}$, which in this case can be selected from -1.172 to $1,583,778$. Of course, negative starting points do not represent physical meaning. A similar exercise should be done for Example 2, in which $Re = 4.6 \times 10^7$ and $\frac{\epsilon}{D} = 0.037$.

For this task, at least 90 min should be assigned, as multi-point iterative methods are more complex than two-point methods.

2.2. Special Iterative Methods

While simple operations, such as adding, subtracting, multiplication, and division, require minimal system resources of computers, more complex operations, such as logarithmic, non-integer power, and similar function calls, require multiple floating point operations to be executed in the Central Processing Unit (CPU), which represents a time- and energy-consuming task [15,25–28]. For example, students should understand that the execution of one non-integer power operation in a computer goes through execution of one logarithmic and one exponential function. Some numbers are too big or too small to be placed in registers of computers, which have 32 or 64 bits [37,38].

2.2.1. Padé Approximant

As noted, one iterative cycle involves the evaluation of one logarithmic function when solving the Colebrook's law. The Colebrook's law is in its nature logarithmic, but here, students will learn how to avoid multiple evaluations of the logarithmic function by replacing it with a Padé approximant in all iterations, with the exception of the first one [30]. A Padé approximant is the "best" approximation of the given function by a rational function of given order. For example, $\log_{10}(95)$ can be accurately approximated by its Padé approximant at the expansion point $x = 1$, keeping in mind that $\log_{10}(100) = 2$ and using the fact that $\log_{10}(95) = \log_{10}\left(\frac{100}{95}\right) = \log_{10}(100) - \log_{10}\left(\frac{100}{95}\right)$ where

$\frac{100}{95} \approx 1.0526$ is close to 1. The Padé approximant will be very accurate in this case, because 1.0526 is very close to the expansion point $x = 1$.

Evaluation of the decadic logarithm can be expressed by the natural logarithm $\log_{10}(z) = \frac{\ln(z)}{\ln(10)}$, where $\ln(10) \approx 2.302585093$ is constant. The first task for students would be to make research related to Padé approximants. For students with interest in programming, the additional task would be to develop a code in appropriate software, such as Matlab. Padé approximants of different orders can be used for approximation of $\ln(z)$, for arguments z close to 1, i.e., $z \approx 1$. As the expansion point $z_0 = 1$ is a root of $\ln(z)$, the accuracy of the Padé approximant decreases. Setting the “OrderMode” option in the Matlab Padé command to relative compensates for the decreased accuracy. Thus, here, the Padé approximant of the (m,n) order uses the form $\ln(z) \approx \frac{(z-z_0) \cdot (\alpha_0 + \alpha_1 \cdot (z-z_0) + \dots + \alpha_m \cdot (z-z_0)^m)}{1 + \beta_1 \cdot (z-z_0) + \dots + \beta_n \cdot (z-z_0)^n}$, where α and β are coefficients (the coefficients of the polynomials need not be rational numbers). The Horner algorithm transforms polynomials into a computationally efficient form [30]. One possible procedure for solving the Colebrook equation using Padé approximants and the simple fixed-point iterative method is shown in Figure 2. Detailed step-by-step calculation using a table can be found in [30].

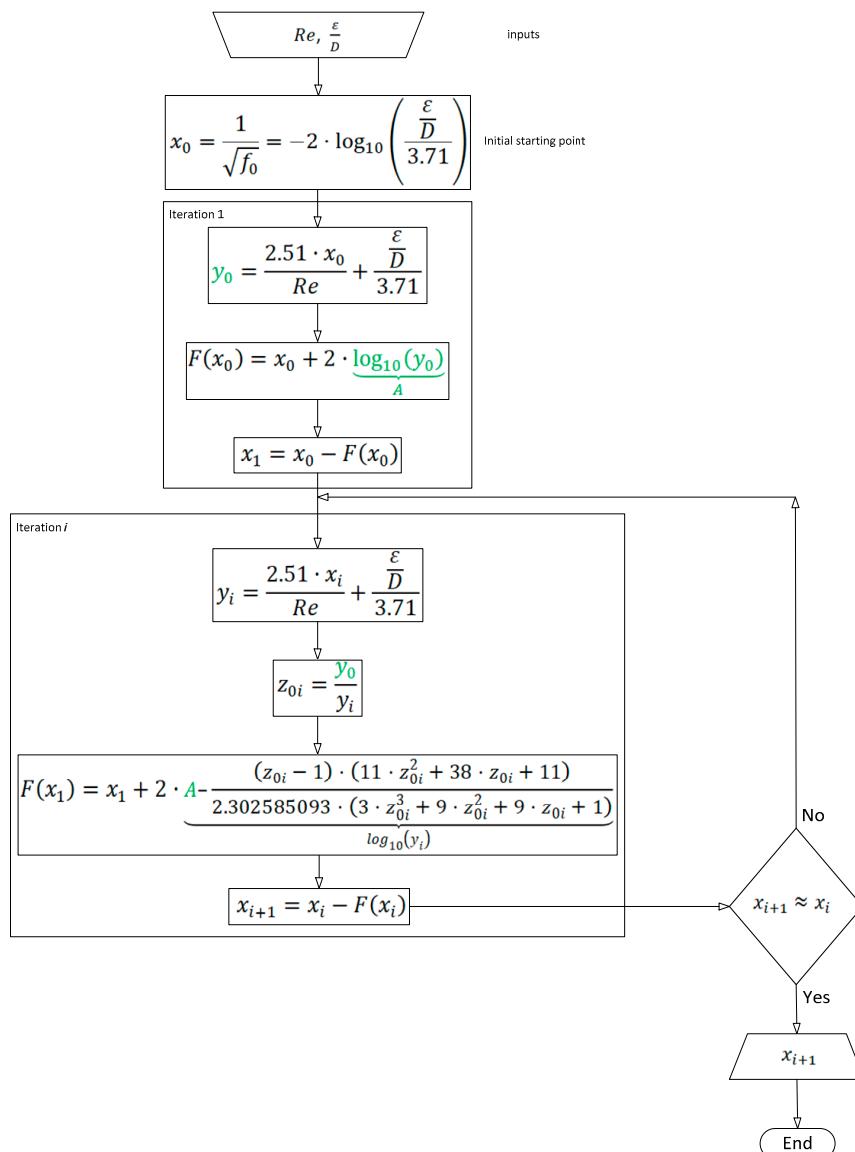


Figure 2. A possible procedure for solving the Colebrook equation using Padé approximants and the simple fixed-point iterative method.

Students interested in programming can use tic and toc functions of Matlab and GNU/Octave [49,50] to measure the speed of calculations using the classical fixed-point method vs. the fixed-point method combined with Padé approximants (let us remind readers that GNU/Octave software is available for free [49]). Students can combine the Newton–Raphson method with Padé approximants and check the accuracy of Padé approximants of different orders [30].

For a general understanding of the concept of Padé approximants, 90 min is required (45 min for an introduction and a further 45 min for numerical tests). Then, that additional 90 min would be required for the application of Padé polynomials for solving the Colebrook equation (again, 45 min for an introduction and 45 min for numerical examples). Students with an interest in programming should have an additional 45-min exercise (a presentation of complexity vs. accuracy of Padé approximants).

2.2.2. Approximations by Multi-Point Methods with Internal Cycles

Some iterative methods belong to the group of multi-point methods [8] that are very powerful and that can reach the accurate solution of the Colebrook equation in the first iteration. An example of an approximation of the Colebrook equation based on internal cycles is given by Serghides [51]. Serghides' methods belong to Steffensen types of methods, which do not require derivatives (other Steffensen types of methods can be used in addition [52,53]).

In order to develop an approximation with the internal cycle, students can choose one initial fixed point f_0 , for example $f_0 = \frac{f_{min} + f_{max}}{2} = \frac{0.0064 + 0.077}{2} = 0.0417$, and then calculate $\frac{1}{\sqrt{f_0}} \approx 4.9$, which will give $\frac{1}{\sqrt{f}} = -2 \cdot \log_{10} \left(\frac{2.51 \cdot 4.9}{Re} + \frac{\epsilon}{3.71} \right)$ and after one additional cycle $\frac{1}{\sqrt{f}} \approx -2 \cdot \log_{10} \left(\left(\frac{\frac{\epsilon}{D}}{3.71} - \frac{5.02}{Re} \cdot \log_{10} \left(\frac{12.3}{Re} + \frac{\epsilon}{3.71} \right) \right) \right)$. Using this exercise, students should see how important constant 12.3 is. Students should evaluate the relative error over the domain of applicability of the Colebrook's equation, then replace the constant 12.3 with a more appropriate value. Students can compare their chosen constant with values shown by Shacham [54] and by Zigrang and Sylvester [24]. For Example 1, in which $Re = 2.3 \cdot 10^5$ and $\frac{\epsilon}{D} = 10^{-4}$, the approximate value is $\frac{1}{\sqrt{f}} \approx -2 \cdot \log_{10} \left(\left(\frac{10^{-4}}{3.71} - \frac{5.02}{2.3 \cdot 10^5} \cdot \log_{10} \left(\frac{12.3}{2.3 \cdot 10^5} + \frac{10^{-4}}{3.71} \right) \right) \right)$. Students should calculate this value and evaluate the introduced error compared with the accurate iterative solution [20]. An additional task could be to find additional approximations with internal cycles, such as Romeo et al. [55] or Offor and Alabi [56] and to discuss what additional strategies were used in these cases.

For this task, 45 min is required.

2.3. Special Functions

The Colebrook equation is implicit with respect to the unknown flow friction factor. However, the Colebrook equation can be rearranged in the explicit form using the Lambert W function, where further, this special function can be evaluated only approximately [5]. Some numerical constraints in using the Lambert W function, which can occur in our case, will be detected [37] and a way to mitigate this inconvenience will be shown [27–29].

For tasks related to special functions, students need 45 min for a general introduction and an additional 90 min to work on examples.

2.3.1. Lambert W Function

Interested readers may find more information on the Lambert W function through [5,6,57,58]. Then, students should understand why some functions, such as logarithmic, square root, or exponential, are available in pocket calculators and spreadsheet solvers, while the Lambert W function is not [6]. In this task, students should understand how computers use a series expansion to interpret functions. Moreover, they need to understand how to use Visual Basic for Applications (VBA) modules in order to introduce new functions in MS Excel [59] (related to this task, students interested in programming should take at least 90 min).

The Lambert W function is used in mathematics to solve equations in which the unknown appears both outside and inside an exponential function or a logarithm, such as $3x + 2 = e^x$ or $x = \ln(4 \cdot x)$. According to Fukushima [60], the Lambert W function is defined as the solution of a transcendental equation with $\kappa = W(\kappa) \cdot e^{W(\kappa)}$. Such equations cannot be, except in special cases, solved explicitly in terms of the finite number of arithmetic operations, nor in terms of exponentials or logarithms. In the case of the Colebrook equation, students can consult scientific literature, while here we will show formulation from [28,29], Equation (7), while a similar formulation can be found in [27]:

$$\left. \begin{aligned} \frac{1}{\sqrt{f}} &= \frac{2}{\ln(10)} \cdot \left(\ln\left(\frac{Re}{2.51} \cdot \frac{\ln(10)}{2}\right) + W(e^x) - x \right) \\ x &= \ln\left(\frac{Re}{2.51} \cdot \frac{\ln(10)}{2}\right) + \frac{Re \cdot \frac{\epsilon}{D}}{2.51 \cdot 3.71} \cdot \frac{\ln(10)}{2} \end{aligned} \right\} \quad (7)$$

Our carefully chosen examples, $Re = 2.3 \times 10^5$ and $\frac{\epsilon}{D} = 10^{-4}$, (2) $Re = 4.6 \times 10^7$ and $\frac{\epsilon}{D} = 0.037$, respectively, are chosen to show that using floating-point arithmetic, Equation (7), gives the accurate solution for Example 1, while for Example 2, an arithmetic overflow error will occur. It is because $W(e^x)$ becomes too large for ordinary computer registers.

2.3.2. Wright ω Function and Related Approximations

As shown in Biberg [27] and Brkić and Praks [28,29], the problem with too large numbers in computer registers can be solved by introducing a cognate of the Lambert W function—the Wright ω function. The Wright ω function $y = \omega(x) = W(e^x)$ is a solution to the equation $y + \ln(y) = x$. Here, students should reformulate Equation (7) through the Wright ω function [50].

For Example 2, which uses $Re = 4.6 \times 10^7$ and $\frac{\epsilon}{D} = 0.037$, x is approximately 210441.7. In Matlab, the command $y = \text{wrightOmega}(210441.7)$ returns the value $y = 210429.44$.

In Matlab, but also in the GNU/Octave software [49], students can use the free WrightOmegaq library [50]. In this case, the command is $y = \text{wrightOmegaq}(210441.7)$.

Students can easily verify that the solution y satisfies the relation $y + \ln(y) = x$. Let us reiterate that for Example 2, the value of x is approximately 210441.7.

In the following task, students should explore how to estimate the Wright ω function without Matlab or GNU/Octave. The simplification $W(e^x) - x \approx \ln(x) \cdot \left(\frac{1}{x} - 1\right)$ gives an accurate approximation suitable for engineering practice—Equation (8) [28,29]:

$$\left. \begin{aligned} \frac{1}{\sqrt{f}} &\approx 0.8686 \cdot \left[B + \ln(B + A) \cdot \left(\frac{1}{B+A} - 1 \right) \right] \\ A &\approx \frac{Re \cdot \frac{\epsilon}{D}}{0.0878} \\ B &\approx \ln\left(\frac{Re}{2.18}\right) \approx \ln(Re) - 0.7794 \end{aligned} \right\} \quad (8)$$

where $\frac{2.251}{\ln(10)} \approx 2.18$ and $2.18 \cdot 3.71 \approx 8.0878$.

Students should evaluate the relative error of the approximation, Equation (8), and develop a more accurate approximation using the enhanced model $W(e^x) - x \approx c + \ln(x) \cdot \left(\frac{1}{x} - 1\right)$, where c is a constant. Of course, if $c = 0$, the enhanced model is equivalent to the model (8). For the general case, the constant c can be found by the MS Excel Solver, in order to minimize the relative error of the model with the iterative solution of the Colebrook equation provided by MS Excel. Students should do this optimization task with randomly selected input values of the Colebrook model $(Re, \frac{\epsilon}{D})$, which are sampled from the domain of applicability of the Colebrook equation. Finally, students can compare the relative error of this enhanced model with the relative error of the original model.

2.3.3. Tania Function and Related Approximations

Students should examine the Tania function, which is defined as $Tania(x) = \omega(x + 1)$, $Tania(x) = W(e^x)$, all with reference to [61] (the Euler T function should be an appropriate task to be further examined [39]).

For this task, 45–90 min should be assigned, during which students need to try to develop ability accepting new functions with new concepts [62].

3. Conclusions

Here we present how the relatively simple Colebrook equation, widely known by all students of hydraulics, can be used for introducing in curricula various iterative methods, special functions such as Lambert W, polynomials such as Padé approximants, optimization methods, etc.

The proposed exercises are suitable for students of hydraulics (civil, mechanical, chemical, or petroleum engineering curriculum, including the exploitation of oil and gas), but can be used as practical examples in teaching of numerical methods for students of mathematics. The proposed exercises can be extended for the teaching of pipe network analysis [31–36]. These methods can be combined with recent experimental works [63]. Furthermore, instead of pipes, a modified version of the Colebrook equation can be used for fuel cells [64,65].

Author Contributions: The paper is a product of the joint efforts of the authors, who worked together on models of natural gas distribution networks. P.P. has a scientific background in applied mathematics and programming, while D.B. has a background in control and applied computing in mechanical and petroleum engineering. They used the experience gained from their previous papers related to the Colebrook equation to develop proposed teaching exercises. D.B. wrote a draft of this article and prepared figures.

Funding: This work has been partially funded by the Technology Agency of the Czech Republic, partially by the National Centre for Energy TN01000007 and partially by the project “Energy System for Grids” TK02030039. Work of D.B. has been also supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPS II) project “IT4Innovations excellence in science - LQ1602” and by the Ministry of Education, Science, and Technological Development of the Republic of Serbia through the project “Development of new information and communication technologies, based on advanced mathematical methods, with applications in medicine, telecommunications, power systems, protection of national heritage and education” iii44006.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Colebrook, C.F. Turbulent flow in pipes with particular reference to the transition region between the smooth and rough pipe laws. *J. Inst. Civ. Eng.* **1939**, *11*, 133–156. [[CrossRef](#)]
2. Colebrook, C.F.; White, C.M. Experiments with fluid friction in roughened pipes. *Proc. R. Soc. Lond. Ser. A-Math. Phys. Sci.* **1937**, *161*, 367–381. [[CrossRef](#)]
3. Hager, W.H. Cedric Masey White and his solution to the pipe flow problem. *Proc. Inst. Civ. Eng. Water Manag.* **2010**, *163*, 529–537. [[CrossRef](#)]
4. Keady, G. Colebrook-White formula for pipe flows. *J. Hydraul. Eng.* **1998**, *124*, 96–97. [[CrossRef](#)]
5. Corless, R.M.; Gonnet, G.H.; Hare, D.E.; Jeffrey, D.J.; Knuth, D.E. On the LambertW function. *Adv. Comput. Math.* **1996**, *5*, 329–359. [[CrossRef](#)]
6. Hayes, B. Why W? On the Lambert W function, a candidate for a new elementary function in mathematics. *Am. Sci.* **2005**, *93*, 104–108. [[CrossRef](#)]
7. More, A.A. Analytical solutions for the Colebrook and White equation and for pressure drop in ideal gas flow in pipes. *Chem. Eng. Sci.* **2006**, *61*, 5515–5519. [[CrossRef](#)]
8. Praks, P.; Brkić, D. Choosing the optimal multi-point iterative method for the Colebrook flow friction equation. *Processes* **2018**, *6*, 130. [[CrossRef](#)]
9. Praks, P.; Brkić, D. Advanced iterative procedures for solving the implicit Colebrook equation for fluid flow friction. *Adv. Civ. Eng.* **2018**, *2018*, 5451034. [[CrossRef](#)]
10. Brkić, D. Determining friction factors in turbulent pipe flow. *Chem. Eng. N. Y.* **2012**, *119*, 34–39.
11. Brkić, D. Review of explicit approximations to the Colebrook relation for flow friction. *J. Pet. Sci. Eng.* **2011**, *77*, 34–48. [[CrossRef](#)]
12. Brkić, D.; Čoibašić, Ž. Evolutionary optimization of Colebrook’s turbulent flow friction approximations. *Fluids* **2017**, *2*, 15. [[CrossRef](#)]
13. Brkić, D. New explicit correlations for turbulent flow friction factor. *Nucl. Eng. Des.* **2011**, *241*, 4055–4059. [[CrossRef](#)]

14. Brkić, D. An explicit approximation of Colebrook's equation for fluid flow friction factor. *Pet. Sci. Technol.* **2011**, *29*, 1596–1602. [[CrossRef](#)]
15. Winning, H.K.; Coole, T. Improved method of determining friction factor in pipes. *Int. J. Numer. Methods Heat Fluid Flow* **2015**, *25*, 941–949. [[CrossRef](#)]
16. Pimenta, B.D.; Robaina, A.D.; Peiter, M.X.; Mezzomo, W.; Kirchner, J.H.; Ben, L.H. Performance of explicit approximations of the coefficient of head loss for pressurized conduits. *Rev. Bras. Eng. Agrícola Ambiental* **2018**, *22*, 301–307. [[CrossRef](#)]
17. Vatankhah, A.R. Approximate analytical solutions for the Colebrook equation. *J. Hydraul. Eng.* **2018**, *144*, 06018007. [[CrossRef](#)]
18. Moody, L.F. Friction factors for pipe flow. *Trans. ASME* **1944**, *66*, 671–684.
19. Brkić, D.; Praks, P. Unified friction formulation from laminar to fully rough turbulent flow. *Appl. Sci.* **2018**, *8*, 2036. [[CrossRef](#)]
20. Brkić, D. Solution of the implicit Colebrook equation for flow friction using Excel. *Spreadsheets Educ.* **2017**, *10*, 2.
21. Santos-Ruiz, I.; Bermúdez, J.R.; López-Estrada, F.R.; Puig, V.; Torres, L. Estimación experimental de la rugosidad y del factor de fricción en una tubería. In Proceedings of the Memorias del Congreso Nacional de Control Automático, San Luis Potosí, Mexico, 10–12 October 2018. Available online: <https://www.researchgate.net/publication/328332798> (accessed on 19 April 2019). (In Spanish)
22. Olivares, A.; Guerra, R.; Alfaro, M.; Notte-Cuello, E.; Puentes, L. Experimental evaluation of correlations used to calculate friction factor for turbulent flow in cylindrical pipes. *Rev. Int. Métodos Numér. Cálculo. Diseño Ing.* **2019**, *35*, 15. (In Spanish) [[CrossRef](#)]
23. Olivares Gallardo, A.P.; Guerra Rojas, R.A.; Alfaro Guerra, M.A. Evaluación experimental de la solución analítica exacta de la ecuación de Colebrook-White. *Ing. Investig. Tecnol.* **2019**, *20*, 1–11. (In Spanish) [[CrossRef](#)]
24. Zigrang, D.J.; Sylvester, N.D. Explicit approximations to the solution of Colebrook's friction factor equation. *AIChE J.* **1982**, *28*, 514–515. [[CrossRef](#)]
25. Winning, H.K.; Coole, T. Explicit friction factor accuracy and computational efficiency for turbulent flow in pipes. *Flow Turbul. Combust.* **2013**, *90*, 1–27. [[CrossRef](#)]
26. Clamond, D. Efficient resolution of the Colebrook equation. *Ind. Eng. Chem. Res.* **2009**, *48*, 3665–3671. [[CrossRef](#)]
27. Biberg, D. Fast and accurate approximations for the Colebrook equation. *J. Fluids Eng.* **2017**, *139*, 031401. [[CrossRef](#)]
28. Brkić, D.; Praks, P. Accurate and efficient explicit approximations of the Colebrook flow friction equation based on the Wright ω -function. *Mathematics* **2019**, *7*, 34. [[CrossRef](#)]
29. Brkić, D.; Praks, P. Accurate and efficient explicit approximations of the Colebrook flow friction equation based on the Wright ω -function: Reply to discussion. *Mathematics* **2019**, *7*, 410. [[CrossRef](#)]
30. Praks, P.; Brkić, D. One-log call iterative solution of the Colebrook equation for flow friction based on Padé polynomials. *Energies* **2018**, *11*, 1825. [[CrossRef](#)]
31. Brkić, D.; Praks, P. An efficient iterative method for looped pipe network hydraulics free of flow-corrections. *Fluids* **2019**, *4*, 73. [[CrossRef](#)]
32. Brkić, D.; Praks, P. Short overview of early developments of the Hardy Cross type methods for computation of flow distribution in pipe networks. *Appl. Sci.* **2019**, *9*, 2019. [[CrossRef](#)]
33. Brkić, D. Spreadsheet-based pipe networks analysis for teaching and learning purpose. *Spreadsheets Educ.* **2016**, *9*, 4.
34. Brkić, D. A gas distribution network hydraulic problem from practice. *Pet. Sci. Technol.* **2011**, *29*, 366–377. [[CrossRef](#)]
35. Brkić, D. Iterative methods for looped network pipeline calculation. *Water Resour. Manag.* **2011**, *25*, 2951–2987. [[CrossRef](#)]
36. Brkić, D. An improvement of Hardy Cross method applied on looped spatial natural gas distribution networks. *Appl. Energy* **2009**, *86*, 1290–1300. [[CrossRef](#)]
37. Sonnad, J.R.; Goudar, C.T. Constraints for using Lambert W function-based explicit Colebrook–White equation. *J. Hydraul. Eng.* **2004**, *130*, 929–931. [[CrossRef](#)]

38. Brkić, D. Comparison of the Lambert W-function based solutions to the Colebrook equation. *Eng. Comput.* **2012**, *29*, 617–630. [[CrossRef](#)]
39. Belkić, D. The Euler T and Lambert W functions in mechanistic radiobiological models with chemical kinetics for repair of irradiated cells. *J. Math. Chem.* **2018**, *56*, 2133–2193. [[CrossRef](#)]
40. Brkić, D. Lambert W function in hydraulic problems. *Math. Balk.* **2012**, *26*, 285–292.
41. Brkić, D. W solutions of the CW equation for flow friction. *Appl. Math. Lett.* **2011**, *24*, 1379–1383. [[CrossRef](#)]
42. Čojašić, Ž.; Brkić, D. Very accurate explicit approximations for calculation of the Colebrook friction factor. *Int. J. Mech. Sci.* **2013**, *67*, 10–13. [[CrossRef](#)]
43. Brkić, D.; Čojašić, Ž. Intelligent flow friction estimation. *Comput. Intell. Neurosci.* **2016**, *2016*, 5242596. [[CrossRef](#)] [[PubMed](#)]
44. Brkić, D. Discussion of “Gene expression programming analysis of implicit Colebrook–White equation in turbulent flow friction factor calculation” by Saeed Samadianfar [J. Pet. Sci. Eng. 92–93 (2012) 48–55]. *J. Pet. Sci. Eng.* **2014**, *124*, 399–401. [[CrossRef](#)]
45. Praks, P.; Brkić, D. Symbolic regression-based genetic approximations of the Colebrook equation for flow friction. *Water* **2018**, *10*, 1175. [[CrossRef](#)]
46. Ettema, R. Hunter Rouse—His work in retrospect. *J. Hydraul. Eng.* **2006**, *132*, 1248–1258. [[CrossRef](#)]
47. Brkić, D. Can pipes be actually really that smooth? *Int. J. Refrig.* **2012**, *35*, 209–215. [[CrossRef](#)]
48. Brkić, D. A note on explicit approximations to Colebrook’s friction factor in rough pipes under highly turbulent cases. *Int. J. Heat Mass Transf.* **2016**, *93*, 513–515. [[CrossRef](#)]
49. Eaton, J.W. GNU Octave Manual, U.K., Bristol: Network Theory Ltd. 1997. Available online: [Ftp://ftp.eeng.dcu.ie/pub/ee454/cygwin/usr/share/doc/octave-2.1.73/pdf/Octave-FAQ.pdf](ftp://ftp.eeng.dcu.ie/pub/ee454/cygwin/usr/share/doc/octave-2.1.73/pdf/Octave-FAQ.pdf) (accessed on 13 June 2019).
50. Horchler, A.D. WrightOmegaq: Complex Double-Precision Evaluation of the Wright Omega Function, a Solution of $W + \text{LOG}(W) = Z$. Version 1.0, 3-12-13. Available online: <https://github.com/horchler/wrightOmegaq> (accessed on 13 June 2019).
51. Serghides, T.K. Estimate friction factor accurately. *Chem. Eng. N. Y.* **1984**, *91*, 63–64.
52. Khdhr, F.W.; Saeed, R.K.; Soleymani, F. Improving the computational efficiency of a variant of Steffensen’s method for nonlinear equations. *Mathematics* **2019**, *7*, 306. [[CrossRef](#)]
53. Behl, R.; Salimi, M.; Ferrara, M.; Sharifi, S.; Alharbi, S.K. Some real-life applications of a newly constructed derivative free iterative scheme. *Symmetry* **2019**, *11*, 239. [[CrossRef](#)]
54. Shacham, M. Comments on: “An explicit equation for friction factor in pipe”. *Ind. Eng. Chem. Fundam.* **1980**, *19*, 228–229. [[CrossRef](#)]
55. Romeo, E.; Royo, C.; Monzón, A. Improved explicit equations for estimation of the friction factor in rough and smooth pipes. *Chem. Eng. J.* **2002**, *86*, 369–374. [[CrossRef](#)]
56. Offor, U.H.; Alabi, S.B. An accurate and computationally efficient explicit friction factor model. *Adv. Chem. Eng. Sci.* **2016**, *6*, 66711. [[CrossRef](#)]
57. Barry, D.A.; Parlange, J.Y.; Li, L.; Prommer, H.; Cunningham, C.J.; Stagnitti, F. Analytical approximations for real values of the Lambert W-function. *Math. Comput. Simul.* **2000**, *53*, 95–103. [[CrossRef](#)]
58. Boyd, J.P. Global approximations to the principal real-valued branch of the Lambert W-function. *Appl. Math. Lett.* **1998**, *11*, 27–31. [[CrossRef](#)]
59. Demir, S.; Duman, S.; Demir, N.M.; Karadeniz, A. An MS Excel add-in for calculating Darcy friction factor. *Spreadsheets Educ.* **2018**, *10*, 2. Available online: <https://sie.scholasticahq.com/article/4669> (accessed on 12 May 2019).
60. Fukushima, T. Precise and fast computation of Lambert W-functions without transcendental function evaluations. *J. Comput. Appl. Math.* **2013**, *244*, 77–89. [[CrossRef](#)]
61. Kouznetsov, D. Doya Function and Tania Function. Available online: <http://www.ils.uec.ac.jp/~dima/PAPERS/2013taniadoya.pdf> (accessed on 12 May 2019).
62. Stewart, S. A new elementary function for our curricula? *Aust. Sr. Math. J.* **2005**, *19*, 8–26. Available online: <https://files.eric.ed.gov/fulltext/EJ720055.pdf> (accessed on 12 May 2019).
63. Pal, R. Teaching Fluid Mechanics and Thermodynamics Simultaneously through Pipeline Flow Experiments. *Fluids* **2019**, *4*, 103. [[CrossRef](#)]

64. Barreras, F.; López, A.M.; Lozano, A.; Barranco, J.E. Experimental study of the pressure drop in the cathode side of air-forced open-cathode proton exchange membrane fuel cells. *Int. J. Hydrol. Energy* **2011**, *36*, 7612–7620. [[CrossRef](#)]
65. Brkić, D. Comments on “Experimental study of the pressure drop in the cathode side of air-forced open-cathode proton exchange membrane fuel cells” by Barreras et al. *Int. J. Hydrol. Energy* **2012**, *37*, 10963–10964. [[CrossRef](#)]



© 2019 by the authors. 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 (<http://creativecommons.org/licenses/by/4.0/>).