

MATLAB for Iteration: Hydraulic modeling for environmental engineering students

Abstract

The wide spread of obtainable, personal computers has opened doors to vast computational capabilities which enforced educational systems especially engineering programs to develop proficiency in using computers along with experiments and theories. Teaching techniques have changed accordingly, and appropriate methods were required. An example of that development was shown in the course of Hydraulics given to Environmental Engineering undergraduate students. Students in the past spent much time on the iterative solving method used in hydraulic problems and expended a lot of energy to find the solutions. Today instead, a commonly known program called MATLAB has proven efficiency and brought great easiness which not only assisted engineers in their work but also allowed them to drive more attention on reasoning and interpretation. In this regard, three examples of hydraulic problems related to determining the minimum diameter of a pipe, the depth of uniform flow in an open channel, and the hydraulic load over a sharp-crested weir are explained with fully executable MATLAB codes for the solution. This study requires basic understanding of Hydraulics and familiarity with MATLAB. It benefits any branch of engineering concerned with hydraulics.

Keywords: iterative-based problems, hydraulics, MATLAB, engineering problems, calculations, solving methods

Volume 3 Issue 1 - 2019

Tasnim Almoulki,¹ Kaan Yetilmezsoy²

¹Undergraduate Student, Yildiz Technical University, Turkey

²Associate Professor, Yildiz Technical University, Turkey

Correspondence: Kaan Yetilmezsoy, Associate Professor, Department of Environmental Engineering, Faculty of Civil Engineering, Yildiz Technical University, Davutpasa Campus, 34220, Esenler, Istanbul, Turkey, Tel +902123835376, Email yetilmez@yildiz.edu.tr, kyetilmezsoy@gmail.com

Received: March 14, 2019 | Published: March 22, 2019

Introduction

The importance and advantages of technology and computers are familiar to everyone. In fact, not implementing computer-based tasks and technological activities in our educational systems would be a contradictory behavior to the rhythm of our everyday-life. Thus, it is essential to teach students about computational methods, modeling knowledge and simulation experience that will allow them to gain keen insights on the real-world problems that they will face in their future careers.^{1,2}

When solving engineering problems (especially iterative-based problems) using conventional hand calculations, time is consumed significantly, and big amounts of energy are expended. Furthermore, consistent results become elusive when the numbers of inputs increase, or simple blunders arise.

Therefore, instead of getting trapped in such endless cycles, and wasting time on the solving process itself, computer programs provide an alternative for such calculations and eventually, great easiness for engineers. This way, more emphasis can be placed on the interpretation of data and the development of intuition and common sense among students.³

In this study, the solution of three hydraulic iterative-based problems is shown using one of the most important programs among engineers which is MATLAB. This program helps obtaining numerous, accurate and quick values for the solutions.

Representation of iterative hydraulic problems

The term Hydraulics is related to the application of Fluid Mechanics' principles to water engineering structures, and civil and environmental engineering facilities. The main parameters of a hydraulic study are the geometry of the channel, the properties of the flowing fluid and the flow parameters.^{4,5} For the following examples, primary recognition of MATLAB is considered in addition to the

governing system of equations used in Hydraulics since our aim here is to show a methodology rather than discuss prior explanations.

Problem 1: Determining the minimum diameter of a pipe (Figure 1).

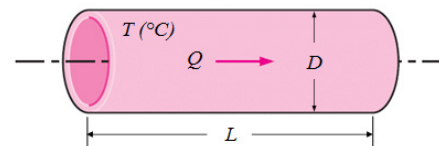


Figure 1 Schematic for Problem 1. Water at 20°C is to be transported in a 120-m-long plastic pipeline at a flow rate of 0.25m³/s. The roughness height for this pipe's material is 0.0015mm. If the head loss in the pipe is not supposed to exceed 10m, determine the minimum diameter (D) of the pipe to ensure the assigned value of the maximum head loss.

Problem 2: Determining the depth of uniform flow in an open channel (Figure 2).

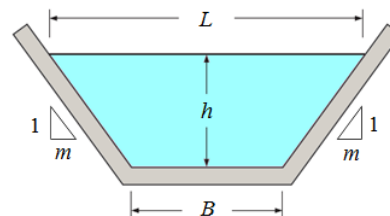


Figure 2 Schematic for Problem 2. The flow rate passing through an open finished-concrete channel of a trapezoidal cross section is 6.4m³/sec. The side surface slope (m) is 2:1 and the energy grade line slope (bed slope for normal flow) is 1.4m/km. The Manning coefficient for an open channel with finished-concrete surfaces is 0.012, and the bottom width of this channel is 4 m. The flow is steady and uniform, and the bottom slope is constant.⁵ The roughness of the wetted surface of the channel and the friction coefficient are assumed to be constant. Determine the depth of uniform flow and other dimensions of the channel.

Problem 3: Determining the hydraulic load over a sharp-crested weir (Figure 3).

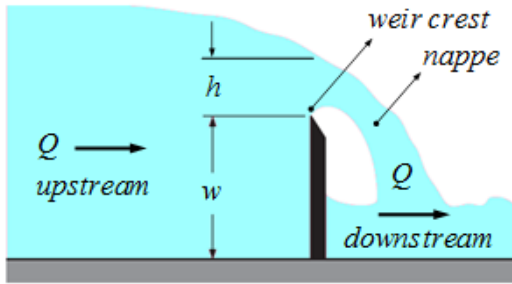


Figure 3 Schematic for Problem 3. A sharp-crested weir is a vertical plate placed in a channel that forces the liquid to flow through an opening to measure the flow rate.⁵ The flow rate passing over a sharp-crested Bazin weir⁷ is 2.2m³/s. If the height of this weir (w) that will be located into a 3-m wide rectangular channel is 3.2m, determine the hydraulic load (h) or the effective head flowing over the weir crest.

Computational procedure

The use of computer graphic visualization and animation is an effective method for displaying large volumes of model input and output.⁸ In this study, computer-based solutions are conducted by writing scripts in the M-file Editor within the framework of: MATLAB® R2018a software (V9.4.0.813654, 64-bit (win64), Academic License Number: 40578168, MathWorks Inc., Natick, MA) running on a Casper Excalibur (Intel® Core™ i7-7700HQ CPU, 2.81GHz, 16GB of RAM, 64-bit) Windows 10 PC platform. For a visual understanding of the implemented computational procedure, pictorial diagrams for each hydraulic problem are depicted in (Figure 4–6).

MATLAB-based solutions

Calculations are conducted for each problem by running the MATLAB scripts presented in (Table 1–3) of the Appendix. The scripts are evaluated in the Editor Window and the outputs for various input variables are shown in (Figure 7–9).

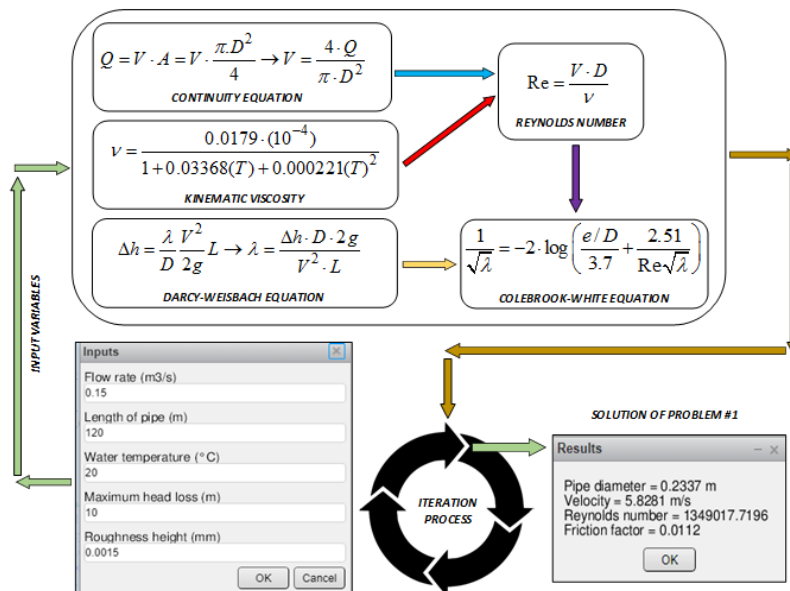


Figure 4 A visual representation of the solution methodology for problem 1.

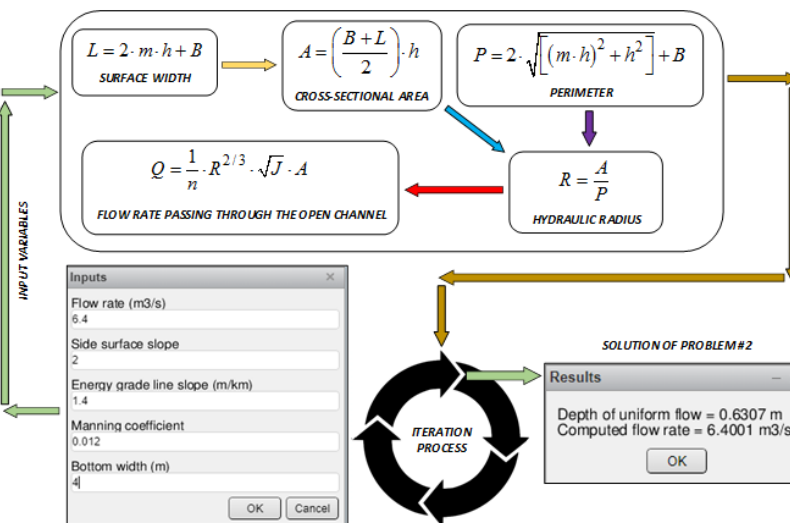


Figure 5 A visual representation of the solution methodology for problem 2.

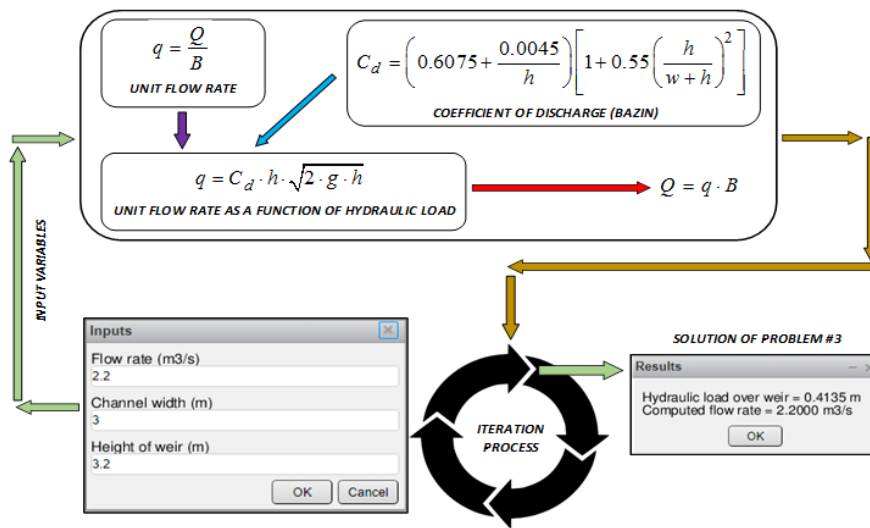


Figure 6 A visual representation of the solution methodology for problem 3.

Table 1 Script for determining the minimum diameter of the pipe

```
%Determining the minimum diameter of pipe
%STEP #1
clear; clc
format short
disp(datestr(now,0))

%STEP #2
a1 = {'Flow rate (m3/s)', 'Length of pipe (m)', ...
      'Water temperature (°C)', 'Maximum head loss (m)', ...
      'Roughness height (mm)'};
a2 = 'Inputs';
a3 = [1 40; 1 40; 1 40; 1 40; 1 40];
a4 = inputdlg(a1,a2,a3);
tic

%STEP #3
b = char(a4);
c = str2num(b);
Q = c(1); L = c(2); T = c(3); h = c(4); e = c(5);
e1 = e./1000;
nu = (0.01779./(1+0.03368.*T+0.000221.*(T.^2))).*(10.^(-4));
g = 9.807;

%STEP #4
Dp = 0.001;
EI = 0;
E = 1;
while EI < E
    V = (4*Q)/(pi.*(Dp.^2));
    Re = (V*Dp)./nu;
    f = (h.*Dp.*(2.*g))/(L.*(V.^2));
    EI = sqrt(f)*(-2)*log10(((e1./Dp)./3.7)+(2.51./(Re.*sqrt(f))));
    Dp = Dp + 0.000001;
end

%STEP #5
%Convert numbers to character representation
Dp_s = sprintf('Pipe diameter = %1.4f m', Dp);
V_s = sprintf('Velocity = %1.4f m/s', V);
Re_s = sprintf('Reynolds number = %1.4f', Re);
f_s = sprintf('Friction factor = %1.4f', f);
d = char(Dp_s, V_s, Re_s, f_s);
msgbox(d, 'Results');
toc
```

Table 2 Script for determining the depth of uniform flow in an open channel

```
%Determining the depth of uniform flow in an open channel
%STEP #1
clear; clc
format short
disp(datestr(now,0))

%STEP #2
a1 = {'Flow rate (m3/s)', 'Side surface slope', ...
      'Energy grade line slope (m/km)', 'Manning coefficient', ...
      'Bottom width (m)'};
a2 = 'Inputs';
a3 = [1 40; 1 40; 1 40; 1 40; 1 40];
a4 = inputdlg(a1,a2,a3);
tic

%STEP #3
b = char(a4);
c = str2num(b);
Q = c(1); m = c(2); J = c(3); n = c(4); B = c(5);
J1 = J./1000;

%STEP #4
h = 0.01;
Q1 = 0;
while Q1 < Q
    L = 2.*m.*h+B;
    A = ((B+L)./2).*h;
    P = abs(m.*h+h.*i).*2+B;
    R = A./P;
    Q1 = (1./n).*(R.^(2./3)).*sqrt(J1).*A;
    h = h + 0.00001;
end

%STEP #5
%Convert numbers to character representation
h_s = sprintf('Depth of uniform flow = %1.4f m', h);
Q1_s = sprintf('Computed flow rate = %1.4f m3/s', Q1);
d = char(h_s, Q1_s);
msgbox(d, 'Results');
toc
```

Table 3 Script for determining the hydraulic load over a sharp-crested weir

```
%Determining the hydraulic load over a sharp-crested weir
%STEP #1
clear, clc
format short
disp(datestr(now,0))

%STEP #2
a1 = {'Flow rate (m3/s)', 'Channel width (m)', 'Height of weir (m)'};
a2 = 'Inputs';
a3 = [1 40; 1 40; 1 40];
a4 = inputdlg(a1,a2,a3);
tic

%STEP #3
b = char(a4);
c = str2num(b);
Q = c(1); B = c(2); w = c(3);
q = Q./B;
g = 9.807;

%STEP #4
h = 0.01;
q1 = 0;
while q1 < q
part1 = 0.6075 + (0.0045./h);
part2 = 1 + 0.55.*((h.^2)./((w+h).^2));
Cd = part1.*part2;
q1 = Cd.*h.*sqrt(2.*g.*h);
h = h + 0.00001;
end

%STEP #5
%Convert numbers to character representation
h_s = sprintf('Hydraulic load over weir = %1.4f m',h);
Q1_s = sprintf('Computed flow rate = %1.4f m3/s',q1.*B);
d = char(h_s,Q1_s);
msgbox(d,'Results');
toc
```

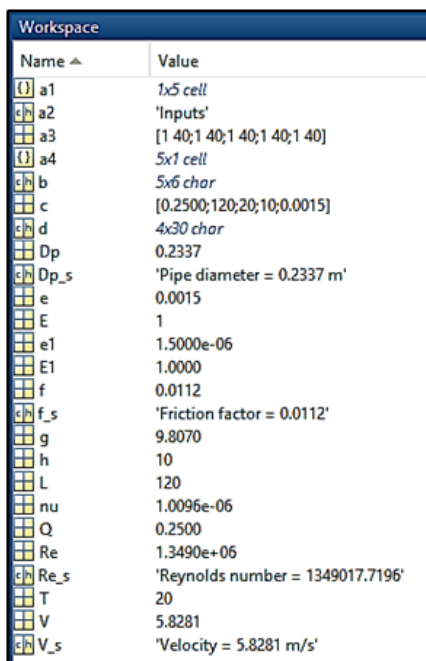


Figure 7 Contents of the workspace window after running of the script for problem 1.

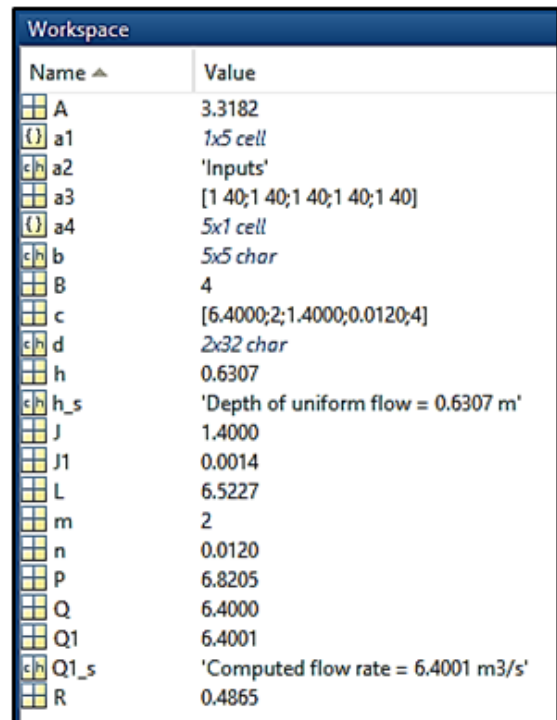


Figure 8 Contents of the workspace window after running of the script for problem 2.

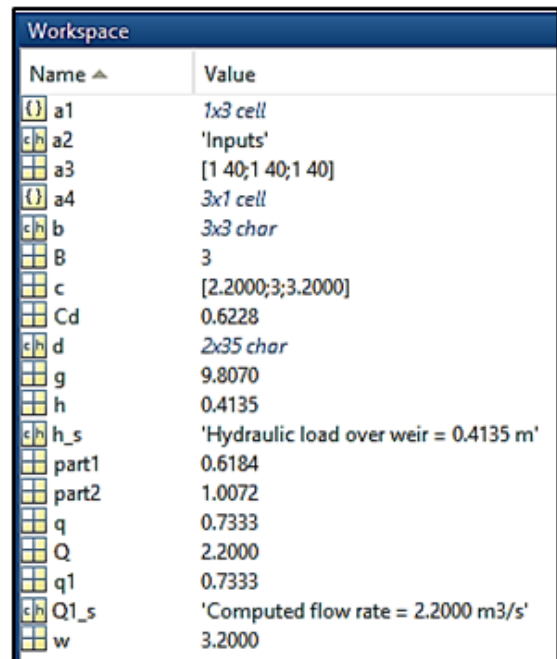


Figure 9 Contents of the workspace window after running of the script for problem 3.

Conclusion

By using the solving methods above, we reduced time, got more detailed, less error-prone solutions. And because of its accuracy, more comparisons and design trials can be performed. The solution time for the studied hydraulic cases was obtained as 0.2374, 0.2053, and 0.1446 (seconds) respectively, which reveals that the proposed MATLAB methodology gave quantitative results in a computational

time that doesn't exceed a few seconds. As a conclusion, using a program such as MATLAB in hydraulic problems gives us a chance to explore more design options in which allows us to find the best and most efficient design every time.

Appendix

Fully executable MATLAB codes of the three iterative-based problems are presented in the following three tables (Table 1–3).

Acknowledgments

None.

Conflicts of interest

The authors declare there is no conflicts of interest.

References

1. Yetilmezsoy K, Mungan CE. MATLAB time-based simulations of projectile motion, pendulum oscillation, and water discharge. *European Journal of Physics*. 2018;39(6):1–18.
2. Yetilmezsoy K. IMECE Implementation of mathematical, experimental, and computer-based education: A special application of fluid mechanics for civil and environmental engineering students. *Computer Applications in Engineering Education*. 2017;25(5):833–860.
3. Chapra SC, Canale RP. *Numerical methods for engineers*. Boston: McGraw-Hill Higher Education. 2010.
4. Chanson H. *Hydraulics of open channel flow: An introduction*. Elsevier. 2004.
5. Çengel YA, Cimbala JM. *Fluid mechanics fundamentals and applications*. New York: McGraw-Hill. 2006.
6. Kijjarvi J. Darcy friction factor formulae in turbulent pipe flow. 2011.
7. Wielogorski, JWJ. *Influence of a Dip Plate on the Discharge of Water over a Sharp Crested Rectangular Weir*. PhD Thesis. University of Surrey. 1981.
8. Tsanis I, Wu J, Shen H, et al. *Environmental hydraulics: Hydrodynamic and pollutant transport models of lakes and coastal waters*. Elsevier. 2007.