Hydraulics and Water Resources: Examples Using R

true

2022-11-18

Contents

#

Preface

This is a compilation of various R exercises and examples created over many years. They have been used mostly in undergraduate civil engineering classes including fluid mechanics, hydraulics, and water resources. This is a dynamic work, and will be regularly updated as errors are identified, improved presentation is developed, or new topics or examples are introduced. I welcome any suggestions or comments.

In what follows, text will be intentionally brief. More extensive discussion and description can be found in any fluid mechanics, applied hydraulics, or water resources engineering text. Symbology in this reference generally follows that Finnemore and Franzini (**?**). Fundamental equations will be introduced though the emphasis will be on applications to solve common problems. Also, since this is written by a civil engineer, the only fluids included are water and air, since that accounts for nearly all problems encountered in the field.

Solving water problems is rarely done by hand calculations, though the importance of performing order of magnitude 'back of the envelope' calculations cannot be overstated. Whether using a hand calculator, spreadsheet, or a programming language to produce a solution, having a sense of when an answer is an outlier will help catch errors.

Scripting languages are powerful tools for performing calculations, providing a fully traceable and reproducible path from your input to a solution. Open source languages have the benefit of being free to use, and invite users to be part of a community helping improve the language and its capabilities. The language of choice for this book is R (**?**), chosen for its straightforward syntax, powerful graphical capabilities, wide use in engineering and in many other disciplines, and by using the RStudio interface, it can look and feel a lot like Matlab® with which most engineering students have some experience.

No introduction to R or RStudio is provided here. It is assumed that the reader has installed R (and RStudio), is comfortable installing and updating packages, and understands the basics of R scripting. A brief overview is provided here, aimed at students at Santa Clara University. More comprehensive materials are readily available, including An Introduction to R by the R Core Team. An excellent set of materials for a course Introduction to Programming with R by Stauffer et al. is also freely available and includes interactive exercises.

As I developed these exercises and text, I learned R through the work of many others, and the excellent help offered by skilled people sharing their knowledge on stackoverflow. The methods shown here are not the only ways to solve these problems, and users are invited to share alternative or better solutions.

Copyright

Chapter 1

Units in Fluid Mechanics

Before beginning with problem solving methods it helps to recall some important quantities in fluid mechanics and their associated units. While the world has generally moved forward into standardizing the use of the SI unit system, the U.S. stubbornly holds onto the antiquated US (sometimes called the British Gravitational, BG) system. This means practicing engineers must be familiar with both systems, and be able to convert between the two systems.

These important quantities are shown in Table **??**.

Dimensions and units for common quantities.

Quantity

Symbol

Dimensions

US (or BG) Units

SI Units

US to SI multiply by

Length

L

L

ft

m

0.3048

Acceleration

a

LT^{-2}

ft/s^2

m/s^2

0.3048

Mass

m

M

$slug$

kg

14.59

Force

F

F

lb

N

4.448

Density

ρ

ML^{-3}

$slug/ft^3$

kg/m^3

515.4

Energy/Work

FL

$ft \cdot lb$

$N \cdot m = joule(J)$

1.356

Flowrate

Q

L^3/T

ft^3/s=cfs

m^3/s

0.02832

Kinematic viscocity

ν

L^2/T

ft^2/s

m^2/s

0.0929

Power

FLT^{-1}

$ft \cdot lb/s$

$N \cdot m/s = watt(W)$

1.356

Pressure

p

FL^{-2}

$lb/in^2 = psi$

$N/m^2 = Pa$

6895

Specific Weight

γ

FL^{-3}

lb/ft^3

N/m^3

157.1

Velocity

V

LT^{-1}

ft/s

m/s

0.3048

(Dynamic) Viscocity

μ

FTL^{-2}

$lb \cdot s/ft^2$

$N \cdot s/m^2 = Pa \cdot s$

47.88

Volume

$\forall$

L^3

ft^3

m^3

0.02832

There are many other units that must be accommodated. For example, one may encounter the *poise* to describe (dynamic) viscosity ($1\ Pa*s = 10\ poise$), or the *stoke* for kinematic viscocity ($1\ m^2/s = 10^4\ stokes$). Many hydraulic systems use gallons per minute (gpm) as a unit of flow ($1\ ft^3/s = 448.8\ gpm$), and larger water systems often use millions of gallons per day (mgd) ($1\ mgd = 1.547\ ft^3/s$). For volume, the SI system often uses liters (l) instead of m^3 ($1\ m^3 = 1000\ l$).

One regular conversion that needs to occur is the translation between mass (m) and weight (W), where $W = mg$, where g is gravitational acceleration on the earth's surface: $g = 9.81\ m/s^2 = 32.2\ ft/s^2$. When working with forces (such as with momentum problems or hydrostatic forces) be sure to work with weights/forces, not mass.

It is straightforward to use conversion factors in the table to manipulate values between the systems, multiplying by the factor shown to go from US to SI units, or dividing to do the

$$1 * 10^{-6}\ m^2/s * \frac{1\ ft^2/s}{0.0929\ m^2/s} = 1.076 * 10^{-5}\ ft^2/s$$

Another example converts between two quantities in the US system: 100 gallons per minute to cfs:

$$100\ gpm * \frac{1\ cfs}{448.8\ gpm} = 0.223\ cfs$$

The *units* package in R can do these conversions and more, and also checks that conversions are permissible (producing an error if incompatible units are used).

```
units::units_options(set_units_mode = "symbols")
Q_gpm <- units::set_units(100, gallon/min)
Q_gpm
#> 100 [gallon/min]
Q_cfs <- units::set_units(Q_gpm, ft^3/s)
Q_cfs
#> 0.2228009 [ft^3/s]
```

Repeating the unit conversion of viscosity using the *units* package:

Example 1.1. Convert kinematic viscosity from SI to Eng units.

```
nu <- units::set_units(1e-6, m^2/s)
nu
#> 1e-06 [m^2/s]
units::set_units(nu, ft^2/s)
#> 1.076391e-05 [ft^2/s]
```

The *units* package also produces correct units during operations. For example, multiplying mass by g should produce weight.

Example 1.2. Using the *units* package to produce correct units during mathematical operations.

```
#If you travel at 88 ft/sec for 1 hour, how many km would you travel?
v <- units::set_units(88, ft/s)
t <- units::set_units(1, hour)
d <- v*t
d
#> 316800 [ft]
units::set_units(d, km)
#> 96.56064 [km]

#What is the weight of a 4 slug mass, in pounds and Newtons?
m <- units::set_units(4, slug)
g <- units::set_units(32.2, ft/s^2)
w <- m*g
#Notice the units are technically correct, but have not been simplified in this case
w
#> 128.8 [ft*slug/s^2]
#These can be set manually to verify that lbf (pound-force) is a valid equivalent
units::set_units(w, lbf)
#> 128.8 [lbf]
units::set_units(w, N)
#> 572.9308 [N]
```

Chapter 2

Properties of water (and air)

Fundamental properties of water allow the description of the forces it exerts and how it behaves while in motion. Many were listed in Chapter .

A summary of basic water properties, which vary with temperature, is shown in Table **??** for SI units and Table **??** for US (or Eng) units.

Water properties in SI units

Temp

Density

Spec_Weight

Viscosity

Kinem_Visc

Sat_VP

Surf_Tens

Bulk_Mod

C

kg m-3

N m-3

N s m-2

m2 s-1

Pa

N m-1

Pa

0

999.9

9.809×10^3

1.734×10^{-3}

1.734×10^{-6}

611.2

75.7×10^{-3}

2.02×10^9

5

1000

9.810×10^3

1.501×10^{-3}

1.501×10^{-6}

872.6

74.9×10^{-3}

2.06×10^9

10

999.7

9.807×10^3

1.310×10^{-3}

1.311×10^{-6}

1.228×10^3

74.2×10^{-3}

2.10×10^9

15

999.1

9.801×10^3

1.153×10^{-3}

1.154×10^{-6}

1.706×10^3

73.5×10^{-3}

2.14×10^9

20

998.2

9.793×10^3

1.021×10^{-3}

1.023×10^{-6}

2.339×10^3

72.7×10^{-3}

2.18×10^9

25

997.1

9.781×10^3

910.8×10^{-6}

913.5×10^{-9}

3.170×10^3

72.0×10^{-3}

2.22×10^9

30

995.7

9.768×10^3

817.4×10^{-6}

821.0×10^{-9}

4.247×10^3

71.2×10^{-3}

2.25×10^9

35

994.1

9.752×10^3

738.0×10^{-6}

742.4×10^{-9}

5.629×10^{3}

70.4×10^{-3}

2.26×10^{9}

40

992.2

9.734×10^{3}

669.9×10^{-6}

675.1×10^{-9}

7.385×10^{3}

69.6×10^{-3}

2.28×10^{9}

45

990.2

9.714×10^{3}

611.2×10^{-6}

617.3×10^{-9}

9.595×10^{3}

68.8×10^{-3}

2.28×10^{9}

50

988.1

9.693×10^{3}

560.5×10^{-6}

567.2×10^{-9}

12.35×10^{3}

67.9×10^{-3}

2.29×10^{9}

55

985.7

9.670×10^{3}

516.2×10^{-6}

523.7×10^{-9}

15.76×10^{3}

67.1×10^{-3}

2.28×10^{9}

60

983.2

9.645×10^{3}

477.6×10^{-6}

485.7×10^{-9}

19.95×10^{3}

66.2×10^{-3}

2.28×10^{9}

65

980.6

9.619×10^{3}

443.5×10^{-6}

452.3×10^{-9}

25.04×10^{3}

65.4×10^{-3}

2.26×10^{9}

70

977.7

9.592×10^{3}

413.5×10^{-6}

422.9×10^{-9}

31.20×10^{3}

64.5×10^{-3}

2.25×10^{9}

75

974.8

9.563×10^3

386.9×10^{-6}

396.9×10^{-9}

38.60×10^3

63.6×10^{-3}

2.22×10^9

80

971.7

9.533×10^3

363.1×10^{-6}

373.7×10^{-9}

47.42×10^3

62.7×10^{-3}

2.20×10^9

85

968.5

9.501×10^3

341.9×10^{-6}

353.0×10^{-9}

57.87×10^3

61.8×10^{-3}

2.17×10^9

90

965.2

9.468×10^3

322.9×10^{-6}

334.5×10^{-9}

70.18×10^3

60.8×10^{-3}

2.14×10^9

95

961.7

9.434×10^3

305.7×10^{-6}

317.9×10^{-9}

84.61×10^3

59.9×10^{-3}

2.10×10^9

100

958.1

9.399×10^3

290.2×10^{-6}

302.9×10^{-9}

101.4×10^3

58.9×10^{-3}

2.07×10^9

Water properties in US units

Temp

Density

Spec_Weight

Viscosity

Kinem_Visc

Sat_VP

Surf_Tens

Bulk_Mod

F

slug ft-3

lbf ft-3

lbf s ft-2

ft2 s-1

lbf ft-2

lbf ft-1

lbf ft-2

32

1.938

62.42

36.21×10^{-6}

18.73×10^{-6}

12.77

5.18×10^{-3}

42.2×10^{6}

42

1.939

62.43

30.87×10^{-6}

15.96×10^{-6}

18.94

5.13×10^{-3}

43.1×10^{6}

52

1.938

62.40

26.58×10^{-6}

13.75×10^{-6}

27.62

5.07×10^{-3}

44.0×10^{6}

62

1.937

62.36

23.11×10^{-6}

11.96×10^{-6}

39.64

5.02×10^{-3}

45.0×10^{6}

72

1.934

62.29

20.26×10^{-6}

10.50×10^{-6}

56.00

4.96×10^{-3}

45.9×10^{6}

82

1.932

62.20

17.90×10^{-6}

9.290×10^{-6}

77.99

4.90×10^{-3}

46.7×10^{6}

92

1.928

62.09

15.94×10^{-6}

8.286×10^{-6}

107.2

4.84×10^{-3}

47.2×10^{6}

102

1.925

61.97

14.29×10^{-6}

7.443×10^{-6}

145.3

4.78×10^{-3}

47.5×10^{6}

112

1.920

61.83

12.89×10^{-6}

6.732×10^{-6}

194.7

4.72×10^{-3}

47.7×10^{6}

122

1.916

61.68

11.71×10^{-6}

6.126×10^{-6}

258.0

4.66×10^{-3}

47.8×10^{6}

132

1.911

61.52

10.69×10^{-6}

5.608×10^{-6}

338.1

4.59×10^{-3}

47.7×10^{6}

142

1.905

61.34

9.808×10^{-6}

5.162×10^{-6}

438.5

4.53×10^{-3}

47.5×10^{6}

152

1.899

61.16

9.046×10^{-6}

4.775×10^{-6}

563.2

4.46×10^{-3}

47.2×10^{6}

162

1.893

60.96

8.381×10^{-6}

4.438×10^{-6}

716.9

4.39×10^{-3}

46.8×10^{6}

172

1.887

60.75

7.797×10^{-6}

4.144×10^{-6}

904.5

4.32×10^{-3}

46.2×10^{6}

182

1.880

60.53

7.283×10^{-6}

3.884×10^{-6}

1.132×10^{3}

4.25×10^{-3}

45.5×10^{6}

192

1.873

60.30

6.828×10^{-6}

3.655×10^{-6}

1.405×10^{3}

4.18×10^{-3}

44.8×10^{6}

202

1.865

60.06

6.423×10^{-6}

3.452×10^{-6}

1.731×10^{3}

4.11×10^{-3}

44.0×10^{6}

212

1.858

59.81

6.061×10^{-6}

3.271×10^{-6}

2.118×10^{3}

4.04×10^{-3}

43.2×10^{6}

What follows is a brief discussion of some of these properties, and how they can be applied in R. All of the properties shown in the tables above are produced