

6-29-2016

Simple Models for Underdamped Slug Tests in High Permeability Aquifers

Maria E. Marquez

Florida International University, mmarq002@fiu.edu

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>



Part of the [Geology Commons](#), and the [Hydrology Commons](#)

Recommended Citation

Marquez, Maria E., "Simple Models for Underdamped Slug Tests in High Permeability Aquifers" (2016).
FIU Electronic Theses and Dissertations. 2570.
<https://digitalcommons.fiu.edu/etd/2570>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

SIMPLE MODELS FOR UNDERDAMPED SLUG TESTS IN HIGH PERMEABILITY
AQUIFERS

A thesis submitted in partial fulfillment of

the requirements for the degree of

MASTER OF SCIENCE

in

GEOSCIENCES

by

Maria E. Marquez

2016

To: Dean Michael R. Heithaus
College of Arts, Sciences, and Education

This thesis, written by Maria E. Marquez, and entitled Simple Models for Underdamped Slug Tests in High Permeability Aquifers, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

Grenville Draper

Julian Edward

Michael C. Sukop, Major Professor

Date of Defense: June 29, 2016

The thesis of Maria E. Marquez is approved.

Dean Michael R. Heithaus
College of Arts, Sciences and Education

Andrés G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2016

DEDICATION

I dedicate this thesis to my children, Gabriel Roberts and Marie Roberts. Their love, inspiration, and patience made this work possible.

ACKNOWLEDGMENTS

I wish to thank my major professor, Dr. Michael Sukop, for his guidance, support, and mentorship. This work would not be possible if not for his insight and encouragement. I also wish to thank my committee members Dr. Grenville Draper and Dr. Julian Edward for their patience and willingness to impart their knowledge time and again.

I also wish to express my gratitude to Florida International University's Graduate Minority Opportunity Program fellowship, which supported my travel to conferences to present this research and obtain valuable feedback and experience.

I wish to also thank Diane Pirie, Vanessa Londono, and Alejandro Garcia for their assistance in carrying out the necessary slug tests, without which this research would not have been possible.

Finally, I wish to thank my family for their unwavering love and support.

ABSTRACT OF THE THESIS
SIMPLE MODELS FOR UNDERDAMPED SLUG TESTS IN HIGH PERMEABILITY
AQUIFERS

by

Maria E. Marquez

Florida International University, 2016

Miami, Florida

Professor Michael C. Sukop, Major Professor

Accurate hydraulic conductivity values are necessary for understanding groundwater flow. Methods for estimating hydraulic conductivity show limitations because measured values vary several orders of magnitude in high permeability aquifers. Slug tests, while cost and time efficient, result in values lower than expected. It is proposed that underdamped behavior of water in a well is similar to mass on a damped spring; hence, models constructed to simulate behavior independent of aquifer effects might replicate some tests. The Poiseuille and Darcy-Weisbach models, and extensions of these models considering entry/exit effects, are applied to an aquifer-free laboratory test, and real wells. Aquifer-free laboratory tests are modeled well using both Poiseuille and Darcy-Weisbach models with entry/exit effects. The Poiseuille model for wells does not agree with observed data, possibly because of high Reynolds numbers. The Darcy-Weisbach model does agree with well data significantly better, although the friction factor relies on a single Reynolds number.

TABLE OF CONTENTS

CHAPTER	PAGE
1 INTRODUCTION	1
1.1 Biscayne Aquifer.....	2
1.2 Slug Tests	4
1.3 Prior Studies	4
1.4 Preliminary Research	6
1.4.1 FIU Nature Preserve Slug Tests.....	6
1.4.2 Underdamped Slug Test Results from the literature.....	8
2 AIMS/OBJECTIVES OF RESEARCH.....	10
3 MODELS	10
3.1 Classical Damped Spring Model.....	11
3.2 Relating Classical Model to Pipe Model.....	12
3.3 Poiseuille Model.....	14
3.4 Darcy-Weisbach Model.....	16
3.5 Entry/Exit Losses Models	18
4 METHODS	19
4.1 Aquifer-free Laboratory Experimental Design	19
4.2 Snapper Creek Wells.....	22
5 RESULTS	23
5.1 Aquifer-free Laboratory Data.....	23
5.1.1 Poiseuille with Entry/Exit Losses Model.....	23
5.1.2 Darcy-Weisbach entry/exit losses Model	27
5.2 Snapper Creek Wells.....	28
5.2.1 Poiseuille Model	28
5.2.2 Darcy-Weisbach Model	30
6 DISCUSSION.....	31
7 CONCLUSIONS.....	33
LIST OF REFERENCES.....	35
APPENDICES	38

LIST OF TABLES

TABLE	PAGE
Table 1. Maximum Biscayne Aquifer hydraulic conductivities for different test methods.	5
Table 2. Hydraulic conductivities based on slug tests in the Nature Preserve in the Biscayne aquifer.....	8
Table 3. Literature maximum hydraulic conductivities from underdamped slug tests.	9
Table 4. Summary of Test values	25
Table 5. Comparison of normal and adjusted kinematic viscosities for Poiseuille model applied to wells	29
Table 6. Comparison of normal and adjusted friction factor for Darcy Weisbach model applied to wells	31

LIST OF FIGURES

FIGURE	PAGE
Figure 1. Map depicting areal extent of Biscayne Aquifer in South Florida from USGS http://maps.waterdata.usgs.gov/mapper/index.html	2
Figure 2. Sample of Miami Limestone touching-vug facies. Note ~2-cm scale of many vugs. (Photo by Mike Wacker, USGS)	3
Figure 3. Typical slug test type-curve result for FIU Nature Preserve study. 80ft depth, 8 inch diameter well, with slug generated by 3 inch pipe.	7
Figure 4. Definition diagram for L, r, and w variables	13
Figure 5. Moody Diagram. Friction factor as a function of the Reynolds number. (Matos and Valerio, 2009.)	17
Figure 6. Aquifer-free end member laboratory experiment displaying entry/exit head loss	18
Figure 7. Laboratory aquifer-free tank and submerged pipe	20
Figure 8. Laboratory experiment tank with pipe and slug	21
Figure 9. Still images from videos of tests	21
Figure 10. Aerial photograph of Snapper Creek Well Field showing location of wells used in this study	23
Figure 11. Aquifer-free 0.01 m pipe laboratory data and Poiseuille model (test 7713) without entry/exit losses (blue line) and with entry/exit losses (red line)	26
Figure 12. Aquifer-free 0.024 m pipe laboratory data and Poiseuille model (test 7717) without entry/exit losses (blue line) and with entry/exit losses (red line)	26
Figure 13. Aquifer-free 0.01 m pipe laboratory data and Darcy-Weisbach model (test 7713) without entry/exit losses (blue line) and with entry/exit losses (red line)	27
Figure 14. Aquifer-free 0.024 m pipe laboratory data and Darcy-Weisbach model (test 7717) without entry/exit losses (blue line) and with entry/exit losses	

(red line).....	28
Figure 15. Poiseuille Model USGS Well G3902. Blue dots show observed peak displacement (in meters) as a function of time (seconds). Red line shows Poiseuille based model with standard water viscosity. Blue line shows model with adjusted viscosity for improved fit.	29
Figure 16. Darcy Weisbach Model USGS Well G3902. Blue dots show observed peak displacement (in meters) as a function of time (seconds). Red line shows maximum-velocity-based Reynolds number model. Blue line shows model with adjusted friction factor for improved fit.	30
Figure 17. Comparison of Darcy-Weisbach Model with L vs. L+w as seen in test 7740.....	32

NOTATIONS

damping coefficient	c
damping ratio	ζ
density	ρ
diameter of pipe or well	d
displacement from the equilibrium water level	w
dynamic viscosity	μ
effective length of the water column in the pipe	L
Friction factor	f_D
gravitational acceleration	g
Hooke's Law constant	k_H
hydraulic conductivity	K
kinematic viscosity	ν
minor head loss	h
minor losses coefficient	k
Pressure loss	ΔP
radius of pipe or well	r
Reynolds number	Re
spring natural frequency	ω
time	t
velocity	u

1 INTRODUCTION

An aquifer system's hydraulic conductivity provides essential information necessary for understanding the flow of groundwater. Accurate representation of an aquifer's hydraulic conductivity values is essential in the development of reliable models used to determine groundwater behavior.

Methods for estimating the hydraulic conductivity vary, and include aquifer tests, Lattice Boltzmann model cores, specialized laboratory cores, and slug tests. Slug tests are both cost and time efficient. Estimating hydraulic conductivity in an aquifer system through slug tests is a fast and simple technique. It requires inducing a change in the elevation of the water level in a monitoring well, often by inserting or withdrawing a slug into or from the well, then measuring the change in water level with time as it equilibrates again. These well response tests were initially developed by Hvorslev (1951). The data are compiled and fitted curves are used to estimate the hydraulic conductivity.

Well response tests in highly permeable aquifers behave similarly to a mass in a viscous medium attached to a damped spring, where the water level oscillates around the equilibrium, and is characterized as an underdamped oscillator (Cooper et al 1967, van der Kamp 1976).

Van der Kamp (1976) noted that his own theory of the underdamped case must assume "exponentially damped cyclic fluctuation", so the well response test analysis may not be reliable for other damped fluctuations. Building off van der Kamp's work and the work of Shinohara and Ramey (1979) based on Laplace transforms, Kipp (1985) solved for a more complete solution for various cases, including the underdamped case, while

Butler (2000) developed type curves to analyze the hydraulic conductivity in highly permeable aquifers.

1.1 Biscayne Aquifer

The Biscayne Aquifer in Florida is a significant and prolific aquifer system (Figure 1). An eastward-thickening wedge of limestone, up to 240 feet thick along Biscayne Bay. Composed principally of Miami Limestone and the Fort Thompson Formation/Tertiary Quaternary Shelly Unit, the Biscayne Aquifer serves Miami and Fort Lauderdale metropolitan areas as the primary source of drinking water.

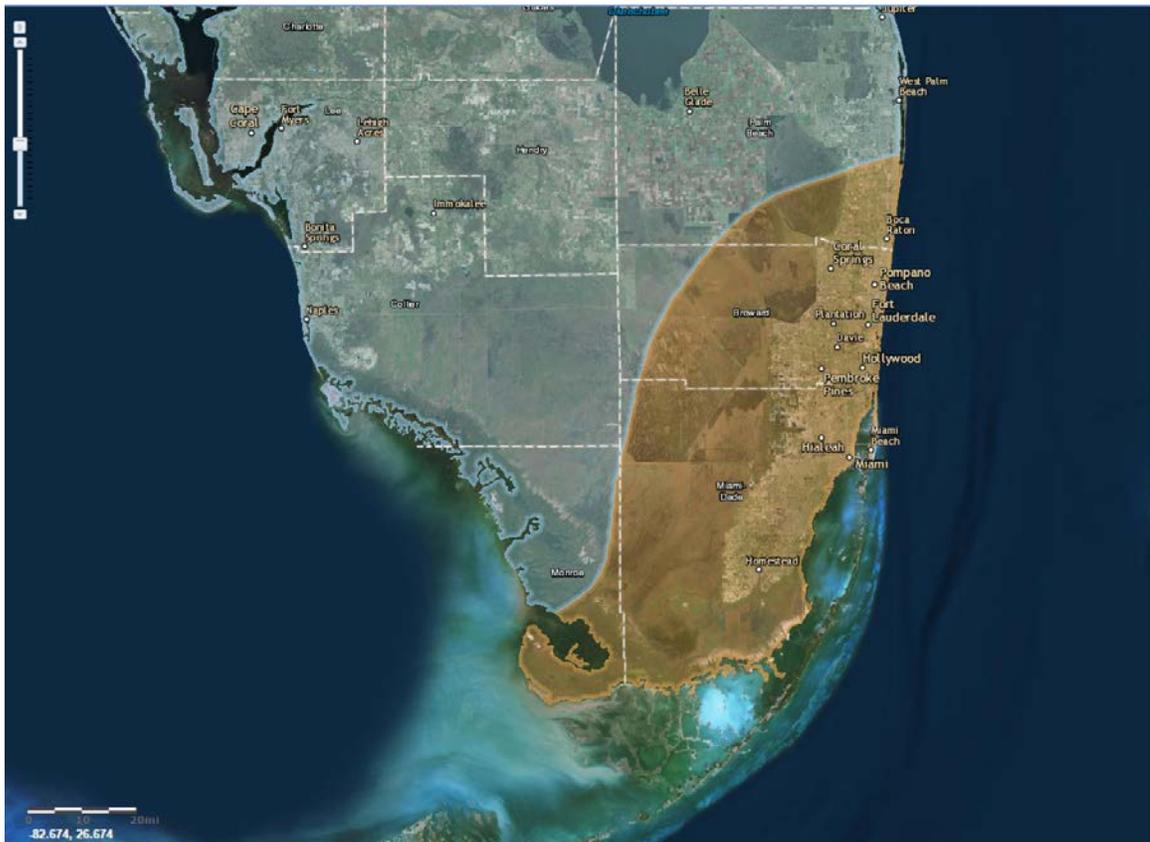


Figure 1. Map depicting areal extent of Biscayne Aquifer in South Florida from USGS <http://maps.waterdata.usgs.gov/mapper/index.html>

In this study area, the Fort Thompson Formation is characterized largely by medium to very thick beds of limestone, aranceous limestone, and some beds of skeletal quartz sandstone, along with minor quartz sand. (Wacker et al. 2014) The uniqueness of this aquifer is a result of its high permeability. Its karst is characterized by thick and laterally extensive zones of vug-to-vug connecting porosity created by burrowing shrimp in a shallow water marine environment (Figure 2). The vugs are commonly 2 centimeters in diameter and the porosity is frequently 50% or more. Recognizing that there is no single value for the hydraulic conductivity of the Biscayne Aquifer, the widespread existence of the extreme permeability, large diameter touching-vug facies has repeatedly been observed and its relevance to the overall hydraulic conductivity of the aquifer has been demonstrated (Sukop and Cunningham, 2014).



Figure 2. Sample of Miami Limestone touching-vug facies. Note ~2-cm scale of many vugs. (Photo by Mike Wacker, USGS)

1.2 Slug Tests

A slug test is an aquifer characterization method used to estimate transmissivity and hydraulic conductivity of a geologic unit. Transmissivity is a measure of the amount of water that can be transmitted through a unit width of the aquifer under a hydraulic gradient of one (Schwartz and Zhang, 2003). Transmissivity is used to calculate hydraulic conductivity when the aquifer thickness is known.

Advantages of slug tests include cost efficiency, quick test time, and the lack of contamination spread risk as there is not any water to dispose from the test (Weight and Wittman, 1999). Slug tests are performed by suddenly inducing a change in the water level in a well (van der Kamp, 1976). The slug, a known volume of water or a solid volume, is suddenly injected or removed from the well. The resulting displacement of the water level is measured as a function of time as it returns to the equilibrium water level. Hvorslev (1951) originally used the straight-line method to estimate the hydraulic conductivity of the aquifer based on the ratio of drawdown. There are two conditions that can lead to oscillation; either the aquifer transmissivity is very high or the oscillating mass is large (generally due to a long water column). Aquifers with high permeability often result in underdamped slug tests with oscillatory behavior, while overdamped slug tests generally result in aquifers with lower permeability.

1.3 Prior Studies

Studies of the Biscayne Aquifer have frequently been unable to determine the value of transmissivity. Maximum hydraulic conductivity (K) values in the Biscayne Aquifer obtained from five methods, including detailed Lattice Boltzmann Modeling

(LBM) at pore scale, laboratory measurements at core scale, borehole-scale LBM measurements, borehole-scale slug testing, and aquifer test meta-analysis are reviewed in Table 1. There are 5 orders of magnitude variations in the maximum hydraulic conductivity values obtained from these different techniques. Some of this is a result of real variations in the specific aquifer materials tested, but there is great disagreement of the hydraulic conductivity values among the conventional methods that leaves the true values undetermined. The exceptionally low core K values are readily explained by limits of the methods applied to create that data set. The Lattice Boltzmann and specialized laboratory methods consistently give $K > 10 \text{ m s}^{-1}$ and aquifer tests are generally inconclusive (Fish and Stewart, 1991) but indicate $K > 0.1 \text{ m s}^{-1}$. These results raise concerns as to why slug tests results are consistently lower than results from aquifer tests and Lattice Boltzmann simulations.

Table 1. Maximum Biscayne Aquifer hydraulic conductivities for different test methods.

Measurement	K (m s^{-1})
Standard laboratory cores (Cunningham et al., 2006)	<0.0005
3-D printed laboratory core with viscous fluid and LBM (Garcia, 2013)	15
LBM ‘Cores’ (Sukop et al., 2013; Cunningham et al., 2009)	50
17-m scale LBM from borehole images (Sukop and Cunningham, 2014)	50
Underdamped slug tests (Wacker et al., 2014)	0.03
Aquifer tests (Fish and Stewart, 1991)	>0.1

Some would argue that high permeability aquifers, which are characterized by large connected pores, conduits of various sizes, and potentially large-aperture fractures, fall outside the realm of applicability of traditional Darcy Law-based models. However, Darcy-based models are widely used in many critical applications such as assessing seawater intrusion and water supply in South Florida and most other areas. Only recently have extensions to these models incorporating non-linear effects been developed (e.g., Mayaud et al., 2014; Shoemaker et al., 2007 and 2008) and applications of the extensions are in their infancy (e.g., Gallegos et al., 2013; Reimann et al., 2011).

1.4 Preliminary Research

1.4.1 FIU Nature Preserve Slug Tests

In 2013, I performed a preliminary study of slug tests using uncased wells in the Biscayne Aquifer at Florida International University's Nature Preserve. A replicated series of slug tests were conducted on three adjacent unscreened wells of varying depths. A 1.25 inch, 2 inch, and 3 inch diameter pipe were used. Each of the pipes was submerged into the wells with a cap in place at the top. Once the water level stabilized, a perturbation was imposed by rapidly removing the cap from the top of the pipe. Water then flowed into the submerged pipe and reduced the water level in the well. Oscillations in the water level in the wells were monitored with a data logger, and the data were analyzed using the methods of Butler and Garnett (2000). Figure 3 shows a typical fit of

the data to a type curve of Butler et al., (2003). Though the fits are not perfect, the hydraulic conductivity of the aquifer was initially considered effectively determined.

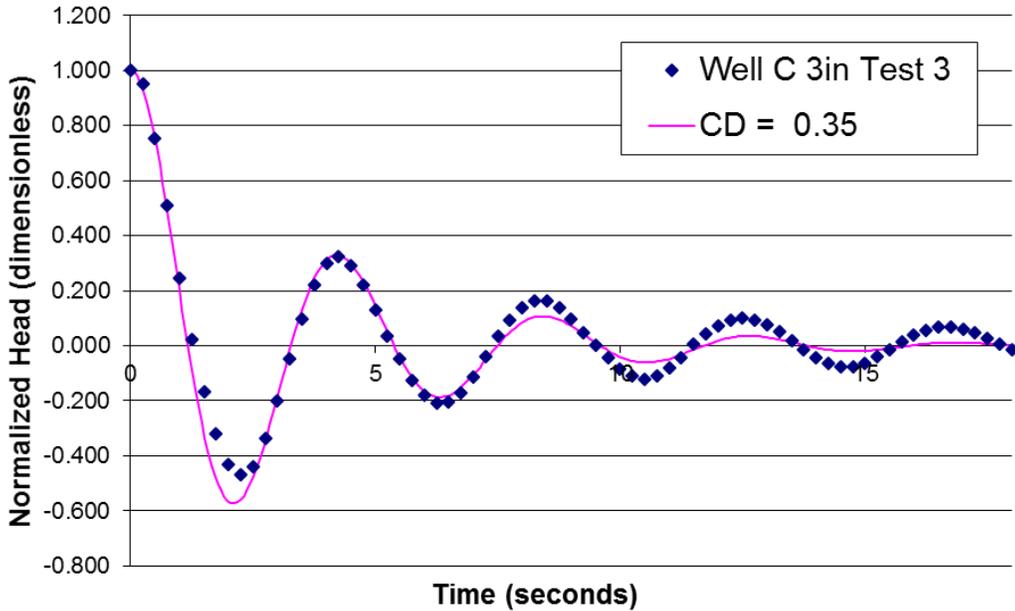


Figure 3. Typical slug test type-curve result for FIU Nature Preserve study. 80ft depth, 8 inch diameter well, with slug generated by 3 inch pipe.

Hydraulic conductivity values yielded mostly consistent results when analyzed using Butler type curves (Table 2). These hydraulic conductivity values are much lower in comparison to aquifer test values, Lattice Boltzmann model “cores” and specialized laboratory cores in Table 1. The slug test hydraulic conductivity values represent what literature often indicates as hydraulic conductivities comparable to those of well-sorted sand, rather than limestone (Freeze and Cherry, 1979). Consistent with observations from Butler (1998), these discrepancies in the hydraulic conductivity values further support reassessment of conventional slug test analysis methods and their reliability and possible limitations for accurately determining a highly permeable aquifer’s hydraulic conductivity.

Table 2. Hydraulic conductivities based on slug tests in the Nature Preserve in the Biscayne aquifer

Hydraulic conductivity (m/s)	80 ft. Well	40 ft. Well	20 ft. Well
3 in Pipe	0.013 +/- 0.0003	0.012 +/-0.0007	0.016 +/-0.0003
2 in Pipe	0.014 +/- 0.0001	0.015 +/-0.0004	0.016 0.0001 m/s
1.25 in Pipe	0.014 +/-0.0004	0.015 +/- 0.0002	0.0048 +/- 0.00007

A second set of slug tests were conducted using slugs constructed of pipes of different diameters capped on both ends and filled with water. Both slug insertion and slug removal tests were carried out. Oscillatory responses to the slugs were observed and recorded. However, despite the large volume of some of the slugs (0.026 m³), it was not possible to impose an equivalent water level perturbation on these wells due to the wells being open across the water table and most likely also due to the very high hydraulic conductivity: inserting or removing the slugs from these particular wells appears to be like dropping them into a swimming pool and the aquifer is effectively transparent.

1.4.2 Underdamped Slug Test Results from the literature

A review of hydraulic conductivities of oscillatory slug tests results from the literature is compiled in Table 3. These are maximum reported values. The Nature Preserve well results are on the high end and consistent with the recent results of Wacker et al., (2014). The highest reported values are all relatively low comparable to the

hydraulic conductivity of a well-sorted sand -- considerably lower than might be anticipated for many systems exhibiting oscillatory responses. Weight and Wittman (1999) concluded that “Slug tests tend to underestimate the hydraulic properties of aquifers”.

Black (2010) proposed that a practical upper limit for slug tests was $5 \times 10^{-1} \text{ m s}^{-1}$, though this argument is based on short test duration and the time needed to initiate a test. However, in the very high transmissivity cases of interest in this study, oscillation can persist for a long time.

Table 3. Literature maximum hydraulic conductivities from underdamped slug tests.

Source	Hydraulic Conductivity K (m s^{-1})
DiFrenna, 2005	0.00044
Audouin and Bodin, 2007	0.0006
Brauchler et al., 2010	0.0013
McElwee, 2002	0.0015
Ostendorf et al., 2005	0.0017
Zurbuchen et al., 2002	0.0024
Zemansky and McElwee, 2005	0.003
Kungui et al., 2000	0.01
Nature Preserve Slug tests, 2013	0.01
van der Kamp, 1976	0.02
Biscayne Aquifer results (Wacker et al 2014)	0.035

2 AIMS/OBJECTIVES OF RESEARCH

Existing analysis methods for underdamped slug tests appear to be inadequate to capture the true hydraulic conductivity of highly permeable materials. Re-examining oscillatory slug tests and developing an end-member numerical model to assess if the observations obtained from slug test measurements are representative of the hydraulic conductivity of a porous medium, rather than simply dependent on the flow in a pipe such as a well casing, will expose potential problems with the existing analyses, improve upon current aquifer testing methods, and allow for accurate hydraulic conductivity values to be measured. The hypothesis tested in this study is that water in a pipe, such as a well casing, behaves as a mass on a damped spring and its behavior can be predicted by a model independent of aquifer effects. Development of new models, and collection and analysis of laboratory and field measurements are used to test this hypothesis.

The development of a model using a Poiseuille damping parameter and a model using a Darcy-Weisbach damping parameter, and hybrids of these that incorporate entry/exit effects will allow for a thorough comparison that evaluates the conditions of aquifers and end-member lab experiments simulating an aquifer-free medium. The approach should give insight into current model deficiencies for highly permeable aquifers and eventually yield an approach more suitable for determining true hydraulic conductivities.

3 MODELS

To determine if the slug test method gives results that correspond to true hydraulic conductivity values of highly permeable aquifers, models were developed to

consider effects of water flowing through a well system without a porous medium. Such models can serve as a way to determine if aquifer effects are being measured at all in cases of highly permeable aquifers. Models were constructed based on simple force balance equations to predict the behavior of water as a mass on a damped spring in a pipe. To understand the validity of current slug test methods, the models were designed to be applied to real wells and to an aquifer-free laboratory tank.

In the following sections, the classical mass damped spring model is briefly reviewed and its connections with the new pipe model are explained.

3.1 Classical Damped Spring Model

In developing an appropriate numerical model for the oscillatory, underdamped case that represents highly permeable aquifer conditions, the classical mass damped spring model force balance equation begins with:

$$F_{total} = -F_{damping} - F_{spring} . \quad (1)$$

The total force is the product of the mass and acceleration and is given by

$$F_{total} = m \frac{d^2 w}{dt^2} , \quad (2)$$

where m is the mass, w is the displacement, and t is the time. The damping force is often proportional to velocity and can be written as

$$F_{damping} = c \frac{dw}{dt} , \quad (3)$$

where c is a damping coefficient. Finally, the spring force is proportional to the displacement according to Hooke's Law

$$F_{spring} = k_H w, \quad (4)$$

where k_H is the Hooke's Law constant. Inserting these expressions into Equation (1) leads to the full expression for the damped spring model:

$$m \frac{d^2 w}{dt^2} = -c \frac{dw}{dt} - k_H w. \quad (5)$$

The natural frequency ω_0 is

$$\omega_0 = \sqrt{\frac{k_H}{m}}, \quad (6)$$

Which is an angular frequency expressed as radians per second. Hence, to convert it to the period

$$Period = \frac{2\pi}{\sqrt{\frac{k_H}{m}}}. \quad (7)$$

The damping ratio ζ is given by

$$\zeta = \frac{c}{2\sqrt{mk_H}}, \quad (8)$$

which is the parameter that determines if the mass spring damper-system is overdamped ($\zeta > 1$), critically damped ($\zeta = 1$), or underdamped ($0 < \zeta < 1$). Overdamped and critically damped systems evolve monotonically towards their equilibrium positions while underdamped systems oscillate and are the primary interest in this thesis.

3.2 Relating Classical Model to Pipe Model

For development of the pipe models, assumptions include that the change in water level as a result of displacement (w) within the column of the pipe is much smaller than the overall water column length (L) in the pipe, and therefore negligible in the context of the water column length. Hence the mass of moving water is assumed constant. In the results section, a model is tested that considers the fully variable column length in order

to relax this assumption. In addition, previous work on slug tests has made use of an effective column length (L_e) that is computed as $L \frac{b}{2}$ where b is the aquifer thickness (eg. van der Kamp, 1976).

The force of the spring for the pipe submerged in water is constructed using Hooke's law where the weight of the displaced water is the force

$$F_{spring} = \pi r^2 \rho g w. \quad (9)$$

As shown in Figure 4, w is the displacement of the water level from its static equilibrium position and r is the radius of the pipe. The density is denoted by ρ and g is the gravitational acceleration. This equation indicates that the spring constant is $\pi r^2 \rho g$.

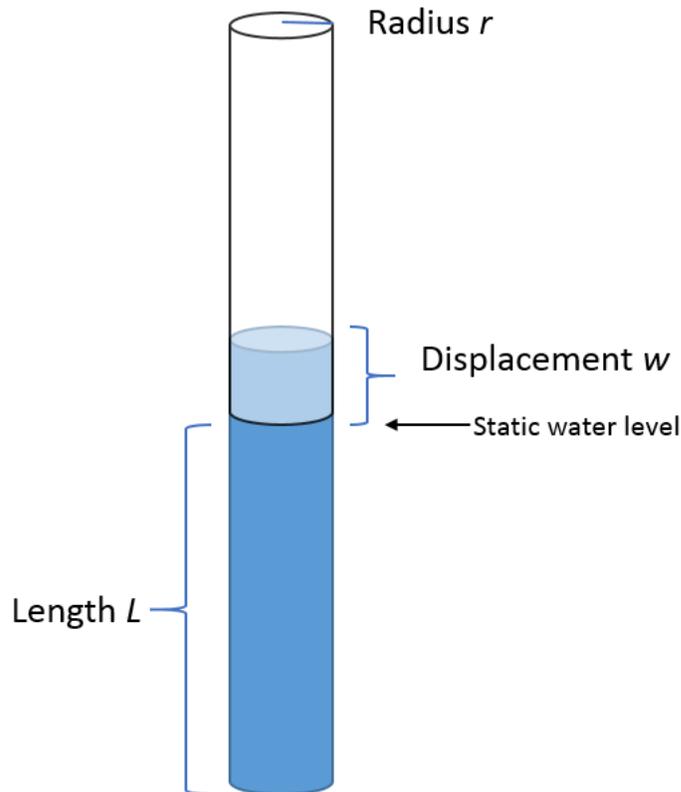


Figure 4. Definition diagram for L , r , and w variables

For the pipe model, the mass is $\pi r^2 \rho L$ and based on Equation (6) the natural

frequency is $\omega_0 = \sqrt{\frac{k_H}{m}}$ or

$$\omega_0 = \sqrt{\frac{\pi r^2 \rho g}{\pi r^2 \rho L}}, \quad (10)$$

which simplifies to

$$\omega_0 = \sqrt{\frac{g}{L}}. \quad (11)$$

A check of the period can determine if the model is appropriate. As before, the natural frequency is inverted and multiplied by 2π to obtain the period:

$$Period = \frac{2\pi}{\sqrt{\frac{g}{L}}}. \quad (12)$$

Similarly, the damping ratio $\zeta = \frac{c}{2\sqrt{mk_H}}$ can be expressed in terms of the pipe

parameters once a damping model is selected. In the following sections, Poiseuille-based and Darcy-Weisbach-based damping models.

3.3 Poiseuille Model

The following assumptions apply when using Poiseuille's Law: the fluid in the cylindrical pipe is Newtonian, incompressible, and the flow in the pipe is laminar and there is no acceleration of fluid in the pipe.

The damping force is considered using Poiseuille's law, where the average velocity across the cross-section of the pipe is given by

$$u_{avg} = \frac{\Delta P}{L} \frac{r^2}{8\mu}. \quad (13)$$

ΔP is the pressure drop over the length of pipe L , r is the pipe radius, and μ is the dynamic viscosity. Replacing u_{avg} with the time derivative of the water level displacement from equilibrium, replacing μ with the product of the density ρ and the kinematic viscosity ν , and re-arranging to solve for ΔP gives

$$\Delta P = \frac{8\nu\rho L}{r^2} \frac{dw}{dt}. \quad (14)$$

Force = Pressure \times area, and the damping force is given by

$$F_{damping} = \Delta P \pi r^2 = 8\pi\nu\rho L \frac{dw}{dt}. \quad (15)$$

The total force is from Newton's second law, $F_{total} = ma$, where the mass m of the water column in the pipe is $\rho\pi r^2 L$ and the acceleration a is the second derivative of the water level displacement $\frac{d^2w}{dt^2}$.

Inserting the Poiseuille equation for the laminar pipe flow damping force and expanding the individual force terms in Equation (1) results in the following model:

$$\rho\pi r^2 L \frac{d^2 w}{dt^2} = -8\pi\nu\rho L \frac{dw}{dt} - g\rho\pi r^2 w, \quad (16)$$

which reduces to:

$$\frac{d^2 w}{dt^2} = -\frac{8\nu}{r^2} \frac{dw}{dt} - \frac{g}{L} w. \quad (17)$$

3.4 Darcy-Weisbach Model

The Reynolds number represents the ratio of inertial forces to viscous forces and therefore characterizes flow as laminar, transitional, or turbulent. The Reynolds number is obtained from

$$\text{Re} = \frac{ud}{\nu}, \quad (18)$$

where u is velocity, d is a characteristic length (pipe diameter for pipe flows), and ν is the kinematic viscosity.

In using Poiseuille's law in the first model, flow is limited to the laminar regime. To account for test conditions with higher Reynold's numbers, the Darcy-Weisbach approach to damping force provides a friction factor coefficient for use in place of the Poiseuille damping parameter. The Darcy-Weisbach model constructed thus expresses flows in the turbulent regime with a Reynolds number higher than 2000. It replaces the Poiseuille pressure drop term (14) as

$$\Delta P = f_D \frac{L \rho u^2}{2d}, \quad (19)$$

where the friction factor (f_D) term is based on the calculated Reynolds number and the corresponding value on the Moody diagram (Figure 5). From the parameters derived from the force balance equation, the following equation comprises our numerical model in the Darcy-Weisbach case:

$$\frac{d^2 w}{dt^2} = -f_D \frac{1}{4r} \frac{dw}{dt} \left| \frac{dw}{dt} \right| - \frac{g}{L} w. \quad (20)$$

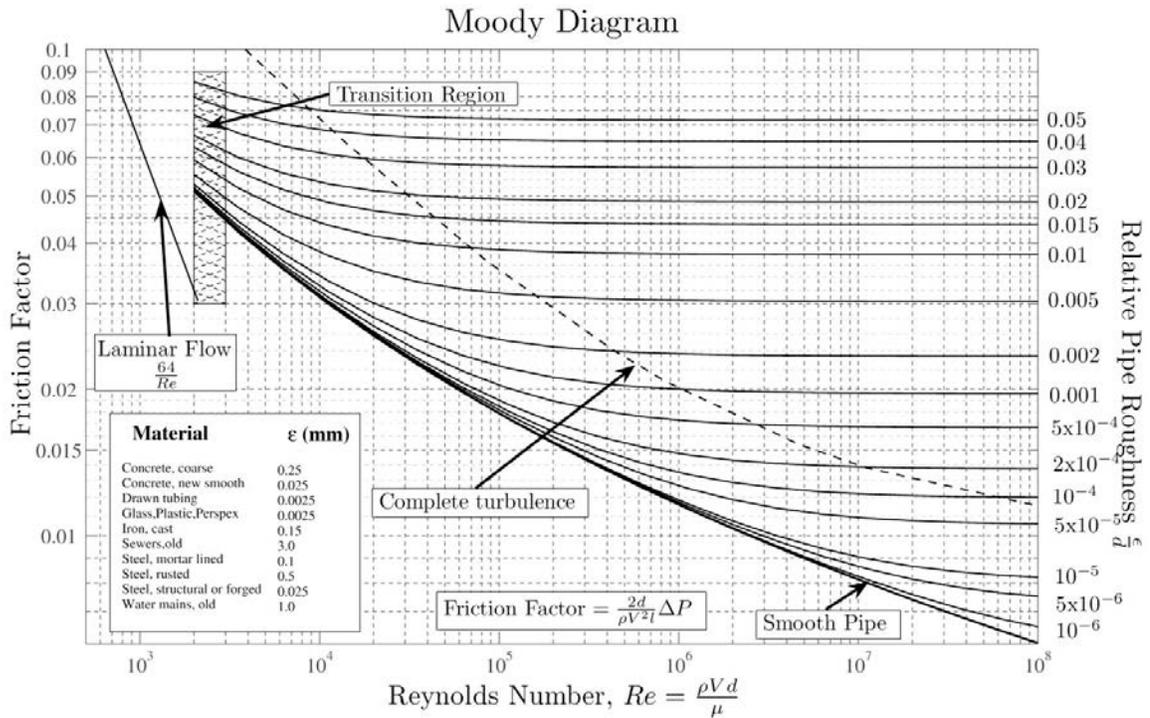


Figure 5. Moody Diagram. Friction factor as a function of the Reynolds number. (Matos and Valerio, 2009.)

3.5 Entry/Exit Losses Models

The Poiseuille model and the Darcy-Weisbach model are further developed in this section to consider the effects of kinetic energy loss, particularly for the end-member aquifer-free laboratory conditions where the pipe is submerged in a tank as seen in Figure 6.

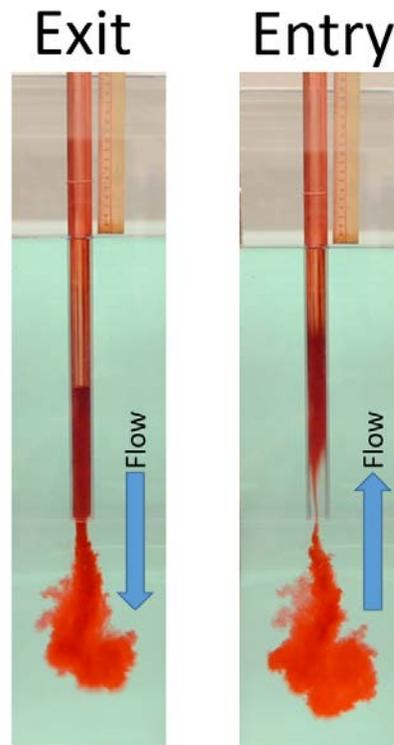


Figure 6. Aquifer-free end member laboratory experiment displaying entry/exit head loss

To account for these entry/exit losses, the model includes what are referred to in the literature as ‘minor’ losses, derived from the kinetic energy of the fluid, as an additional model term derived from the equation for minor head loss where

$$\Delta h = \frac{ku^2}{2g} \quad (21)$$

is the minor head loss. The minor loss coefficient $k = 1$ because the flow velocity drops to zero when the flow exits the pipe into the tank and has to accelerate from zero to enter

the pipe. Incorporating $\Delta P = \rho g \Delta h$ gives $\Delta P = \frac{\rho}{2} \frac{dw}{dt} \left| \frac{dw}{dt} \right|$ and, because

Force = pressure \times area, entry/exit terms $F_{entry/exit} = \pi r^2 \frac{\rho}{2} \frac{dw}{dt} \left| \frac{dw}{dt} \right|$, which then reduces

to the following for the Poiseuille case (22, 23) and the Darcy-Weisbach case (24):

$$\rho \pi r^2 L \frac{d^2 w}{dt^2} = -8 \pi \nu \rho L \frac{dw}{dt} - g \rho \pi r^2 w - \frac{\rho \pi r^2}{2} \frac{dw}{dt} \left| \frac{dw}{dt} \right| \quad (22)$$

$$\frac{d^2 w}{dt^2} = -\frac{8 \nu}{r^2} \frac{dw}{dt} - \frac{g}{L} w - \frac{1}{2L} \frac{dw}{dt} \left| \frac{dw}{dt} \right| \quad (23)$$

$$\frac{d^2 w}{dt^2} = -\frac{f_D}{4r} \frac{dw}{dt} \left| \frac{dw}{dt} \right| - \frac{g}{L} w - \frac{1}{2L} \frac{dw}{dt} \left| \frac{dw}{dt} \right|. \quad (24)$$

Note that in the Darcy-Weisbach case, this can be viewed as simply an additional coefficient preceding the velocity product term.

4 METHODS

4.1 Aquifer-free Laboratory Experimental Design

To apply the models to an aquifer-free end-member, a laboratory experiment was designed to produce a slug test in an open medium. The set up consisted of a partially-

submerged pipe in a 1.375 m \times 0.705 m \times 0.91 m tank of water. For the experiments, pipe sizes, submergence depths, and maximum displacements were varied. The two pipes used in the experiments were 108 cm in length and 0.02 m in diameter, and 112 cm in length and 0.047 m in diameter. The submergence depths were varied between 30 cm to 40 cm. The maximum displacement was varied between 5.4 cm to 6.5 cm. Details regarding the submergence depths of the pipe, the maximum displacements of the water level in the pipe, and the sizes of pipe used for each test are available in Table 4. A slug was induced with a vacuum hose at the open end of the top of the pipe as shown in Figure 7 and depicted in Figure 8.

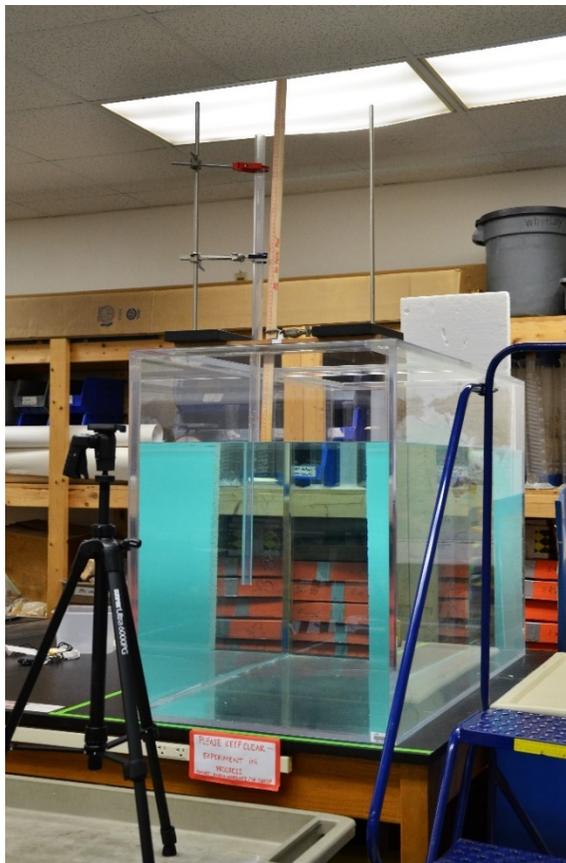


Figure 7. Laboratory aquifer-free tank and submerged pipe

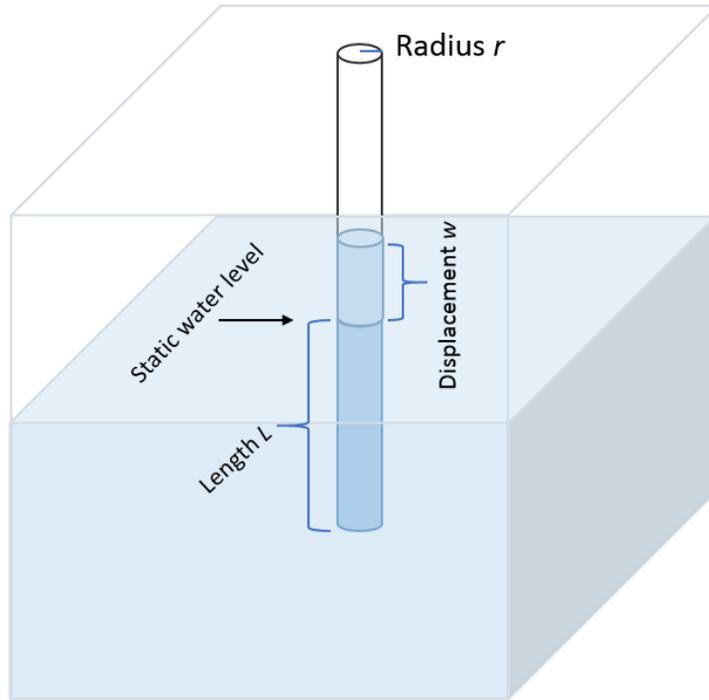


Figure 8. Laboratory experiment tank with pipe and slug

The water level changes were recorded with a digital single-lens reflex camera using the full HD video function with a frame rate of 24p, and a ruler in the tank was used to visually determine maximum displacement values over time from still frames using VideoPad editor (Figure 9). Food coloring and rubber bands on the pipe were sometimes used for visual aid.

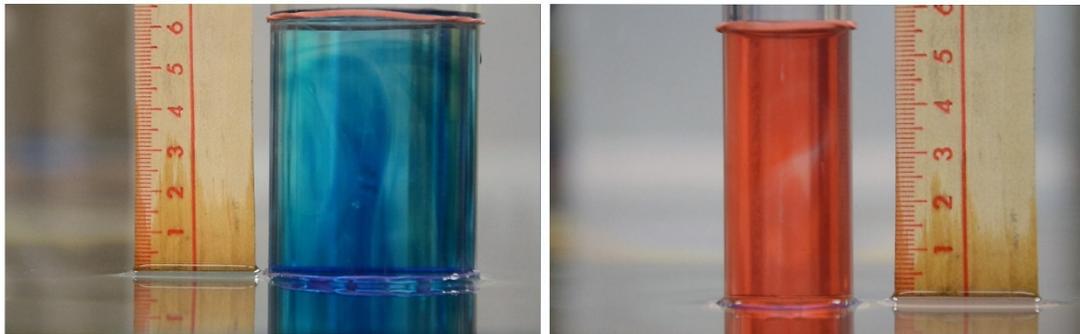


Figure 9. Still images from videos of tests

To determine the Reynolds number in the laboratory experiment, the still frame images depicting the last two water level displacements just before the water level passed the static water level were selected and a finite difference of their positions was used to calculate a velocity representing the maximum velocity. The calculated velocity was then used in Equation (18) to determine the Reynolds number, from which the friction factor f_D term was determined using the Moody Diagram (Figure 5). The results of each Reynolds number calculation can be seen in Table 4.

4.2 Snapper Creek Wells

Well data and slug test results were obtained from United States Geological Survey's (USGS) report, Geologic and Hydrologic Frameworks of the Biscayne Aquifer in Central Miami-Dade County, Florida (Wacker et al. 2014). The report's assessment site was the Snapper Creek Well Field (SCWF) (Figure 10). From the USGS SCWF reported results, wells with high hydraulic conductivity value, oscillatory response, and large effective length of water in the well pipe relative to the initial slug induced were identified and selected for model application. A summary of the periods of the oscillations for the wells was compiled (Appendix A). Time and displacement were adjusted for each well test to represent an oscillation that passed through zero, and from the data points nearest the crossing of the time axis, maximum velocity was calculated as a finite difference. For each test, the Reynolds number was calculated using the raw data points from the USGS report. The time and depth for two consecutive data points representing the water level as it passed through the static water level after the initial induction of the slug were used to calculate the maximum velocity as a finite difference.

The velocity determined was then used in Equation (18) to determine the Reynolds number for each test (see Table 4), and the necessary parameters for each model were calculated for use in Mathematica. The data from the selected wells were then analyzed to compute the parameters necessary to run each model, which are shown in Appendix B.

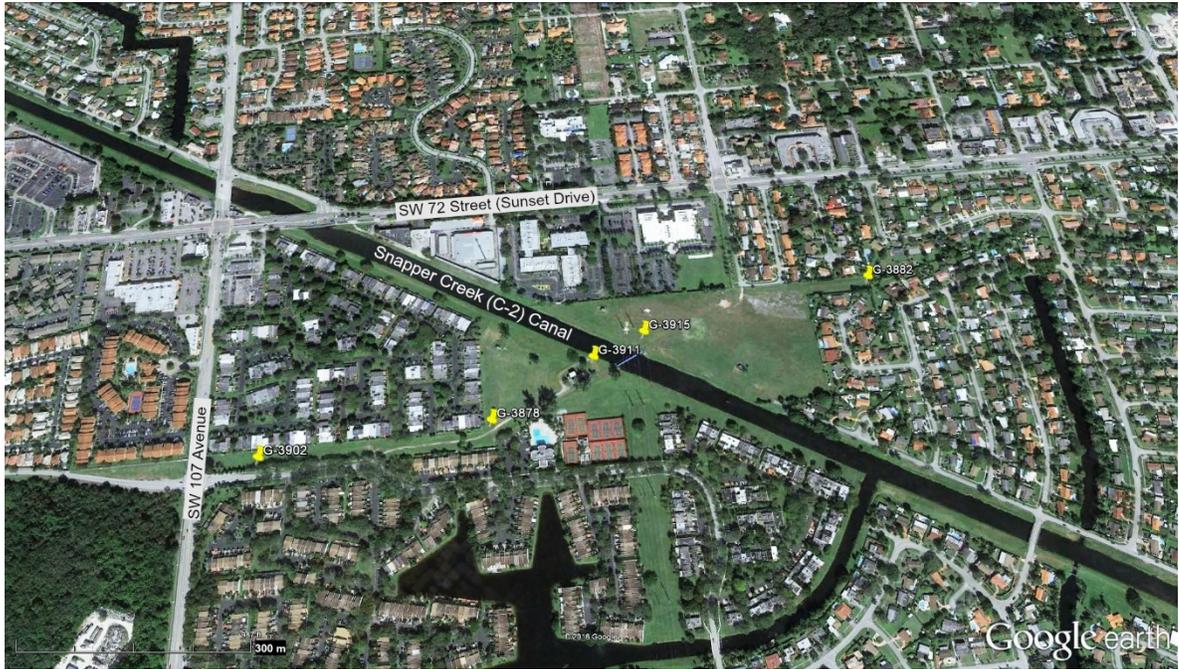


Figure 10. Aerial photograph of Snapper Creek Well Field showing location of wells used in this study

5 RESULTS

5.1 Aquifer-free Laboratory Data

5.1.1 Poiseuille with Entry/Exit Losses Model

Figure 11 and Figure 12 show data from two different sized pipes in the aquifer-free test tank depicted above, along with the results of the Poiseuille model both with and without entry/exit effects described above. While the Poiseuille model without entry/exit effects predicts significantly less damping of the water level oscillation in the larger pipe, the data and the model with entry/exit effects are similar for both pipes. Despite some

differences in the timing of the peaks that may be related to data collection and analysis, the Poiseuille model with entry/exit effects does a good job of capturing the decay of the oscillations in both cases, as seen in the agreement of the data observed and the entry/exits effects curve (Figure 11 and Figure 12). These results are similar to all of the aquifer-free test cases as shown in detail in Appendix C.

Table 4. Summary of Test values

Test #	L (m)	Le (m)	max displacement (m)	g/L (1/s ²)	g/Le (1/s ²)	Reported hydraulic conductivity in ft ² /d from USGS report	Radius of pipe (m)	Re	Expected flow type
Well Tests									
G3878	21.2	22.31	0.995	0.462	0.4398	8000	0.0508	58,852	turbulent
G3902	19.0	20.15	0.942	0.516	0.4869	9000	0.0508	59,441	turbulent
G3907	18.1	19.87	1.004	0.542	0.4937	2000	0.0508	51,034	turbulent
G3911	16.8	17.22	0.723	0.585	0.5697	9000	0.0508	39,976	turbulent
G3915	1.30	2.835	0.083	7.57	3.46	10000	0.0508	20,178	turbulent
Lab Tank									
7713	0.3		0.054	32.7		n/a	0.010	3,000	transitional
7717	0.3		0.063	32.7		n/a	0.0239	2,500	transitional
7725	0.3		0.058	32.7		n/a	0.0239	10,494	turbulent
7727	0.35		0.061	28.0		n/a	0.0239	11,448	turbulent
7732	0.4		0.057	24.5		n/a	0.0239	6,678	turbulent
7735	0.3		0.059	32.7		n/a	0.010	4,000	turbulent
7740	0.35		0.065	28.0		n/a	0.010	5,200	turbulent

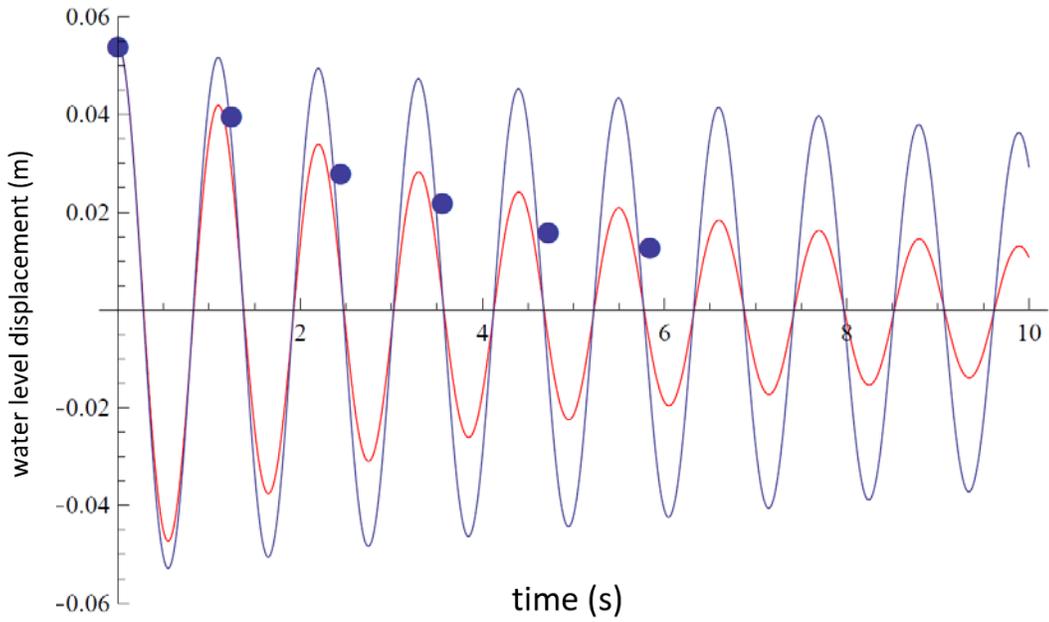


Figure 11. Aquifer-free 0.01 m pipe laboratory data and Poiseuille model (test 7713) without entry/exit losses (blue line) and with entry/exit losses (red line)

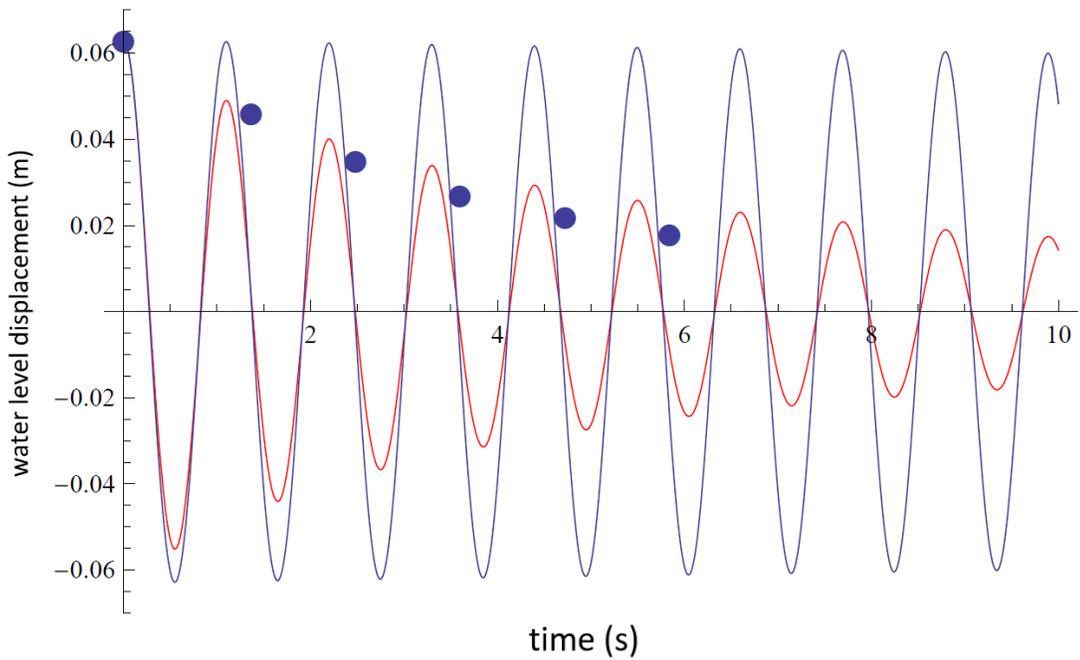


Figure 12. Aquifer-free 0.024 m pipe laboratory data and Poiseuille model (test 7717) without entry/exit losses (blue line) and with entry/exit losses (red line)

5.1.2 Darcy-Weisbach entry/exit losses Model

Figure 13 and Figure 14 show the Darcy-Weisbach model with and without entry/exit effects for the same data as above. The results are similar to those of the Poiseuille model above: the model with the entry/exit effects captures the essence of the aquifer-free data without parameter adjustment. Any difference between the large and small pipe is far less prominent than in the case using the Poiseuille model.

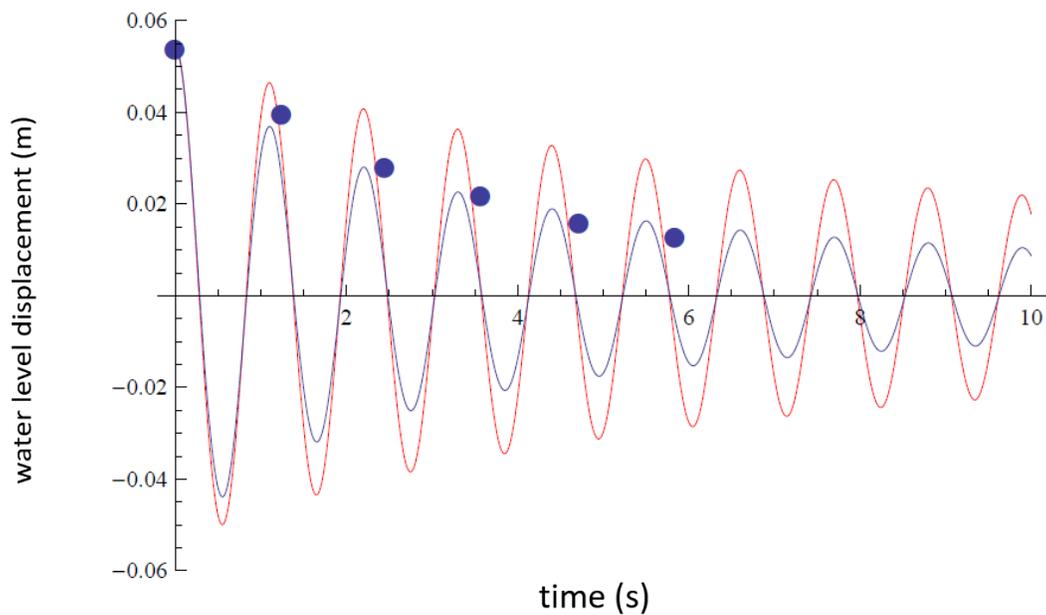


Figure 13. Aquifer-free 0.01 m pipe laboratory data and Darcy-Weisbach model (test 7713) without entry/exit losses (blue line) and with entry/exit losses (red line)

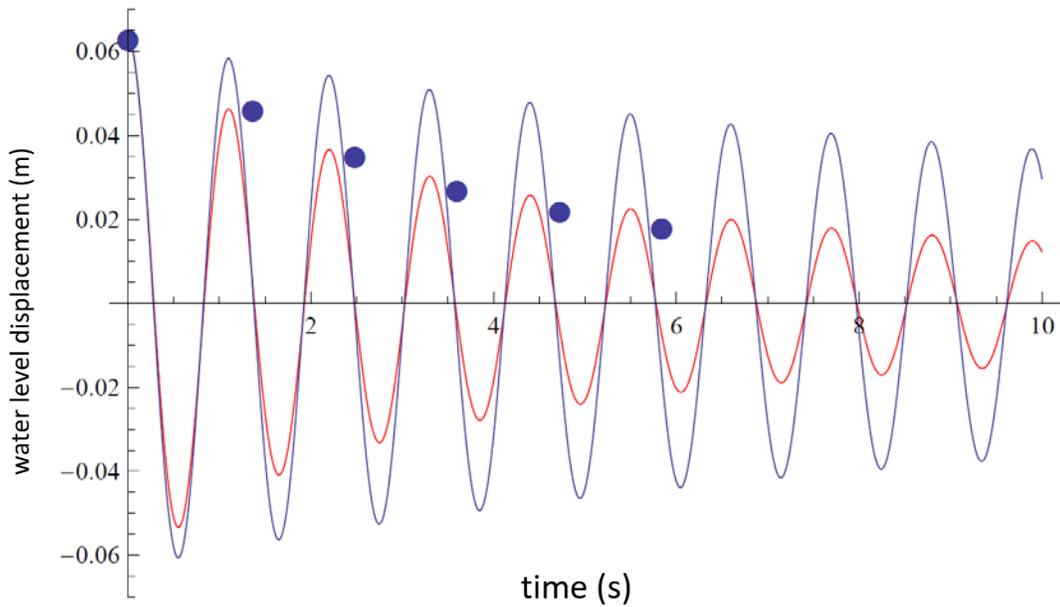


Figure 14. Aquifer-free 0.024 m pipe laboratory data and Darcy-Weisbach model (test 7717) without entry/exit losses (blue line) and with entry/exit losses (red line)

5.2 Snapper Creek Wells

5.2.1 Poiseuille Model

Figure 15 shows the observed peaks of the oscillations for well G3902 along with two Poiseuille-based models, and is typical for all of the comparable analyses for each of the USGS wells considered, as shown in detail in Appendix C. The Poiseuille model which uses standard water viscosity does not display the same degree of damping that the data indicate, which is not surprising because, based on the high Reynolds number, the flow conditions in the wells are turbulent, while the Poiseuille equation applies to laminar flow conditions. Therefore, the viscosity in the model was adjusted to obtain a better fit and the effective column length (L_e) was used to adjust the timing of the peaks. Table 5 shows the normal and adjusted kinematic viscosities, as well as the ratio between the two for each of the analyzed wells.

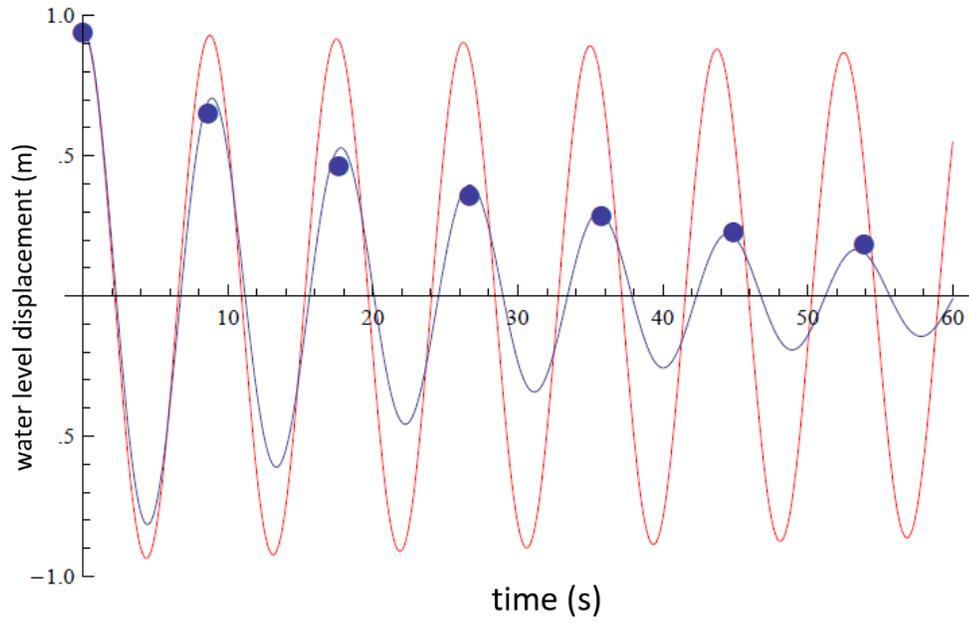


Figure 15. Poiseuille Model USGS Well G3902. Blue dots show observed peak displacement (in meters) as a function of time (seconds). Red line shows Poiseuille based model with standard water viscosity. Blue line shows model with adjusted viscosity for improved fit.

Table 5. Comparison of normal and adjusted kinematic viscosities for Poiseuille model applied to wells

Well number	Kinematic viscosity (m ² /s)	Adjusted kinematic viscosity (m ² /s)	Ratio of adjusted to normal viscosity
G3878	1E-06	1.9E-05	19
G3902	1E-06	2.1E-05	21
G3907	1E-06	5.8E-05	58
G3911	1E-06	5.8E-05	58
G3915	1E-06	1.8E-04	180

5.2.2 Darcy-Weisbach Model

Figure 16 shows the data for well G3902 with the Darcy-Weisbach numerical model applied and, as in the case of the Poiseuille model results for the wells, these results are typical for most of the analyses of similar data, shown in detail in Appendix C. The fit of the unadjusted model (red line) is substantially better than that of the unadjusted Poiseuille model above. The better fit of the Darcy-Weisbach model is also evident by the ratios of the adjusted to unadjusted friction factors (f_D), seen in Table 6, which are much smaller than the viscosity ratios in Table 5 for the Poiseuille model. Consequently, the Darcy-Weisbach model appears to be superior despite the fact that it is based on a single f_D value that corresponds to a single constant velocity through the Reynolds number.

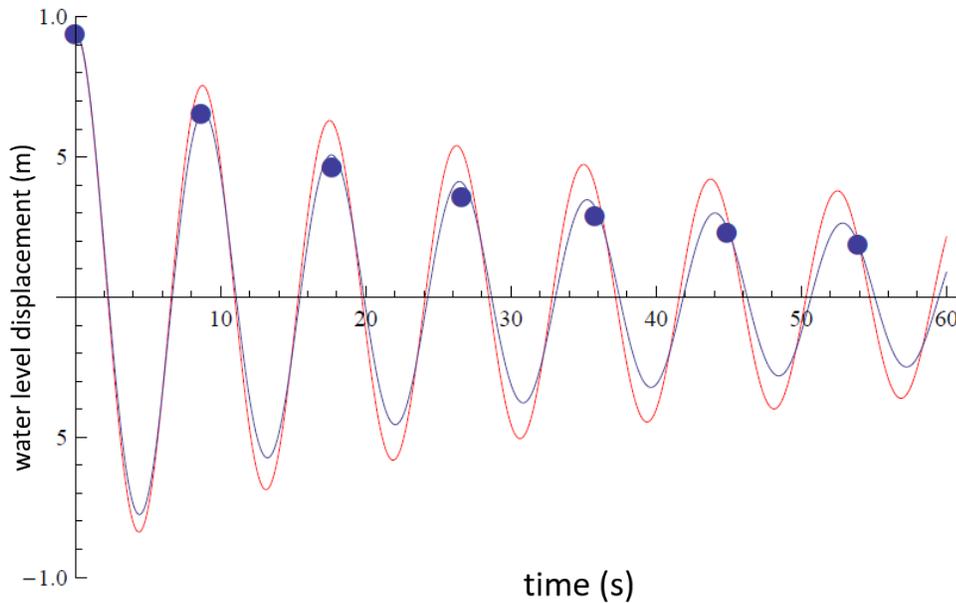


Figure 16. Darcy Weisbach Model USGS Well G3902. Blue dots show observed peak displacement (in meters) as a function of time (seconds). Red line shows maximum-velocity-based Reynolds number model. Blue line shows model with adjusted friction factor for improved fit.

Table 6. Comparison of normal and adjusted friction factor for Darcy Weisbach model applied to wells

Well number	f_D	Adjusted f_D	Ratio of adjusted to normal f_D
G3878	0.020	0.038	1.9
G3902	0.020	0.035	1.7
G3907	0.021	0.13	6.3
G3911	0.022	0.15	6.7
G3915	0.026	1.3	51

6 DISCUSSION

In general, the peak timing of the proposed models fit the well data better than the laboratory data. Some experimentation with the model parameters suggested that an initial velocity other than zero might improve the fits of the peak timing. However, the required initial velocities are not considered physically realistic for the actual laboratory experiments.

Invalidation of the assumption $L \gg w$ was considered as a possible source of the observed discrepancy in peak timing. However, solution of the full equation including $L+w$ for the particular experimental data (Figure 17) collected suggests that the impact of including $L+w$ in the equation is very small.

The entry/exit effects included in the models used to simulate the aquifer-free data appear to dominate the solutions because they work well with either the Poiseuille or

Darcy-Weisbach models. No parameter adjustments are required to achieve relatively good agreement with the observations when these effects are included. Experiments with a range of pipe sizes, maximum displacements, and water column lengths all produced similar results.

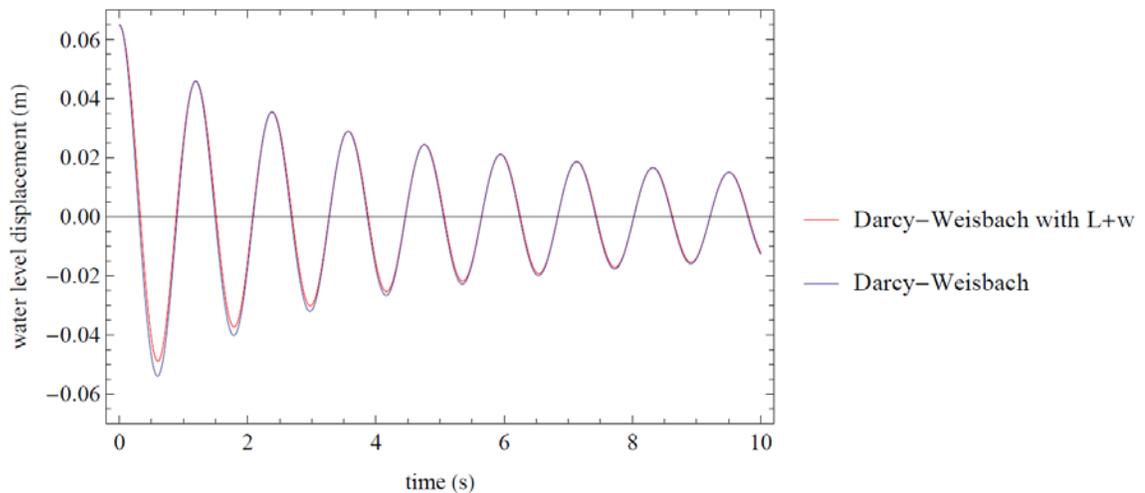


Figure 17. Comparison of Darcy-Weisbach Model with L vs. $L+w$ as seen in test 7740.

Reynolds numbers were calculated and resulted in values greater than 2000 in all laboratory experiments and well data, which signifies that the flow regime exceeds laminar conditions, and the values fall into the transitional and turbulent flow regimes. The Poiseuille model assumes laminar conditions, therefore it cannot be expected to effectively model the underdamped oscillations. The inability of the Poiseuille model to model the underdamped oscillations for the well can also be seen by the large viscosity adjustments necessary to achieve reasonable descriptions of the data for the wells (and for the aquifer-free case without entry/exit effects included in Appendix C).

The Darcy-Weisbach model is more effective at describing the data with less extensive parameter adjustments. The differences between the large viscosity ratios in Table 5 and the significantly smaller friction factor ratios in Table 6 illustrate the benefit of using the Darcy-Weisbach model. However, the Darcy-Weisbach model is based on a constant friction factor which comes from a constant Reynolds number and flow velocity. The rapid oscillations observed in the experiments and the USGS well data are not consistent with the empirical approach underlying the Moody diagram and the concept of the friction factor. The standard Poiseuille and Darcy-Weisbach approaches are widely applied to pipe flow problems, but are likely inadequate for the underdamped oscillations observed in these well tests and laboratory experiments. There are no comparable standard approaches for the oscillating flow problem.

7 CONCLUSIONS

Current analyses of slug tests conducted in extremely permeable materials may be underestimating the actual hydraulic conductivity. This presents potential problems for regions and engineering works that may rely on models founded on good estimates of the hydraulic conductivity to predict groundwater behavior.

Several new simple models for the aquifer-free description of underdamped oscillations in pipes were developed and tested against laboratory and field data. These include a Poiseuille equation-based model, a Darcy-Weisbach-based model, and extensions of these models incorporating entry/exit effects.

Both the Poiseuille and Darcy-Weisbach models incorporating entry/exit effects are effective at capturing the behavior of an aquifer-free experimental condition. Possibly

as a result of the relatively high Reynolds numbers observed in the well flows, the Darcy-Weisbach model was more effective at describing the observations from wells than the Poiseuille model. The Darcy-Weisbach model does not provide a satisfying description of the oscillating flows however, because it is based on a single constant velocity. Although no standard approaches appear to be available for such oscillating flows, future work will need to assess this issue more thoroughly.

LIST OF REFERENCES

- Audouin, O., and J. Bodin. 2007. Analysis of slug-tests with high-frequency oscillations. *Journal of Hydrology* 334 (1-2), 282-289. doi:10.1016/j.jhydrol.2006.10.009
- Black, J.H. (2010). The practical reasons why slug tests (including falling and rising head tests) often yield the wrong value of hydraulic conductivity. *Quarterly Journal of Engineering Geology and Hydrogeology*, 43:345–358. <http://dx.doi.org/10.1144/1470-9236/08-094>
- Brauchler, R., R. Hu, P. Dietrich, and M. Sauter (2011), A field assessment of high-resolution aquifer characterization based on hydraulic travel time and hydraulic attenuation tomography, *Water Resources Research*, 47, W03503, doi:10.1029/2010WR009635.
- Butler, J. Jr., (1998) *The design, Performance, and Analysis of Slug Tests*. Boca Raton, Florida: Lewis Publishers.
- Butler, J. Jr. and E.J. Garnett., (2000) Simple procedures for analysis of slug tests in formations of high hydraulic conductivity using spreadsheet and scientific graphics software. Kansas Geological Survey Open-File Rep. 2000-40 (available at www.Kgs.ke.edu/Hydro/Publications/OFR00_40/index.html)
- Choquette, P.W. and L.C. Pray 1970. Geological nomenclature and classification of porosity in sedimentary carbonates. *American Association of Petroleum Geologists Bulletin*, v. 54, no. 2, p. 207-250.
- Cooper, H. H., Bredehoeft, J. D., & Papadopoulos, I. S., (1967). Response of a finite-diameter well to an instantaneous charge of water. *Water Resources Research*, 3(1), 263–269. doi:10.1029/WR003i001p00263
- Cunningham, K.J., M.C. Sukop, H. Huang, P.F. Alvarez, H. A. Curran, J.F. Dixon, and R.A. Renken, 2009. Prominence of ichnologically-influenced macroporosity in the karst Biscayne aquifer: stratiform "super-K" zones, *Geological Society of America Bulletin* 121:1-2, 164-235. <http://dx.doi.org/10.1130/B26392.1>
- Cunningham, K.J., Wacker, M.A., Robinson, E., Dixon, J.F., and Wingard, G.L., (2006), *A Cyclostratigraphic and Borehole-Geophysical Approach to Development of a Three-Dimensional Conceptual Hydrogeologic Model of the Karstic Biscayne Aquifer, Southeastern Florida*: U.S. Geological Survey Scientific Investigations Report 2005-5235, 69 p., plus CD.

Fish, J. E. and M. Stewart (1991) Hydrogeology of the Surficial Aquifer System, Dade County, Florida, Water-Resources Investigations Report 90-4108 Department of the Interior, US Geological Survey, WRI 90-4108.

Freeze, R. A., and Cherry, J. A. (1979). Groundwater. Englewood Cliffs, NJ: Prentice-Hall.

Hvorslev, M. J. (1951). Time lag and soil permeability in ground-water observations. Vicksburg, MS: Waterways Experiment Station, Corps of Engineers, U.S. Army.

Gallegos, J.J., B.X. Hu, H. Davis (2013), Simulating flow in karst aquifers at laboratory and sub-regional scales using MODFLOW-CFP, Hydrogeology Journal 21:1749-1760, <http://dx.doi.org/10.1007/s10040-013-1046-4>

Garcia, S. M., (2013) "Lattice Boltzmann Modeling and Specialized Laboratory Techniques to Determine the Permeability of Megaporous Karst Rock". FIU Electronic Theses and Dissertations. Paper 918. <http://digitalcommons.fiu.edu/etd/918>

Kipp, K. L. Jr., (1985) Type Curve Analysis of Inertial Effects in the Response of a Well to a Slug Test. Water Resources Research, 21 (9), 1397-1408

Matos, D., and Valerio, C. (2009). Fluid mechanics and pipe flow: Turbulence, simulation, and dynamics. New York: Nova Science. McElwee, C. (2002). Improving the analysis of slug tests. Journal of Hydrology, 269(3-4), 122-133. doi:10.1016/s0022-1694(02)00214-7

Mayaud, C., Walker, P., Hergarten, S., Birk, S. (2014): Nonlinear Flow Process: A New Package to Compute Nonlinear Flow in MODFLOW. Groundwater. <http://dx.doi.org/10.1111/gwat.12243>

Ostendorf, D. W., Degroot, D. J., Dunaj, P. J., & Jakubowski, J. (2005). A Closed Form Slug Test Theory for High Permeability Aquifers. Ground Water, 43(1), 87-101. doi:10.1111/j.1745-6584.2005.tb02288.x

Reimann, T., C. Rehl, W. B. Shoemaker, T. Geyer, and S. Birk (2011), The significance of turbulent flow representation in single-continuum models, Water Resources Research, 47, W09503, <http://dx.doi.org/10.1029/2010WR010133>.

Schwartz, F. W., and Zhang, H. (2003). Fundamentals of Ground Water. John Wiley & Sons.

Shinohara, K., and H. J. Ramey., (1979) Slug Test data analysis, including the inertial effect of the fluid in the wellbore, paper presented at the 54th Annual Fall Technical Conference and Exhibition of the Society of Petroleum Engineers, Am. Inst. Mech. Eng., Las Vegas, Nev., September 23-26.

Shoemaker, W. B., K. J. Cunningham, E. L. Kuniandy, and J. Dixon, 2008, Effects of turbulence on hydraulic heads and parameter sensitivities in preferential groundwater flow layers, *Water Resources Research*, 44, W03501, <http://dx.doi.org/10.1029/2007WR006601>.

Shoemaker, W.B., Kuniandy, E.L., Birk, S., Bauer, S., and Swain, E.D., 2007, Documentation of a Conduit Flow Process (CFP) for MODFLOW-2005: U.S. Geological Survey Techniques and Methods, Book 6, Chapter A24, 50 p.

Sukop, M. C., Cunningham, K. J., (2014). Lattice Boltzmann methods applied to large-scale three-dimensional virtual cores constructed from digital optical borehole images of the karst carbonate Biscayne aquifer in southeastern Florida. *Water Resources Research*, 50(11), 8807–8825. doi:10.1002/2014WR015465

Sukop, M. C., H. Huang, P. F. Alvarez, E. A. Variano, and K. J. Cunningham, 2013, Evaluation of permeability and non-Darcy flow in vuggy macroporous limestone aquifer samples with lattice Boltzmann methods, *Water Resources Research*, 49(1), 216- 230, <http://dx.doi.org/10.1029/2011WR011788>.

van der Kamp, Garth., (1976). Determining aquifer transmissivity by means of well response tests: the underdamped case: *Water Resources Research*, v. 12(1), 71–77.

Wacker, M., Cunningham, K., and Williams, J., (2014) Geologic and Hydrogeologic Frameworks of the Biscayne Aquifer in Central Miami-Dade County, Florida; USGS.

Werner, A. D., Simmons, C. T., (2009), Impact of Sea Level Rise on Salt Water Intrusion in Coastal Aquifers. *Ground Water*, v. 47 (2), 197-204.

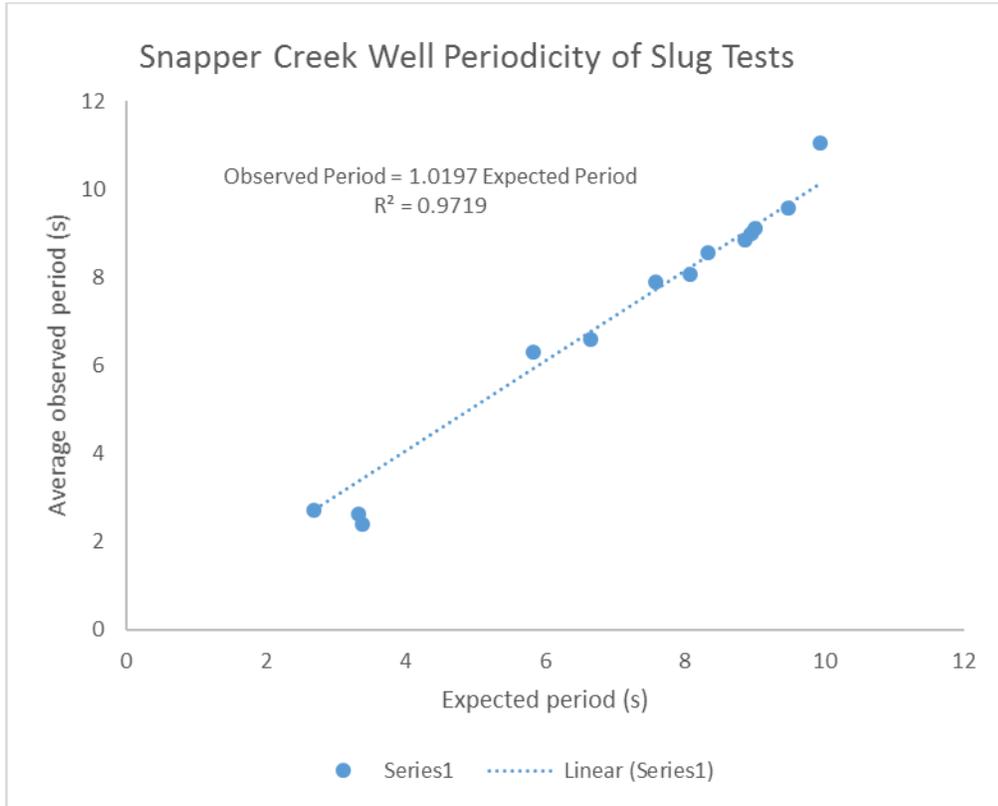
Weight, W. D. and Wittman, G. P. (1999), Oscillatory Slug-Test Data Sets: A Comparison of Two Methods. *Groundwater*, 37: 827–835. <http://dx.doi.org/10.1111/j.1745-6584.1999.tb01181.x>

Zemansky, G.M. and McElwee, C.D. (2005), High-resolution slug testing. *Ground Water*, 43: 222–230. doi:10.1111/j.1745-6584.2005.0008.x

Zurbuchen, B. R., Zlotnik, V.A., Butler Jr., J. J. (2002). Dynamic interpretation of slug tests in highly permeable aquifers. *Water Resources Research*. doi:10.1029/2001WR000354

APPENDICES

APPENDIX A. Selected Snapper Creek wells period analysis



Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
G3902	9.01	test 1	10.896		9.122167
			20.001	9.105	
			29.001	9	
			38.001	9	
			47.201	9.2	
		test 2	9.59		
			19.251	9.661	
			28.001	8.75	
			37.251	9.25	
			46.251	9	
		test 3	20.251		

Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
			29.501	9.25	
			38.251	8.75	
			47.496	9.245	
			56.751	9.255	
g3878	9.48	test 1	13.5		9.579083
			23.001	9.501	
			32.501	9.5	
			42	9.499	
			51.75	9.75	
		test 2	8.301		
			18	9.699	
			27.5	9.5	
			37	9.5	
			46.75	9.75	
		test 3	13		
			22.501	9.501	
			32.25	9.749	
			41.5	9.25	
			51.25	9.75	
G3911	8.33	test 1	16.25		8.560083
			24.751	8.501	
			33.251	8.5	
			42.001	8.75	
			50.501	8.5	
		test 2	11.116		
			19.751	8.635	
			28.251	8.5	
			37.251	9	
			45.251	8	
		test 3	5.602		
			14.013	8.411	

Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
			22.751	8.738	
			31.115	8.361	
			39.94	8.825	
G3915	3.38	test 1	2.344		2.398417
			4.708	2.364	
			7	2.292	
			9.5	2.5	
			12	2.5	
		test 2	2.177		
			4.501	2.324	
			6.751	2.25	
			9.329	2.578	
			11.802	2.473	
		test 3	5		
			7.597	2.597	
			9.802	2.205	
			12.25	2.448	
			14.5	2.25	
G3882	9.94	test 1	13.251		11.04167
			24.001	10.75	
			36.001	12	
		test 2	9.251		
			20.751	11.5	
			31.251	10.5	
		test 3	11.251		
			21.75	10.499	
			32.751	11.001	
G3880	2.88	test 1	20		10.36742
			30.25	10.25	
			40.747	10.497	

Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
			51.269	10.522	
			61.501	10.232	
		test 2	22.75		
			33.5	10.75	
			43.5	10	
			54.25	10.75	
			64.251	10.001	
		test 3	27		
			37.251	10.251	
			47.806	10.555	
			58.284	10.478	
			68.407	10.123	
G3910	9.08	test 1	6.811		9.041
			15.906	9.095	
			24.845	8.939	
			33.75	8.905	
			43	9.25	
		test 2	17.202		
			26.001	8.799	
			34.751	8.75	
			43.927	9.176	
			53.001	9.074	
		test 3	10.996		
			20.32	9.324	
			29.25	8.93	
			38.25	9	
			47.5	9.25	
G3914	8.06	test 1	2.863		8.06
			11	8.137	
			18.751	7.751	
			26.75	7.999	

Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
			35	8.25	
		test 2	2.835		
			11.001	8.166	
			19.001	8	
			27.001	8	
			35.251	8.25	
		test 3	3.084		
			11.251	8.167	
			19.251	8	
			27.501	8.25	
			35.251	7.75	
G3917	6.65	test 1	7.001		6.6025
			13.501	6.5	
			20.001	6.5	
			26.501	6.5	
			33.251	6.75	
		test 2	4.021		
			10.751	6.73	
			17.251	6.5	
			23.751	6.5	
			30.501	6.75	
		test 3	4.5		
			11	6.5	
			17.75	6.75	
			24.5	6.75	
			31	6.5	
G3918	3.32	test 1	10		2.643417
			12.5	2.5	
			15.251	2.751	
			18.001	2.75	
			20.501	2.5	

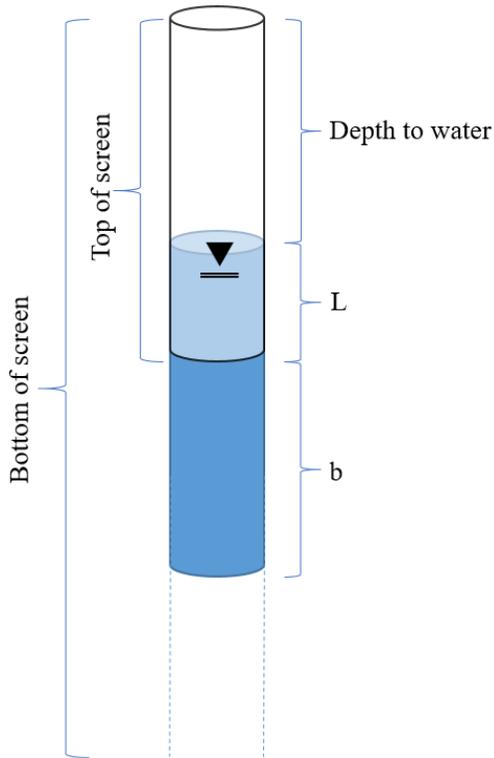
Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
		test 2	3.751		
			6.501	2.75	
			9.001	2.5	
			11.501	2.5	
			14.501	3	
		test 3	3.531		
			6.001	2.47	
			8.751	2.75	
			11.501	2.75	
			14.001	2.5	
G3904	8.85	test 1	5.908		8.841167
			14.75	8.842	
			23.5	8.75	
			32.501	9.001	
			41.501	9	
		test 2	11.5		
			20.25	8.75	
			29.25	9	
			38.001	8.751	
			46.75	8.749	
		test 3	11		
			19.751	8.751	
			28.5	8.749	
			37.25	8.75	
			46.251	9.001	
G3905	5.83	test 1	9.583		6.294889
			16.001	6.418	
			22.251	6.25	
			29.251	7	
		test 2	9.85		

Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
			16	6.15	
			22.001	6.001	
			28.5	6.499	
		test 3	4.664		
			11	6.336	
			17.25	6.25	
			23	5.75	
G3906	2.69	test 1	13.44		2.730083
			15.96	2.52	
			18.96	3	
			21.3	2.34	
			23.88	2.58	
		test 2	14.22		
			16.921	2.701	
			20.101	3.18	
			22.561	2.46	
			25.321	2.76	
		test 3	12.66		
			15.06	2.4	
			17.88	2.82	
			21.3	3.42	
			23.88	2.58	
G3907	8.95	test 1	9.501		8.979167
			18.501	9	
			27.501	9	
			36.501	9	
			45.251	8.75	
		test 2	9.751		
			18.751	9	
			28.001	9.25	
			36.751	8.75	

Well	Expected period (s)		Time (s)	Period (s)	Average period (s)
			45.751	9	
		test 3	8.5		
			17.25	8.75	
			26.25	9	
			35.5	9.25	
			44.5	9	
G3908	7.58	test 1	8.251		7.90625
			16.251	8	
			23.751	7.5	
		test 2	7.75		
			15.75	8	
			23.75	8	
			31	7.25	
		test 3	9.001		
			17.001	8	
			25.001	8	
			33.501	8.5	

APPENDIX B. Summary of selected Snapper Creek well parameters

Well Number	Velocity (m/s)	Friction factor (f_b)	$f_b/4r$	g/L (s^2)	Poiseuille damping term $8\nu/r^2$
G3878	0.58	0.02	0.1	0.462	0.0031
G3902	0.59	0.02	0.1	0.516	0.0031
G3907	0.5	0.021	0.1	0.542	0.0031
G3911	0.4	0.022	0.11	0.585	0.0031
G3915	0.2	0.026	0.13	7.57	0.0031



L is obtained from TOS-DTW

b (the aquifer thickness) is obtained from BOS-TOS

$$L_e = L + b/2$$

APPENDIX C. Models

Test 7713 Models

```
In[1009]= Remove["Global`*"]
In[1010]= L = 0.3
Out[1010]= 0.3

In[1011]= fD = 0.045
Out[1011]= 0.045

In[1012]= g = 9.81
Out[1012]= 9.81

In[1013]= r = 0.01
Out[1013]= 0.01

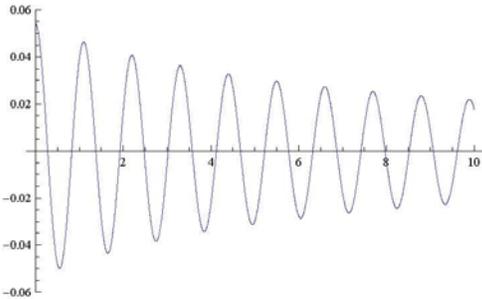
In[1014]= fD / (4 r)
Out[1014]= 1.125

In[1015]= g / L
Out[1015]= 32.7

In[1016]= 1 / (2 L)
Out[1016]= 1.66667

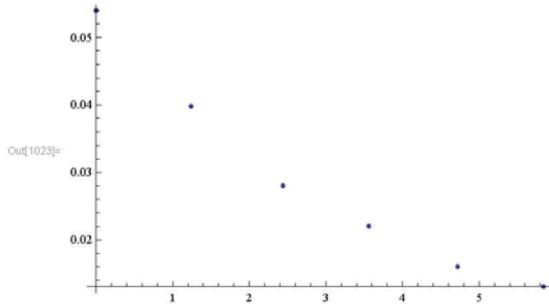
In[1017]=
In[1018]= solution =
NDSolve[{D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
w[0] == 0.054, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1018]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1019]=
In[1020]= Plot[Evaluate[w[t] /. First[solution]],
{t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[1020]=
```

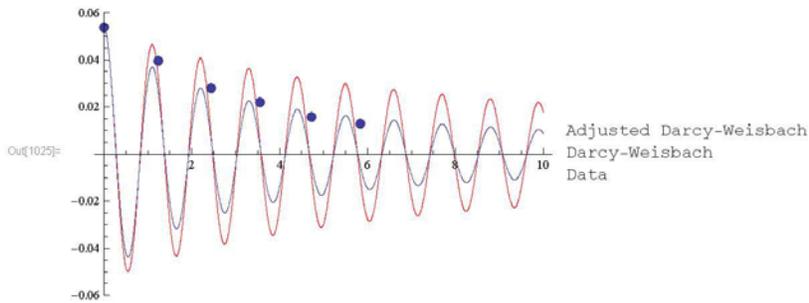


```
In[1021]= solution2 =
  NDSolve[{D[w[t], t, t] == -3.2 * (D[w[t], t]) * Abs[D[w[t], t]] - 32.7 * w[t],
    w[0] == 0.054, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1021]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[1022]=
In[1023]= ListPlot[{{0, 0.054}, {1.24, 0.0398},
  {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016}, {5.84, 0.013}}]
```



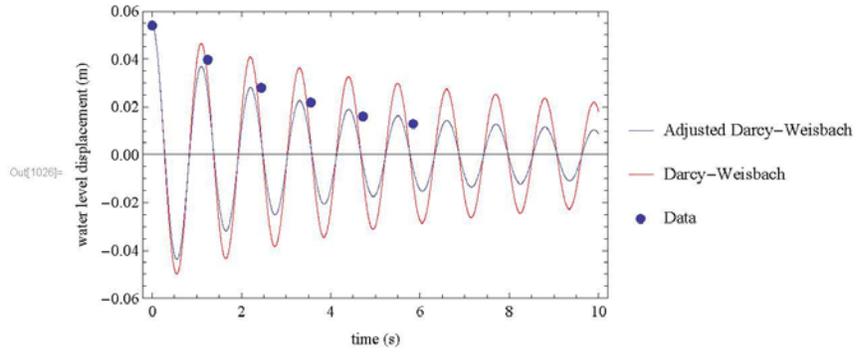
```
In[1024]=
In[1025]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
  ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
  {5.84, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1026]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
  {5.84, 0.013}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[1027]= Remove["Global`*"]

In[1028]= L = 0.3
          fD = 0.045
          g = 9.81
          r = 0.01

Out[1028]= 0.3

Out[1029]= 0.045

Out[1030]= 9.81

Out[1031]= 0.01

In[1032]= fD / (4 r)
Out[1032]= 1.125

In[1033]= g / L
Out[1033]= 32.7

In[1034]= 1 / (2 L)
Out[1034]= 1.66667

In[1035]= solution = NDSolve[
          { D[w[t], t, t] == - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
            (1 / (2 * L)) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] == 0.054, w'[0] == 0},
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]

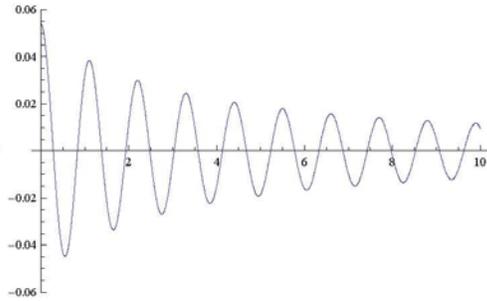
Out[1035]= {w -> InterpolatingFunction[{{0., 10.}}, <>]}

In[1036]=

In[1037]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {- .06, .06}, PlotPoints -> 30]

Out[1037]=

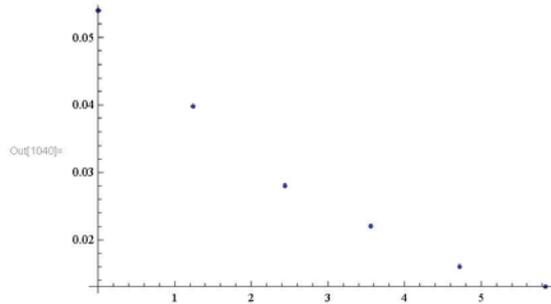
```



```
In[1038]= solution2 = NDSolve[
  { D[w[t], t, t] = - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] = 0.054, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1038]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

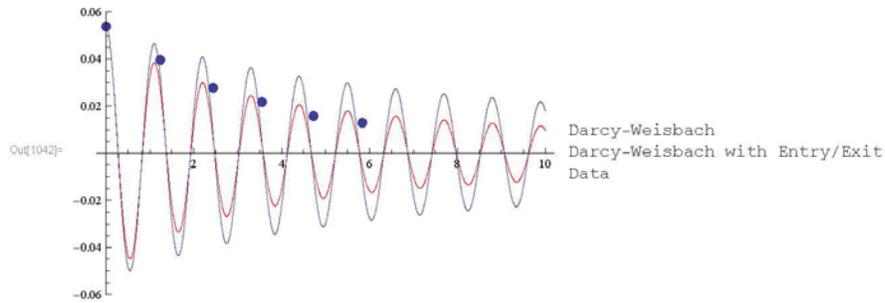
In[1039]=

```
In[1040]= ListPlot[{{0, 0.054}, {1.24, 0.0398},
  {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016}, {5.84, 0.013}}]
```



In[1041]=

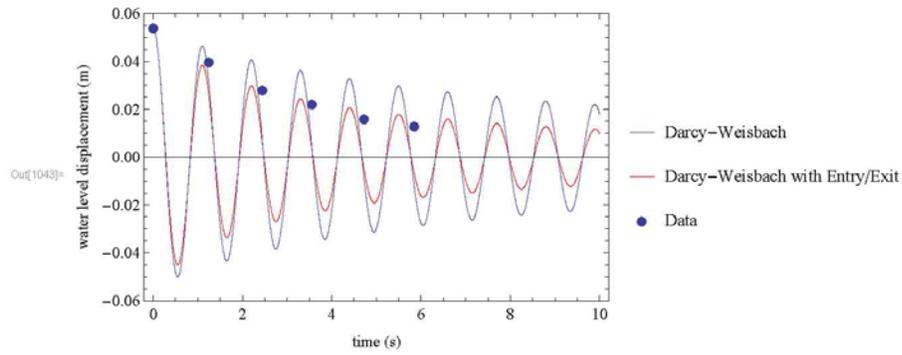
```
In[1042]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 10}, PlotRange -> {- .06, .06}, PlotPoints -> 30,
  PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach with Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
  {5.84, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1043]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022},
  {4.72, 0.016}, {5.84, 0.013}}, PlotLegends -> {Style["Data" ]},
  PlotMarkers -> {Automatic, Medium}], Frame -> True,
FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1044]= Remove["Global`*"]

In[1045]= L = 0.3
Out[1045]= 0.3

In[1046]= g = 9.81
Out[1046]= 9.81

In[1047]= r = 0.01
Out[1047]= 0.01

In[1048]= v = 10^-6
Out[1048]=  $\frac{1}{1\,000\,000}$ 

In[1049]= g/L
Out[1049]= 32.7

In[1050]= (8*v)/(r^2)
Out[1050]= 0.08

In[1051]= 1/(2*L)
Out[1051]= 1.66667

In[1052]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
    1/(2*L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.054, w'[0] == 0},
    w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[1052]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1053]=

In[1054]= Plot[Evaluate[w[t] /. First[solution]],
    {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[1054]=

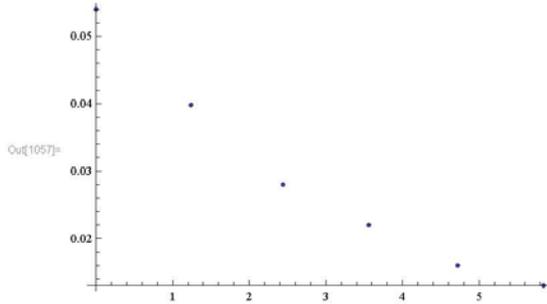
```

```
In[1055]= solution2 = NDSolve[{D[w[t], t, t] = -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] = 0.054, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1055]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[1056]=
```

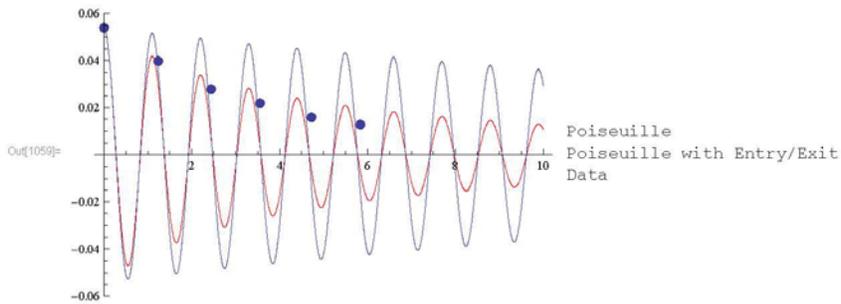
```
In[1057]= ListPlot[{{0, 0.054}, {1.24, 0.0398},
{2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016}, {5.84, 0.013}}]
```



```
Out[1057]=
```

```
In[1058]=
```

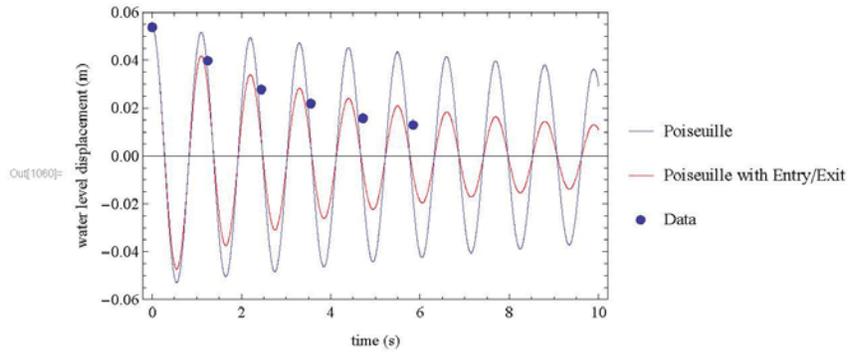
```
In[1059]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille with Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
{5.84, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1060]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
  {5.84, 0.013}}, PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1]= Remove["Global`*"]

Remove::rmnm : There are no symbols matching "Global`*":>>

In[2]= L = 0.3
g = 9.81
x = 0.01
Out[2]= 0.3
Out[3]= 9.81
Out[4]= 0.01

In[5]= v = 10^-6
Out[5]=  $\frac{1}{1\,000\,000}$ 

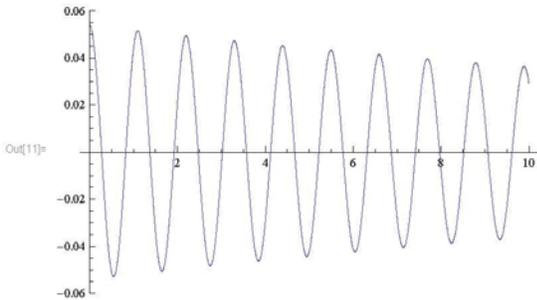
In[6]= g / L
Out[6]= 32.7

In[7]= (8 * v) / (x^2)
Out[7]= 0.08

In[8]= solution = NDSolve[{D[w[t], t, t] == -((8 * v) / (x^2)) * D[w[t], t] - (g / L) * w[t],
w[0] == 0.054, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[8]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[9]=
In[10]=
In[11]= Plot[Evaluate[w[t] /. First[solution]],
{t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[11]=

```

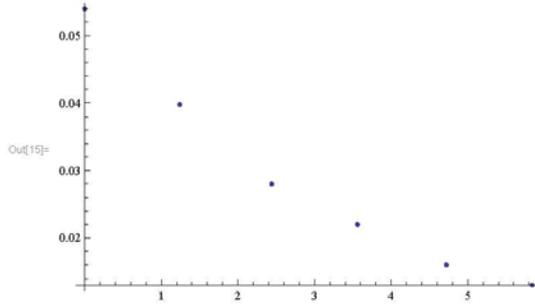


```
In[12]= solution2 =
  NDSolve[{D[w[t], t, t] == -0.5*D[w[t], t] - 32.7*w[t], w[0] == 0.054, w'[0] == 0},
  w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[12]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

In[13]=

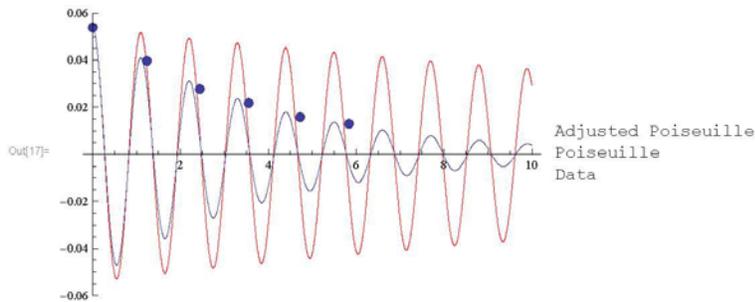
In[14]=

```
In[15]= ListPlot[{{0, 0.054}, {1.24, 0.0398},
  {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016}, {5.84, 0.013}}]
```



In[16]=

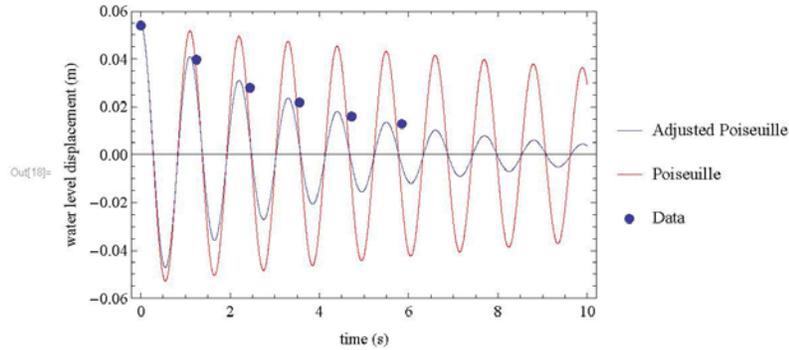
```
In[17]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
  ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
  {5.84, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[18]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.054}, {1.24, 0.0398}, {2.44, 0.028}, {3.56, 0.022}, {4.72, 0.016},
  {5.84, 0.013}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test 7717 Models

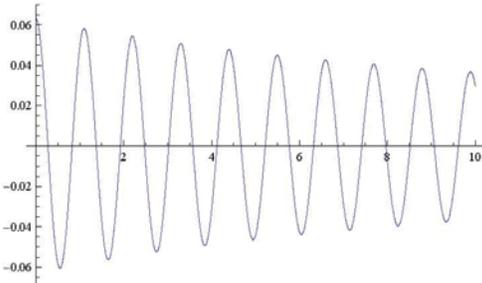
```
In[1113]= Remove["Global`*"]
In[1114]= L = 0.3
Out[1114]= 0.3

In[1115]= fD = 0.045
          g = 9.81
          r = 0.02385
Out[1115]= 0.045
Out[1116]= 9.81
Out[1117]= 0.02385

In[1118]= g / L
Out[1118]= 32.7

In[1119]= fD / (4 r)
Out[1119]= 0.471698

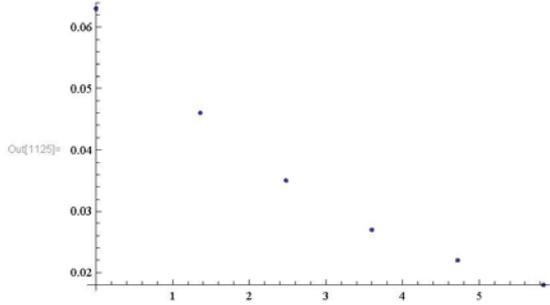
In[1120]= solution = NDSolve[
          { D[w[t], t, t] = -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
            w[0] = 0.063, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1120]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1121]=
In[1122]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {- .07, .07}, PlotPoints -> 30]
Out[1122]= 

In[1123]= solution2 = NDSolve[{ D[w[t], t, t] = -2 * (D[w[t], t]) * Abs[D[w[t], t]] - 32.7 * w[t],
          w[0] = 0.063, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1123]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

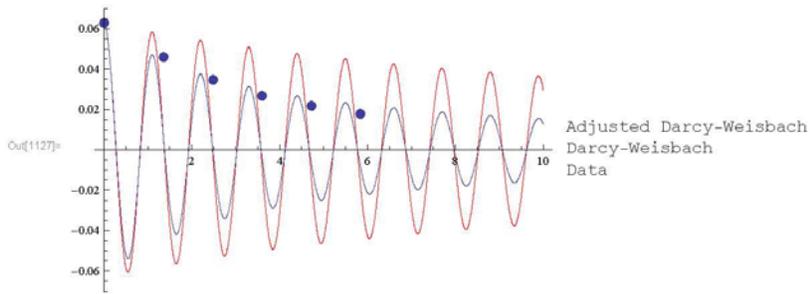
In[1124]=
```

```
In[1125]= ListPlot[{{0, 0.063}, {1.36, 0.046},
{2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022}, {5.84, 0.018}}]
```



```
In[1126]=
```

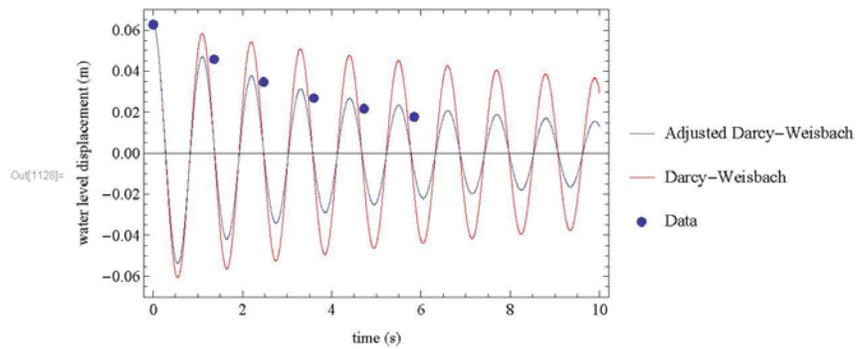
```
In[1127]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"},
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"},
ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
{5.84, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1128]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
  {5.84, 0.018}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[1195]= Remove["Global`*"]

In[1196]= L = 0.3
Out[1196]= 0.3

In[1197]= fD = 0.045
Out[1197]= 0.045

In[1198]= g = 9.81
          r = 0.02385
Out[1198]= 9.81
Out[1199]= 0.02385

In[1200]= fD / (4 r)
Out[1200]= 0.471698

In[1201]= g / L
Out[1201]= 32.7

In[1202]= 1 / (2 L)
Out[1202]= 1.66667

In[1203]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
            (1 / (2 * L)) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] == 0.063, w'[0] == 0,
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1203]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1204]=

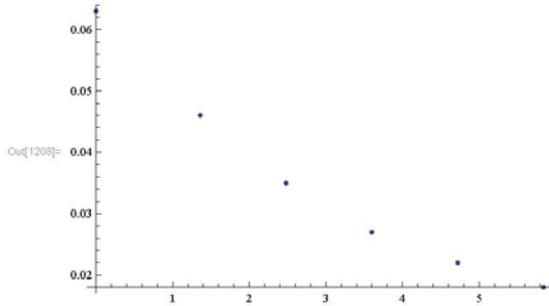
In[1205]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]
Out[1205]=

```

```
In[1206]= solution2 = NDSolve[
  { D[w[t], t, t] = - (fD / (4 x)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] = 0.063, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1206]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

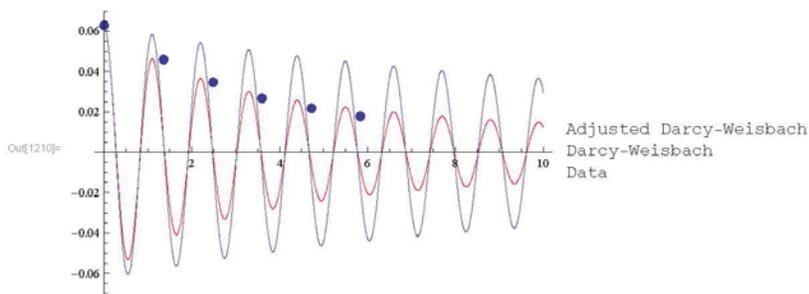
In[1207]=

```
In[1208]= ListPlot[{{0, 0.063}, {1.36, 0.046},
  {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022}, {5.84, 0.018}}]
```



In[1209]=

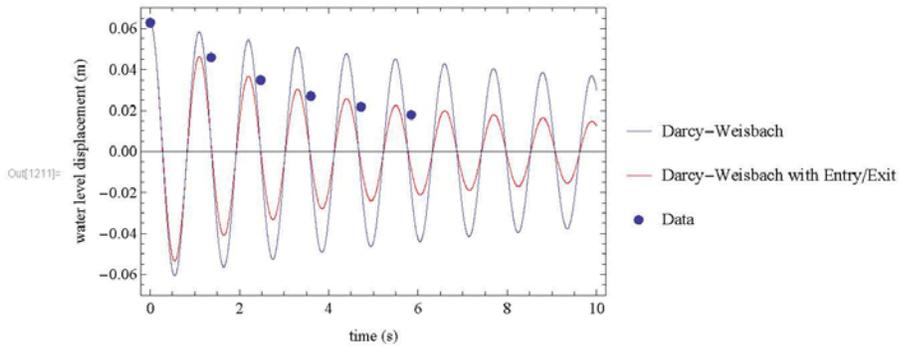
```
In[1210]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
  PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
  ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
  {5.84, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1211]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .07, .07}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
{5.84, 0.018}}, PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1212]=

```

```

In[1265]= Remove["Global`*"]

In[1266]= L = 0.3
          g = 9.81
Out[1266]= 0.3

Out[1267]= 9.81

In[1268]= r = 0.02385
Out[1268]= 0.02385

In[1269]= v = 10^-6
Out[1269]=  $\frac{1}{1\,000\,000}$ 

In[1270]= g/L
Out[1270]= 32.7

In[1271]= (8*v)/(r^2)
Out[1271]= 0.0140641

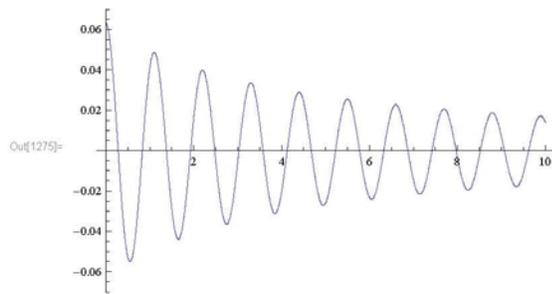
In[1272]= 1/(2*L)
Out[1272]= 1.66667

In[1273]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
                             1/(2*L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.063, w'[0] == 0},
                             w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[1273]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1274]=

In[1275]= Plot[Evaluate[w[t] /. First[solution]],
               {t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]

```

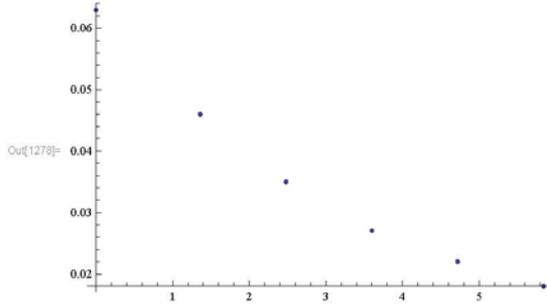


```
In[1276]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] == 0.063, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1276]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

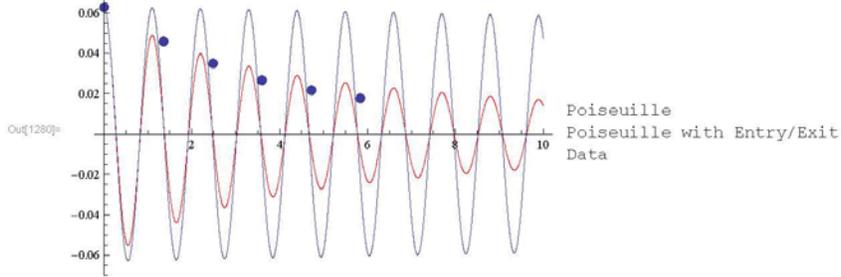
```
In[1277]=
```

```
In[1278]= ListPlot[{{0, 0.063}, {1.36, 0.046},
{2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022}, {5.84, 0.018}}]
```



```
In[1279]=
```

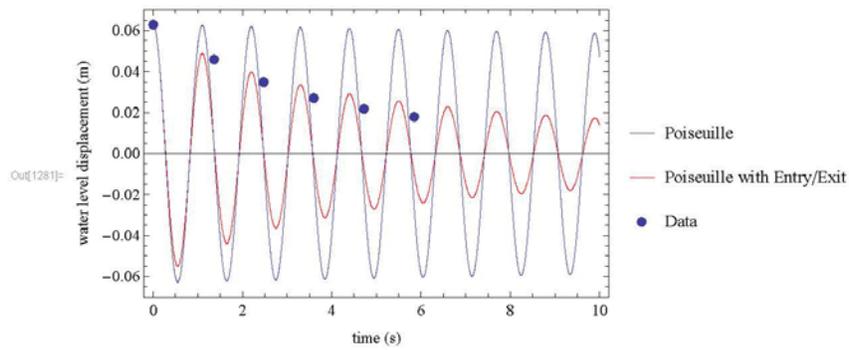
```
In[1280]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille with Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
PlotRange -> {- .07, .07}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
{5.84, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1281]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .07, .07}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
  PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
  {5.84, 0.018}}, PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1341]= Remove["Global`*"]

In[1342]= L = 0.3
          g = 9.81
          r = 0.02385

Out[1342]= 0.3

Out[1343]= 9.81

Out[1344]= 0.02385

In[1345]= v = 10^-6

Out[1345]=  $\frac{1}{1\,000\,000}$ 

In[1346]= g / L
Out[1346]= 32.7

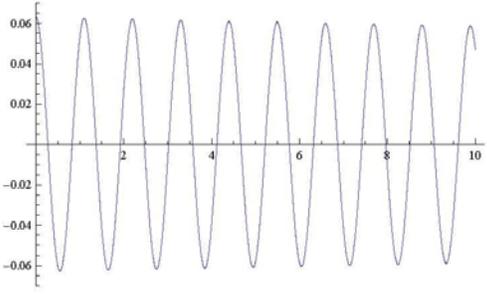
In[1347]= (8 * v) / (r^2)
Out[1347]= 0.0140641

In[1348]= solution = NDSolve[{D[w[t], t, t] == -((8 * v) / (r^2)) * D[w[t], t] - (g / L) * w[t],
                              w[0] == 0.063, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]

Out[1348]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1349]=

In[1350]= Plot[Evaluate[w[t] /. First[solution]],
              {t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]

Out[1350]= 

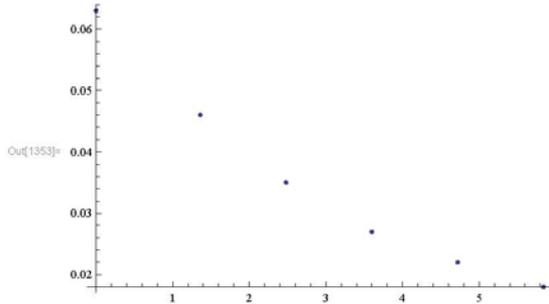
In[1351]= solution2 =
          NDSolve[{D[w[t], t, t] == -0.5 * D[w[t], t] - 32.7 * w[t], w[0] == 0.063, w'[0] == 0},
                w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]

Out[1351]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1352]=

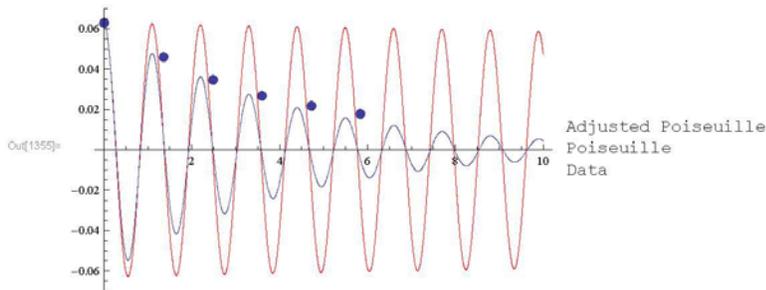
```

```
In[1353]= ListPlot[{{0, 0.063}, {1.36, 0.046},
{2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022}, {5.84, 0.018}}]
```



```
In[1354]=
```

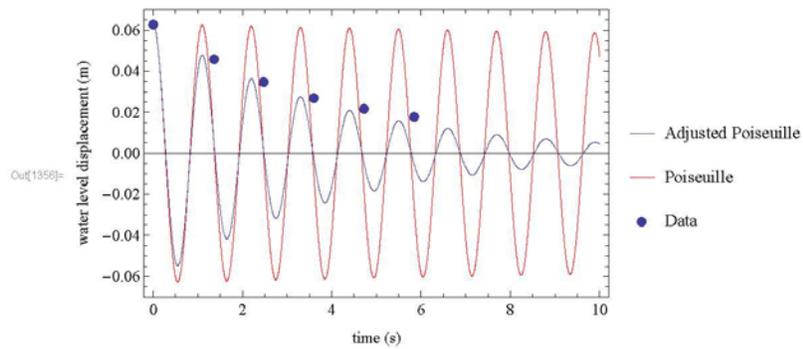
```
In[1355]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
{5.84, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1356]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.063}, {1.36, 0.046}, {2.48, 0.035}, {3.6, 0.027}, {4.72, 0.022},
  {5.84, 0.018}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test 7725 Models

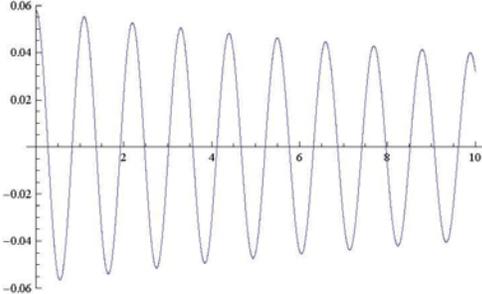
```
In[1549]= Remove["Global`*"]
In[1550]= L = 0.3
Out[1550]= 0.3

In[1551]= fD = 0.031
          g = 9.81
          r = 0.02385
Out[1551]= 0.031
Out[1552]= 9.81
Out[1553]= 0.02385

In[1554]= fD / (4 r)
Out[1554]= 0.324948

In[1555]= g / L
Out[1555]= 32.7

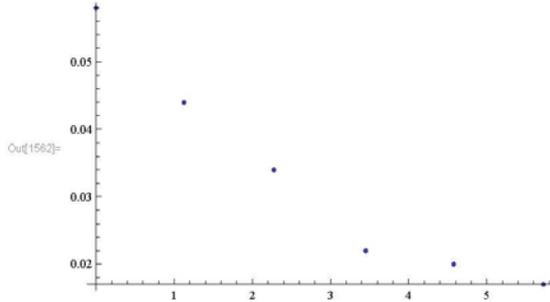
In[1556]=
In[1557]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
            w[0] == 0.058, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1557]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1558]=
In[1559]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[1559]= 

In[1560]= solution2 = NDSolve[{ D[w[t], t, t] == -2 * (D[w[t], t]) * Abs[D[w[t], t]] - 32.7 * w[t],
          w[0] == 0.058, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1560]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

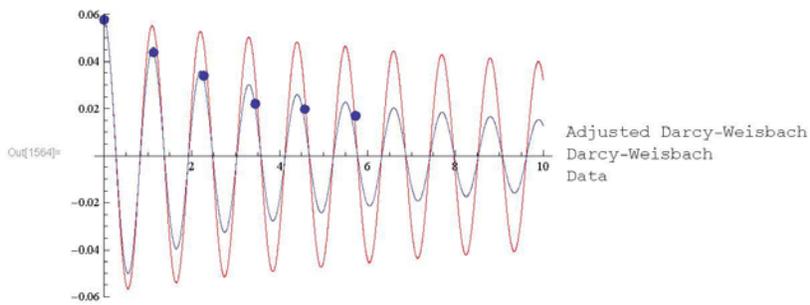
In[1561]=
```

```
In[1562]= ListPlot[{{0, 0.058}, {1.125, 0.044},
{2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}}]
```



In[1563]=

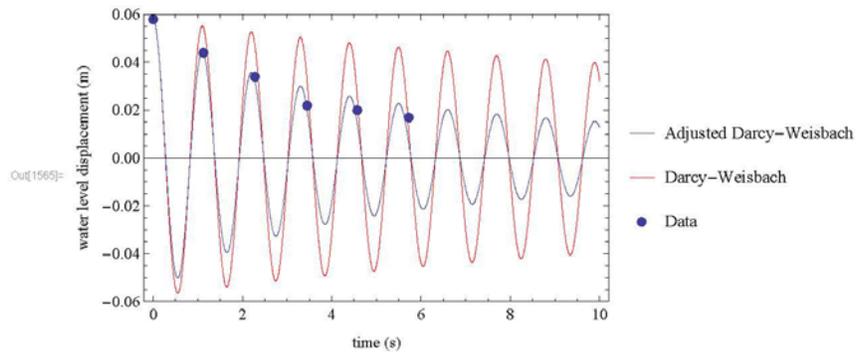
```
In[1564]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"},
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"},
ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02},
{5.725, 0.017}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1565]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034},
  {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[1531]= Remove["Global`*"]

In[1532]= L = 0.3
          fD = 0.031
          g = 9.81
          r = 0.02385

Out[1532]= 0.3

Out[1533]= 0.031

Out[1534]= 9.81

Out[1535]= 0.02385

In[1536]= fD / (4 r)
Out[1536]= 0.324948

In[1537]= g / L
Out[1537]= 32.7

In[1538]= 1 / (2 L)
Out[1538]= 1.66667

In[1539]=

In[1540]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
            (1 / (2 * L)) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] == 0.058, w'[0] == 0,
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]

Out[1540]= {w -> InterpolatingFunction[{{0., 10.}}, <>]}

In[1541]=

In[1542]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]

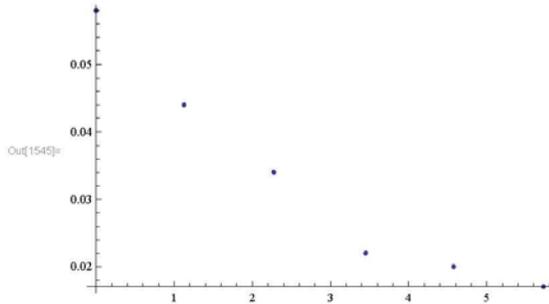
Out[1542]=

```

```
In[1543]= solution2 = NDSolve[
  { D[w[t], t, t] = - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] = 0.058, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1543]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

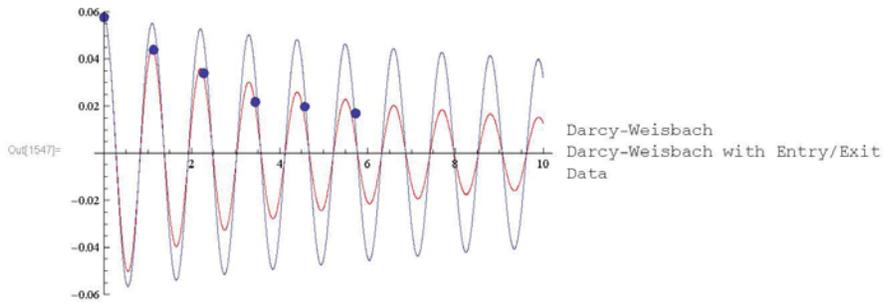
In[1544]=

```
In[1545]= ListPlot[{{0, 0.058}, {1.125, 0.044},
  {2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}}]
```



In[1546]=

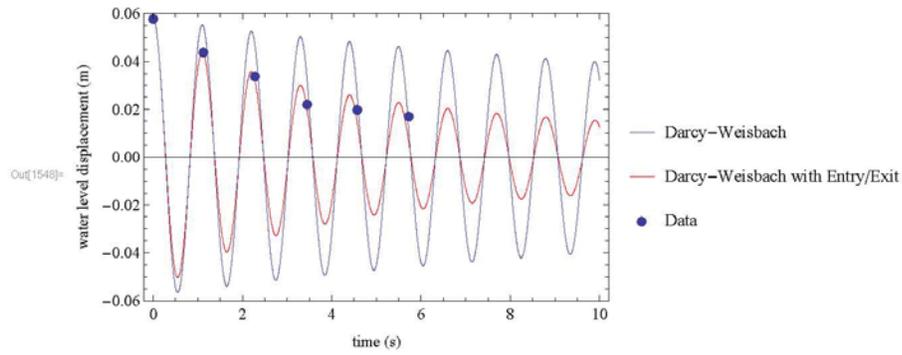
```
In[1547]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 10}, PlotRange -> {- .06, .06}, PlotPoints -> 30,
  PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach with Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02},
  {5.725, 0.017}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1548]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034},
  {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1744]= Remove["Global`*"]

In[1745]= L = 0.3
Out[1745]= 0.3

In[1746]= v = 10^-6
Out[1746]=  $\frac{1}{1\,000\,000}$ 

In[1747]= g = 9.81
           r = 0.02385
Out[1747]= 9.81
Out[1748]= 0.02385

In[1749]= g / L
Out[1749]= 32.7

In[1750]= (8 * v) / (r^2)
Out[1750]= 0.0140641

In[1751]= 1 / (2 L)
Out[1751]= 1.66667

In[1752]= solution = NDSolve[{ D[w[t], t, t] = -((8 * v) / (r^2)) * D[w[t], t] - (g / L) * w[t] -
                             1 / (2 L) * D[w[t], t] + Abs[D[w[t], t]], w[0] = 0.058, w'[0] == 0 },
                             w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[1752]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1753]=
In[1754]= Plot[Evaluate[w[t] /. First[solution]],
              {t, 0, 10}, PlotRange -> {- .06, .06}, PlotPoints -> 30]
Out[1754]=

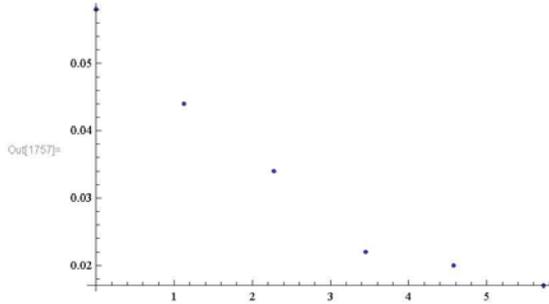
```

```
In[1755]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] = 0.058, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1755]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

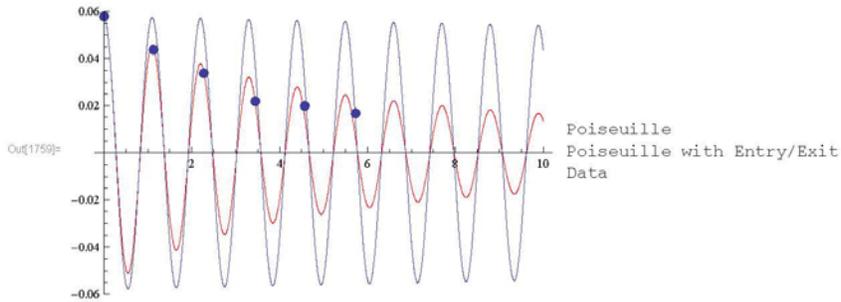
```
In[1756]=
```

```
In[1757]= ListPlot[{{0, 0.058}, {1.125, 0.044},
{2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}}]
```



```
In[1758]=
```

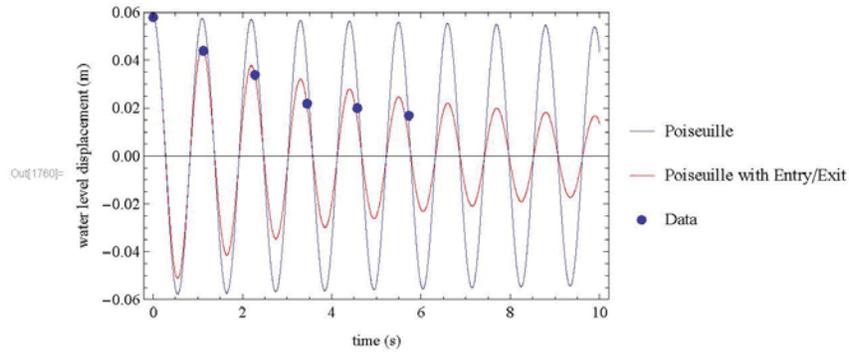
```
In[1759]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille with Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02},
{5.725, 0.017}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1760]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}], ListPlot[{{0, 0.058},
{1.125, 0.044}, {2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}},
PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1761]= Remove["Global`*"]

In[1762]= L = 0.3
Out[1762]= 0.3

In[1763]= g = 9.81
          r = 0.02385
Out[1763]= 9.81
Out[1764]= 0.02385

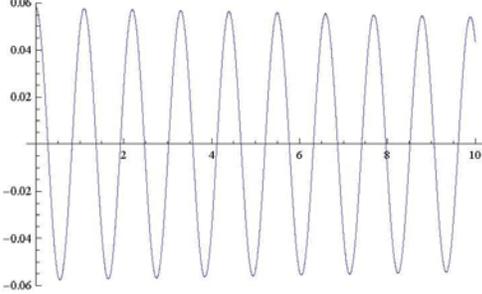
In[1765]= v = 10^-6
Out[1765]=  $\frac{1}{1\,000\,000}$ 

In[1766]= g/L
Out[1766]= 32.7

In[1767]= (8*v)/(r^2)
Out[1767]= 0.0140641

In[1768]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
          w[0] == 0.058, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1768]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1769]=

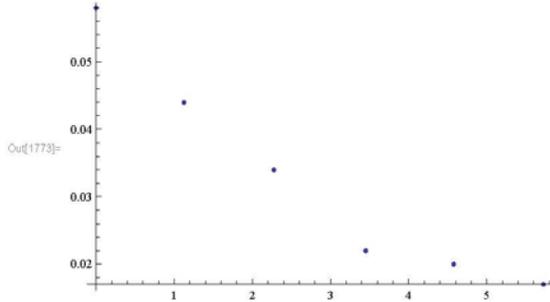
In[1770]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[1770]= 

In[1771]= solution2 =
          NDSolve[{D[w[t], t, t] == -0.5*D[w[t], t] - 32.7*w[t], w[0] == 0.058, w'[0] == 0},
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1771]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1772]=

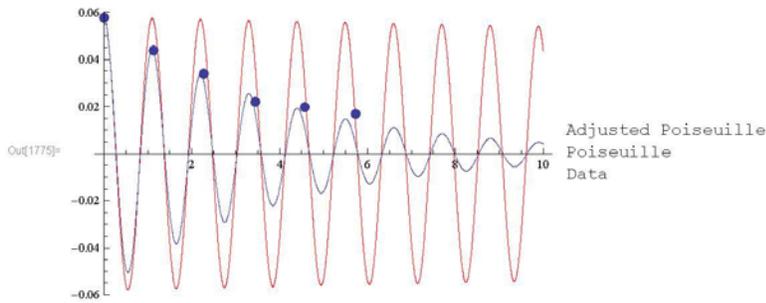
```

```
In[1773]= ListPlot[{{0, 0.058}, {1.125, 0.044},
{2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}}]
```



In[1774]=

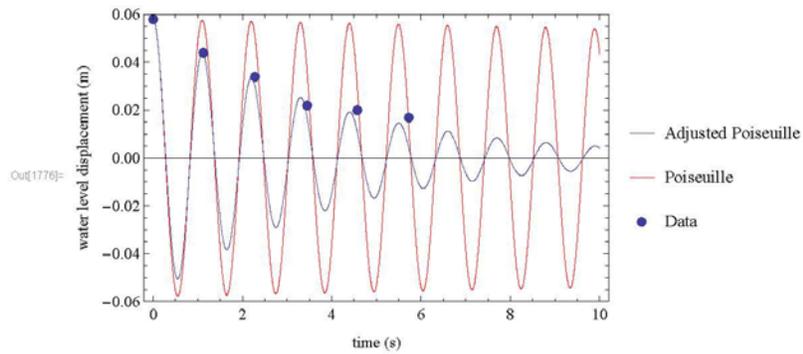
```
In[1775]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034}, {3.45, 0.022}, {4.575, 0.02},
{5.725, 0.017}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1776]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.058}, {1.125, 0.044}, {2.275, 0.034},
  {3.45, 0.022}, {4.575, 0.02}, {5.725, 0.017}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test 7727 Models

```
In[1808]= Remove["Global`*"]
In[1809]= L = 0.35
Out[1809]= 0.35

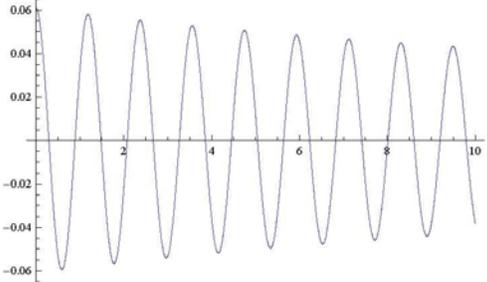
In[1810]= fD = 0.03
Out[1810]= 0.03

In[1811]= g = 9.81
           r = 0.02385
Out[1811]= 9.81
Out[1812]= 0.02385

In[1813]= fD / (4 r)
Out[1813]= 0.314465

In[1814]= g / L
Out[1814]= 28.0286

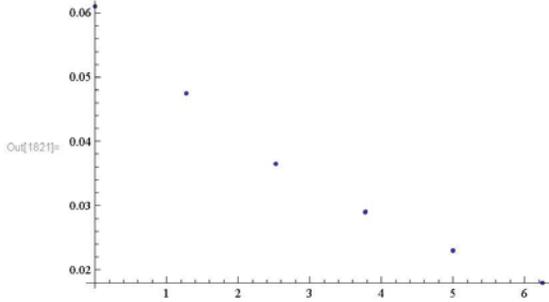
In[1815]=
In[1816]= solution = NDSolve[
           { D[w[t], t, t] = - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
           w[0] = 0.061, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1816]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[1817]=
In[1818]= Plot[Evaluate[w[t] /. First[solution]],
           {t, 0, 10}, PlotRange -> {- .065, .065}, PlotPoints -> 30]
Out[1818]= 

In[1819]= solution2 =
           NDSolve[{ D[w[t], t, t] = -2 * (D[w[t], t]) * Abs[D[w[t], t]] - 28.03 * w[t],
           w[0] = 0.061, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1819]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

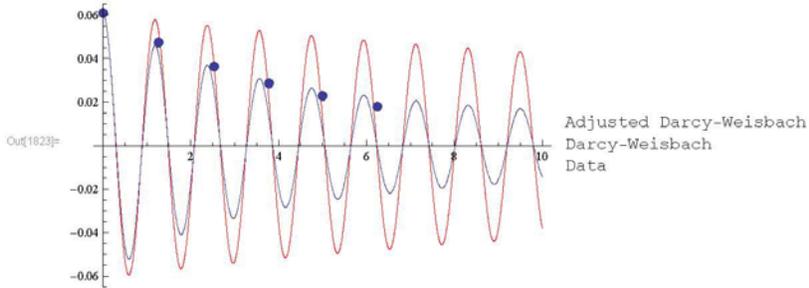
In[1820]=

```
In[1821]= ListPlot[{{0, 0.061}, {1.275, 0.0475},
{2.525, 0.0365}, {3.775, 0.029}, {5, 0.023}, {6.25, 0.018}}]
```



In[1822]=

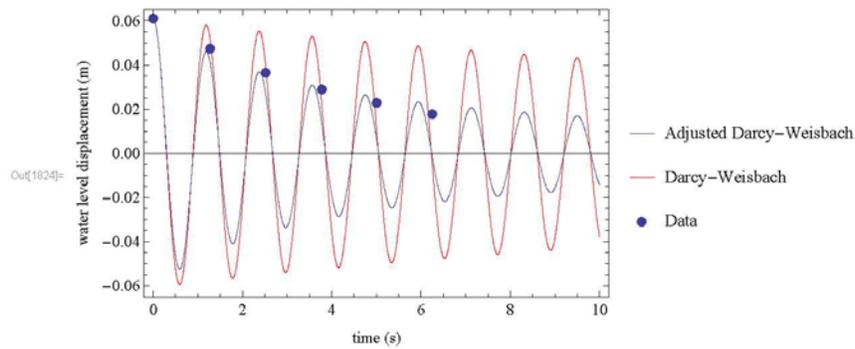
```
In[1823]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .065, .065},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .065, .065},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
{6.25, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1824]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .065, .065},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .065, .065},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
  {6.25, 0.018}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[1912]= Remove["Global`*"]

In[1913]= L = 0.35
          fD = 0.03
Out[1913]= 0.35
Out[1914]= 0.03

In[1915]= g = 9.81
          r = 0.02385

Out[1915]= 9.81
Out[1916]= 0.02385

In[1917]= fD / (4 r)
Out[1917]= 0.314465

In[1918]= g / L
Out[1918]= 28.0286

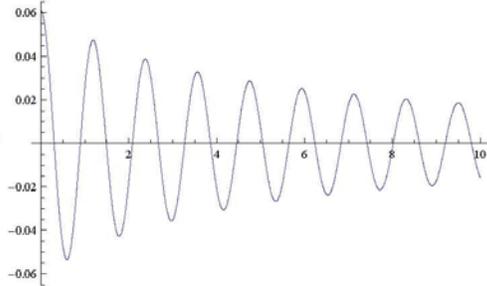
In[1919]= 1 / (2 L)
Out[1919]= 1.42857

In[1920]= solution = NDSolve[
          { D[w[t], t, t] = -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
            1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] = 0.061, w'[0] == 0},
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1920]= {w -> InterpolatingFunction[{{0., 10.}}, <>]}

In[1921]=

In[1922]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.065, 0.065}, PlotPoints -> 30]
Out[1922]=

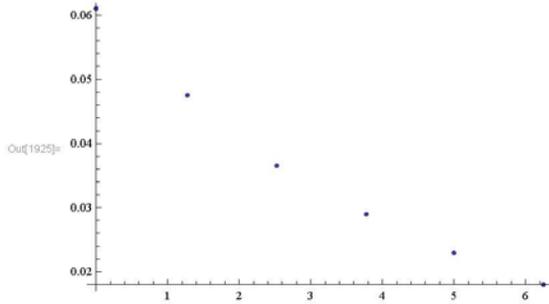
```



```
In[1923]= solution2 = NDSolve[
  { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] == 0.061, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1923]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

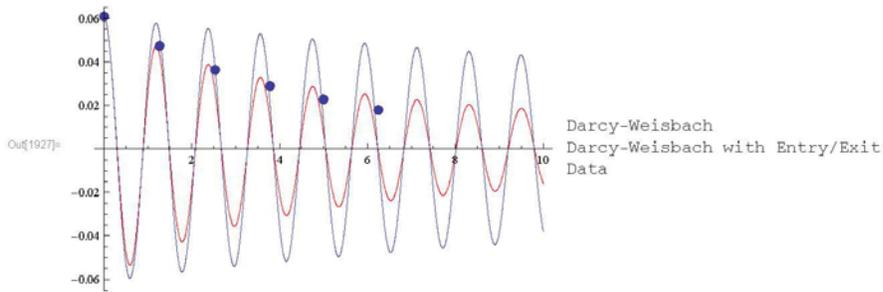
In[1924]=

```
In[1925]= ListPlot[{{0, 0.061}, {1.275, 0.0475},
  {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023}, {6.25, 0.018}}]
```



In[1926]=

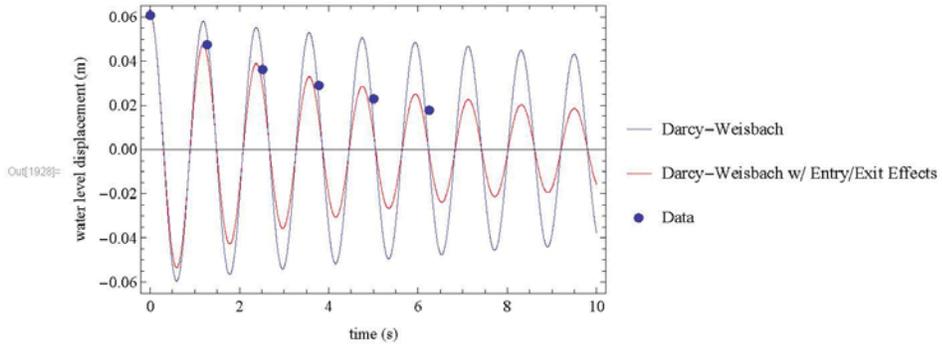
```
In[1927]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 10}, PlotRange -> {- .065, .065}, PlotPoints -> 30,
  PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach with Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .065, .065},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
  {6.25, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1928]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .065, .065}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit Effects"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .065, .065},
  PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
  {6.25, 0.018}}, PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```
In[2014]= Remove["Global`*"]
```

```
In[2015]= L = 0.35  
v = 10^-6  
g = 9.81  
r = 0.02385  
g/L  
(8*v)/(r^2)
```

```
Out[2015]= 0.35
```

```
Out[2016]=  $\frac{1}{1000000}$ 
```

```
Out[2017]= 9.81
```

```
Out[2018]= 0.02385
```

```
Out[2019]= 28.0286
```

```
Out[2020]= 0.0140641
```

```
In[2021]= 1/(2L)
```

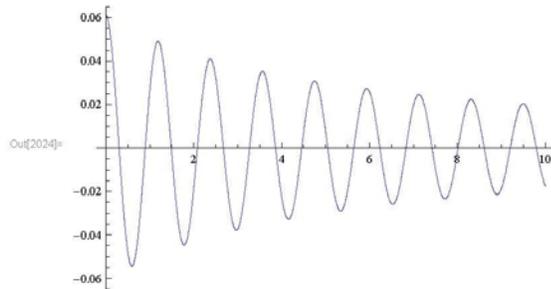
```
Out[2021]= 1.42857
```

```
In[2022]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -  
1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.061, w'[0] == 0},  
w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
```

```
Out[2022]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[2023]=
```

```
In[2024]= Plot[Evaluate[w[t] /. First[solution]],  
{t, 0, 10}, PlotRange -> {-0.065, 0.065}, PlotPoints -> 30]
```

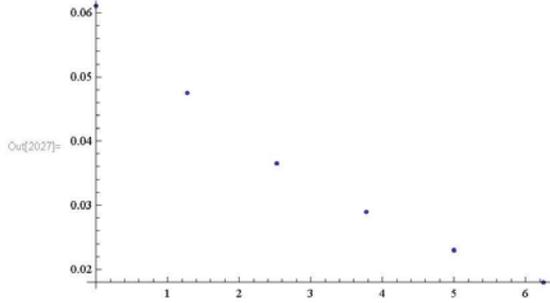


```
In[2025]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],  
w[0] == 0.061, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[2025]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

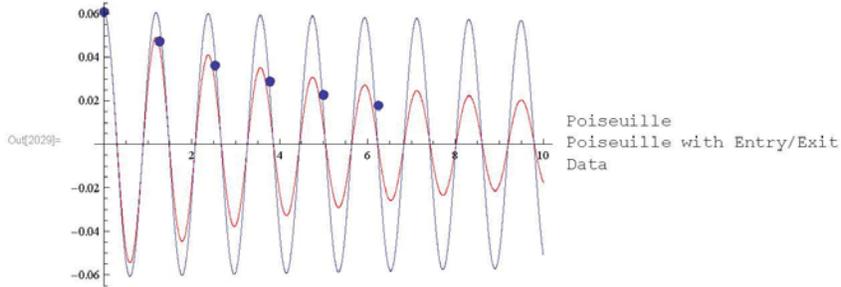
```
In[2026]=
```

```
In[2027]= ListPlot[{{0, 0.061}, {1.275, 0.0475},
{2.525, 0.0365}, {3.775, 0.029}, {5, 0.023}, {6.25, 0.018}}]
```



In[2028]=

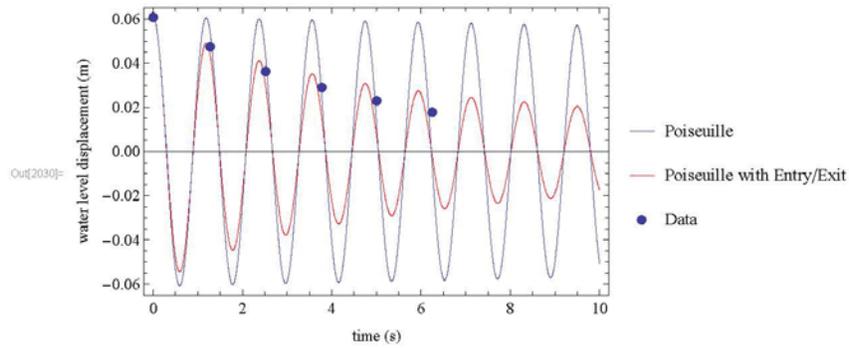
```
In[2029]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .065, .065},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille with Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
PlotRange -> {- .065, .065}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
{6.25, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[2030]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .065, .065}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .065, .065},
PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
{6.25, 0.018}}, PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```
In[1946]= Remove["Global`*"]
```

```
In[1947]= L = 0.35  
v = 10^-6  
g = 9.81  
r = 0.02385  
g/L  
(8*v)/(r^2)
```

```
Out[1947]= 0.35
```

```
Out[1948]=  $\frac{1}{1000000}$ 
```

```
Out[1949]= 9.81
```

```
Out[1950]= 0.02385
```

```
Out[1951]= 28.0286
```

```
Out[1952]= 0.0140641
```

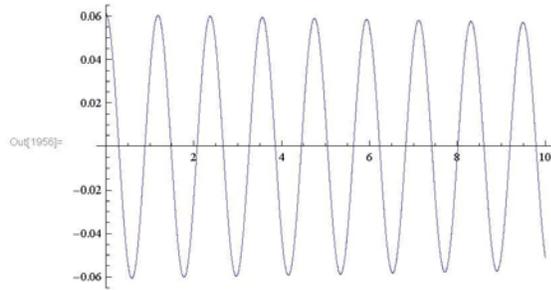
```
In[1953]=
```

```
In[1954]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],  
w[0] == 0.061, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1954]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[1955]=
```

```
In[1956]= Plot[Evaluate[w[t] /. First[solution]],  
{t, 0, 10}, PlotRange -> {-0.065, 0.065}, PlotPoints -> 30]
```

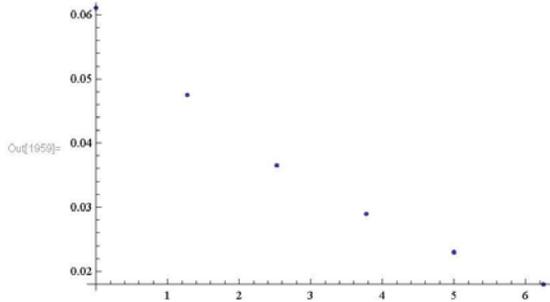


```
In[1957]= solution2 =  
NDSolve[{D[w[t], t, t] == -0.40*D[w[t], t] - 28.03*w[t], w[0] == 0.061, w'[0] == 0},  
w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1957]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

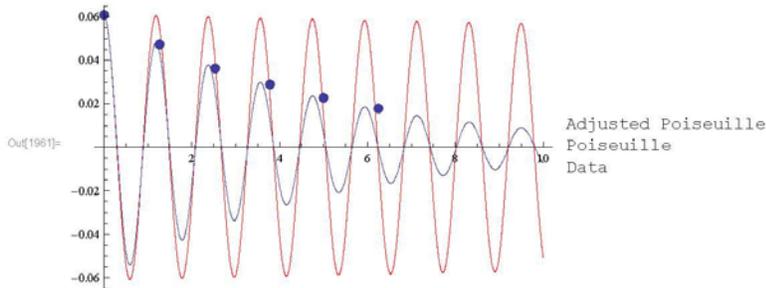
```
In[1958]=
```

```
In[1959]= ListPlot[{{0, 0.061}, {1.275, 0.0475},
{2.525, 0.0365}, {3.775, 0.029}, {5, 0.023}, {6.25, 0.018}}]
```



```
In[1960]=
```

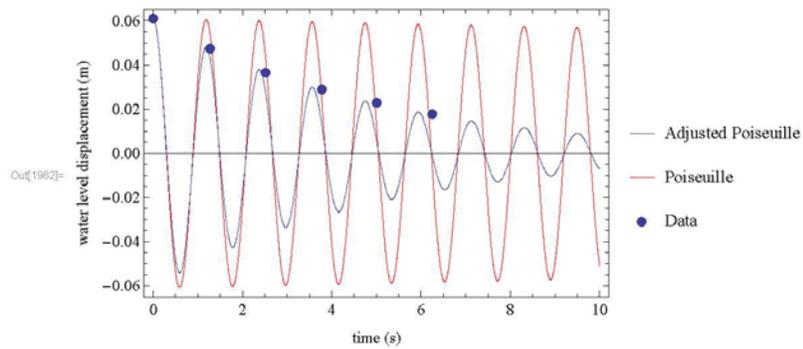
```
In[1961]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .065, .065},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .065, .065},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
{6.25, 0.018}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1962]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .065, .065},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .065, .065},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.061}, {1.275, 0.0475}, {2.525, 0.0365}, {3.775, 0.029}, {5, 0.023},
  {6.25, 0.018}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test 7732 Models

```
In[2048]= Remove["Global`*"]
In[2049]= L = 0.4
Out[2049]= 0.4

In[2050]= fD = 0.035
Out[2050]= 0.035

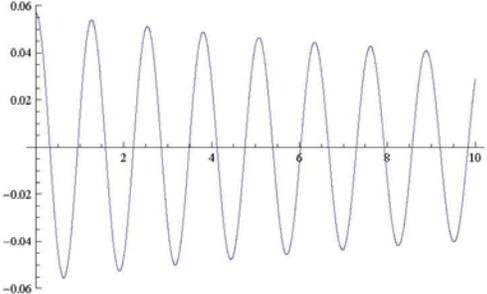
In[2051]= g = 9.81
           r = 0.02385
Out[2051]= 9.81
Out[2052]= 0.02385

In[2053]= fD / (4 r)
Out[2053]= 0.366876

In[2054]= g / L
Out[2054]= 24.525

In[2055]=
In[2056]= solution = NDSolve[
           { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
             w[0] == 0.057, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2056]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

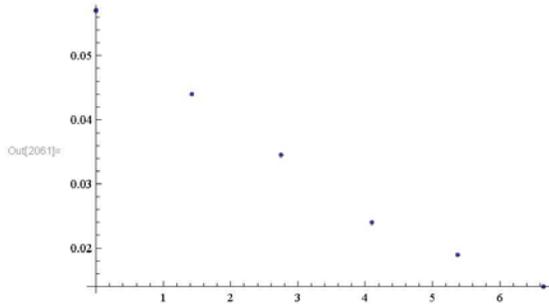
In[2057]=
In[2058]= Plot[Evaluate[w[t] /. First[solution]],
           {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[2058]=
```



```
In[2059]= solution2 =
  NDSolve[{D[w[t], t, t] == -2 * (D[w[t], t]) * Abs[(D[w[t], t])] - 24.525 * w[t],
    w[0] == 0.057, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2059]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

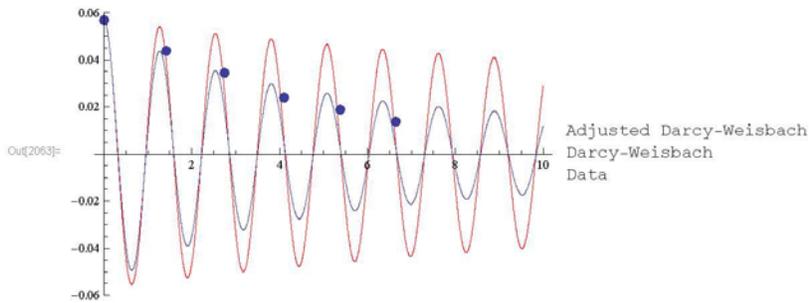
```
In[2060]=
```

```
In[2061]= ListPlot[{{0, 0.057}, {1.425, 0.044},
  {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019}, {6.65, 0.014}}]
```



```
In[2062]=
```

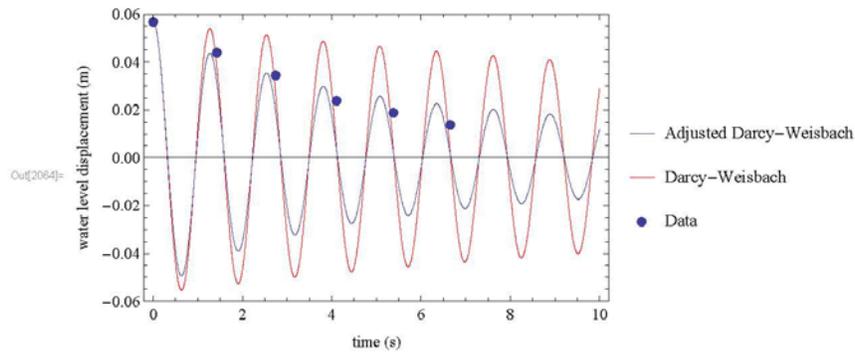
```
In[2063]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
  ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
    {6.65, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[2064]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
  {6.65, 0.014}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[2137]= Remove["Global`*"]

In[2138]= L = 0.4
Out[2138]= 0.4

In[2139]= fD = 0.035
Out[2139]= 0.035

In[2140]= g = 9.81
          r = 0.02385
Out[2140]= 9.81
Out[2141]= 0.02385

In[2142]= fD / (4 r)
Out[2142]= 0.366876

In[2143]= g / L
Out[2143]= 24.525

In[2144]= 1 / (2 L)
Out[2144]= 1.25

In[2145]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
            1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]] , w[0] == 0.057, w'[0] == 0 },
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2145]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[2146]=

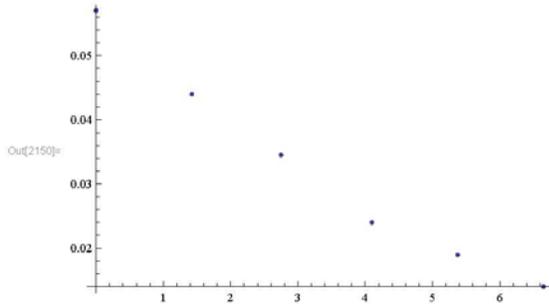
In[2147]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[2147]=

```

```
In[2148]= solution2 = NDSolve[
  { D[w[t], t, t] = - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] = 0.057, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2148]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

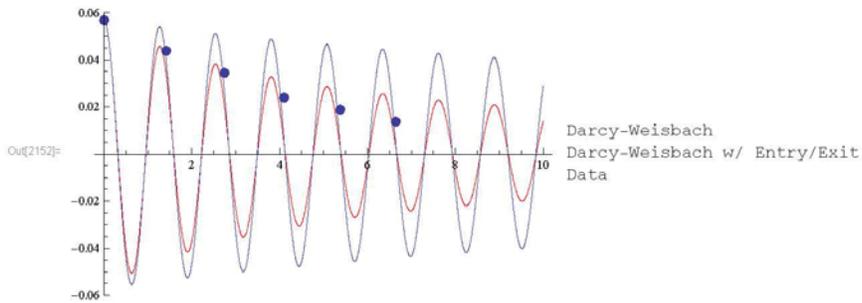
In[2149]=

```
In[2150]= ListPlot[{{0, 0.057}, {1.425, 0.044},
  {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019}, {6.65, 0.014}}]
```



In[2151]=

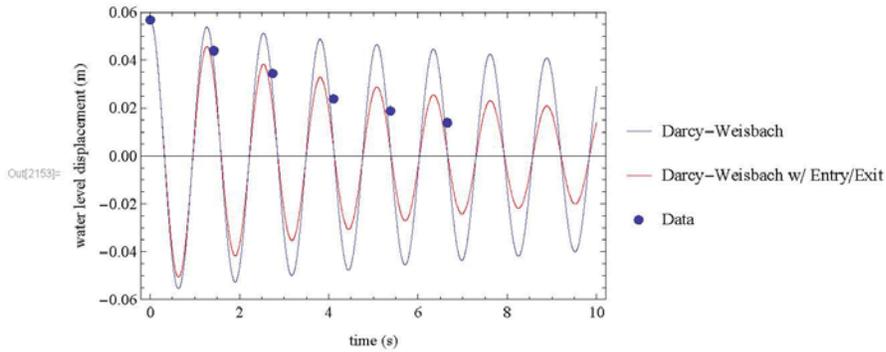
```
In[2152]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
  {6.65, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[2153]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
  {6.65, 0.014}}, PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[2220]= Remove["Global`*"]

In[2221]= L = 0.4
          v = 10^-6
          g = 9.81
          r = 0.02385
          g/L
          (8*v)/(r^2)
Out[2221]= 0.4

          1
Out[2222]= ----
          1000000

Out[2223]= 9.81

Out[2224]= 0.02385

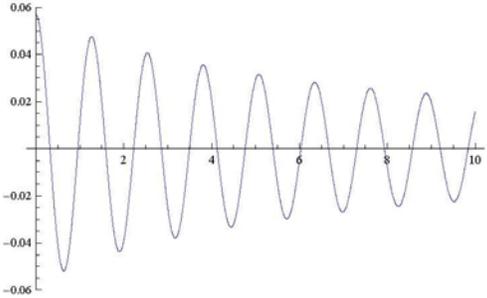
Out[2225]= 24.525

Out[2226]= 0.0140641

In[2227]= 1/(2L)
Out[2227]= 1.25

In[2228]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
          1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.057, w'[0] == 0},
          w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[2228]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

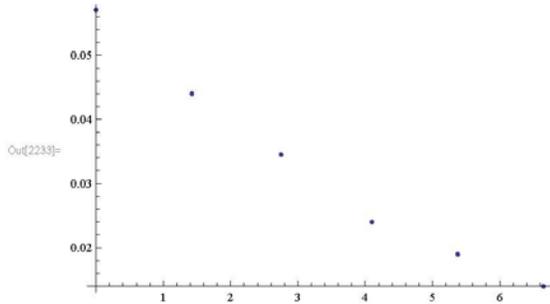
In[2229]=

In[2230]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[2230]=

In[2231]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
          w[0] == 0.057, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2231]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[2232]=

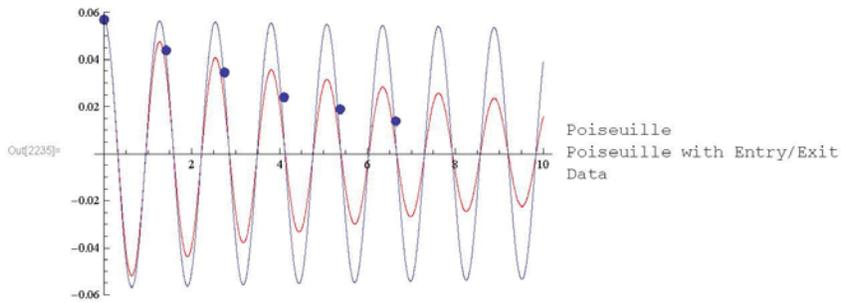
```

```
In[2233]= ListPlot[{{0, 0.057}, {1.425, 0.044},
{2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019}, {6.65, 0.014}}]
```



In[2234]=

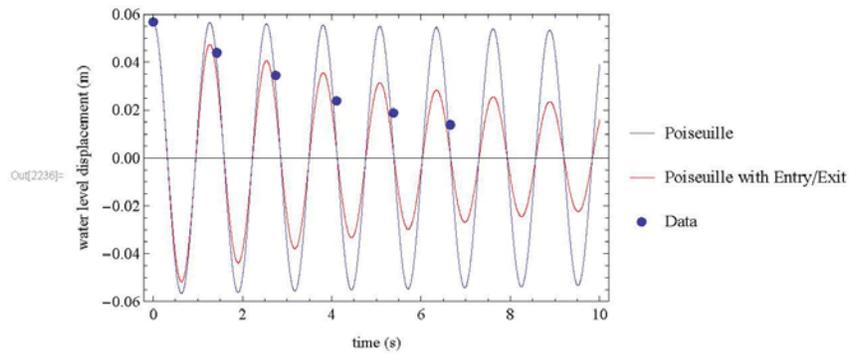
```
In[2235]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille with Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
{6.65, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[2236]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange → {- .06, .06}, PlotPoints → 30, PlotStyle → {Red},
  PlotLegends → {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
PlotPoints → 30, PlotLegends → {Style["Poiseuille"]}],
ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
{6.65, 0.014}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[2271]= Remove["Global`*"]

In[2272]= L = 0.4
          v = 10^-6
          g = 9.81
          r = 0.02385
          g/L
          (8*v)/(r^2)
Out[2272]= 0.4

          1
Out[2273]= ----
          1 000 000

Out[2274]= 9.81

Out[2275]= 0.02385

Out[2276]= 24.525

Out[2277]= 0.0140641

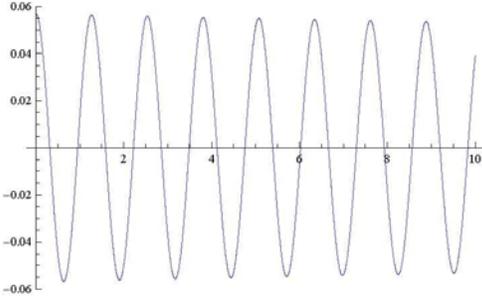
In[2278]=

In[2279]=

In[2280]=

In[2281]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
          w[0] == 0.057, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2281]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[2282]=

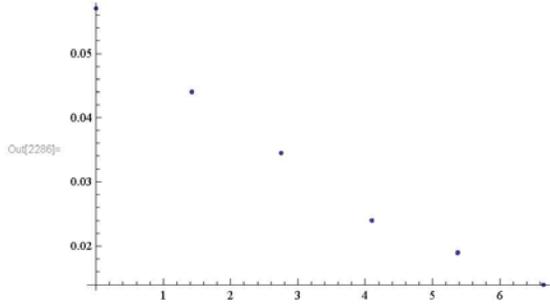
In[2283]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[2283]=


In[2284]= solution2 = NDSolve[
          {D[w[t], t, t] == -0.285*D[w[t], t] - 24.525*w[t], w[0] == 0.057, w'[0] == 0},
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[2284]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

```

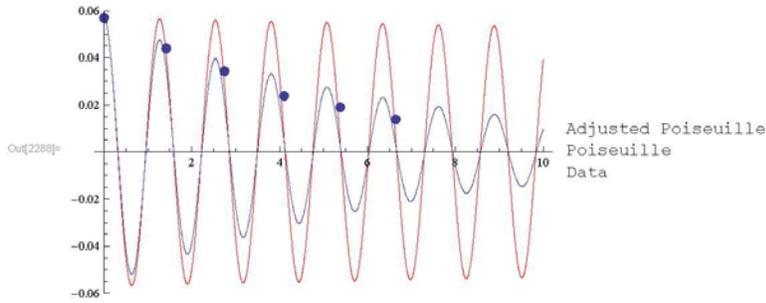
In[2285]=

```
In[2286]= ListPlot[{{0, 0.057}, {1.425, 0.044},
                  {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019}, {6.65, 0.014}}]
```



In[2287]=

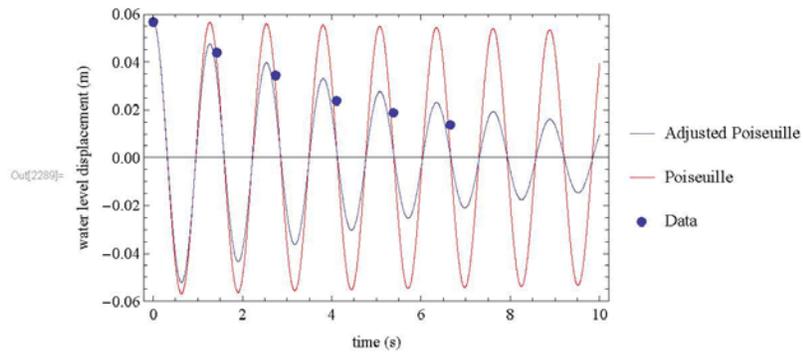
```
In[2288]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
                  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
               Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
                  PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
               ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
                       {6.65, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[2289]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.057}, {1.425, 0.044}, {2.75, 0.0345}, {4.1, 0.024}, {5.375, 0.019},
  {6.65, 0.014}}, PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test 7735 Models

```
In[170]= Remove["Global`*"]
```

```
In[171]= L = 0.3
```

```
Out[171]= 0.3
```

```
In[172]= fD = 0.04
```

```
Out[172]= 0.04
```

```
In[173]= g = 9.81
```

```
r = 0.01
```

```
Out[173]= 9.81
```

```
Out[174]= 0.01
```

```
In[175]= fD / (4 r)
```

```
Out[175]= 1.
```

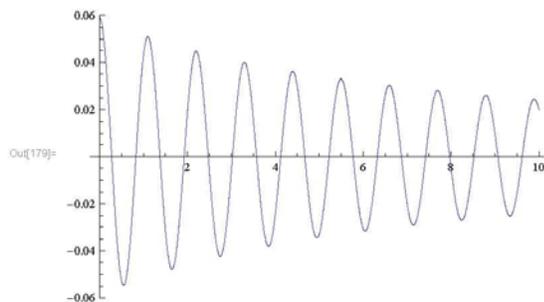
```
In[176]= g / L
```

```
Out[176]= 32.7
```

```
In[177]= solution = NDSolve[
  { D[w[t], t, t] == - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
    w[0] == 0.059, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[177]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[178]=
```

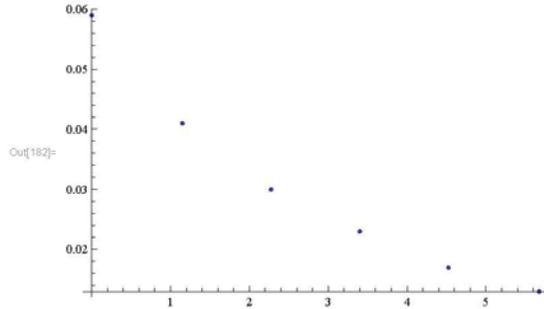
```
In[179]= Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 10}, PlotRange -> {- .06, .06}, PlotPoints -> 30]
```



```
In[180]= solution2 = NDSolve[{ D[w[t], t, t] == -3 * (D[w[t], t]) * Abs[D[w[t], t]] - 32.7 * w[t],
  w[0] == 0.059, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[180]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

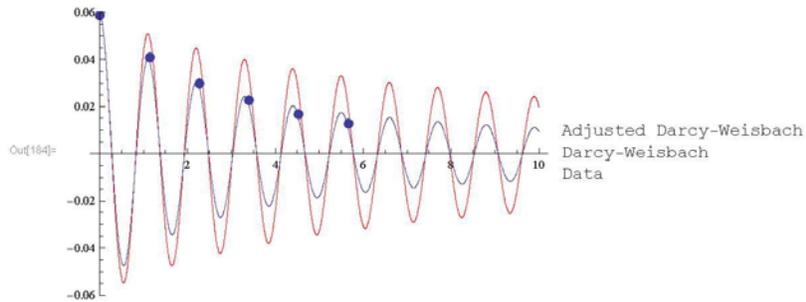
```
In[181]=
```

```
In[182]= ListPlot[{{0, 0.059}, {1.15, 0.041},
{2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017}, {5.675, 0.013}}]
```



```
In[183]=
```

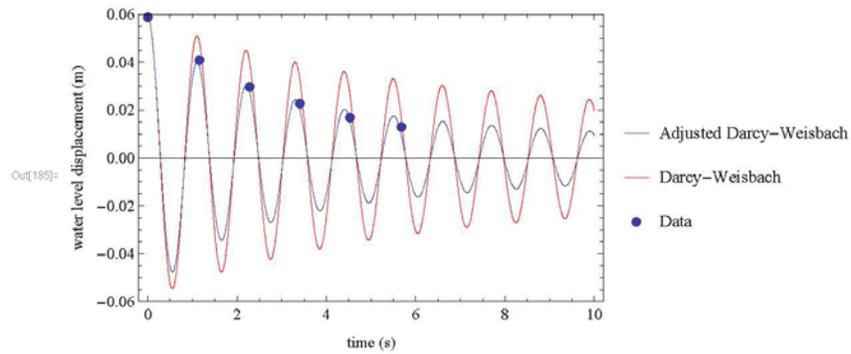
```
In[184]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"},
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"},
ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017},
{5.675, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[185]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023},
  {4.525, 0.017}, {5.675, 0.013}}, PlotLegends → {Style["Data"]},
  PlotMarkers → {Automatic, Medium}], Frame → True,
  FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[68]= Remove["Global`*"]

In[69]= L = 0.3
Out[69]= 0.3

In[70]= fD = 0.04
Out[70]= 0.04

In[71]= g = 9.81
        r = 0.01
Out[71]= 9.81

Out[72]= 0.01

In[73]= fD / (4 r)
Out[73]= 1.

In[74]= g / L
Out[74]= 32.7

In[75]= 1 / (2 L)
Out[75]= 1.66667

In[76]= solution = NDSolve[
        { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
          1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]], w[0] == 0.059, w'[0] == 0 },
        w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[76]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

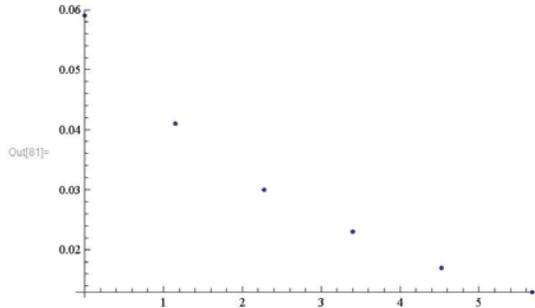
In[77]=

In[78]= Plot[Evaluate[w[t] /. First[solution]],
        {t, 0, 10}, PlotRange -> {- .06, .06}, PlotPoints -> 30]
Out[78]=

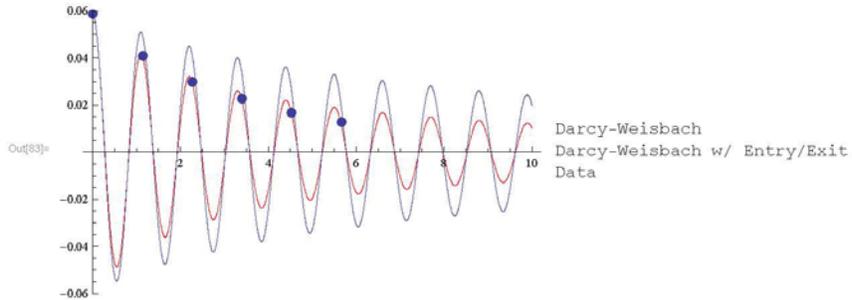
```

```
In[79]= solution2 = NDSolve[
  { D[w[t], t, t] = - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] = 0.059, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[79]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[80]=
In[81]= ListPlot[{{0, 0.059}, {1.15, 0.041},
  {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017}, {5.675, 0.013}}]
```



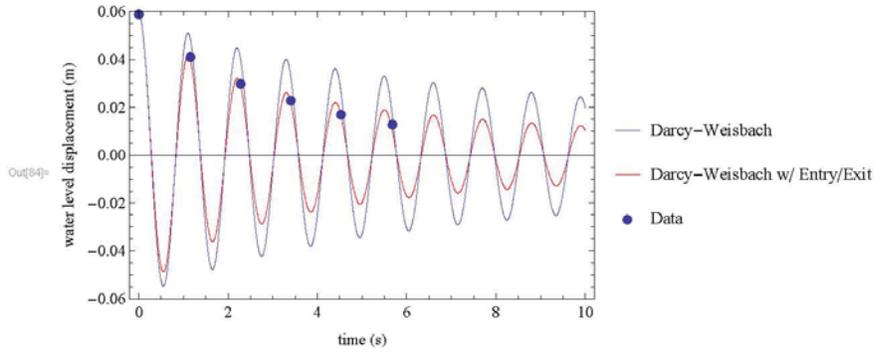
```
In[82]=
In[83]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017},
  {5.675, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[84]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023},
  {4.525, 0.017}, {5.675, 0.013}}, PlotLegends -> {Style["Data"]},
  PlotMarkers -> {Automatic, Medium}], Frame -> True,
FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[50]= Remove["Global`*"]

In[51]= L = 0.3
v = 10^-6
g = 9.81
r = 0.01
g/L
(8*v)/(r^2)
Out[51]= 0.3

Out[52]=  $\frac{1}{1000000}$ 

Out[53]= 9.81

Out[54]= 0.01

Out[55]= 32.7

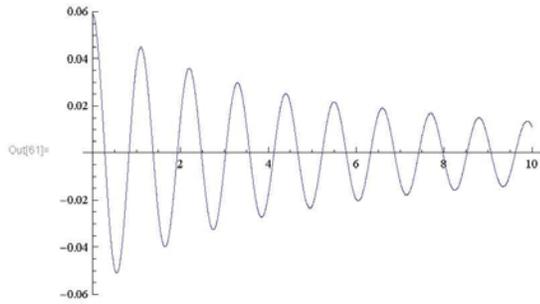
Out[56]= 0.08

In[57]= 1/(2L)
Out[57]= 1.66667

In[58]=

In[59]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.059, w'[0] == 0},
w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[59]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[60]=

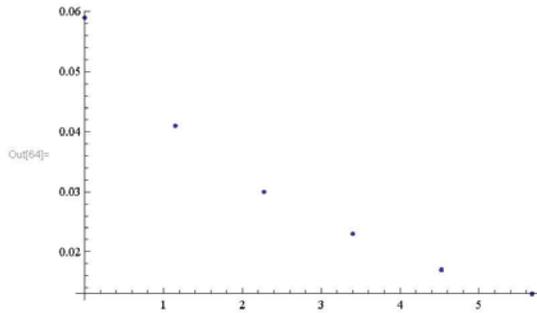
In[61]= Plot[Evaluate[w[t] /. First[solution]],
{t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[61]= 

In[62]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] == 0.059, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[62]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

```

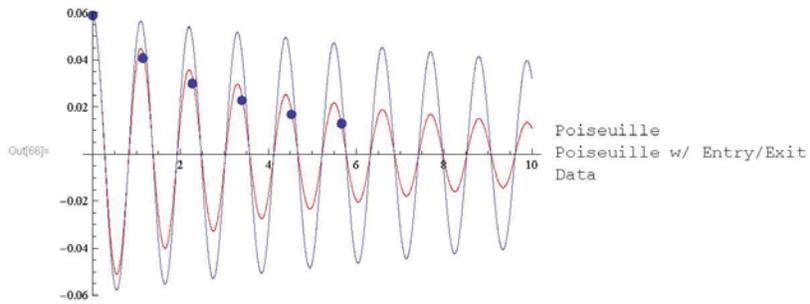
In[63] =

```
In[64] = ListPlot[{{0, 0.059}, {1.15, 0.041},
                 {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017}, {5.675, 0.013}}]
```



In[65] =

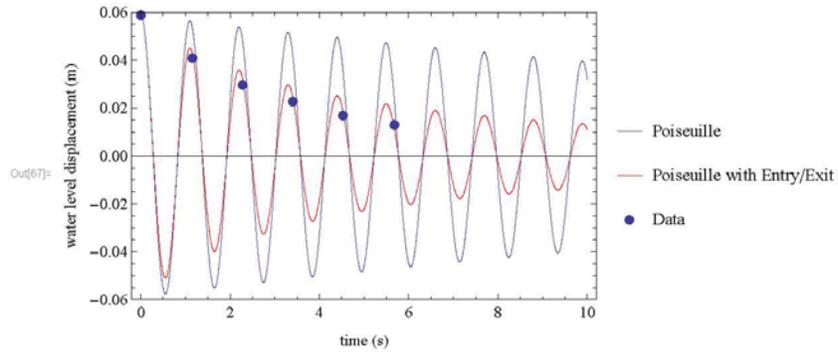
```
In[66] = Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
                  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille w/ Entry/Exit"],
              Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
                  PlotRange -> {- .06, .06}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
              ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017},
                       {5.675, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[67]:= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange → {- .06, .06}, PlotPoints → 30, PlotStyle → {Red},
  PlotLegends → {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Poiseuille"]}], ListPlot[{{0, 0.059},
  {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017}, {5.675, 0.013}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```

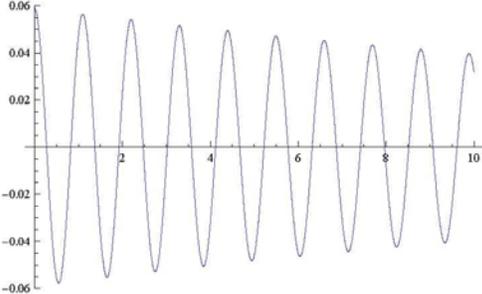


```

In[85]= Remove["Global`*"]

In[86]= L = 0.3
v = 10^-6
g = 9.81
r = 0.01
g/L
(8*v)/(r^2)
Out[86]= 0.3
Out[87]=  $\frac{1}{1000000}$ 
Out[88]= 9.81
Out[89]= 0.01
Out[90]= 32.7
Out[91]= 0.08

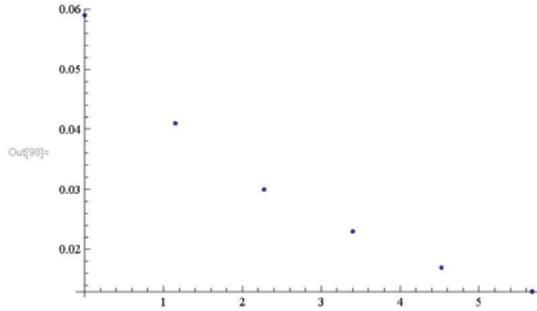
In[92]=
In[93]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] = 0.059, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[93]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[94]=
In[95]= Plot[Evaluate[w[t] /. First[solution]],
{t, 0, 10}, PlotRange -> {-0.06, 0.06}, PlotPoints -> 30]
Out[95]= 
In[96]= solution2 =
NDSolve[{D[w[t], t, t] == -0.53*D[w[t], t] - 32.7*w[t], w[0] = 0.059, w'[0] == 0},
w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[96]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

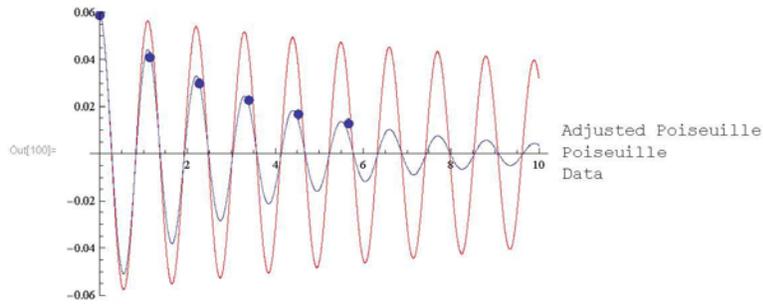
In[97]=

```

```
In[98]= ListPlot[{{0, 0.059}, {1.15, 0.041},
{2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017}, {5.675, 0.013}}]
```



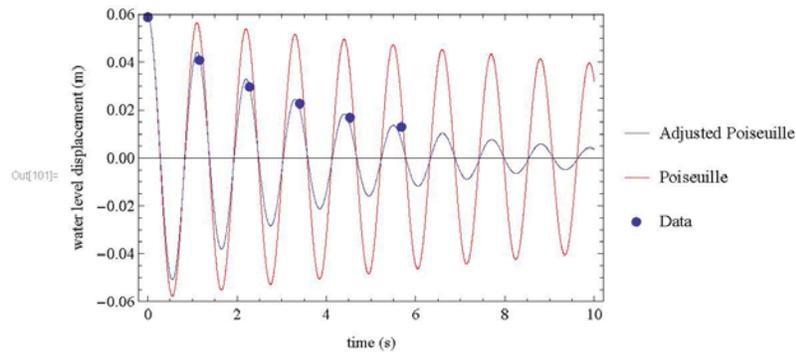
```
In[99]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .06, .06},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023}, {4.525, 0.017},
{5.675, 0.013}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[101]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .06, .06},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.059}, {1.15, 0.041}, {2.275, 0.03}, {3.4, 0.023},
  {4.525, 0.017}, {5.675, 0.013}}, PlotLegends → {Style["Data"]},
  PlotMarkers → {Automatic, Medium}], Frame → True,
  FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test 7740 Models

```
In[137]:= Remove["Global`*"]
In[138]:= L = 0.35
Out[138]= 0.35

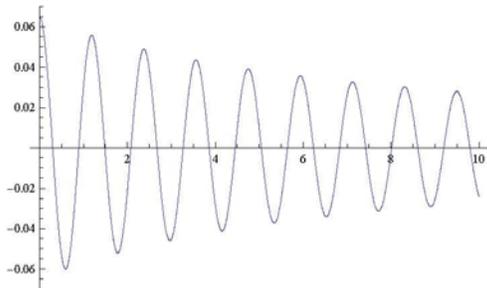
In[139]:= fD = 0.038
Out[139]= 0.038

In[140]:= g = 9.81
r = 0.01
Out[140]= 9.81
Out[141]= 0.01

In[142]:= fD / (4 r)
g / L
Out[142]= 0.95
Out[143]= 28.0286

In[144]:= solution = NDSolve[
  { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
    w[0] == 0.065, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[144]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

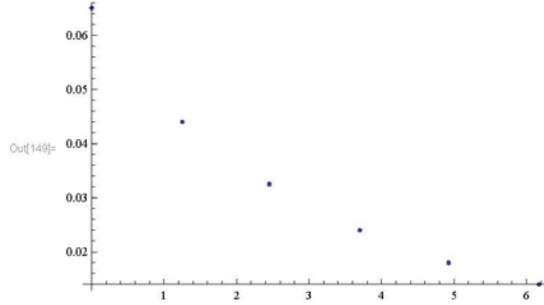
In[145]=

In[146]:= Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]
Out[146]= 

In[147]:= solution2 =
  NDSolve[{D[w[t], t, t] == -2.9 * (D[w[t], t]) * Abs[D[w[t], t]] - 28.03 * w[t],
    w[0] == 0.065, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[147]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

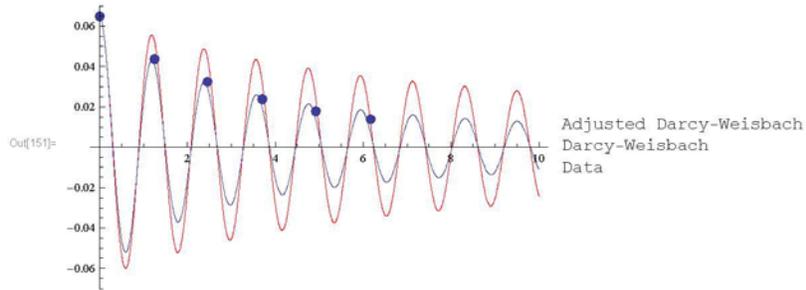
In[148]=
```

```
In[149]= ListPlot[{{0, 0.065}, {1.25, 0.044},
{2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018}, {6.175, 0.014}}]
```



```
In[150]=
```

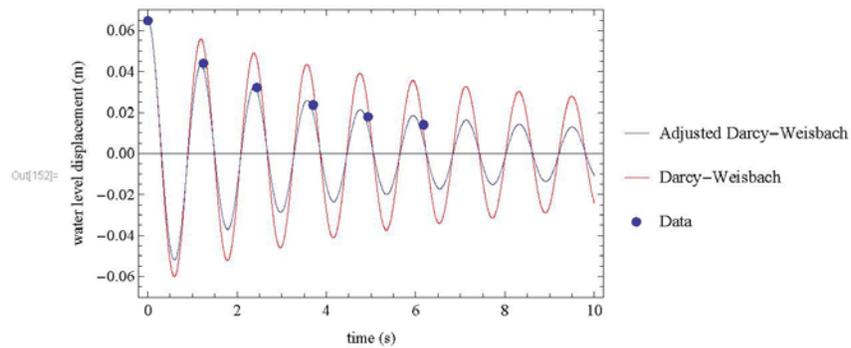
```
In[151]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"},
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"},
ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018},
{6.175, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[152]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024},
  {4.925, 0.018}, {6.175, 0.014}}, PlotLegends → {Style["Data"]}],
  PlotMarkers → {Automatic, Medium}], Frame → True,
  FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[201]= Remove["Global`*"]

In[202]= L = 0.35
Out[202]= 0.35

In[203]= fD = 0.038
Out[203]= 0.038

In[204]= g = 9.81
          r = 0.01
Out[204]= 9.81
Out[205]= 0.01

In[206]= fD / (4 r)
          g / L
Out[206]= 0.95
Out[207]= 28.0286

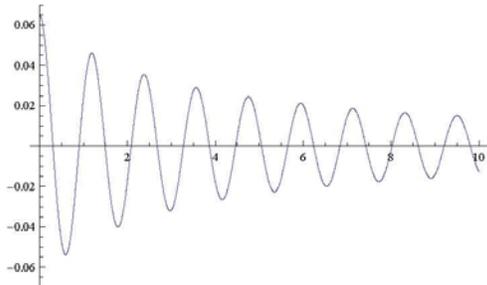
In[208]= 1 / (2 L)
Out[208]= 1.42857

In[209]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t] -
            1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]], w[0] == 0.065, w'[0] == 0 },
          w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[209]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[210]=

In[211]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]
Out[211]=

```



```

In[212]= solution2 = NDSolve[
  { D[w[t], t, t] == -(fD / (4 x)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] == 0.065, w'[0] == 0 }, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[212]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

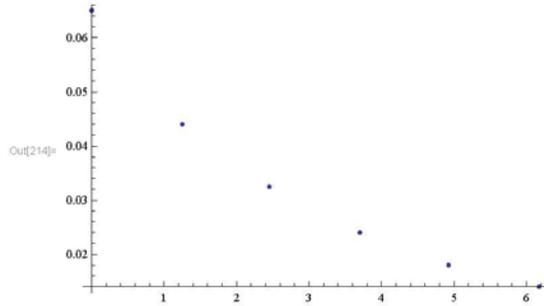
```

In[213]=

```

In[214]= ListPlot[{{0, 0.065}, {1.25, 0.044},
  {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018}, {6.175, 0.014}}]

```

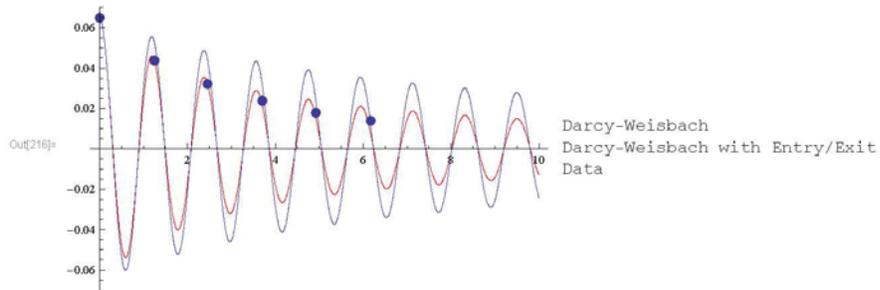


In[215]=

```

In[216]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 10}, PlotRange -> {- .07, .07}, PlotPoints -> 30,
  PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach with Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018},
  {6.175, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]

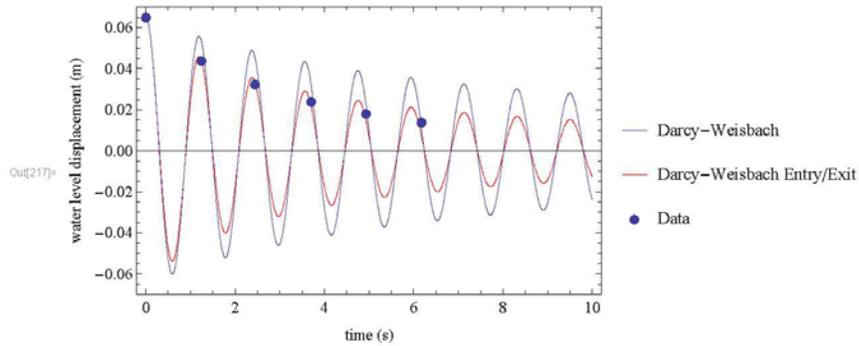
```



```

In[217]:= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .07, .07}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach Entry/Exit"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
  PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
  ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024},
  {4.925, 0.018}, {6.175, 0.014}}, PlotLegends -> {Style["Data"]},
  PlotMarkers -> {Automatic, Medium}], Frame -> True,
  FrameLabel -> {"time (s)", "water level displacement (m)"},
  LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

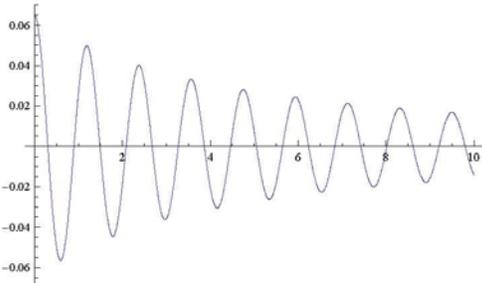
In[218]= Remove["Global`*"]

In[219]= L = 0.35
v = 10^-6
g = 9.81
r = 0.01
g/L
(8*v)/(r^2)
Out[219]= 0.35

Out[220]=  $\frac{1}{1000000}$ 
Out[221]= 9.81
Out[222]= 0.01
Out[223]= 28.0286
Out[224]= 0.08

In[225]= 1/(2L)
Out[225]= 1.42857

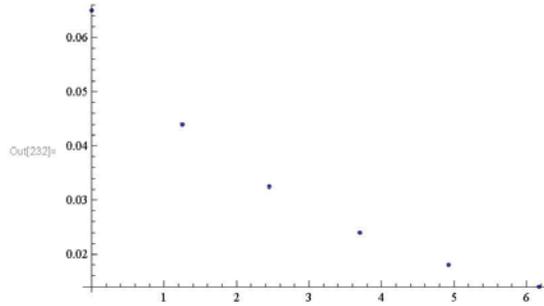
In[226]=
In[227]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.065, w'[0] == 0},
w, {t, 0, 10}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[227]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

In[228]=
In[229]= Plot[Evaluate[w[t] /. First[solution]],
{t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]
Out[229]=

In[230]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] == 0.065, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[230]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}

```

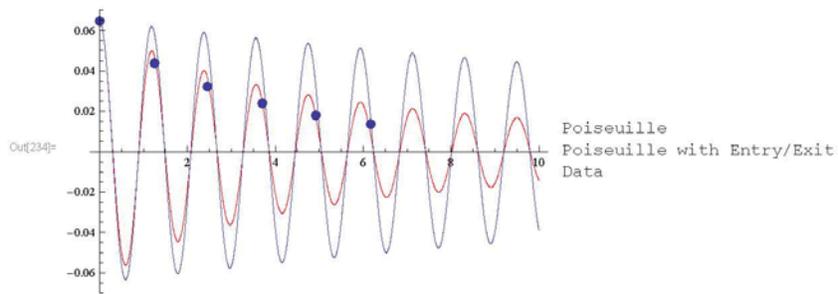
In[231]=

```
In[232]= ListPlot[{{0, 0.065}, {1.25, 0.044},
                 {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018}, {6.175, 0.014}}]
```



In[233]=

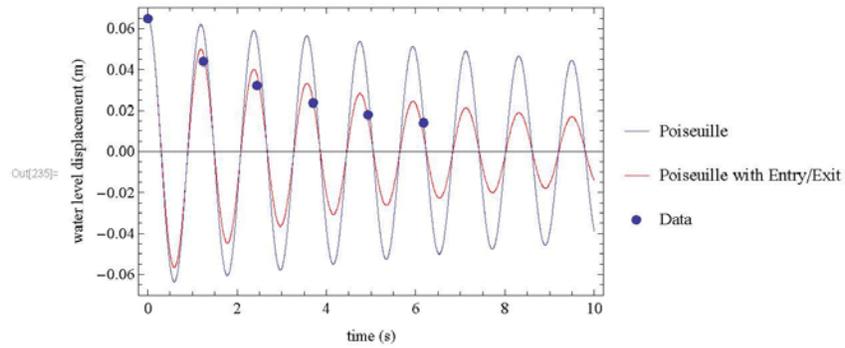
```
In[234]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
                  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille with Entry/Exit"],
              Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10},
                  PlotRange -> {- .07, .07}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
              ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018},
                       {6.175, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[235]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10},
  PlotRange -> {- .07, .07}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
  PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}], ListPlot[{{0, 0.065},
  {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018}, {6.175, 0.014}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```
In[236]= Remove["Global`*"]
```

```
In[237]= L = 0.35  
v = 10^-6  
g = 9.81  
r = 0.01  
g/L  
(8*v)/(r^2)
```

```
Out[237]= 0.35
```

```
Out[238]=  $\frac{1}{1000000}$ 
```

```
Out[239]= 9.81
```

```
Out[240]= 0.01
```

```
Out[241]= 28.0286
```

```
Out[242]= 0.08
```

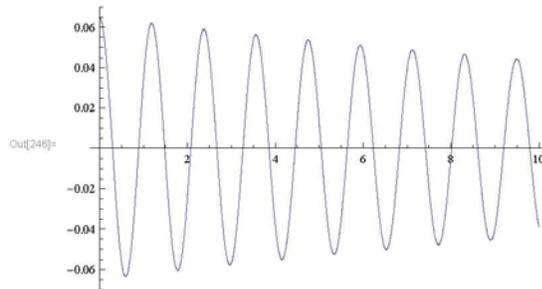
```
In[243]=
```

```
In[244]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],  
w[0] == 0.065, w'[0] == 0}, w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[244]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

```
In[245]=
```

```
In[246]= Plot[Evaluate[w[t] /. First[solution]],  
{t, 0, 10}, PlotRange -> {-0.07, 0.07}, PlotPoints -> 30]
```

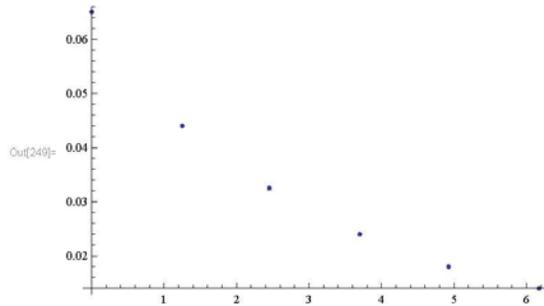


```
In[247]= solution2 =  
NDSolve[{D[w[t], t, t] == -0.54*D[w[t], t] - 28.03*w[t], w[0] == 0.065, w'[0] == 0},  
w, {t, 0, 10}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[247]= {{w -> InterpolatingFunction[{{0., 10.}}, <>]}}
```

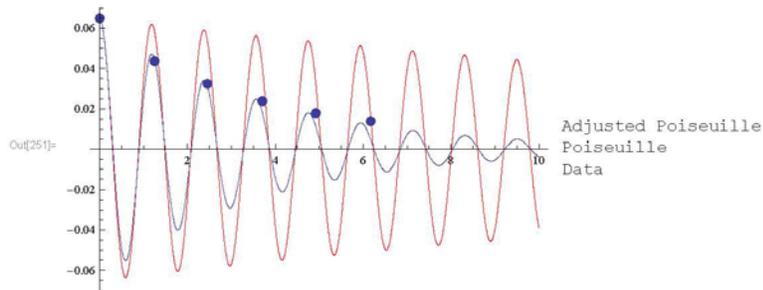
```
In[248]=
```

```
In[249]= ListPlot[{{0, 0.065}, {1.25, 0.044},
{2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018}, {6.175, 0.014}}]
```



In[250]=

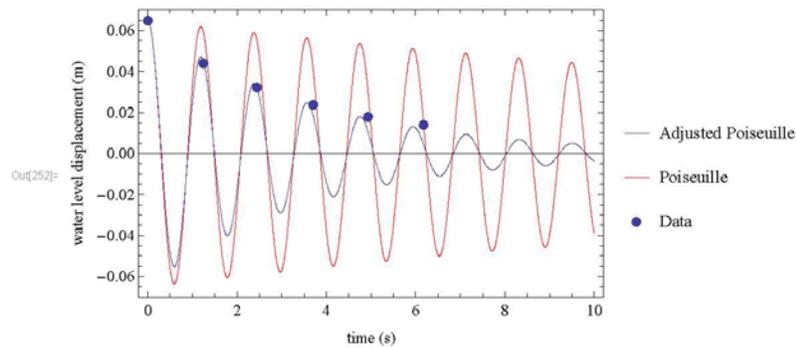
```
In[251]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange -> {- .07, .07},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024}, {4.925, 0.018},
{6.175, 0.014}}, PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

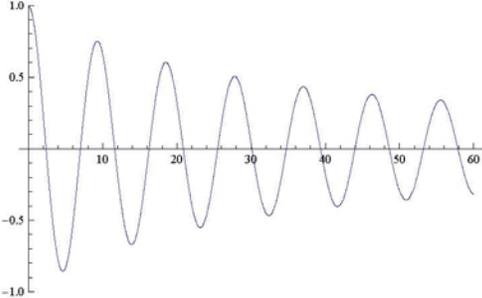
In[252]:= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 10}, PlotRange → {- .07, .07},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.065}, {1.25, 0.044}, {2.45, 0.0325}, {3.7, 0.024},
  {4.925, 0.018}, {6.175, 0.014}}, PlotLegends → {Style["Data"]},
  PlotMarkers → {Automatic, Medium}], Frame → True,
  FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test G3878 Models

```
In[97]= Remove["Global`*"]
In[98]= L = 21.24151
Out[98]= 21.2415
In[99]= fD = 0.02
Out[99]= 0.02
In[100]= g = 9.81
r = 0.0508
Out[100]= 9.81
Out[101]= 0.0508
In[102]= fD / (4 r)
Out[102]= 0.0984252
In[103]= g / L
Out[103]= 0.461832
In[104]= 1 / (2 L)
Out[104]= 0.0235388
In[105]= solution = NDSolve[
  { D[w[t], t, t] = -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t] -
    1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] = 0.995, w'[0] == 0},
  w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[105]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
In[106]=
In[107]=
In[108]= Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[108]=
```



```

In[109]= solution2 = NDSolve[
  { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] == 0.995, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[109]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

```

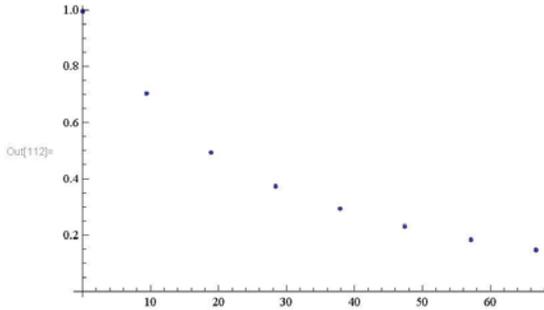
In[110]=

In[111]=

```

In[112]= ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
  {37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}}]

```

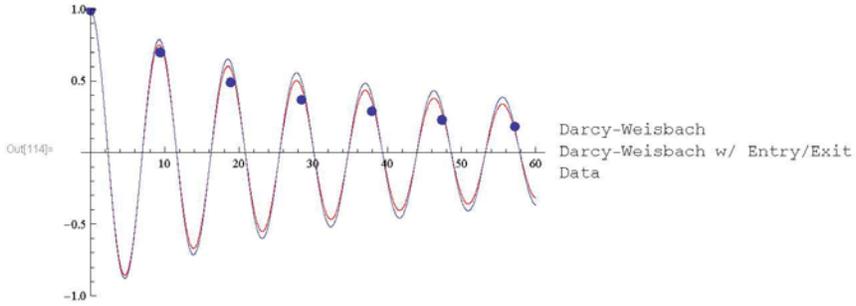


In[113]=

```

In[114]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"},
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"},
  ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
  {37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]

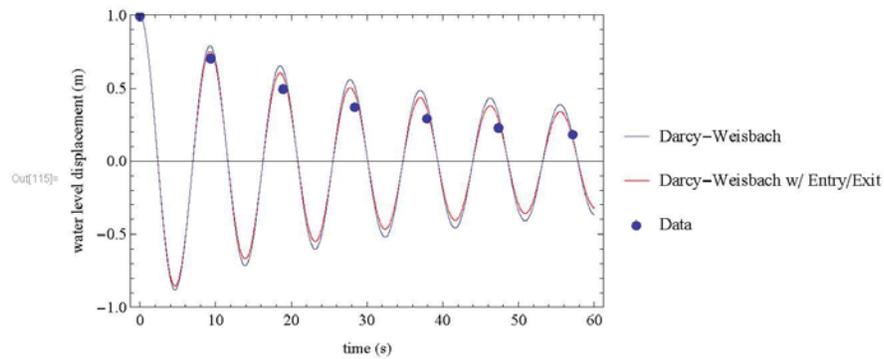
```



```

In[115]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
  {37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[63]= Remove["Global`*"]

In[64]= L = 21.24151
      fD = 0.02
Out[64]= 21.2415
Out[65]= 0.02

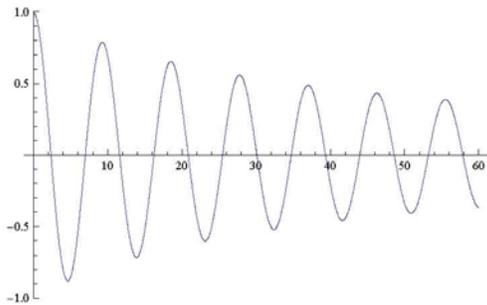
In[66]= g = 9.81
      r = 0.0508
Out[66]= 9.81
Out[67]= 0.0508

In[68]= (fD / (4 r))
      g / L
Out[68]= 0.0984252
Out[69]= 0.461832

In[70]=

In[71]= solution = NDSolve[
      { D[w[t], t, t] == - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
      w[0] == 0.995, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[71]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

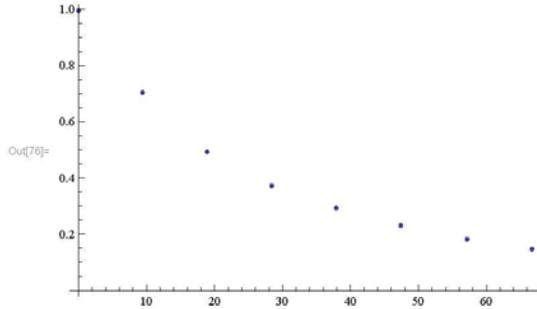
In[72]=

In[73]= Plot[Evaluate[w[t] /. First[solution]],
      {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[73]=

In[74]= solution2 =
      NDSolve[{ D[w[t], t, t] == -0.185 * (D[w[t], t]) * Abs[D[w[t], t]] - 0.4397 * w[t],
      w[0] == 0.995, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[74]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[75]=

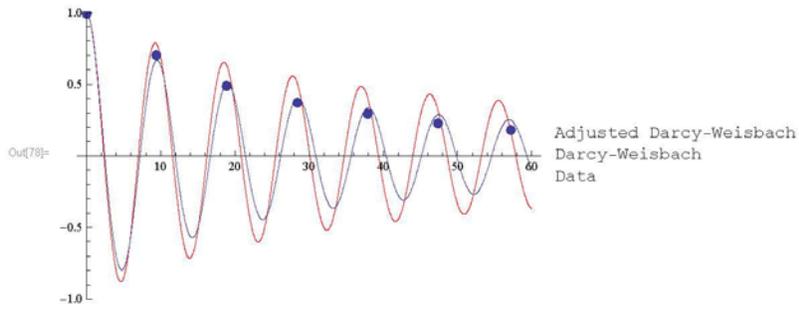
```

```
In[76]= ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
{37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}}]
```



```
In[77]=
```

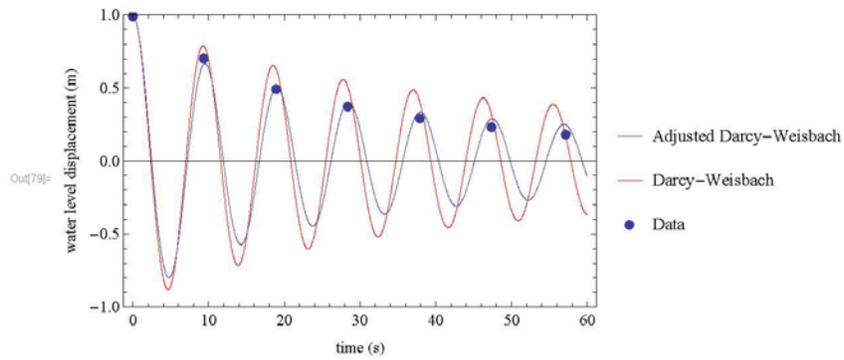
```
In[78]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
{37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[79]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
  {37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[113]= Remove["Global`*"]

In[114]= L = 21.24151
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[114]= 21.2415

Out[115]=  $\frac{1}{1000000}$ 

Out[116]= 9.81

Out[117]= 0.0508

Out[118]= 0.461832

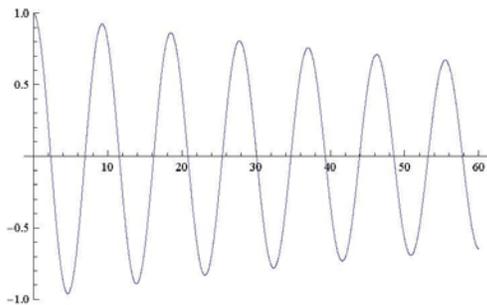
Out[119]= 0.00310001

In[120]= 1/(2L)
Out[120]= 0.0235388

In[121]=
In[122]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
          1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.995, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[122]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[123]=
In[124]=
In[125]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[125]=

```

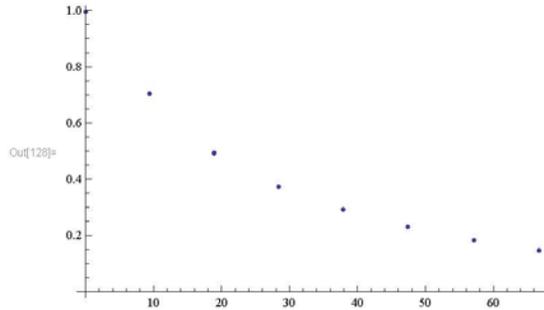


```
In[126]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] == 0.995, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[126]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

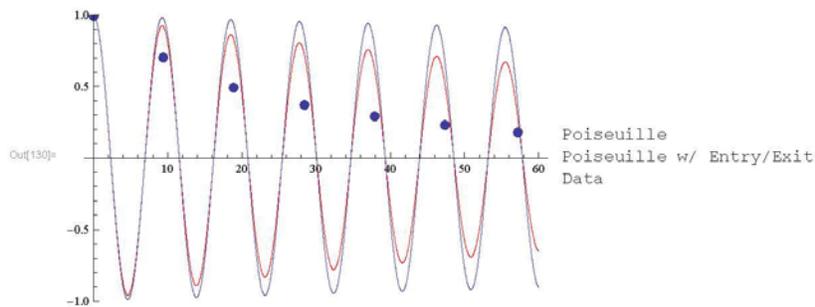
```
In[127]=
```

```
In[128]= ListPlot[{{(0, 0.995), {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
{37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}}]
```



```
In[129]=
```

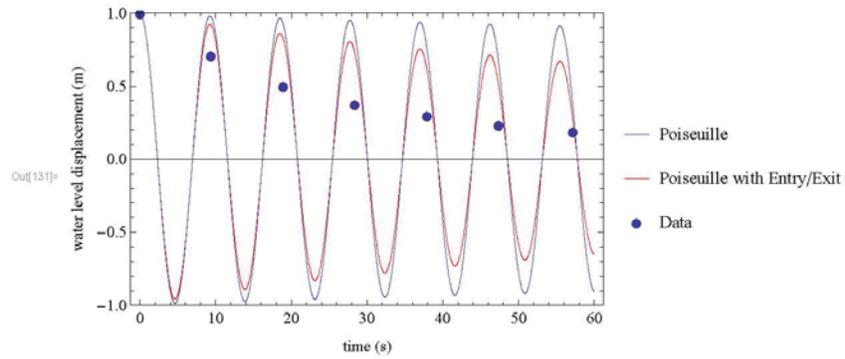
```
In[130]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille w/ Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60},
PlotRange -> {-1, 1}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{(0, 0.995), {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
{37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[131]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
  {37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```
In[274]= Remove["Global`*"]
```

```
In[275]= L = 21.24151
```

```
Out[275]= 21.2415
```

```
In[276]= v = 10^-6  
g = 9.81  
r = 0.0508
```

```
Out[276]=  $\frac{1}{1000000}$ 
```

```
Out[277]= 9.81
```

```
Out[278]= 0.0508
```

```
In[279]= g/L
```

```
Out[279]= 0.461832
```

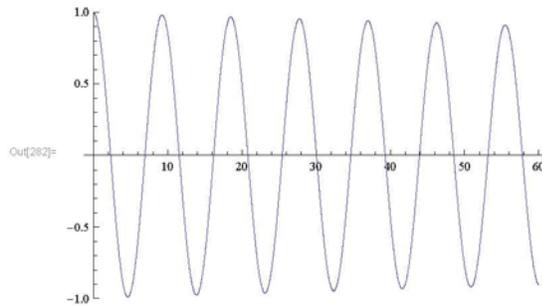
```
In[280]= (8*v)/(r^2)
```

```
Out[280]= 0.00310001
```

```
In[281]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],  
w[0] = 0.995, w'[0] = 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[281]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

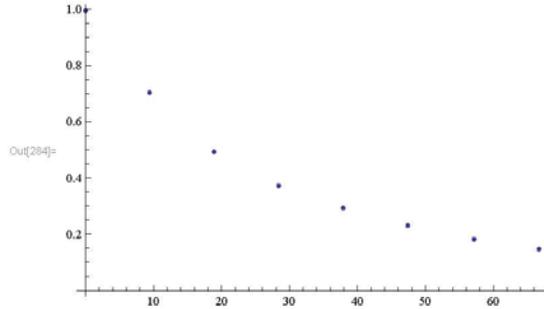
```
In[282]= Plot[Evaluate[w[t] /. First[solution]],  
{t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
```



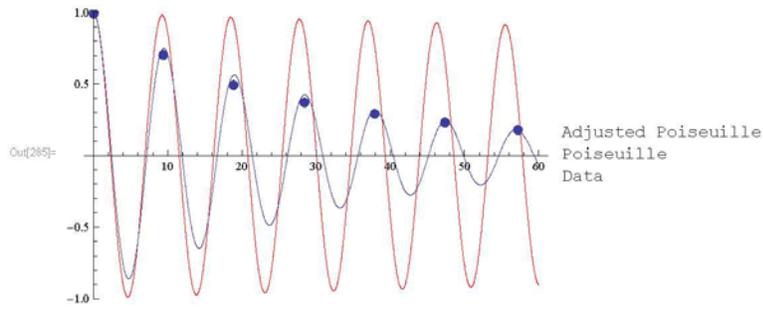
```
In[283]= solution2 = NDSolve[{D[w[t], t, t] == -0.06*D[w[t], t] - 0.4397*w[t], w[0] = 0.995, w'[0] == 0},  
w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[283]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

```
In[284]= ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
{37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}}]
```



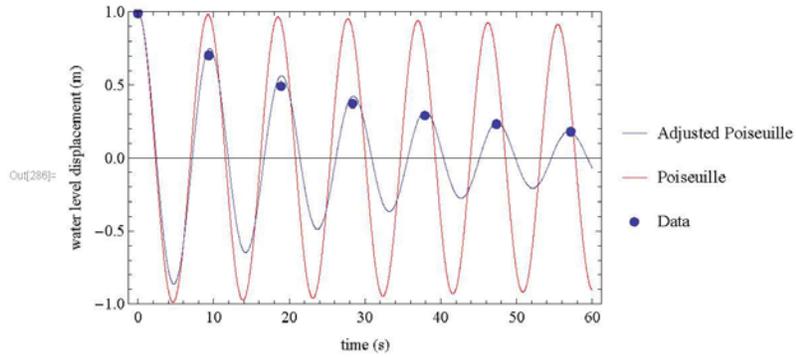
```
In[285]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
{37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[286]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.995}, {9.393, 0.704}, {18.893, 0.493}, {28.394, 0.373},
  {37.894, 0.293}, {47.394, 0.231}, {57.143, 0.183}, {66.714, 0.147}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



In[287]=

In[288]=

In[289]=

Test G3902 Models

```
In[307] = Remove["Global`*"]
In[308] = L = 19.00428
Out[308] = 19.0043

In[309] = fD = 0.02
Out[309] = 0.02

In[310] = g = 9.81
          r = 0.0508
Out[310] = 9.81
Out[311] = 0.0508

In[312] = fD / (4 r)
Out[312] = 0.0984252

In[313] = g / L
Out[313] = 0.5162

In[314] = 1 / (2 L)
Out[314] = 0.0263099

In[315] = solution = NDSolve[
  { D[w[t], t, t] = -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t] -
    1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] = 0.942, w'[0] == 0,
  w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[315] = {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

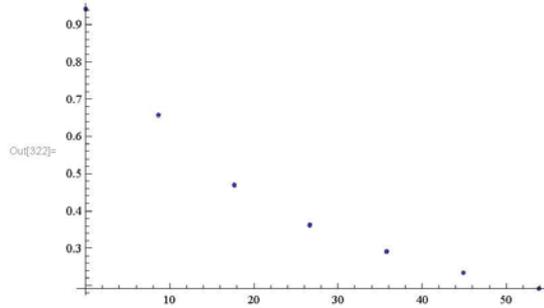
In[316] =
In[317] =
In[318] = Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[318] = 
```

```
In[319]= solution2 = NDSolve[
  { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] == 0.942, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[319]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

In[320]=

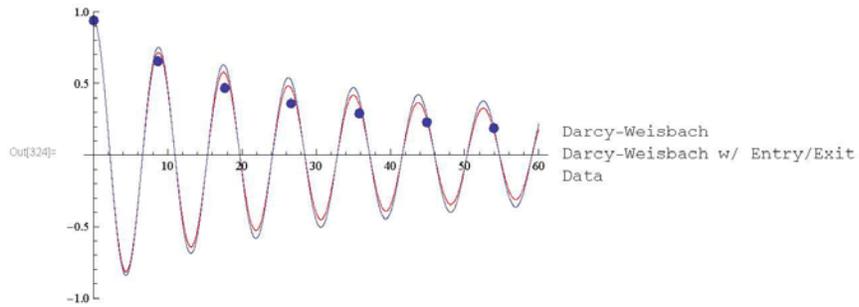
In[321]=

```
In[322]= ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469},
  {26.647, 0.362}, {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}}]
```



In[323]=

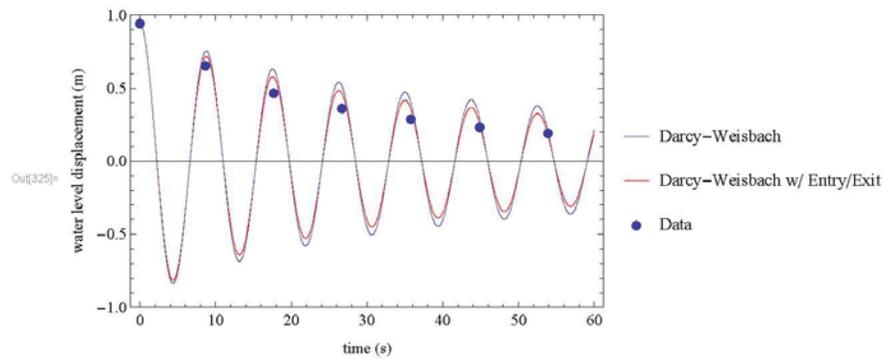
```
In[324]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
  {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[325]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
  {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[361]= Remove["Global`*"]
In[362]= L = 19.00428
Out[362]= 19.0043
In[363]= fD = 0.02
Out[363]= 0.02
In[364]= g = 9.81
Out[364]= 9.81
In[365]= r = 0.0508
Out[365]= 0.0508
In[366]= fD / (4 r)
          g / L
Out[366]= 0.0984252
Out[367]= 0.5162
In[368]=
In[369]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
            w[0] == 0.942, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[369]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
In[370]=
In[371]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[371]=

```

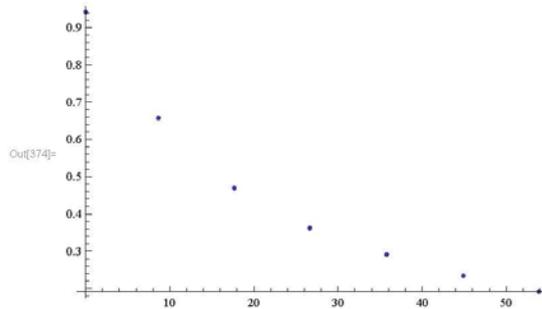
```

In[372]= solution2 =
  NDSolve[{D[w[t], t, t] == -0.17 * (D[w[t], t]) * Abs[D[w[t], t]] - 0.51 * w[t],
    w[0] == 0.942, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[372]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[373]= {{w -> InterpolatingFunction[{{0., 60.}}, "<>"]}}
Out[373]= {{w -> InterpolatingFunction[]}}

In[374]= ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469},
  {26.647, 0.362}, {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}}]

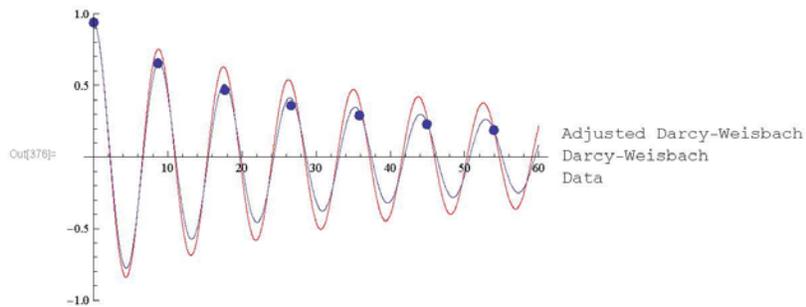
```



```

In[375]=
In[376]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
  ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
    {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]

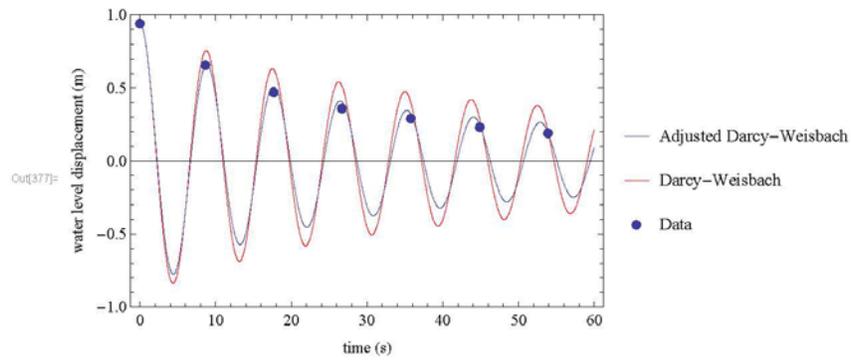
```



```

In[377]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
  {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[378]= Remove["Global`*"]

In[379]= L = 19.00428
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[379]= 19.0043

Out[380]=  $\frac{1}{1000000}$ 

Out[381]= 9.81

Out[382]= 0.0508

Out[383]= 0.5162

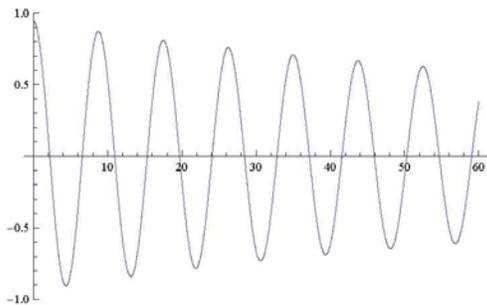
Out[384]= 0.00310001

In[385]= 1/(2L)
Out[385]= 0.0263099

In[386]=
In[387]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
          1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.942, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[387]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[388]=
In[389]=
In[390]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[390]=

```

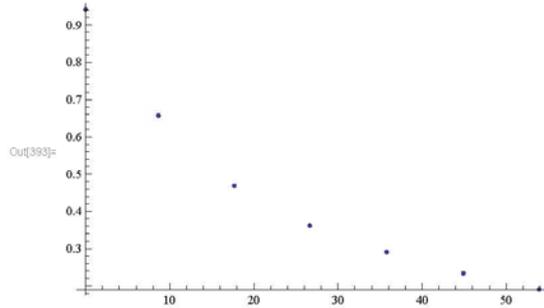


```
In[391]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] = 0.942, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[391]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

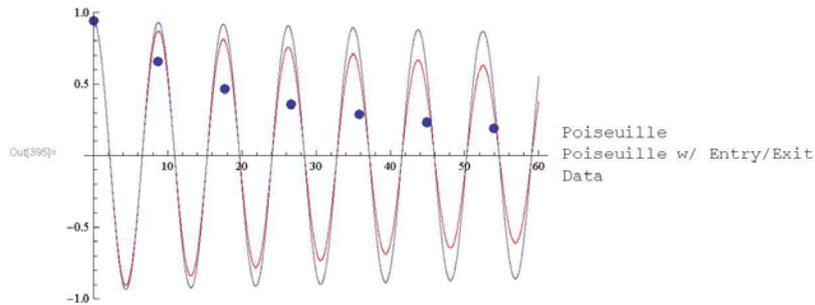
```
In[392]=
```

```
In[393]= ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469},
{26.647, 0.362}, {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}}]
```



```
In[394]=
```

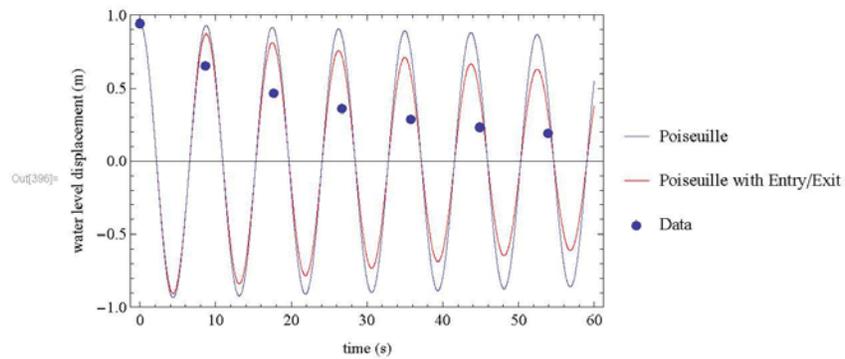
```
In[395]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille w/ Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60},
PlotRange -> {-1, 1}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
{35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[396]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange → {-1, 1}, PlotPoints → 30, PlotStyle → {Red},
  PlotLegends → {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
PlotPoints → 30, PlotLegends → {Style["Poiseuille"]}],
ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
  {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[516]= Remove["Global`*"]

In[517]= L = 19.00428
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[517]= 19.0043

Out[518]=  $\frac{1}{1000000}$ 

Out[519]= 9.81

Out[520]= 0.0508

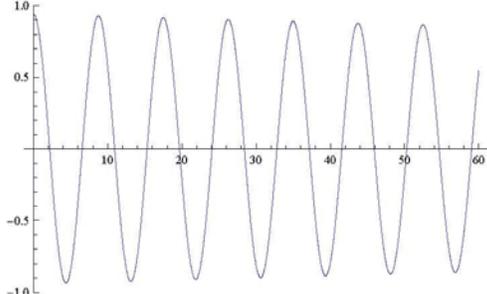
Out[521]= 0.5162

Out[522]= 0.00310001

In[523]=

In[524]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
                             w[0] == 0.942, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[524]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[525]=

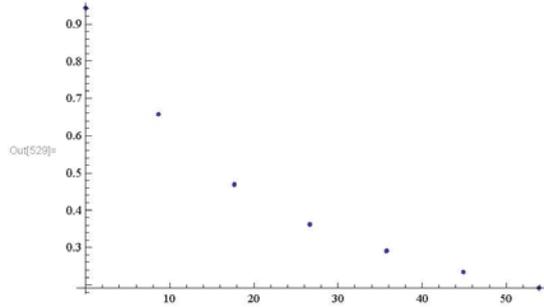
In[526]= Plot[Evaluate[w[t] /. First[solution]],
              {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[526]= 

In[527]= solution2 =
          NDSolve[{D[w[t], t, t] == -0.065*D[w[t], t] - 0.5*w[t], w[0] == 0.942, w'[0] == 0},
                  w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[527]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[528]=

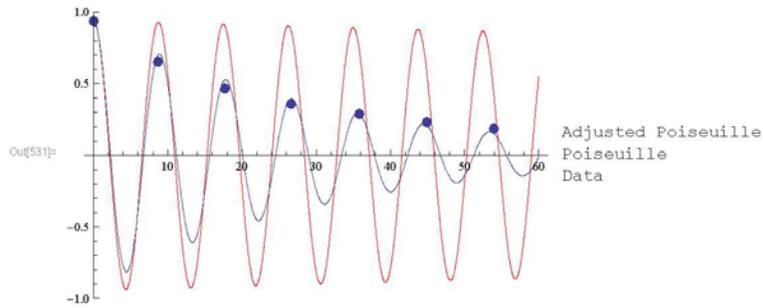
```

```
In[529]= ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469},
{26.647, 0.362}, {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}}]
```



In[530]=

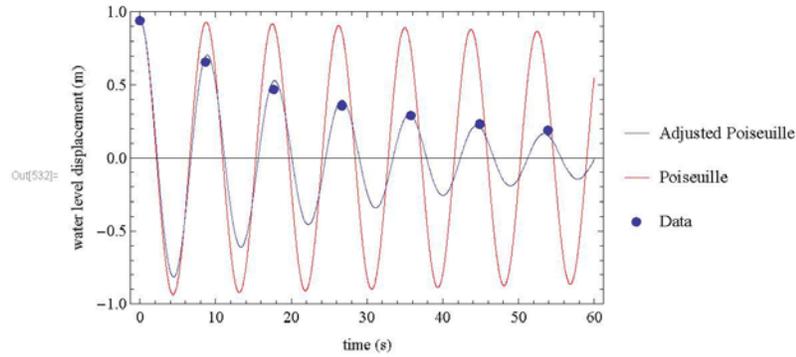
```
In[531]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
{35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[532]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.942}, {8.647, 0.657}, {17.674, 0.469}, {26.647, 0.362},
  {35.773, 0.291}, {44.897, 0.234}, {53.897, 0.191}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test G3907 Models

```
In[533]= Remove["Global`*"]
In[534]= L = 18.0863
Out[534]= 18.0863

In[535]= fD = 0.021
Out[535]= 0.021

In[536]= g = 9.81
          r = 0.0508
Out[536]= 9.81
Out[537]= 0.0508

In[538]= fD / (4 r)
Out[538]= 0.103346

In[539]= g / L
Out[539]= 0.542399

In[540]= 1 / (2 L)
Out[540]= 0.0276452

In[541]= solution = NDSolve[
          { D[w[t], t, t] = -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t] -
            1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] = 1, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[541]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[542]=
In[543]=
In[544]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[544]= 
```

```

In[545]= solution2 = NDSolve[
  { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] == 1, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[545]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

```

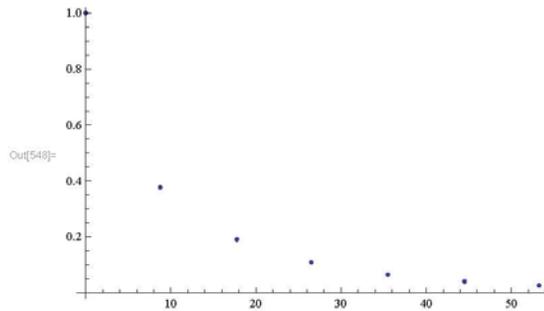
In[546]=

In[547]=

```

In[548]= ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}}]

```

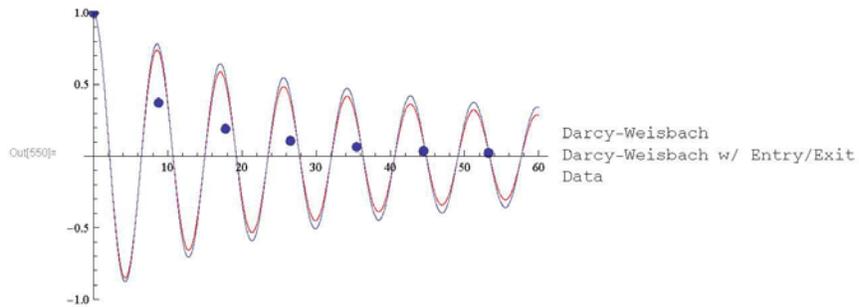


In[549]=

```

In[550]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
    {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]

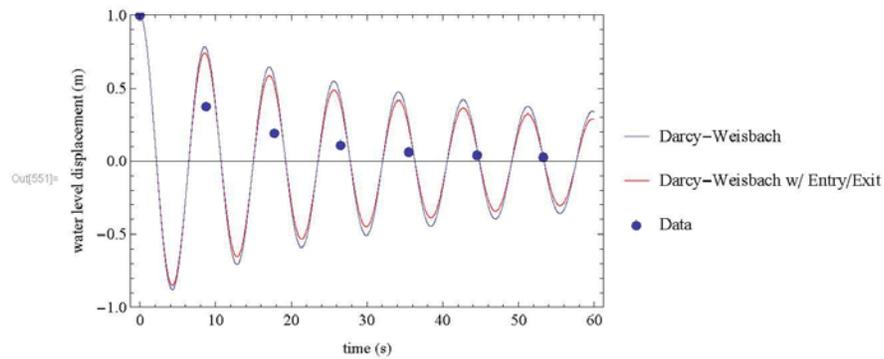
```



```

In[551]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[713]= Remove["Global`*"]

In[714]= L = 18.0863
Out[714]= 18.0863

In[715]= fD = 0.021
Out[715]= 0.021

In[716]= g = 9.81
Out[716]= 9.81

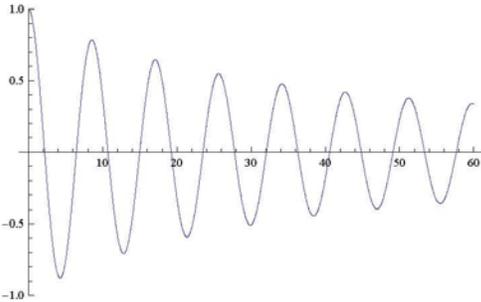
In[717]= r = 0.0508
Out[717]= 0.0508

In[718]= fD / (4 r)
          g / L

Out[718]= 0.103346
Out[719]= 0.542399

In[720]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
            w[0] == 1.00, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[720]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[721]=

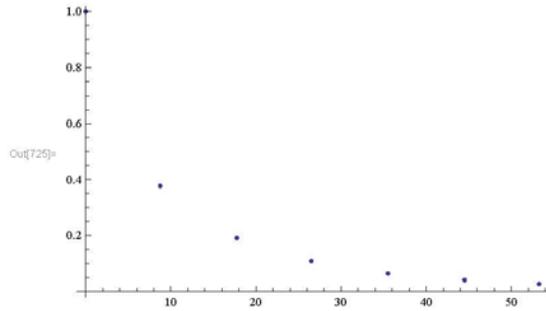
In[722]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[722]=


In[723]= solution2 =
          NDSolve[{ D[w[t], t, t] == -0.65 * (D[w[t], t]) * Abs[D[w[t], t]] - 0.56 * w[t],
                    w[0] == 1.00, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[723]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

```

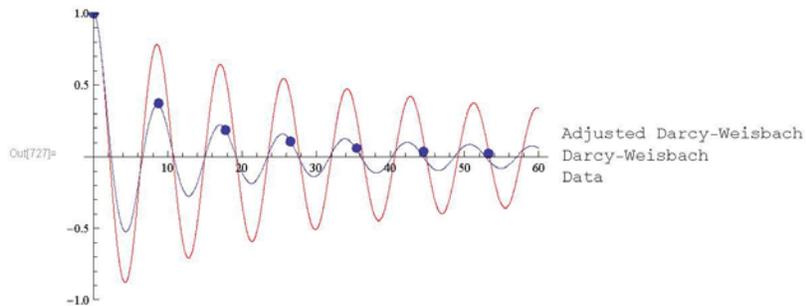
In[724]=

```
In[725]= ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}}]
```



In[726]=

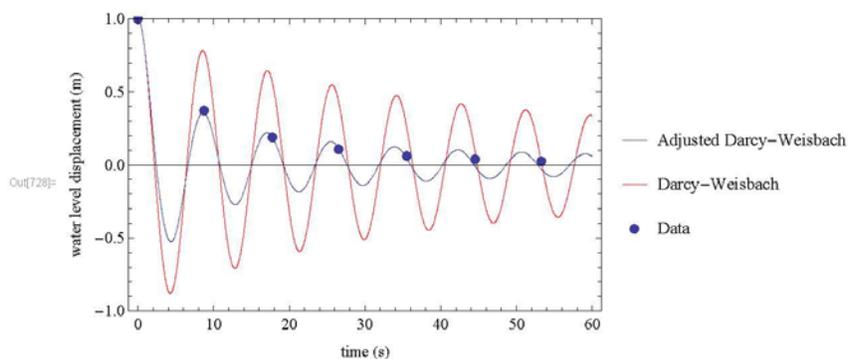
```
In[727]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"},
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"},
  ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[728]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[747]= Remove["Global`*"]

In[748]= L = 18.0863
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[748]= 18.0863

Out[749]=  $\frac{1}{1000000}$ 

Out[750]= 9.81

Out[751]= 0.0508

Out[752]= 0.542399

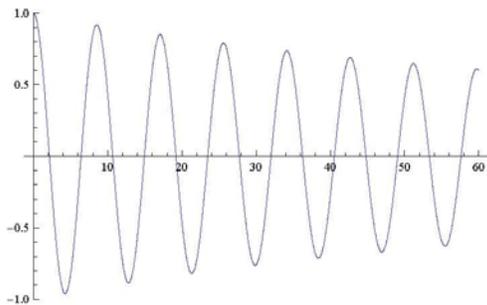
Out[753]= 0.00310001

In[754]= 1/(2L)
Out[754]= 0.0276452

In[755]=
In[756]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
          1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 1.0, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 5, PrecisionGoal -> 5]
Out[756]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[757]=
In[758]=
In[759]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[759]=

```

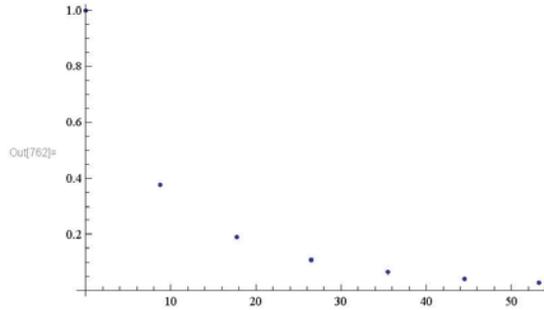


```
In[760]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] == 1.0, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[760]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

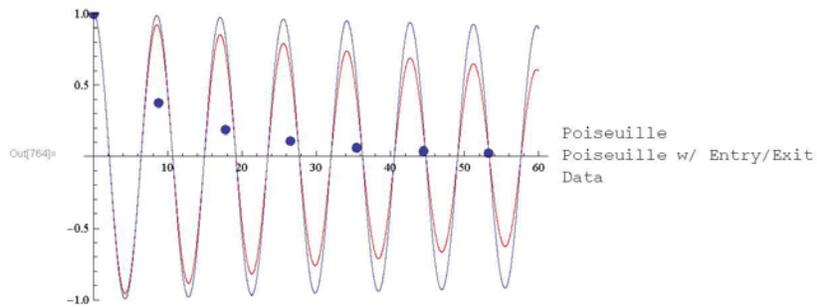
```
In[761]=
```

```
In[762]= ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
{26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}}]
```



```
In[763]=
```

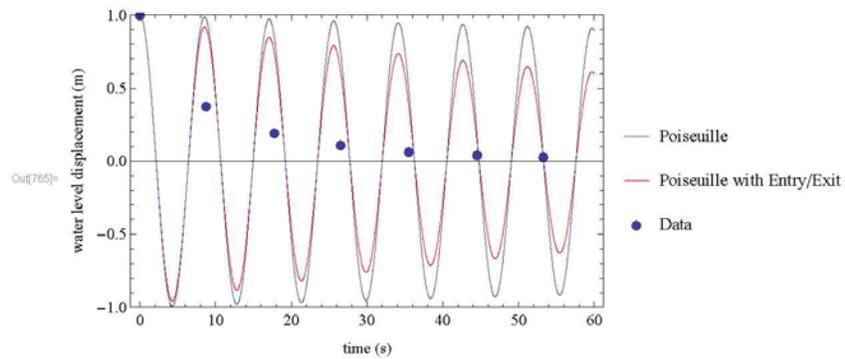
```
In[764]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille w/ Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60},
PlotRange -> {-1, 1}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
{26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[765]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[962]= Remove["Global`*"]

In[963]= L = 18.0863
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[963]= 18.0863

Out[964]=  $\frac{1}{1000000}$ 

Out[965]= 9.81

Out[966]= 0.0508

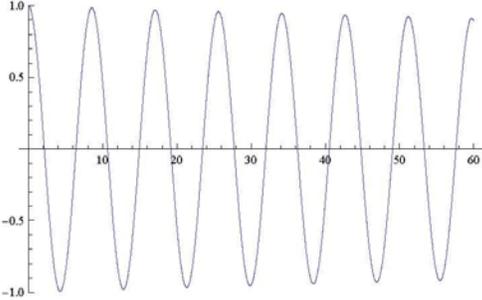
Out[967]= 0.542399

Out[968]= 0.00310001

In[969]=

In[970]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
                             w[0] == 1.00, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[970]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[971]=

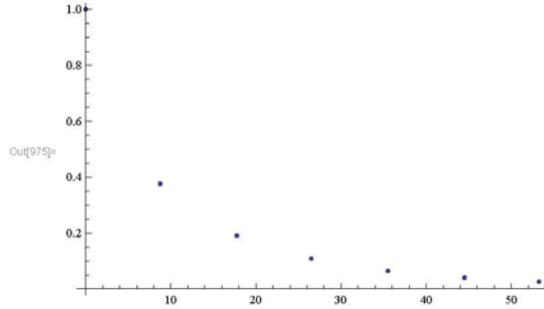
In[972]= Plot[Evaluate[w[t] /. First[solution]],
              {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[972]= 

In[973]= solution2 =
          NDSolve[{D[w[t], t, t] == -0.18*D[w[t], t] - 0.51*w[t], w[0] == 1.00, w'[0] == 0},
                  w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[973]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

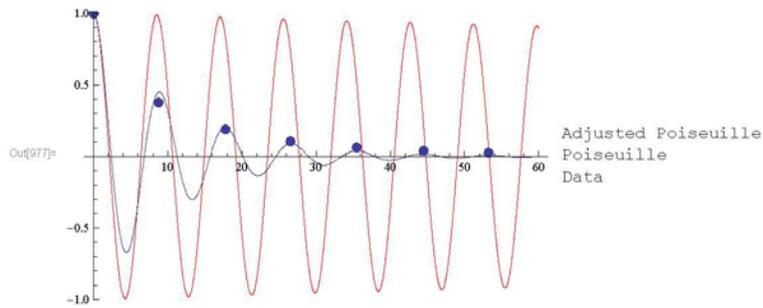
In[974]=

```

```
In[975]= ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
{26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}}]
```



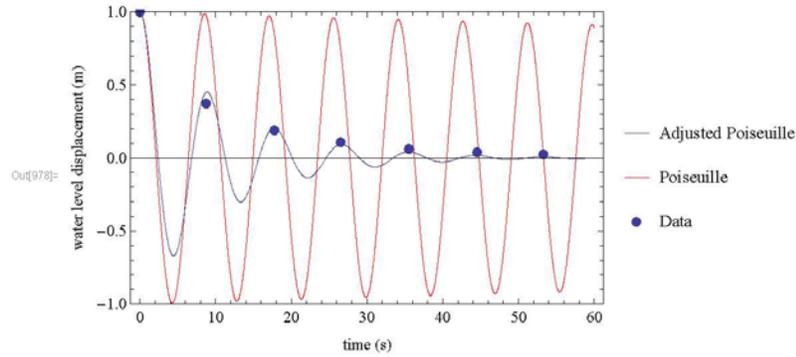
```
In[976]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
{26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[978]:= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 1.00}, {8.75, 0.377}, {17.75, 0.191},
  {26.5, 0.109}, {35.5, 0.065}, {44.5, 0.041}, {53.25, 0.027}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



Test G3911 Models

```
In[1000]= Remove["Global`*"]
In[1001]= L = 16.77924
Out[1001]= 16.7792

In[1002]= fD = 0.022
Out[1002]= 0.022

In[1003]= g = 9.81
           r = 0.0508
Out[1003]= 9.81
Out[1004]= 0.0508

In[1005]= fD / (4 r)
Out[1005]= 0.108268

In[1006]= g / L
Out[1006]= 0.584651

In[1007]= 1 / (2 L)
Out[1007]= 0.0297987

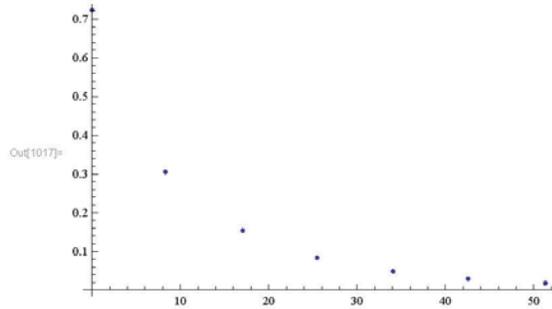
In[1008]= solution = NDSolve[
           { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t] -
             1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] == 0.723, w'[0] == 0,
           w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1008]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[1009]=
In[1010]=
In[1011]=
In[1012]= Plot[Evaluate[w[t] /. First[solution]],
           {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[1012]=
```

```
In[1013]= solution2 = NDSolve[
  { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] == 0.995, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1013]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

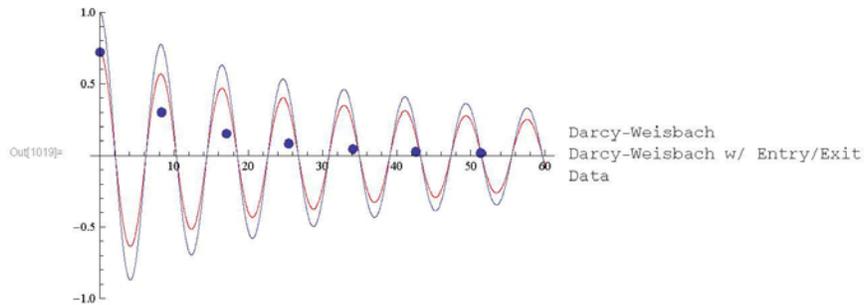
```
In[1014]=
In[1015]=
In[1016]=
```

```
In[1017]= ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154},
  {25.48, 0.084}, {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}}]
```



```
In[1018]=
```

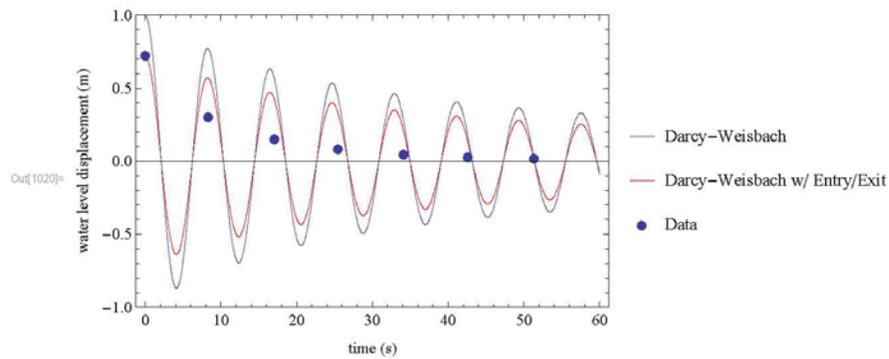
```
In[1019]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
  {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1020]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
  {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1025]= Remove["Global`*"]

In[1026]= L = 16.77924
Out[1026]= 16.7792

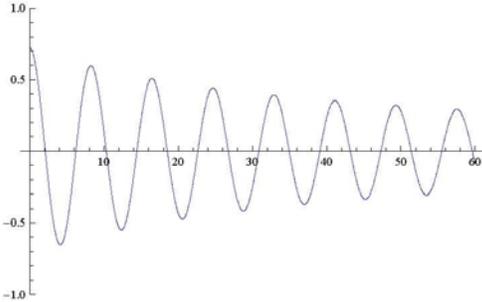
In[1027]= fD = 0.022
Out[1027]= 0.022

In[1028]= g = 9.81
Out[1028]= 9.81

In[1029]= r = 0.0508
Out[1029]= 0.0508

In[1030]= g / L
          fD / (4 r)
Out[1030]= 0.584651
Out[1031]= 0.108268

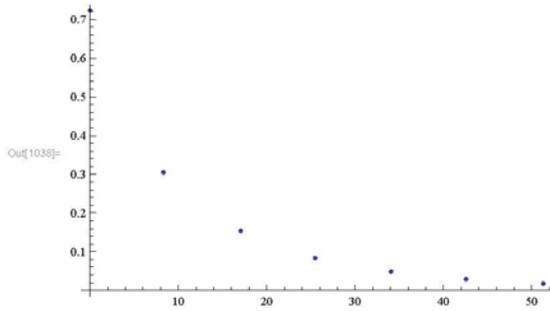
In[1032]=
In[1033]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
            w[0] == 0.723, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1033]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[1034]=
In[1035]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
Out[1035]=

In[1036]= solution2 =
          NDSolve[{ D[w[t], t, t] == -0.72 * (D[w[t], t]) * Abs[D[w[t], t]] - 0.570 * w[t],
            w[0] == 0.723, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1036]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

```

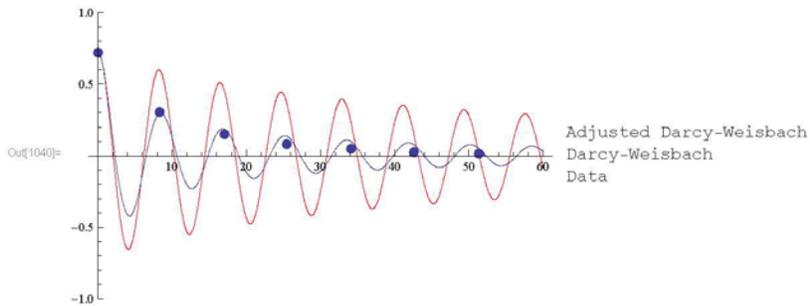
In[1037]=

```
In[1038]= ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154},
{25.48, 0.084}, {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}}]
```



In[1039]=

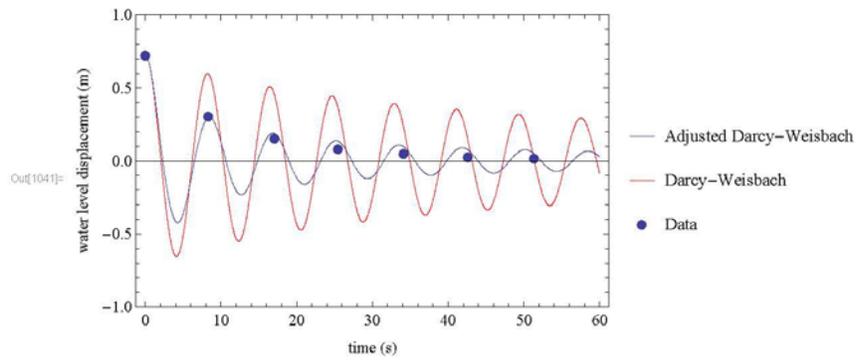
```
In[1040]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
{34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1041]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
  {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```



```

In[1042]= Remove["Global`*"]

In[1043]= L = 16.77924
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)

Out[1043]= 16.7792

Out[1044]=  $\frac{1}{1000000}$ 

Out[1045]= 9.81

Out[1046]= 0.0508

Out[1047]= 0.584651

Out[1048]= 0.00310001

In[1049]= 1/(2L)

Out[1049]= 0.0297987

In[1050]=

In[1051]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
          1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.723, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 5, PrecisionGoal -> 5]

Out[1051]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

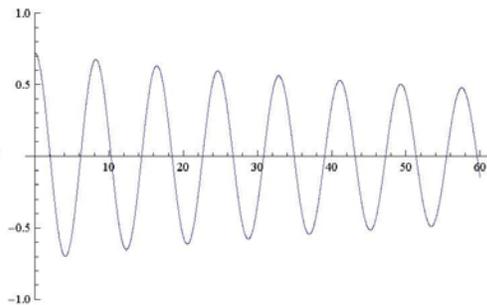
In[1052]=

In[1053]=

In[1054]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]

Out[1054]=

```

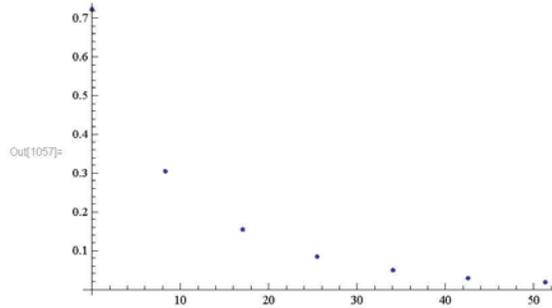


```
In[1055]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] = 0.723, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1055]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

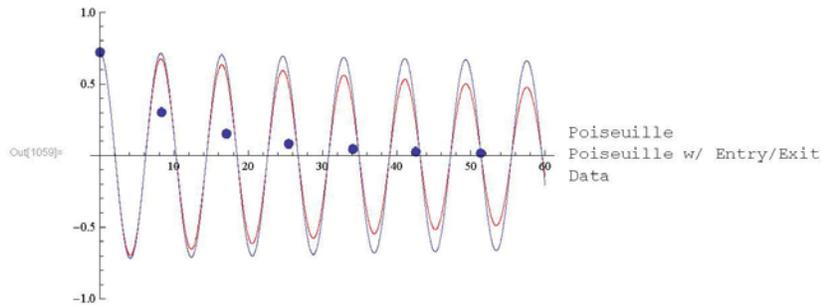
```
In[1056]=
```

```
In[1057]= ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154},
{25.48, 0.084}, {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}}]
```



```
In[1058]=
```

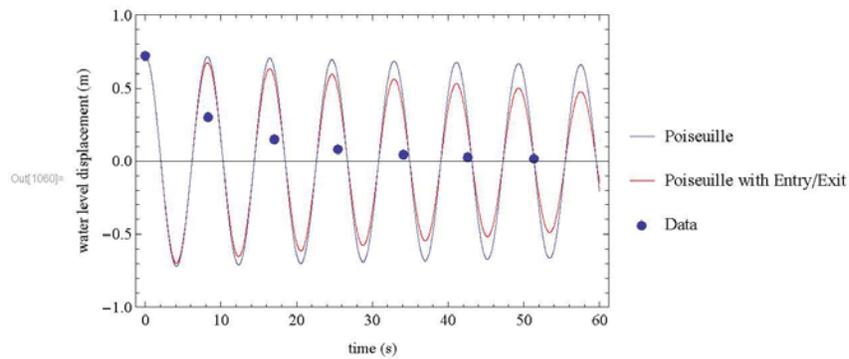
```
In[1059]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille w/ Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60},
PlotRange -> {-1, 1}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
{34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1060]= Show[Plot[Evaluate[w[t] /. First[solution]],
  {t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
  {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```
In[1076]= Remove["Global`*"]
```

```
In[1077]= L = 16.77924  
v = 10^-6  
g = 9.81  
r = 0.0508  
g/L  
(8*v)/(r^2)
```

```
Out[1077]= 16.7792
```

```
Out[1078]=  $\frac{1}{1000000}$ 
```

```
Out[1079]= 9.81
```

```
Out[1080]= 0.0508
```

```
Out[1081]= 0.584651
```

```
Out[1082]= 0.00310001
```

```
In[1083]=
```

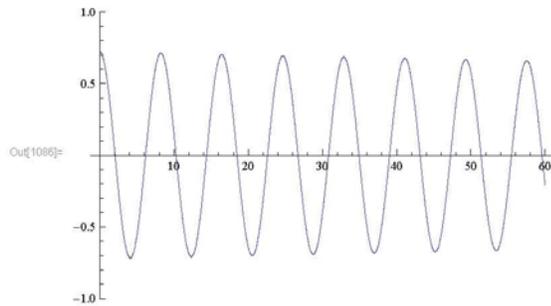
```
In[1084]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],  
w[0] == 0.723, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1084]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

```
In[1085]= {{w -> InterpolatingFunction[{{0., 60.}}, "<>"]}}
```

```
Out[1085]= {{w -> InterpolatingFunction[]}}
```

```
In[1086]= Plot[Evaluate[w[t] /. First[solution]],  
{t, 0, 60}, PlotRange -> {-1, 1}, PlotPoints -> 30]
```



```
In[1087]= solution2 =  
NDSolve[{D[w[t], t, t] == -0.18*D[w[t], t] - 0.570*w[t], w[0] == 0.723, w'[0] == 0},  
w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

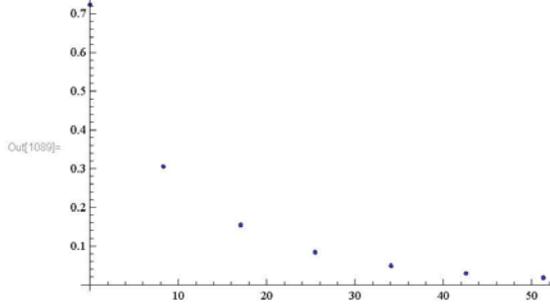
```
Out[1087]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

```
In[1088]= {{w → InterpolatingFunction[{{0., 60.}}, "<>"]}}
```



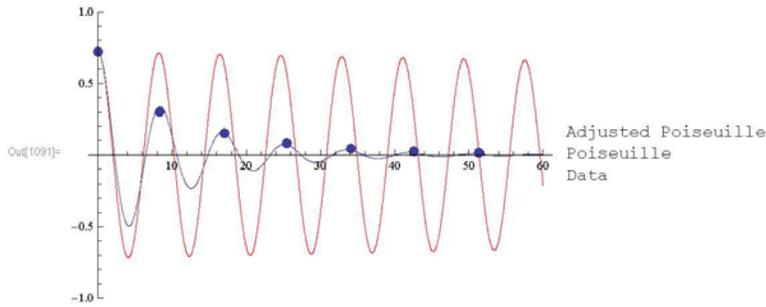
```
Out[1088]= {{w → InterpolatingFunction[]}}
```

```
In[1089]= ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154},
{25.48, 0.084}, {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}}]
```



```
In[1090]=
```

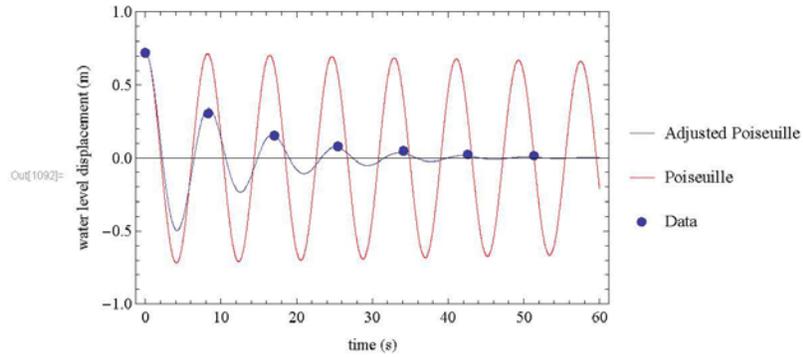
```
In[1091]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {-1, 1},
PlotPoints → 30, PlotStyle → {Red}, PlotLegends → "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {-1, 1},
PlotPoints → 30, PlotLegends → "Adjusted Poiseuille"],
ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
{34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
PlotLegends → "Data", PlotMarkers → {Automatic, Medium}]]
```



```

In[1092]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-1, 1},
  PlotPoints -> 30, PlotLegends -> {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.723}, {8.311, 0.305}, {17.06, 0.154}, {25.48, 0.084},
  {34.074, 0.049}, {42.56, 0.029}, {51.311, 0.018}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
  Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
  LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



Test G3915 Models

```
In[1204]= Remove["Global`*"]
In[1205]= L = 1.2954
Out[1205]= 1.2954

In[1206]= fD = 0.026
Out[1206]= 0.026

In[1207]= g = 9.81
          r = 0.0508
Out[1207]= 9.81
Out[1208]= 0.0508

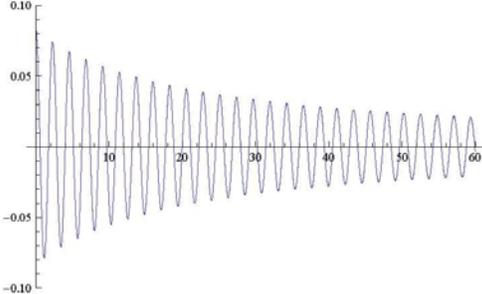
In[1209]= fD / (4 r)
Out[1209]= 0.127953

In[1210]= g / L
Out[1210]= 7.57295

In[1211]= 1 / (2 L)
Out[1211]= 0.385981

In[1212]= solution = NDSolve[
          { D[w[t], t, t] = - (fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t] -
            1 / (2 L) * (D[w[t], t]) * Abs[D[w[t], t]]}, w[0] = 0.083, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1212]= {w -> InterpolatingFunction[{{0., 60.}}, <>]}

In[1213]=
In[1214]=
In[1215]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {- .1, .1}, PlotPoints -> 30]
Out[1215]=
```

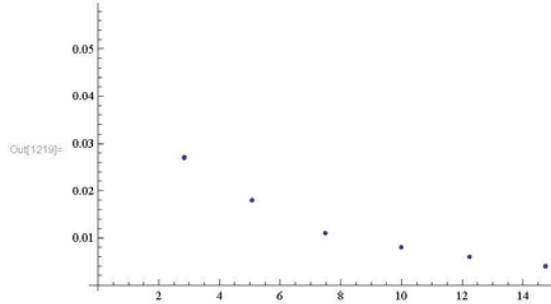


```
In[1216]= solution2 = NDSolve[
  { D[w[t], t, t] = -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / (L)) * w[t],
    w[0] = 0.083, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1216]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

In[1217]=

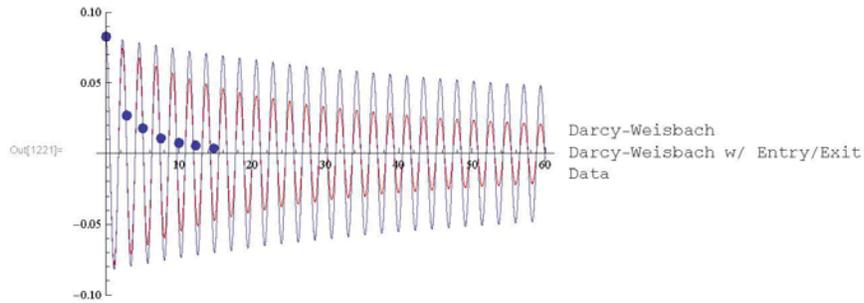
In[1218]=

```
In[1219]= ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018},
  {7.487, 0.011}, {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}}]
```



In[1220]=

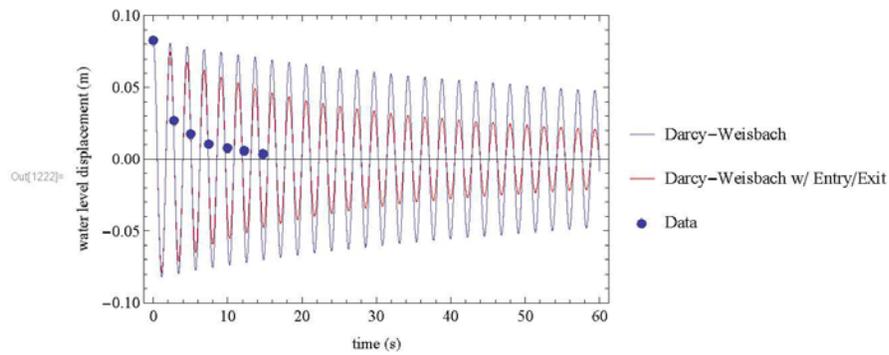
```
In[1221]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {- .1, .1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach w/ Entry/Exit"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {- .1, .1},
  PlotPoints -> 30, PlotLegends -> "Darcy-Weisbach"],
  ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
  {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1222]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60},
  PlotRange -> {- .1, .1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Darcy-Weisbach w/ Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {- .1, .1},
  PlotPoints -> 30, PlotLegends -> {Style["Darcy-Weisbach"]}],
ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
  {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1227]= Remove["Global`*"]

In[1228]= L = 1.2954
Out[1228]= 1.2954

In[1229]= fD = 0.026
          g = 9.81
Out[1229]= 0.026
Out[1230]= 9.81

In[1231]= r = 0.0508
Out[1231]= 0.0508

In[1232]= fD / (4 r)
          g / L
Out[1232]= 0.127953
Out[1233]= 7.57295

In[1234]=

In[1235]= solution = NDSolve[
          { D[w[t], t, t] == -(fD / (4 r)) * (D[w[t], t]) * Abs[D[w[t], t]] - (g / L) * w[t],
            w[0] == 0.083, w'[0] == 0 }, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1235]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[1236]= {{w -> InterpolatingFunction[{{0., 60.}}, "<>"]}}
Out[1236]= {{w -> InterpolatingFunction[]}}

In[1237]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {-0.1, 0.1}, PlotPoints -> 30]
Out[1237]=

```

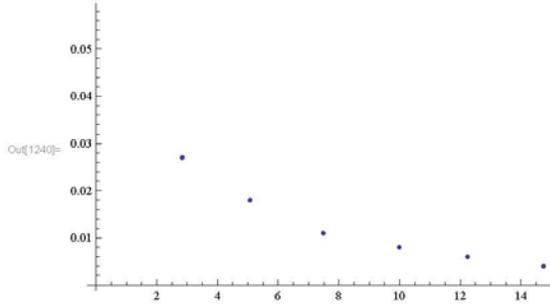
```

In[1238]= solution2 =
  NDSolve[{D[w[t], t, t] == -6.5 * (D[w[t], t]) * Abs[D[w[t], t]] - 3.461 * w[t],
    w[0] == 0.083, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1238]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[1239]= {{w -> InterpolatingFunction[{{0., 60.}}, "<>"]}}
Out[1239]= {{w -> InterpolatingFunction[]}}

In[1240]= ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018},
  {7.487, 0.011}, {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}}]

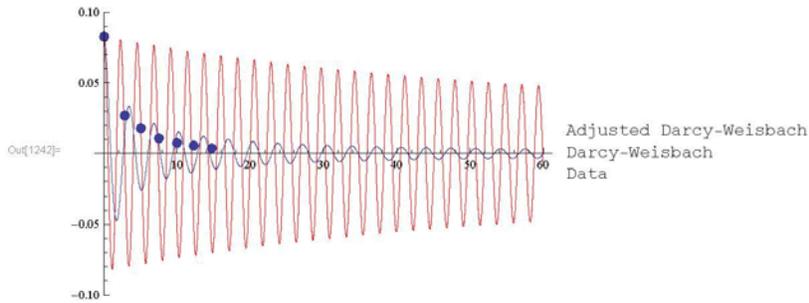
```



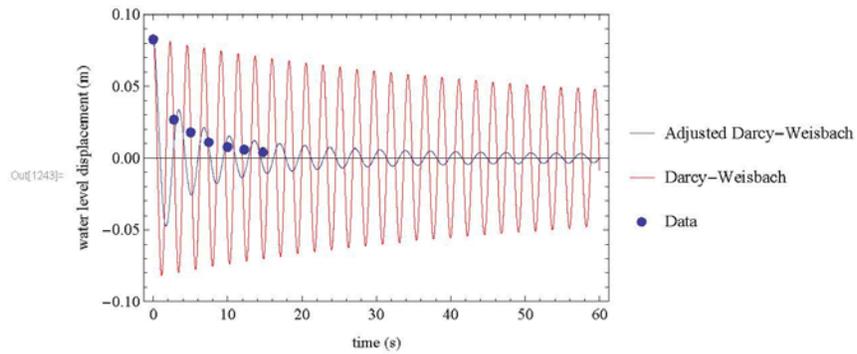
```

In[1241]=
In[1242]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-0.1, 0.1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Darcy-Weisbach"],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-0.1, 0.1},
  PlotPoints -> 30, PlotLegends -> "Adjusted Darcy-Weisbach"],
  ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
    {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
  PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]

```



```
In[1243]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {-0.1, 0.1},
  PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> {Style["Darcy-Weisbach"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {-0.1, 0.1},
  PlotPoints -> 30, PlotLegends -> {Style["Adjusted Darcy-Weisbach"]}],
  ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
  {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
  Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
  LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]
```



```

In[1150]= Remove["Global`*"]

In[1151]= L = 1.2954
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[1151]= 1.2954

Out[1152]=  $\frac{1}{1000000}$ 

Out[1153]= 9.81

Out[1154]= 0.0508

Out[1155]= 7.57295

Out[1156]= 0.00310001

In[1157]= 1/(2L)
Out[1157]= 0.385981

In[1158]=

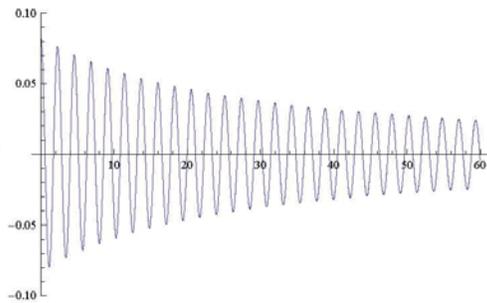
In[1159]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t] -
          1/(2L)*D[w[t], t]*Abs[D[w[t], t]], w[0] == 0.083, w'[0] == 0},
          w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1159]= {w -> InterpolatingFunction[{{0., 60.}}, <>]}

In[1160]=

In[1161]=

In[1162]= Plot[Evaluate[w[t] /. First[solution]],
          {t, 0, 60}, PlotRange -> {- .1, .1}, PlotPoints -> 30]
Out[1162]=

```

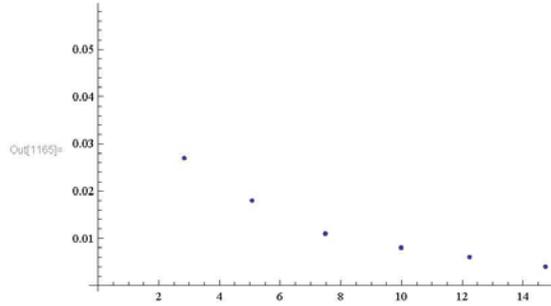


```
In[1163]= solution2 = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
w[0] == 0.083, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
```

```
Out[1163]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}
```

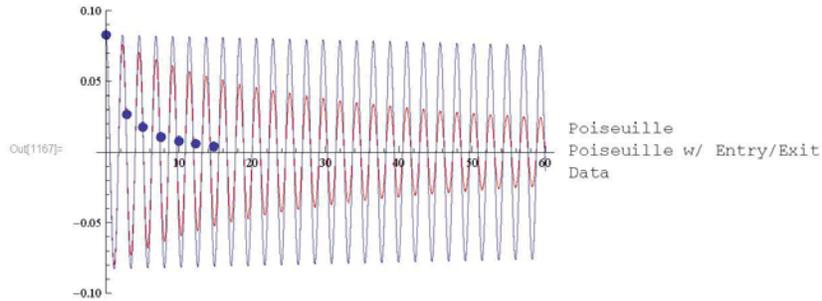
```
In[1164]=
```

```
In[1165]= ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018},
{7.487, 0.011}, {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}}]
```



```
In[1166]=
```

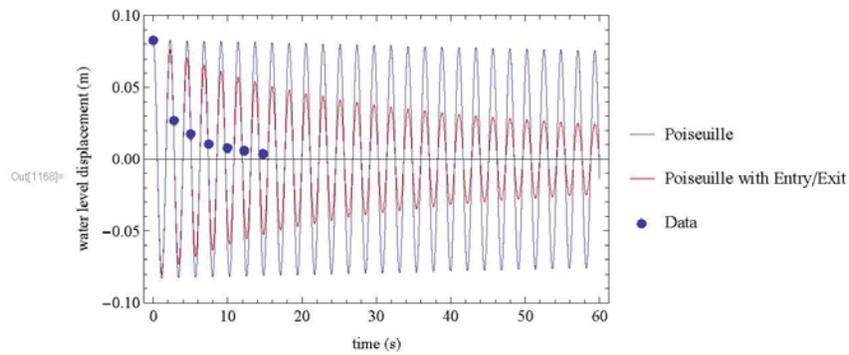
```
In[1167]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {- .1, .1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille w/ Entry/Exit"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60},
PlotRange -> {- .1, .1}, PlotPoints -> 30, PlotLegends -> "Poiseuille"],
ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
{9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1168]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60},
  PlotRange -> {- .1, .1}, PlotPoints -> 30, PlotStyle -> {Red},
  PlotLegends -> {Style["Poiseuille with Entry/Exit"]}],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {- .1, .1},
  PlotPoints -> 30, PlotLegends -> {Style["Poiseuille"]}],
ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
  {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
  PlotLegends -> {Style["Data"]}, PlotMarkers -> {Automatic, Medium}],
Frame -> True, FrameLabel -> {"time (s)", "water level displacement (m)"},
LabelStyle -> {FontSize -> 12}, ImageSize -> {400, 400}]

```



```

In[1169]= Remove["Global`*"]

In[1170]= L = 1.2954
          v = 10^-6
          g = 9.81
          r = 0.0508
          g/L
          (8*v)/(r^2)
Out[1170]= 1.2954

Out[1171]=  $\frac{1}{1000000}$ 

Out[1172]= 9.81

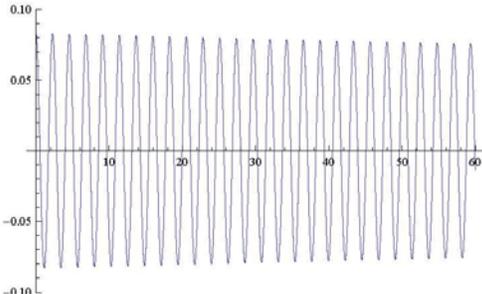
Out[1173]= 0.0508

Out[1174]= 7.57295

Out[1175]= 0.00310001

In[1176]= solution = NDSolve[{D[w[t], t, t] == -((8*v)/(r^2))*D[w[t], t] - (g/L)*w[t],
                              w[0] == 0.083, w'[0] == 0}, w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1176]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[1177]=

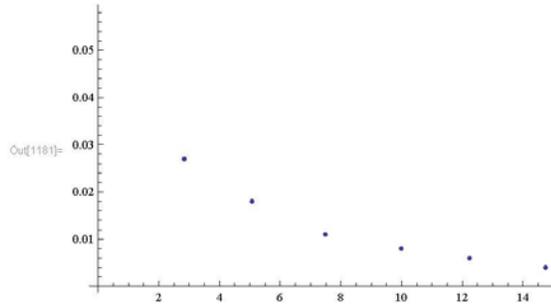
In[1178]= Plot[Evaluate[w[t] /. First[solution]],
              {t, 0, 60}, PlotRange -> {-0.1, 0.1}, PlotPoints -> 30]
Out[1178]= 

In[1179]= solution2 =
          NDSolve[{D[w[t], t, t] == -0.55*D[w[t], t] - 3.461*w[t], w[0] == 0.083, w'[0] == 0},
                w, {t, 0, 60}, AccuracyGoal -> 10, PrecisionGoal -> 10]
Out[1179]= {{w -> InterpolatingFunction[{{0., 60.}}, <>]}}

In[1180]=

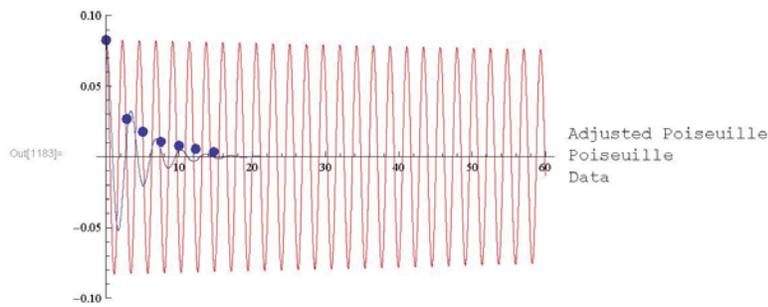
```

```
In[1181]= ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018},
{7.487, 0.011}, {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}}]
```



In[1182]=

```
In[1183]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange -> {- .1, .1},
PlotPoints -> 30, PlotStyle -> {Red}, PlotLegends -> "Poiseuille"],
Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange -> {- .1, .1},
PlotPoints -> 30, PlotLegends -> "Adjusted Poiseuille"],
ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
{9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
PlotLegends -> "Data", PlotMarkers -> {Automatic, Medium}]]
```



```

In[1184]= Show[Plot[Evaluate[w[t] /. First[solution]], {t, 0, 60}, PlotRange → {- .1, .1},
  PlotPoints → 30, PlotStyle → {Red}, PlotLegends → {Style["Poiseuille"]}],
  Plot[Evaluate[w[t] /. First[solution2]], {t, 0, 60}, PlotRange → {- .1, .1},
  PlotPoints → 30, PlotLegends → {Style["Adjusted Poiseuille"]}],
  ListPlot[{{0, 0.083}, {2.841, 0.027}, {5.072, 0.018}, {7.487, 0.011},
  {9.987, 0.008}, {12.237, 0.006}, {14.737, 0.004}},
  PlotLegends → {Style["Data"]}, PlotMarkers → {Automatic, Medium}],
  Frame → True, FrameLabel → {"time (s)", "water level displacement (m)"},
  LabelStyle → {FontSize → 12}, ImageSize → {400, 400}]

```

