

Incorporating hydraulic design software into an introductory fluid mechanics course through virtualized, internet-delivered software applications.

Dr. Isaac W Wait, Marshall University

Dr. Isaac Wait is an associate professor of Engineering in the Division of Engineering at Marshall University in Huntington, West Virginia. He conducts research and teaches courses in water resources and environmental engineering, and is a registered professional engineer in the States of Ohio and West Virginia.

Mr. Mike McSween, Bentley Systems Mr. Bradley E Workman, Bentley Systems Inc,

Incorporating hydraulic design software into an introductory fluid mechanics course through virtualized, internet-delivered software applications.

Abstract

Students utilized a 'virtualized' (i.e., server-hosted, locally-controlled) hydraulic design package (i.e., Bentley WaterGEMS) to conduct an in-class demonstration and out-of-class assignment to support their learning of three fluid mechanics course topics: the hydrostatic equation, energy loss in pipes, and pipe network optimization. Introducing a sophisticated hydraulic design package to students early in their learning, such as during a fluid mechanics course where content typically leans towards 'core knowledge' topics rather than application and/or design, can support teaching of fundamental concepts by enabling students to rapidly solve many different problems and develop a 'feel' for how equations behave. Likewise, students develop a sense of accomplishment and vision for what will later come as they learn how to operate industry-standard design software.

Avoiding most of the inconvenience and confusion associated with having to locally-install software, a virtualized software package functions and appears the same as if the application were instead operating in a stand-alone mode on a lab or personal computer. Additionally, since virtualization clients exist for multiple operating systems (e.g., Mac, Linux), access is enabled for students using otherwise unsupported operating systems. Thus, through virtualization it becomes easy for an instructor to incorporate a short in-class activity, or assign a few problems to be solved on the computer, without the lengthy preparations that would ordinarily be required.

In this paper one approach to software virtualization will be described in the context of a software-utilizing learning activity that was conducted by junior-level undergraduate engineering students enrolled in a fluid mechanics course. Feedback from students indicates an enthusiastic response to both early incorporation of a software package into study of fluid mechanics and virtualization as a means of delivering software availability.

Introduction

Engineers increasingly rely on sophisticated software packages for the design and analysis of hydraulic networks, including drinking water distribution systems, sanitary sewers, and stormwater conveyance channels. The design packages utilized offer a wide array of options and features, such that gaining mastery in how to use the full capability of a program can be a long and potentially intimidating process. Much of the functionality contained in such programs is

above the level of undergraduate engineering students, particularly those enrolled in a first introductory course in fluid mechanics. Likewise, many of the key fundamental concepts that must be taught to students are not readily taught through a program of software education. However, there are certain principles (e.g., hydrostatics, the Reynolds transport theorem, energy loss due to pipe friction, etc.) that can be illustrated simply and effectively using complex hydraulic design software.

After learning governing relationships and solving problems in the traditional 'pencil and paper' way, students can benefit from solving similar and potentially more sophisticated problems using industry-standard hydraulic design software. Utilizing software allows students to experiment with and observe the effects of varying material types, fluid properties, and system element sizes, for example. In this iterative way, students can begin to develop a 'feel' for how a system behaves, and gain an appreciation for the realm of reasonable answers much more quickly than might be possible if they were only able to perform the calculations manually. Besides quick and easy iterative calculations, another benefit to early incorporation of software tools into a content-rich course is that students become comfortable and familiar with software that they may someday utilize in the workplace. This leads to a student impression that they are actively gaining skills that will be relevant and valuable once they graduate and begin a career.

Although many instructors are already aware of the potential benefits of exposing students to design software, some hesitate to actually implement software instruction in their courses because of access barriers, inconvenience, and technical challenges. In some cases, dedicated desktop computers are not available in the classroom, or the software to be taught is not installed on university-owned machines. Distributing large software installation files to students (some of which with incompatible hardware) can also be a challenge, as can be the process of walking users through cumbersome and confusing license activation procedures. Likewise, the prospect of providing technical support to students – each who has installed software on a different computer, and may be experience a different technical problem – can be enough to dissuade an instructor from bothering with the potential morass of asking students to solve problems using software. Fortunately, the rise of cloud computing and the ease of access to server-hosted, virtualized installations of software can eliminate many of the implementation barriers that might otherwise cause instructors to feel that the cost of teaching software outweighs the learning benefits.

Within academic instruction, virtualization has been utilized extensively in computer science settings for a variety of purposes, including unprotected network security testing for which a 'real' hardware environment would represent a security risk¹, and to simulate and investigate the performance implications of different router network configurations without actually needing network hardware². Likewise, virtualization has been utilized for making multiple operating systems available to students within a single physical machine³.

A commercial virtualization product has been used to make a hydraulic design package (i.e., Bentley WaterGEMS) available to students for preliminary testing of this approach. Using the small, lightweight, simple-to-use virtualization client on university-owned and student-owned laptops, students can connect to software applications hosted on a central server. The virtualized software package operates and appears the same as it would if the Windows application was instead installed locally, without most of the inconveniences previously identified.

Traditional Challenges of Software Implementation

Although many instructors recognize the value of incorporating an element of software instruction and familiarity into the courses they teach, there are a variety of technical and logistical hurdles that typically stand in the way. These hurdles introduce inconvenience to the instructor and to students, and serve as powerful dis-incentives to attempting to incorporate software training into courses. Among the challenges are:

<u>Software installation must be coordinated with IT managers</u> – In some cases, instructors do not have administrative rights to themselves install programs onto lab or classroom computers. Thus it is required to coordinate installation with IT managers, who may not be able to install new software onto machines mid-semester. In such instances, the requirement to request the addition of software far in advance of when it will be used in class may constrain the opportunity for new learning activities to be run.

<u>Computer lab crowding</u> – Since computer labs are generally already used for courses that do traditionally contain a software component (e.g., CAD, senior-level design courses, etc.), it may be that there is limited extra capacity in computer labs for new assignments and utilizations to be introduced.

<u>Course is not taught in computer lab</u> – For many core knowledge type courses, where fundamental concepts are of primary concern and software is not typically utilized, the course will often be taught in a classroom that does not include access to computers. Thus it is difficult to provide computer access to students for just one or two class periods when a brief software demonstration is desired.

<u>Heterogeneity of student-owned computers</u> – Among computers that run MS Windows, students generally own a wide variety of different computer types, manufactured by many different manufacturers and running a variety of different operating systems (including 32-bit vs. 64-bit). These differences inevitably lead to errors that must be resolved when students install software onto their local machines. Being thrust into the role of 'software installation technical support' can be a powerful dis-incentive against incorporating a software component into a course for an instructor with limited time or ability to troubleshoot such issues.

<u>Licensing complexities</u> – In some cases where a university has paid for a license to a particular program, that license may not allow students to install versions of the program onto their personally owned computers. In such cases, and where other factors limit availability to software,

<u>Cross-platform non-operability</u> – With the increasing popularity of non-Windows operating systems (e.g., Linux, Mac), more students own computers for which the operating system is not compatible with the software to be taught.

These challenges can be mitigated, and in some cases eliminated entirely through virtualization of software applications. For purposes of comparison, whereas introducing a software application to students previously required the instructor to commit approximately 10 hours per semester (much of which was helping students debug faulty software installations and sort through licensing issues), by implementing software virtualization, the time requirement was reduced to approximately 2 hours, including prep time before class demonstrations.

Benefits of Software Virtualization and Simplified Program Access for Students

"Virtualization" of software, such that it is installed and operated on a remote server to which students connect when they wish to run the program in question, can reduce or eliminate each of the difficulties identified above. During virtualization, the client computer merely acts as a 'window' to the remotely-installed and run software program, such that a user is able to see the software-generated screen, interact with the program, and issue commands to it, but since the program is actually hosted remotely, this means that it is the remote server that must have the hardware and software required to run the program. This virtualization technique simplifies academic implementation of software in a number of ways.

By making it possible for students to operate software without it actually needing to be installed on a computer, instructors no longer need to coordinate installation of software with the IT managers who control classroom and computer lab machines. This introduces flexibility to decide on which programs to utilize during the semester itself, where previously this may not have been possible. Likewise, virtualization can overcome the problems associated with not enough computers in a classroom or computer lab, since most university students already own a PC and through virtualization can easily operate the needed program without having to install it. Since software is run without being installed, the challenges and irritations associated with installation, licensing, and cross-platform non-operability are avoided. Instead, students must only configure the small client applet that allows them to connect to the remote server, and are then able to bring up the program as needed, even if their machine wouldn't otherwise have adequate RAM, a compatible operating system, or any of the other requirements that sometimes stand in the way of their installing software programs locally.

Virtualization – the Student Experience

While running software virtually is significantly less cumbersome than installing it locally, there are still some steps that must be followed for use. In the fluid mechanics course described herein, the following was the workflow that students had to follow in order to utilize WaterGEMS.

<u>Download and install virtualization client</u>. Students clicked on a link that was provided by the instructor. This link is what is used to trigger the virtualization of the WaterGEMS program, and if the requisite client software (i.e., Citrix Receiver) is not installed on the student machine, then it provides an intuitive and easy way to install the client (see Figure 1). Since administrator rights were not required to install the client, this meant that students were able to configure access to the virtualized WaterGEMS even on computers where they have limited accounts.



Figure 1 – Obtaining the virtualization client.

<u>Trigger software virtualization</u>. Once the client software was installed and running in the background as evidenced by an icon displayed in the computer's system tray, students once again clicked on the instructor-provided access link to trigger virtualization of WaterGEMS (see Figures 2 and 3). Upon allowing access in a security warning dialog box (Figure 4), the virtualized version of the program software opens, along with any pre-configured program files that the instructor wishes to have loaded into the program upon initiation of the virtualization

link. Likewise supporting programs can be automatically called to load, and in this case Adobe Reader was used to display an electronic copy of the assignment on screen.

Citrix XenApp ×			
$\label{eq:complexity} \leftarrow \ \Rightarrow \ \mathbf{C} \ \ \fbox{lo-demo.bentley.com/Citrix/XenAppPub/site/directLaunch.aspx}$			☆ =
Logged on as: Guest	<mark>∭</mark> Messages	🔅 Settings	Log Off
LO3 - Network optimization If your application does not appear within a few seconds, <u>click to connect</u> .			
			E

Figure 2 – Triggering virtualization.

LO3 - Network optimization	
Starting	
✓ More information	Cancel

Figure 3 – Initiation of virtualization



Figure 4 – Security warning

As shown in Figure 5, the student's view of the virtualized WaterGEMS program window is identical to the program windows available when the software is installed locally. All of the same program functionality is available, including all program menus and buttons, the option to save program files onto the students' local machine, and the ability to print to locally-connected printers. Thus the student experience when using virtualized WaterGEMS is nearly identical to using a locally-installed version of the program, and in fact users are able to switch back and forth between the two.

For purposes of this pilot test, the server hosting the WaterGEMS software was a cloud-based server managed by Bentley. However, university IT departments often have existing experience with virtualization servers, and could potentially be called upon to host applications that are to be used for instructional purposes.

The technical specifications of the remote server hosting the software (Intel Xeon @ 2.45 GHz with 7.5 GB main memory running 64-bit Windows Server 2008 R2 Datacenter), coupled with the requirements of the program(s) being virtualized and the virtualization software used, can limit the number of simultaneous users that can be accommodated via remote connection from a single server. In the case of the single server that was utilized to host virtualization for this course (which was not designed for many simultaneous users, but rather to offer student's with at-home access), approximately 15-20 users could be connected at any single time. Were additional simultaneous connections attempted above this limit, users were not able to connect to the server and/or performance was slow. Depending on the number of servers used, CPU and RAM configuration of each, the virtualization software utilized, and the specification of program or programs being virtualized, the number of users that can be accommodated may be significantly more than or fewer than those mentioned above.

The initial computational load associated with a user making connection to the server and beginning the virtualized programs (i.e., WaterGEMS and Adobe Reader for each user) was significantly higher than the server resources utilized once the start-up process was finished and the program was in operation. Thus, having all students in a classroom connect to the server at a single time, such as when an instructor begins an in-class demonstration, could lead to reduced performance and/or system difficulties if the virtualization servers are not designed to accommodate simultaneous use.

Network bandwidth is another factor that may limit feasibility of software virtualization. The virtualization software that was utilized is known to be extremely efficient with respect to connection bandwidth requirements; tests have shown that as little as 200 Kbps per user bandwidth is required, and network latency as high as the 100's of milliseconds is acceptable to maintain software interactivity. Only one student reported concerns that may have been related to insufficient bandwidth and its potential for performance degradation.



Figure 5 – Program (and supporting resources) loaded and ready to use.

Teaching and Learning Fluid Mechanics-Support through Software

WaterGEMS was used in three different ways during the semester to illustrate the following fluid mechanics and engineering principles: (1) to illustrate the hydrostatic equation, and the effect of changing various variables in the hydrostatic equation; (2) to support concepts of fluid properties

and introduce the principles of pipe friction and energy loss in a conduit to students; and (3) as a tool for students to experience the need for iterative analysis and optimization in design of a simple water distribution network.

The fluid mechanics course into which these software activities were integrated is a junior-level undergraduate course, following a relatively standard curricular path (i.e., fluid properties, hydrostatics, continuity, momentum, energy equation, dimensional analysis). For the activities described below, the 'Hydrostatics' activity is integrated into the hydrostatics section of the course, the 'Fluid Properties, Pipe Friction, and Energy Loss' activity is given during the momentum section of the course (as a refresher of previous topics and as a preview of energy equation topics that will come), and the 'Network Optimization' activity is conducted during the energy equation section to illustrate applications of the energy principle and future design activities that will be conducted during a subsequent hydraulic engineering course.

<u>Hydrostatics.</u> In the first learning activity, "Hydrostatics" (available in the appendix of this paper), students were asked to consider the case of a single pipe connected to an elevated reservoir, and the static water pressure that would exist if a drinking fountain were connected. During a basic in-class demonstration, students were shown how to draw a single line in WaterGEMS, and how that line could be meant to represent the pipe connected to a reservoir and water fountain. Functionality such as how to input pipe characteristics (e.g., diameter, material type, length, etc.), how to specify water elevation in the reservoir, and how to find the results of the simulation (i.e., pressure at the point of interest) were demonstrated in about 5 minutes. Students were then asked to start the virtualized version of WaterGEMS on their own personally-owned laptops or university-owned laptop computers. Students were asked to solve the same simple problem – finding the pressure at a point connected to an elevated water reservoir, and following this observe a classmate do the same thing.

A short screen-recording video demonstrating the simple steps required to complete this activity was made available to students for subsequent reference, and is available at:

http://www.youtube.com/watch?v=g_CzboAu63A

Following the in-class demonstration, students were assigned several calculation and concept questions that would require them to again utilize the WaterGEMS software to model a hydrostatic case. Students were asked to change variables that, in case of static conditions, should not cause any change in pressure (e.g., pipe diameter, material type, pipe length) and observe the effects of these changes. Similarly, students were asked to vary the water elevation in the reservoir to achieve a desired pressure at the point-of-interest, and comment on the phenomenon being demonstrated. In this way, students were able to conduct and observe many more calculations than would otherwise be possible if these same computations were performed by hand.

<u>Fluid Properties, Pipe Friction, and Energy Loss</u>. As a second activity utilizing WaterGEMS, students were this time asked to model conditions where water was flowing through a pipe from an elevated reservoir to a fire hydrant. Additional program functionality was introduced in that students were taught how to annotate drawings within the program with text related to system properties and conditions (e.g., flow rate, pipe diameter, headloss, pressure, elevation, etc.). As shown in the assignment handout provided in the appendix, students utilized WaterGEMS to calculate the pressure that would be anticipated by varying flow rate, pipe diameter, fluid type, material type, and pipe length. Thus, in just a few moments at the computer, students were able to verify principles that had previously been taught and solved in the traditional way, utilizing pen-and-paper examples. Additionally, through successive iteration with each parameter, students were able to develop a relative sense for which variables had a significant and non-linear effect on system performance (e.g., decreasing pipe diameter from 4-inches to 3-inches), and which variables had a less dramatic effect on pressure change (e.g., increasing pipeline length).

<u>Network Optimization</u>. The third and final learning activity that was conducted introduced students to an increasingly-sophisticated feature of WaterGEMS: to find the pressure at each junction in a water network, where flow-path is indeterminate. Additionally, using an instructor-provided scaled image of a hypothetical town, students were able to create their WaterGEMS network schematic in a scaled mode, such that pipe lengths drawn on screen defined the pipe lengths within the model. Using a map that included elevation contours (see Appendix) students were required to define the elevation of each junction, and input given network attributes (pipe diameter, material, etc.) in order to solve for the pressure at each junction. As a first introduction to the design principle of 'Demand Estimation' students were required to estimate flow demands based on the land use conditions inside of the network.

Following this 'analysis' stage of the assignment, students were given a design criteria of not allowing pressure at any junction to fall below 300 kPa, and were asked to optimize the network in increments of decreasing pipe diameter of 25 mm. For students, not only was this another opportunity to practice creating annotated drawings and operation of an industry-standard design software package, but also a way of experiencing, early in their engineering course work, the idea of balancing network performance with design constraints in an economical way. All students enrolled in the course were successful in identifying a 'better' design for the network than the one originally outlined in the figure that was provided, and several students went to considerable lengths to identify the just-right, barely-meeting-design-criteria solution that minimized the total cost of pipe network materials.

Thus, not only were students once again exposed to iterative calculations to reinforce their 'feel' for what effect on a system different changes will have (e.g., decreasing diameter will increase velocity, increase headloss, and thus decrease pressure at the downstream end), but trough this learning activity they were also able to develop the feeling that they had engaged in 'engineering

design' and were, at the very least, beginning users of a sophisticated and powerful engineering design package.

Student Response and Feedback

A brief survey was administered to students in-class, following their submission of the assignment, to characterize student opinion about the learning experience and to assess whether any technical issues limited their ability to successfully complete the activity. A total of 29 students (out of 31 enrolled in the course) completed the survey. When asked to describe their reaction to WaterGEMS in their own words, the most common responses were that the software was easy to use (N=13), that the software is a powerful tool (N=11), and that they liked using it (N=8). Other feedback was that the program was easy to learn how to operate, helped them to learn fluid mechanics principles, and that they were anxious to learn more. Two comments were provided that are interpreted as negative: that there were technical problems using the software (N=1), that the program was confusing (N=1).

When asked to estimate the amount of time required to complete the homework assignment (provided in the appendix of this paper), the average duration was 32 minutes.

How difficult was it to learn to use WaterGEMS?	N = 29
3 – VERY difficult. This software should not be taught to students in my position.	0
2 – Somewhat challenging, but not too difficult for students in my position.	18
1 – It was easy.	11
In the context of better understanding the hydrostatic equation, what is your opinion about the WaterGEMS assignment that you completed? From an educational standpoint	N = 29
5 - It was a very useful learning activity. It had significant positive impact on my understanding.	10
4 - It was a somewhat useful learning activity.	15
3 - Neutral	4
2 - It was not a useful learning activity.	0
1 - It was a very useless learning activity. It actually made me understand hydrostatics LESS.	0
Did you view the screen-recording video?	N = 29
Yes	7
No	22
To complete the homework assignment, did you use the Virtualized version of WaterGEMS, a lab computer with WaterGEMS, or did you install WaterGEMS on your home computer? (Select all that apply.)	N=29
Virtualized	18
Lab computer	13
Installed on own machine	2
If you used the Virtualized version of WaterGEMS, did you encounter any problems?	N=18
Yes	5
No	13
How interested are you to see more of your academically required engineering software and assignments provided online (i.e., using a remote delivery 'Virtualization' system, rather than having to access software in the lab)?	N=29
5 - Very Interested	14
4 - Somewhat interested	11
3 - Neutral	4
2 - Not interested	0
1 - Very uninterested - would prefer that it not happen at all	0

Table 1 – Results of student feedback on learning activity survey.

As summarized in Table 1, 18 of 29 students (i.e., 62%) completed some or all of the assignment utilizing the "virtualized" (i.e., remotely-operated, server-hosted) version of the program. This is significant because students also had ready access to a local installation of the program in a computer lab adjacent to the classroom where the course is taught, such that they could have completed the homework assignment on campus without the need to configure the virtualization applet on their personal computer. However, the benefit of being able to operate the program on their own PC, without having to actually install it, was enough for 62% of students to choose this option. Of these 18, 16 completed the assignment entirely using the virtualized version of

WaterGEMS, and 2 used both the virtualized version of the program and a lab-installation of the software. 13 of 29 students (45%) completed some or all of the assignment on lab computers, and 2 students installed the full version of the program on their own computers.

Of primary importance when considering student feedback on this experience is that all students characterized the software to be "easy" or "not too difficult", with none selecting the option that the software was "VERY difficult." This outcome highlights the importance of identifying a small, simple introductory exercise with which to initiate student use of new software, and may also reflect the value of a live in-class demonstration where students first see the unfamiliar activity demonstrated, then they complete it themselves, and then they observe a classmate completing the steps a second time. Although the WaterGEMS software does have sophisticated and complex functionality, students can develop a primary viewpoint that it is easy to use through a targeted introductory exercise.

86% of students responded that the software homework assignment that followed the in-class demonstration was "very useful" or "somewhat useful" as a learning activity that supported their understanding of the hydrostatic equation. Thus, even though the WaterGEMS program's functionality goes well beyond calculating hydrostatic pressures, students felt that the assignment – which required them to investigate the effect of manipulating several different parameters within the hydrostatic equation – helped them to better learn the underlying principles.

Of the five students who encountered problems when using the virtualized version of the software, two reported that they were unable to connect to the software after loading the virtualization applet on their computer, one encountered unexpected program termination (possibly due to their internet connection being dropped), one student reported that the program 'didn't work', and one student reported a 'laggy' (i.e., high latency) connection when running the program virtually.

Conclusions

The learning activities described in this paper were undertaken as a pilot test of two principles: (1) introduction of hydraulic design software in the early stages of a student's learning of fundamental principles, and (2) utilization of virtualization systems to simplify the steps necessary to incorporate software instruction into a course. On both accounts, this pilot test was encouraging – students reported a favorable experience using WaterGEMS, and indicated an interest in receiving more software through virtualization. Likewise, instructor resources required for virtualized-delivery of software were less intensive than when software is installed locally, and thus continued future implementation of this methodology is anticipated.

References

- Jones, J.M. and Chou, T. "Work-in-Progress: Creating an Intrusion Detection Experimental Environment Using Cloud-Based Virtualization Technology." Proceedings of the 2012 ASEE Annual Conference and Exposition, San Antonio, Texas, June 10-13, 2012.
- [2] Li, T., Thain, W.E., and Fallon, T.. "On the use of virtualization for router network simulation." Proceedings of the 2010 ASEE Annual Conference and Exposition, Louisville, Kentucky, June 20-23, 2010.
- [3] Bailey, M. and Ekstrom, J. "Teaching web development with OS-virtualization." Proceedings of the 2009 ASEE Annual Conference and Exposition, Austin, Texas, June 14-17, 2009.

Appendix – "Learning Objects"

Learning Objects: Fluid Mechanics with WaterGEMS



FM1 – Hydrostatics

[Note: Before this activity, ensure that the model Friction Method is "Darcy-Weisbach", and not the default "Hazen-Williams" setting. See See Analysis | Calculation Options | Steady State/EPS Solver | Base Calculation Options | <General> | Friction Method.]

A drinking fountain is connected to a reservoir as shown above. The 10 m long pipe is 5 cm diameter cast iron, and conditions are **static** (i.e., no water is flowing).

In-Class Demonstration and Activity: *Use WaterGEMS to find the pressure in the drinking fountain.* [Ans.: 265.3 kPa]

Demonstrate how to:

- Draw network elements: reservoir, pipe, junction
- Edit element parameters individually
- Edit element parameters through FlexTables (including 'User Specified Length')
- Open the Calculation Options window
- Compute model results
- View model results (i.e., pressure at a junction) through FlexTables
- Capture a screenshot, paste it into MS Word, and crop it to include only relevant information.

Homework Problems

For computation problems, print a cropped screenshot showing the results of your WaterGEMS analysis. For explanation questions, please provide a detailed, clear, <u>typed</u> response. Overly brief and/or poorly-reasoned responses will not receive full credit.

- What would be the pressure in the drinking fountain if the reservoir contained 4 °C water, instead of the 20 °C that WaterGEMS initially assumes? (See Analysis | Calculation Options | Steady State/EPS Solver | Base Calculation Options | Hydraulics | Liquid Label ...)
- 2) Write a brief explanation (about one paragraph) about the pressure difference that was observed in Problem 1. (Explain why it was different, and explain the <u>directionality</u> of the difference. For example, if the 4 °C water yielded a pressure that was higher than the 20 °C water, explain specifically why the pressure was higher and not lower.)
- 3) Regarding the pressure value that you found in Problem #1, is that high, low, or about right for what is needed at a drinking fountain? (You may consult the internet for additional information, if needed, to address typical pressure ranges in drinking water networks.)
- 4) What would be the pressure if it was a reservoir of mercury?
- 5) Explain why the pressure in the fountain is so much higher if the reservoir contains mercury instead of water.
- 6) What would be the maximum reservoir water elevation allowed (to the nearest tenth of a meter) if the drinking fountain manufacturer specified a pressure limit of 850 kPa? (Note: be sure to switch the program back to 20 °C water).
- 7) What is the drinking fountain pressure if the pipe length is 1000 m instead of 10 m?
- 8) What happens to pressure if the pipe diameter is 25 cm instead of 5 cm?
- 9) Explain "why" for Problems 7 and 8 (i.e., explain why the results that were computed are the way they are, in terms of the fundamentals of fluid mechanics.)
- 10) Explain the additional factor(s) that would need to be accounted for when calculating pressure if water was <u>flowing</u> through the pipe (instead of the static conditions that were specified)?

Learning Objects: Fluid Mechanics with WaterGEMS

FM2 – Fluid Properties and Flow



[Note: Before this activity, ensure that the model Friction Method is "Darcy-Weisbach", and not the default "Hazen-Williams" setting. See See Analysis | Calculation Options | Steady State/EPS Solver | Base Calculation Options | <General> | Friction Method.]

An elevated water reservoir in Florence, Kentucky is connected to a fire hydrant, as shown above.

Lab Demonstration and Activity: Use WaterGEMS to find the pressure at the hydrant when...

Pipe Length (ft)	Pipe Material	Pipe Diameter (in)	Flow Demand (gpm)	Liquid
1500	Cast Iron	4	180	Water, 68°F

Demonstrate how to:

- Specify flow demand
- Add annotation
- Change FlexTable columns that are displayed

Lab Activity Problems

Condition	Pipe Length	Pipe Material	Pipe Diameter	Flow Demand	Liquid
	(ft)		(in)	(gpm)	
А	2500	Cast Iron	4	180	Water, 68°F
В	1500	Glass	4	180	Water, 68°F
C	1500	Cast Iron	3	180	Water, 68°F
D	1500	Cast Iron	4	250	Water, 68°F
E	1500	Cast Iron	4	180	SAE 30, 100°F

Compute the pressure at the hydrant under the following conditions.

Submit a single printed document that contains an annotated, cropped screenshot for each of the analysis conditions listed above. Clearly identify which condition is being assessed. For each simulation, (a) identify how the situation has changed compared to the baseline example, (b) describe what the resulting effect of that change is on pressure, and (c) briefly explain, in terms of fluid mechanics theory, why you think the pressure changed the way that it did.

Example annotation and screen-shot format:



Figure 1 – Analysis Condition: Demonstration

[Note: the conditions illustrated above may or may not represent the correct answer for Analysis Condition B]



FM3 – Network Analysis and Optimization