
AguaClara Textbook Documentation

Release 0.038

AguaClara Cornell

Jan 02, 2019

ACKNOWLEDGEMENTS

1 Acknowledgements	3
2 Authors	5
3 Introduction to AguaClara Water Treatment Design	7
4 Introduction Design Challenge	27
5 Python Tutorial	29
6 Python Design Challenge	37
7 Review: Fluid Mechanics	43
8 Review: Fluid Mechanics Derivations	63
9 Flow Control and Measurement Introduction	67
10 Flow Control and Measurement Design	71
11 Flow Control and Measurement Derivations	79
12 Flow Control and Measurement Solution	93
13 Flow Control and Measurement Design Challenge	107
14 Rapid Mix Introduction	111
15 Rapid Mix Design	127
16 Rapid Mix Derivations	129
17 Rapid Mix Appendix C: Examples	151
18 Rapid Mix Theory and Future Work	161
19 Rapid Mix Mechanical Solution	171
20 Rapid Mix Design Solution	179
21 Flocculation Introduction	181
22 Flocculation Model	187

23 Flocculation Design	217
24 Flocculation Derivations	239
25 Flocculation Theory and Future Work	241
26 Flocculation Examples	243
27 Mechanical Flocculator Solution	247
28 Flocculation Model Solution	253
29 Flocculator Design Solution	263
30 Flocculation Design Challenge	277
31 Sedimentation Introduction	279
32 Sedimentation Design	281
33 Sedimentation Examples	283
34 Sedimentation Theory and Future Work	285
35 Sedimentation Design Solution	291
36 Sedimentation Design Challenge	303
37 Filtration Introduction	305
38 Filtration Design	317
39 Filtration Derivations	325
40 Filtration Theory and Future Work	339
41 Filtration Design Solution	353
42 Filtration Design Challenge	365
43 Dissolved Gas Introduction	367
44 Fluoride Introduction	369
45 WasteWater Theory and Future Work	371
46 Problems	375
47 Solutions	377
48 Proposed solutions to eliminate iron oxidizing slime bacteria	383
49 Synthesis Design Challenges	385
50 Introduction to RST and Sphinx for Textbook Contributors	387
51 Parameter Convention List	391
Bibliography	393

This textbook is written and maintained in [Github](#) via [Sphinx](#). It uses and refers to AquaClara code and functions in [aguaclara](#). Listed below are the versions of the programs we use:

Table 1: These are the software versions used to compile this textbook

Software	version
Sphinx	1.7.5
aguaclara	0.0.16
Anaconda	4.5.4
Python	3.6.5

**CHAPTER
ONE**

ACKNOWLEDGEMENTS

We gratefully acknowledge the funding provided by the Environmental Protection Agency and the National Science Foundation. Together they have provided over \$1 million in support of developing the next generation of sustainable drinking water treatment technologies.

1.1 Environmental Protection Agency statement

This textbook was developed under numerous Assistance Agreements awarded by the U.S. Environmental Protection Agency to Cornell University. It has not been formally reviewed by EPA. The views expressed in this document are solely those of the authors and do not necessarily reflect those of the Agency. EPA does not endorse any products or commercial services mentioned in this publication.



1.2 National Science Foundation statement

This material is based upon work supported by the National Science Foundation under Grant numbers CBET-1704472 and CBET-1437961. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Table 1.1: Table of funded research projects that contributed to the knowledge in this textbook.

Agency	Proposal Title
NSF	Wrf: Experimental Observation and Modeling of Coagulant Mediated Contaminant Removal: Flocculation, Floc Blankets, and Sedimentation
USEPA	AquaClaras Ram Pump for Zero Electricity Drinking Water Treatment
USEPA	Environment & Community Friendly Wastewater Treatment
USEPA	High Rate Sedimentation
USEPA	Novel Reactor Design for Enhanced Removal of Fluoride Using A Modified Nalgonda Method
USEPA	Novel Reactor Design for Enhanced Removal of Fluoride Using A Modified Nalgonda Method
NSF	Experimental Evaluation And Modeling Of Hydraulic Flocculation Systems Under Conditions of Turbulent Flow
USEPA	Application of Foam Filtration to Water Treatment for Rapid Emergency Response
USEPA	Stacked Rapid Sand Filtration - A Robust Filtration Process for Sustainable Drinking Water
USEPA	Sustainable Water Treatment Facility for Communities with Arsenic Contaminated Groundwater
USEPA	Smart Turbidimeters for Remote Monitoring of Water Quality
USEPA	Stacked Rapid Sand Filtration - A Robust Filtration Process for Sustainable Drinking Water Infrastructure
USEPA	Developing A Point-of-Use Filter Utilizing Polyurethane Foam
USEPA	Dose Controller for AquaClara Water Treatment Plants
USEPA	Dose Controller for AquaClara Water Treatment Plants
USEPA	AquaClara: Clean Water for Small Communities

More gratitude below!

- Ken Brown and the Sanjuan Foundation
- Duane Stiller
- Countless other donors whose contributions have made it possible to develop new technologies and share those technologies with partner organizations and communities
- Many hundreds of students who gave their time and creativity so that others could have safe water on tap
- The Swiss Development Cooperation that has funded the construction of 5 AquaClara plants in Honduras and 2 in Nicaragua
- The Cornell Engineering that provided generous funding in the startup years.

CHAPTER**TWO**

AUTHORS

This text is a collaborative effort involving hundreds of people. Innovation requires collisions of ideas and the AguaClara program was designed to foster global and multidisciplinary interactions between students, faculty, field engineers, plant operators, implementation partner organizations, and community members. These interactions have provided a continuous and rich source of ideas that make it clear that in a social network it is impossible for anyone to claim ownership of an idea. Thus the inventions, equations, and reactor designs that are described in this text are the product of a large, collaborative, open-source community and none of us can claim that we are the sole authors. The list of authors below have contributed directly to this text.

- Monroe Weber-Shirk
- Juan Guzman
- Clare OConnor
- William Pennock
- Leonard Lion
- Yingda Du

This text would not have been possible without the advice, mentoring, and publishing system assembled by Ethan Keller.

INTRODUCTION TO AGUACLARA WATER TREATMENT DESIGN

3.1 A Different Kind of Textbook

This textbook represents our cumulative insights from our journey that has been motivated by a quest to make the world a better place where everyone has access to safe water on tap, the engineering challenge of optimizing the design of drinking water treatment plants, and the curiosity to understand what controls their performance. We would like to understand what determines which contaminants make it the whole way through a water treatment plant. If we could understand what allows some contaminants to sneak the whole way through a water treatment plant, then we suspect that we could create better designs to more effectively remove contaminants.

Engineering textbooks provide a venue for authors to share what theyve learned, to organize ideas, and to provide a guide for engineers as they design solutions for real world problems. Engineering textbooks are often intended to document the established core of knowledge. It seems reasonable to assume that what is in textbooks and in peer reviewed literature is mostly true.

3.1.1 The edge of knowledge may be closer than we thought

The assumption that what is written and passed down in oral history through the scientific community is true can lead to missed opportunities and lost insights. The hypotheses from one generation of scientists can too easily evolve into new theories in the next generation and then into established theories for the next. The history of drinking water treatment science is cloudy (think high turbidity!) with hypotheses that miss or misrepresent key concepts.

You might wonder why we care so much about getting the science right and being as clear as possible about what is known. After all, the core drinking water treatment technologies were invented before we were born and many of us have safe drinking water coming from our taps. Environmental Engineers have known how to design municipal drinking water treatment plants since the early 1900s. We care about getting the science right because we hypothesize that there are many opportunities to significantly improve drinking water treatment technologies and that improved understandings of each unit process have the potential to lead to new breakthroughs.

Our contention is that no one has ever optimized the design of a drinking water treatment plant! We are reasonably certain of this because we dont yet have models (with equations) that describe performance of most of the core unit processes (rapid mix, flocculation, floc blankets, sedimentation, sand filtration) used for surface water treatment. The only possible exception is lamellar sedimentation which can be characterized if we know the size and density distribution of the particles entering the sedimentation tank.

Traditional drinking water treatment textbooks can too easily miss the opportunity to advance the science of drinking water treatment technologies by presenting certainty where there should be skepticism. For example, rapid mix is described as process that occurs in a few seconds, flocculation is described as a process that should be fastest for high turbidity waters and slowest for low turbidity waters, and filtration performance is described by a model that predicts first order removal with respect to filter bed depth. We will demonstrate why each of these assumptions doesnt match observations, we will discuss new insights into these processes, and we will identify high priority research questions that have the potential to lead to major improvements in drinking water treatment.

We want to encourage skepticism and we want to develop insights to guide thoughtful skepticism. A key skill for successful engineering is the ability to identify the location of the edge of knowledge. The ability to distinguish between what is reasonably certain and what is still in question is what powers the scientific method of slowly extending knowledge. New insights are difficult to obtain if the research is based on a faulty premise.



Fig. 3.1: We've learned that we can find the edge of knowledge very soon after we begin researching a water treatment technology (artwork created by Yi Wen Ng in 2012).

There are significant knowledge gaps in every process that we cover in this textbook. We aren't yet able to optimize surface water treatment processes because we don't yet understand the fundamental physics of many of the processes. We are getting closer, join us on the journey.

We need the brightest and the best to create new and better solutions so we can meet the goal of providing everyone with safe drinking water. This challenge is apparently more difficult than building a space station, designing a fuel cell, or inventing the world wide web. So let's roll up our sleeves and begin.

3.1.2 Tools to Find the Edge of Knowledge

- Don't believe everything we say
- Ask lots of questions
- How do you know that? The goal here is to identify the difference between what is known and what is hypothesized.
- What is the equation that describes the physics of this process? If there isn't an equation that describes the process and that can be used to design the reactor for the process, then it is likely that the physics of the process is not yet understood.
- How could we improve this process? If the physics of a process are fully understood, then dimensionally correct equations can be used to obtain the optimal design for that process.

- Is the process design based on rules of thumb or on physics? Rules of thumb or empirical design guidelines often can be identified by the use of physical parameters that have units. For example, if the design guideline specifies a length, time, or velocity then it is likely that the guideline is not based on physics. If the design guidelines are based on a dimensionless parameter then it is possible that it is based on physics.
- Evaluate the data to see if it matches predictions of the hypothesized model. Assess whether the authors acknowledge when their data doesn't match hypothesized models.
- Beware of the use of words that are poorly defined and that hide uncertainty. For example, creating a name for a supposed mechanism to describe all of the observations that don't fit with your theory does NOT mean that you understand that mechanism. The ability to name something doesn't mean it is understood.
- Does this theory provide insights that have led to new discoveries or new applications?
- Does the theory include equations that are based on the fundamental laws of nature?
- Does the theory use dimensionless constants that are close to one?
- Is it an elegant theory with no need for special cases?

3.1.3 Myth in Environmental Engineering

The following list is designed to get you thinking. These are concepts that are present in the Environmental Engineering community.

- Dead bodies cause disease
- Slow sand filters ripen (improve in ability to remove contaminants over time) because of biological growth in the filter bed
- If a 20 cm deep sand filter removes 90% of influent particles, then a 40 cm deep filter will remove 99% of influent particles
- If water is dirty, then you should filter it
- Chlorine disinfects dirty water and makes it safe to drink
- Chlorination and filtration eliminated typhoid fever from the US
- Cessation of chlorination due to fear of disinfection by products caused the cholera outbreak in Peru in 1993
- Sedimentation is simple
- We already know how to solve the problem of the billions of people who do not have access to safe drinking water

3.1.4 Uncertainty in Science and Engineering

A challenge for authors is to recognize the difference between what is known with a reasonably high degree of certainty and what is assumed to be true without a solid basis. We struggle to tell the difference between fact and hypothesis. The time-honored approach in science is to rely on the peer review process. That process for vetting knowledge has been shown to be flawed.

Your question could be whether the distinction between fact and hypothesis really matters. If the hypothesis is widely accepted as fact and if it has been accepted for decades what benefit is there to calling it a hypothesis rather than a fact?

This question is at the core of our educational philosophy. Is this text the repository of knowledge that we are providing for you to drink or is this text a conversation where we invite you to join the effort to discover better ways to provide safe water on tap?

3.1.5 Integrating Educational Philosophy with an Evolving Textbook

This is an evolving textbook. We don't intend to ever print this book. This book has version numbers just like software with the idea that revisions are rapid and frequent. We commit to helping to accelerate the pace of knowledge generation and to revising this text as you help us identify places where we have presented hypotheses as theory and places where research provides a basis for better theoretical models of the water treatment processes.

Our students are co-creators of knowledge and not empty vessels to be filled with our wisdom. AquaClara technologies are inventions that are the result of idea collisions in the AquaClara labs and from observations and reflections with operators, technicians, and engineers in dozens of water treatment plants. Although we've learned a great deal about water treatment since 2005 when AquaClara was founded, there is still much more to be learned. And so it is with a spirit of curiosity that we write this textbook expecting to learn even more in the coming years.

Socrates said [Education is the kindling of a flame, not the filling of a vessel](#). Our goal is to bring the spirit of play, discovery, and mystery into the challenge of improving the quality of life of everyone on the planet by sharing better methods to produce safe drinking water.

In [We Make the Road by Walking: Conversations on Education and Social Change](#), Paulo Freire said, [The more we become able to become a child again, to keep ourselves childlike, the more we can understand](#). We commit to playing together in a relationship where we are all learning and we are all teaching. Education must begin with the solution of the teacher-student contradiction, by reconciling the poles of the contradiction so that both are simultaneously teachers and students. - Paulo Freire

3.1.6 Respect, Empathy, Love and Curiosity power the AquaClara Innovation System

The AquaClara network of organizations has been methodically inventing improved water treatment technologies since 2005. Our success is based on respect, empathy and love. Innovation requires flocculation of ideas. The transport of ideas between organizations and individuals is mediated by respect. Respect as a cornerstone of organizational culture fosters rapid and honest exchange of ideas. The rapid pace of innovation in the AquaClara network is sustained through a shared culture of respect, empathy, and love.

Curiosity can flourish in a culture of love, respect, and empathy. Asking why and why not and pondering an ever growing number of questions has empowered student teams to take on the quest for new knowledge and new solutions.

Any large organization will require a leadership hierarchy and any hierarchy will rely on respect based on fear or respect based

- They'll Teach You, Whether You Like It or Not
- Everyone is a Friend or a Foe
- It's All about the Trophies
- They Don't Step Outside Boxes
- They're Addicted to Yardsticks

Love-based hierarchies foster honesty and a free-flow of information. Reflection is encouraged across the organization and truth, honesty, and integrity are valued. Staff at the bottom of the hierarchy know that their opinions and reflections are valued and thus they will be willing to report problems to organization leaders and share their ideas.

Love-based leaders relate to others based on true respect for the other. They will take the time to converse with people at all levels of the organization and will value the opportunity to speak with people who are the interface between

the organization and the rest of the world. A persons value is based on being a person, not based on position in the hierarchy.

As water treatment plant designers it is critical that we spend time with a diverse set of stakeholders including community members and water treatment plant operators. Those relationships must begin with respect and valuing their insights. As we spend time together we can develop trust so that they communicate both the good and bad.

Weve learned much from plant operators. They figured out how to reduce rising flocs at Agalteca, Honduras where we learned that conventional sedimentation tank inlet manifolds generate large circulation currents. Plant operators added curtains to the windows at Moroceli, Honduras (see Fig. 3.2) because they noticed that direct sunlight on the sedimentation tanks caused an increase in settled water turbidity.



Fig. 3.2: Moroceli AguaClara water treatment plant operators installed curtains to reduce direct sunshine on sedimentation tanks. Solar heating produces density currents that carry flocs to the top of the sedimentation tank.

Empathy is fundamental in design. Empathy enables us to consider reality from another's perspective. Empathy enables us to bring the people who will use or benefit from a technology into the design considerations. Empathy brings the insight that water treatment plants need to have roofs and provide a secure work environment both day and night. Empathy brings the insight that replacement parts must be readily available and that generic components are preferred over specialty proprietary components.

3.2 The Global Context for Drinking Water Treatment

The Sustainable Development Goals: SDGs and specifically SDG 6 provide the context and motivation for this text. The first SDG 6 target is: By 2030, achieve universal and equitable access to safe and affordable drinking water for all. That goal is daunting and won't be met using the approaches of the past 5 decades. This text is about creating a new paradigm for the design of high performing water treatment technologies with the goal of making a real contribution toward SDG 6.1.

The number of people who currently lack access to reliable safe water on tap is not known. Estimates range from 1.8 billion who use a source of drinking water that is contaminated with feces to the Centers for Disease Control recommendations for where it is usually safe to drink tap water.

The UN estimate in 2017 was that 2.1 billion lack access to safe water. By 2030 there will be an additional 1.2 billion from population growth.



Fig. 3.3: Sustainable development goal 6 is all about clean water and sanitation.



Fig. 3.4: There are relatively few countries where it is almost always safe to drink the tap water.

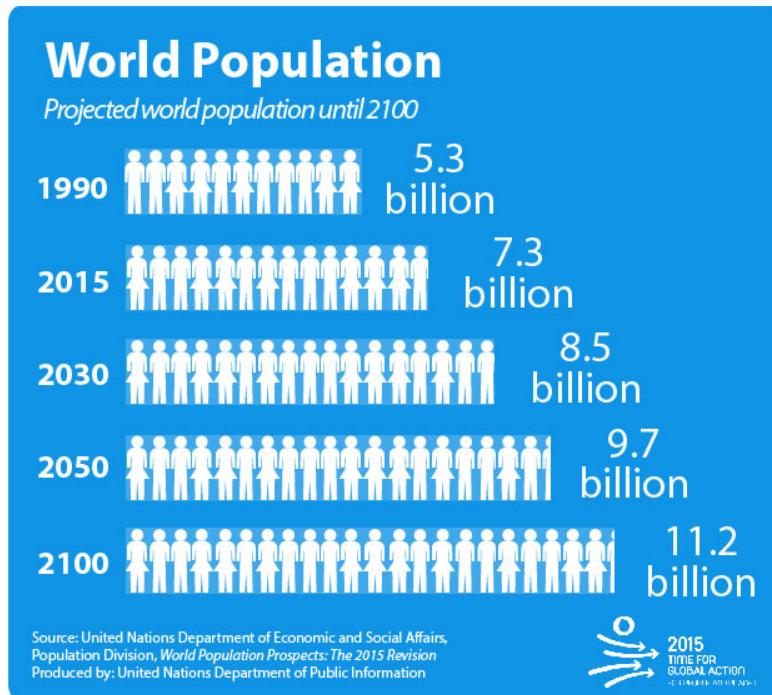


Fig. 3.5: 1.2 billion people will be added to the global population between 2015 and 2030.

Thus by 2030 we need to provide safe water for at least 3.3 billion people AND maintain the water supply systems for the 5.2 billion who currently have access to safe water. That is a daunting number that requires some exploration!

```
from aquaclara.core.units import unit_registry as u
import datetime
People_needing_water_2030 = 3.3*10**9
now = datetime.datetime.now()
Task_time = (2030 - now.year)*u.year
#If we assume we will meet this demand by building the same amount of new capacity_
#each year, then we have
People_per_year = People_needing_water_2030/Task_time
People_per_year
#The per capita demand for water
Per_capita_demand = 3*u.mL/u.s
Per_capita_demand.to(u.L/u.day)
Per_capita_demand
Rate_new_water_supply_capacity = (People_per_year * Per_capita_demand).to(u.L/(u.s*u.
#year))
Rate_new_water_supply_capacity
NYC_water_supply = 44000 * u.L/u.s
NYC_per_year = Rate_new_water_supply_capacity/NYC_water_supply
NYC_per_year
```

If we provide 260 L/day per person, then we need to provide the equivalent of 19 water supplies for New York City every year between now and 2030. The planet needs approximately 800,000 L/s of new capacity each year. AquaClara water treatment plants cost approximately \$10,000 per L/s of treatment capacity. Thus the budget for global water treatment needs to be 8 billion USD per year. Note that this doesn't include any other aspects of supplying water. Managing water sources, transmission lines, storage, and distribution systems are even more expensive than water treatment.

The need for drinking water supplies isn't limited to the global south. The California Urban Water Agencies estimate that 530,000 or more people in rural areas of California are unable to turn on their tap and access clean, safe water.

3.2.1 Why don't 2 billion people have access to safe water?

The simple answer is that they are too poor and are unable to afford safe water on tap. But it isn't that simple! Families without access to safe water on tap often spend more for water than families with safe water on tap. There seem to be two key reasons why those with limited financial resources often have limited access to water, poor quality water, and yet pay a premium for that water.

The first reason for the lack of safe water has been the poor track record of water treatment infrastructure. The frequent failures and high operating costs of municipal scale water treatment systems have led many decision makers to conclude water treatment infrastructure isn't a worthwhile investment. Politicians who invest political capital to bring water treatment to their community often find that after the initial ribbon cutting there is little political benefit because the system doesn't deliver the benefits to the community that they had promised.

The second reason for the lack of safe water is the lack of access to capital for municipal scale infrastructure. Even though an AquaClara water treatment plant would pay for itself in a fraction of its useful life, there is not yet a financial mechanisms for communities to access a loan so that they can make the investment. A community would need to save enough money to be able to purchase a water treatment plant (as was the case for Las Vegas, Honduras), a bilateral donor can finance a plant through a donation, or the national government can use sovereign debt or taxes to finance plants. The challenge for a community is to obtain the financial or political power to access the needed funds.

As we work to solve a global challenge that has been plaguing humanity since the dawn of human civilization, then it will serve us well to understand a bit of the history that has led to our current reality. Water treatment history includes amazing successes, persistent failures, fortuitous discoveries, a heavy reliance on empiricism, and an occasional myth. Our goal is learn from and reflect on our history and then create even better solutions.

3.3 Introduction to Surface Water Treatment

We treat water because it doesn't meet the requirements for its intended use. We need to understand the problem so that we can understand existing and novel water treatment technologies.

3.3.1 Water Contaminants

Many substances are able to dissolve in water and with its high density, water is able to carry suspended solids. The substances may be naturally occurring, anthropogenic, benign, or harmful. The types of contaminants are influenced by the water source. Contaminant concentrations are often highly variable over time.

A water treatment system must be able to handle the likely range of contaminant levels and produce treated water that meets the user requirements. In some cases the user may have the option of switching sources or reducing demand when a source becomes excessively contaminated for a limited period of time. For example, a municipal water supplier may be able to shut the plant down for a few hours to avoid having to treat a very dirty water after a rainstorm. This strategy can work well for water sources that have small watersheds and hence a rapid return to better water after the storm passes. In other cases the water treatment processes must be capable of treating the most contaminated water that the water source provides. In any case, selecting the best unit processes to treat a given water source for a particular use case can be challenging. It is common to find water treatment plants that are unable to adequately treat their water source.

Particles

Surface waters (rivers, streams, lakes) and some ground water (especially ground water under the influence of surface water) inevitable carry some suspended particles. Particles transported by rivers are composed of resistant primary minerals (e.g., quartz and zircon), secondary minerals (clays, metallic oxides and oxyhydroxides) and biogenic remains. Many of these particles may be harmless, but there is good reason to be hesitant to drink water with a high concentration of suspended particles.

Pathogens

Pathogens include viruses (100 nm), bacteria ($1 \mu\text{m}$), and protozoa (several μm). Pathogens are particles and are removed by processes that remove particles along with other microbes, organic and inorganic particles.

Turbidity

Turbidity or cloudiness is an indirect measure of particle concentration. Turbidity is an optical measurement of scattered light. Light scattering by refraction is primarily caused by particles that are smaller than but close to the wavelength of light. Particles that are close larger than the wavelength of light can reflect light. Turbidity measures both of these effects by shining a light into a water sample and then measuring the scattered light with a photodetector at 90° . The meter is then calibrated with standard suspensions.

For a given suspension the turbidity can be directly correlated with the suspended solids concentration. However, that relationship is complicated because the amount of scattered light is related to the particle size distribution because given the same mass concentration, smaller particles have more surface area and thus reflect more light.

Although turbidity would seem to be an odd parameter to use to measure water quality, it turns out to be the most widely used water quality measurement. The reasons are simple. First, turbidity is amazingly easy to measure over a very wide range of particle concentrations (perhaps $10 \mu\text{g/L}$ to 1 g/L). The test doesn't require any reagents and it can be done in a flow through sample cell for real time measurements. Second, particle free water is pathogen free water. Third, disinfection processes (chlorination, ozonation, UV light) are all significantly less effective at inactivating pathogens if there are other particles present in the water.

Dissolved Species

The list of dissolved species that can be present in water in the environment is endless and ranges from natural organic matter (from decay of plants) to caffeine to atrazine. Usually the highest concentration class of molecules is dissolved natural organic matter (NOM). NOM has some similarity to inorganic particles in that it isn't necessarily harmful and yet there are several reasons why removal of NOM is an important water treatment goal.

From an aesthetic perspective, NOM absorbs light at short wavelengths and this results in water that looks yellow or brown. While I enjoy drinking tea with a rich brown color, I'd prefer that my water be clear.

NOM plays a supersized role in influencing performance of surface water treatment plants. NOM has three negative effects:

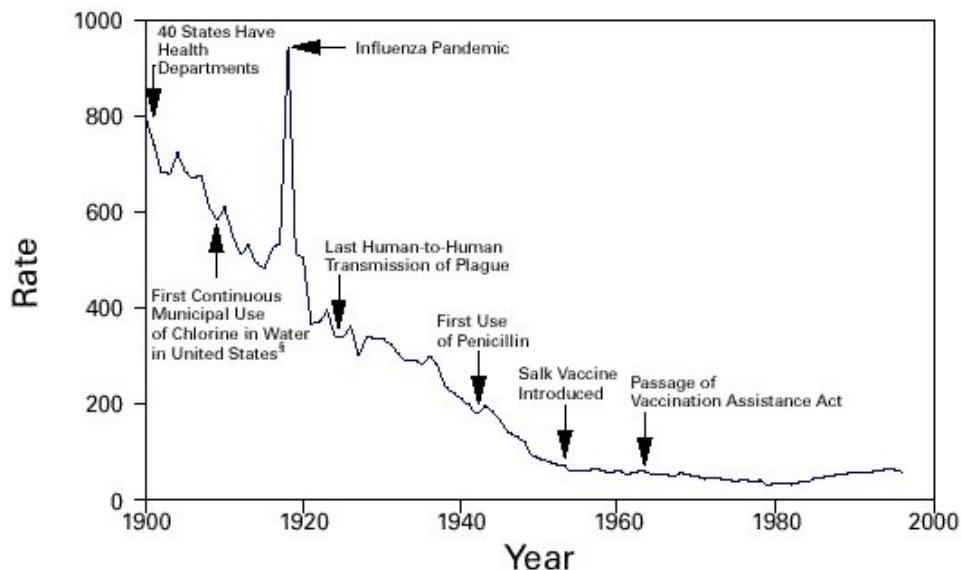
1. It requires higher dosages of coagulant for effective particle removal.
2. It reduces the disinfection effectiveness of chlorine, ozone, and UV. Chlorine partially oxidizes the NOM and thus more chlorine must be used to maintain a residual level of chlorine.
3. It can produce disinfection by-products that are toxic.

Thus removal of NOM is a water treatment goal. Fortunately the same coagulants that are used for particle removal also can remove a significant fraction of NOM. The interactions between NOM and coagulants will be discussed in the *Introduction to Rapid Mix*.

The removal of other dissolved species is beyond the scope of the first release of this textbook. The authors intend to add sections on the removal of some dissolved species in the near future.

3.3.2 Chlorine (Might Have) Saved the World

Chlorine is widely recognized for reducing mortality from water borne disease in the United States. A more careful review of the mortality data and of the ability of chlorine to inactive various pathogens makes it difficult to assess the role of chlorine. A classic graph (see Fig. 3.6) has been used to suggest that chlorination of drinking water supplies resulted in a significant reduction in mortality



*Per 100,000 population per year.

†Adapted from Armstrong GL, Conn LA, Pinner RW. Trends in infectious disease mortality in the United States during the 20th century. JAMA 1999;281:61–6.

‡American Water Works Association. Water chlorination principles and practices: AWWA manual M20. Denver, Colorado: American Water Works Association, 1973.

Fig. 3.6: Classic graph showing the reduction in the death rate for the United States from 1900 to 1996.

Table 3.1: Surface Water Treatment Technologies

Technology	Description	Prerequisite	Owner	Year
Simple sedimentation	particles settle	none	public	unknown
Flocculation	aluminum and iron salts	none	public	1757
Sedimentation	horizontal flow	flocculation	public	unknown
Lamellar sedimentation	plate or tube settlers	flocculation or floc blanket	public	1904
Roughing filter	simple sedimentation in a gravel bed	none	public	unknown
Slow sand filtration	Roughing filter or single step treatment for low NTU water	none	public	1829
Rapid sand filtration	depth filtration	sedimentation	public	1920
Stacked rapid sand filter	gravity powered backwash	lamellar sedimentation	AquaClara Cornell open source	2012
Floc blanket	upflow fluidized suspension of flocs	flocculation	public	1930
Jet reverser floc blanket	first fully fluidized floc blanket	flocculation	AquaClara Cornell open source	2012
Ballasted sedimentation	small sand carry particles downward	•	Actiflo Veolia	1995
Superpulsator	pulsing flow through floc blanket	rapid mix	Degremont	1958 1991
Dissolved air flotation	bubbles carry particles upward	flocculation	Public	1905

See Pretreatment Processes for Potable Water Treatment Plants by Jeff Lindgren for an excellent overview of available technologies, May 2014 (not including AquaClara innovations).

3.3.3 Treatment Trains

The prerequisites for the unit processes in Table 3.1 reveal that surface water treatment almost always requires a series of treatment steps. A treatment train is a series of treatment steps (or unit processes) designed to convert a contaminated source water into a purified water meeting the use requirements.

Example treatment trains include:

- Conventional mechanized treatment: mechanical flocculation, lamellar sedimentation, rapid sand filtration, disinfection
- Superpulsator: rapid mix, floc blanket, lamellar sedimentation, rapid sand filtration
- AquaClara: hydraulic flocculation, floc blanket, lamellar sedimentation, stacked rapid sand filtration, disinfection
- Membrane filtration: flocculation, sedimentation, rapid sand filtration, granular or powdered activated carbon, pre-oxidation (see Review Article)

3.4 The AguaClara Treatment Train

Why does flocculation precedes sedimentation? Which process removes the largest quantity of contaminants?

Sedimentation is the process of particles falling because they have a higher density than the water, and its governing equation is:

$$\bar{v}_t = \frac{D_{particle}^2 g}{18\nu} \frac{\rho_p - \rho_w}{\rho_w} \quad (3.1)$$

Such that:

\bar{v}_t = terminal velocity of a particle, its downwards speed if it were in quiescent (still) water

$D_{particle}$ = particle diameter

ρ = density. The p subscript stands for particle, while w stands for water

```
import aguaclara.core.physchem as pc
import numpy as np
import matplotlib.pyplot as plt
def v_t(D_particle,density_particle,Temperature):
    return (D_particle**2*pc.gravity * (density_particle - pc.density_
    ↪water(Temperature)) / (18*pc.viscosity_kinematic(Temperature)*pc.density_
    ↪water(Temperature))).to(u.m/u.s)
clay = 2650 * u.kg/u.m**3
organic = 1040 * u.kg/u.m**3
Temperature = 20 * u.degC
D_particle = np.logspace(-6,-3)*u.m
fig, ax = plt.subplots()
ax.loglog(D_particle.to(u.m),v_t(D_particle,clay,Temperature).to(u.m/u.s))
ax.loglog(D_particle.to(u.m),v_t(D_particle,organic,Temperature).to(u.m/u.s))
ax.set(xlabel='Particle diameter (m)', ylabel='Terminal velocity (m/s)')
ax.legend(["clay or sand","organic particle"])
imagepath = 'Introduction/Images/'
fig.savefig(imagepath+'Terminal_velocity')
plt.show()
```

The terminal velocities of particles in surface waters range over many orders of magnitude especially if you consider that mountain streams can carry large rocks. But removing rocks from water is easily accomplished, gravity will take care of it for us. Gravity is such a great force for separation of particles from water that we would like to use it to remove small particles too. Unfortunately, gravity becomes rather ineffective at separating pathogens and small inorganic particles such as clay. The terminal velocities ((3.1)) of these particles is given in Fig. 3.7.

The low terminal velocities of particles that we need to remove from surface waters reveals that sedimentation alone will not work. The time required for a small particle to settle even a few mm would require excessively large sedimentation tanks. This is why flocculation, the process of sticking particles together so that they can attain higher sedimentation velocities, is perhaps the most important unit process in surface water treatment plants.

The AguaClara treatment train consists of the following processes

- flow measurement
- metering of the coagulant (and chlorine) that will cause particles to stick together
- mixing of the coagulant with the raw water
- flocculation where the water is deformed to cause particle collisions
- floc blanket where large flocs settle through water that is flowing upward causing collisions between small particles carried by the upward flowing water and the large flocs

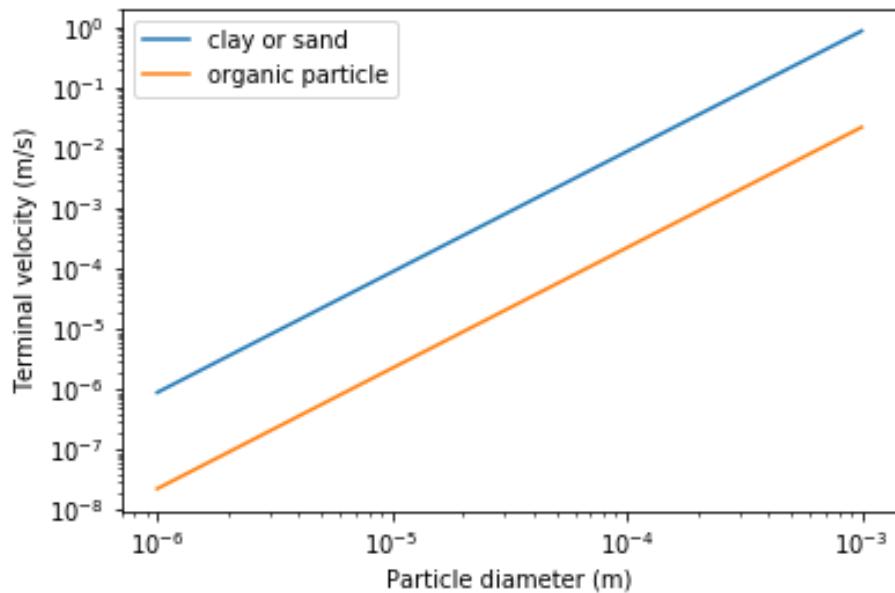


Fig. 3.7: The terminal velocity of a $1 \mu\text{m}$ bacteria cell is approximately 20 nanometers per second. The terminal velocity of a $5 \mu\text{m}$ clay particles is $30 \mu\text{m}/\text{s}$. The velocity estimates for the faster settling particles may be too slow because those particles are transitioning to turbulent flow.

- lamellar sedimentation where gravity causes particles to settle to an inclined plate and then slide back down into the floc blanket
- stacked rapid sand filtration where particles collide with previously deposited particles in a sand filter bed
- disinfection with chlorine to inactivate any pathogens that escaped the previous unit processes

As AquaClara technologies extend to larger and larger cities one of the criticisms could be that the technologies are somehow limited to small scale facilities. To address this question we will compare AquaClara unit processes with one of the most recent large scale water treatment plants, the [Croton Water Treatment Plant \(CWTP\)](#) in NYC.

The CWTP is designed to treat [290 mgd](#) (million gallons per day) which is equivalent to 12,700 L/s. The final cost of the project was \$3.2bn. The cost per L/s of treatment capacity was thus \$250,000. This is approximately 25 times more expensive than AquaClara water treatments. Of course, AquaClara water treatment plants havent been constructed underground in the middle of a major city! Nonetheless, the factor of 25 suggests that AquaClara technologies have a significant cost advantage.

The CWTP has 48 flocculators and 48 dissolved air flotation processes working in parallel. The flow per unit is thus 265 L/s. The current maximum size of the AquaClara Open Stacked Rapid Sand (OStaR) filter is 20 L/s. It would be possible to design larger OStaR filters by simply including multiple sets of inlet/outlet trunk lines into a single filter box. The CWTP filters appear to have 6 outlet trunk lines per filter and thus the flow per trunk line is 44 L/s.

The CWTP uses 2 stage mechanical flocculators with a total residence time of 4.8 minutes and a velocity gradient of 100 Hz. This residence time is much shorter than conventional design requirements, about half of the residence time used by the AquaClara plants built around 2017, significantly larger than the 90 second residence time used in the AquaClara 1 L/s plants.

CWTP uses dissolved air flotation tanks that are located on top of the rapid sand filters. The d

The filter approach velocity (the velocity of water before it enters the sand bed) for CWTP is 4.42 mm/s. This is significantly higher than the 1.85 mm/s filtration velocity currently used in StaRS filters. StaRS filters are a stack of 6 filters and the net filtration velocity is 11 mm/s. Thus by that metric the StaRS filters are significantly smaller than the CWTP filters.

```
#the unit registry has been imported above and does not need to be imported again
import aquaclarar
import aquaclarar.core.physchem as pc
from aquaclarar.core.units import unit_registry as u
Q_Croton = (290 * u.Mgal/u.day).to(u.L/u.s)
Cost_Croton = 3.2 * 10**9 * u.USD
Cost_per_Lps = Cost_Croton/Q_Croton
Cost_per_Lps
N_DAF = 48
Q_per_unit = Q_Croton/N_DAF
Q_per_unit/6
(15.9 * u.m/u.hr).to(u.mm/u.s)
```

3.4.1 Design Evolution

During the later half of the 20th century surface water treatment technologies evolved slowly. The slow evolution was likely a product of the regulatory environment, the high cost of water treatment infrastructure, and the low profit margin. The high cost of municipal scale water treatment infrastructure made experiments at scale infeasible and thus there was no mechanism to introduce disruptive innovations. With little opportunity for a significant return on investment there was little incentive to invest in the research and development that could have advanced the technologies. A final disincentive was the widely held belief that surface water treatment was a mature field with little opportunity for significant advancement. The advances of the latter half of the 20th century focused primarily on mechanization and automation (Supervisory Control and Data Acquisition - SCADA).

Design standards such as the [Great Lakes - Upper Mississippi River Board 10 States Standards](<http://10statesstandards.com/>) are evolving very slowly and retain an empirical approach to design. The empirical design methodology is a direct result of two confounding factors. The physics of particle interactions based on diffusion, fluid shear, and gravity are complex and given the challenges of characterizing surface water particle suspensions it was natural to assume that a mathematical description of the processes would be intractable.

Mechanized and automated water treatment plants performed reasonably well in communities with ready access to technical support services and supply chains that could reliably deliver replacement parts. In the global south municipal water treatment plants havent fared as well. In 2012, one of the main water treatment plants serving Kathmandu, Nepal had failed chlorine pumps and were using a red garden hose to siphon chlorine from the stock tank. They crimped the end of the hose to control the flow rate of the chlorine solution.

The ingenious and simple chemical dosing system that uses a siphon to completely bypass the failed pumps begs the question of whether design engineers could have invented a better option than the short lived pumps that they specified. We will investigate a gravity powered chemical dosing system that is far more reliable than chemical dosing pumps and that borrows from the simplicity of the garden hose solution used by the Nepali plant operators.

Chemical dosing systems are particularly vulnerable and their failures make plant operation very challenging. Providing the right coagulant dose is critical for efficient removal of particle and dissolved organics. Chemical dosing systems commonly rely on pumps and those pumps require regular maintenance and have relatively short mean times between failures.

The AquaClara Cornell program was founded in 2005 with the goal of creating a new generation of sustainable technologies that would perform well even in the rugged settings of rural communities. The goal wasnt simply to create technologies that would work for communities with very limited resources. The goal was to create the next generation of technologies that would both perform well in communities with limited resources and would be the highest performing technologies on multiple metrics for all communities.



Fig. 3.8: Failed chlorine dosing system bypassed with a red tube that siphons the chlorine solution at a plant in Kathmandu, Nepal in 2012.

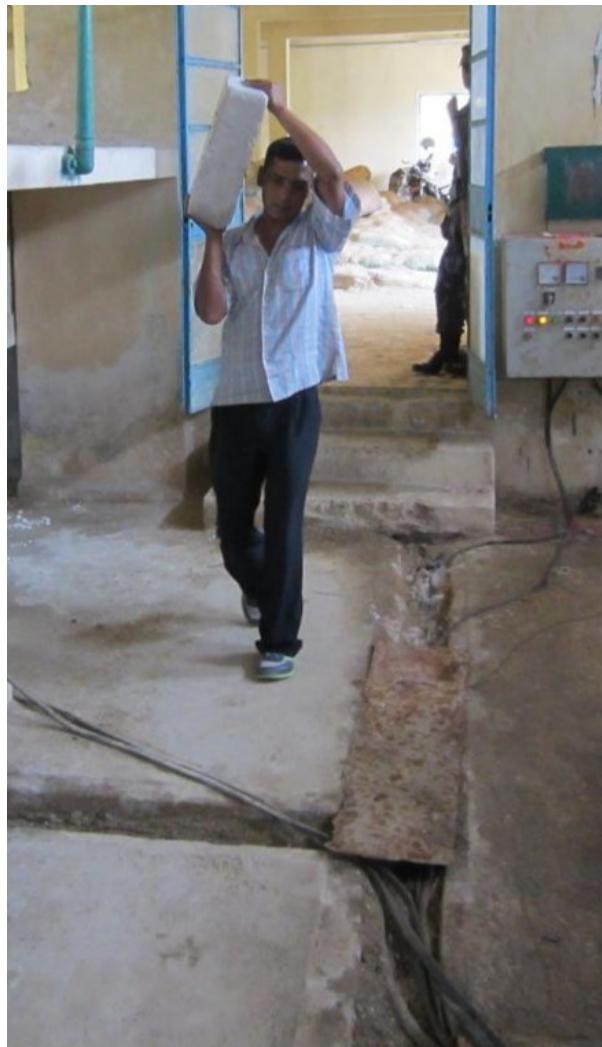


Fig. 3.9: Alum dosing system based on the rate that 25 kg blocks of alum are placed in the inlet channel of the plant.

3.4.2 Empirical Design

For the past several decades surface water treatment technologies have been considered mature and when I (Monroe) took a design course on drinking water treatment in 1985 I had the impression that there was little room for further innovation. This perspective is remarkable given that with the exception of lamellar sedimentation there were no equations describing the core treatment processes.

Empirical design guidelines dont provide insight into how designs could be optimized or even what the performance of a water treatment plant will be.

3.5 Design for the Financers, Venders, Client, or Context?

Tours of water treatment plants suggest that it is common for designs to be driven by the vender goal of a stable revenue stream for replacement parts rather than by a goal of meeting the clients needs. Mandatory software upgrades, mechanical valves, chemical pumps, mixing units provide a steady demand for proprietary components. Financers often prefer projects that can be implemented quickly either because they have target expenditures for a fiscal year or because loan repayment begins when the facility is turned over to the client.

Design for the client would strive to reduce capital, operating, and maintenance expenses. Clients also place a high value on reliability, ease of maintenance, and the ability to handle repairs with their staff. Design for the context would account for the capabilities of local and national supply chains. A key design consideration is to ensure that the treatment capabilities of the treatment plant match the variable water quality of the proposed water source. There are numerous slow sand filtration plants installed in the global south that are attempting to treat water sources that can not be effectively treated by slow sand filtration. The cost of the failure to consider the client and the context is born by the communities who end up with water treatment systems that arent able to provide reliable safe water.

Design for the client requires empathy and a commitment to listen to and learn from plant operators. It also requires attention to detail and watching how plant operators interact with water treatment plants. Empathy leads to the goal of creating a work environment that makes it easy for the plant operators to do their routine tasks. This isnt just to make the plant operators work easy. A plant that is designed with the plant operator in mind will also engender pride and that pride will lead to better plant performance.

An example of design for the operator is the elevation of the walkways in AquaClara plants. Conventional plants often have walkways that are above the tanks. That places the operators eyes several meters above the water surface and makes it difficult to see particles and flocs in the water. AquaClara plants have the walkways approximately 50 cm below the top of the tanks. This makes it easy for the plant operator to look into the tanks for quick visual inspections.

3.6 Design Bifurcations

Seemingly small decisions can have a profound effect on the evolution of design. Often these decisions have a clear logic and a simple analysis would suggest that the decision must be the right one. It is common for design choices to have multiple consequences that can turn a seemingly great choice into a poor performer.

3.6.1 Walls and a Roof

Traditionally in tropical and temperate climates, flocculation and sedimentation units are built without an enclosing building because they arent in danger of freezing. Without protection from the sun the materials used for plant construction must be UV resistant and thus plastic cant be used. This requires use of heavier and more expensive materials such stainless steel and aluminum. Metal plate settlers are heavy and thus they cant be easily removed by the plant operator.



Fig. 3.10: A plant operator built a makeshift ladder to enable easier access to the flocculation and sedimentation tanks in a package plant. This ladder considerably shortened the distance between the coagulant dose controls and the flocculator. The ladder also makes it possible to look closely at the water to see the size of the flocs.

Without the ability to gain access to a sedimentation tank from above, conventional sedimentation tank cleaning must be done by providing operator access below the plate settlers. This in turn requires that the space below the plate settlers be tall enough to accommodate a plant operator. Thus sedimentation tanks that are built in the open have to be deeper than sedimentation tanks that are built under a roof and they are more difficult to maintain because the operator has to enter the tank through a waterproof access port. Operator access to the space below the stainless steel or aluminum plate settlers is through a port in the side of the tank (see the video Fig. 3.11).



Fig. 3.11: Plant operators opening hatch below plate settlers in a traditional sedimentation tank.

AquaClara sedimentation tanks are designed to be taken off line one at a time so the water treatment plant can continue to operate during maintenance. Two plant operators can quickly open a sedimentation tank by removing the plastic plate settlers (see the video Fig. 3.12). The zero settled sludge design of the AquaClara sedimentation tanks also reduces the need for cleaning.

There is another major consequence of building water treatment plants in a secure enclosed building. Many water treatment plants will operate around the clock and that requires plant operators to spend the night at the facility. Having a secure facility provides improved safety for the plant operator. That improved safety is very important for some potential operators and thus providing that safety will increase potential diversity.



Fig. 3.12: Plant operator removing plate settlers from an AquaClara sedimentation tank.

3.6.2 Mechanized or Smart Hydraulics

Dramatically different designs are also created when we choose gravity power and smart hydraulics rather than mechanical mixers, pumps, and mechanical controls for each of the unit processes. It appears that use of electricity in drinking water treatment plants became the popular choice about 100 years ago. Many gravity powered plants have been converted to use mechanical mixers for rapid mix and flocculation. That choice may not have been well founded from a water quality or performance perspective.

Automated plants often move the controls far away from the critical observation locations in the plant. This might be appropriate or necessary in some cases, but it has the disadvantage of making it more difficult for operators to directly observe what is happening in the plant. Direct observations are critical because even highly mechanized water treatment plants are not yet equipped with enough sensors to enable rapid troubleshooting from the control room.

AquaClara plants have a layout that places the coagulant dose controls within a few steps of the best places to observe floc formation in the flocculator. This provides plant operators with rapid feedback that is critical when the raw water changes rapidly at the beginning of a high runoff event. As operators spend time observing the processes in the plant they begin to associate cause and effect and can make operational changes to improve performance. For example, gas bubbles that carry flocs to the surface can indicate sludge accumulation in a sedimentation tank. Rising flocs without gas bubbles can indicate a poor inlet flow distribution for a sedimentation tank or density differences caused by temperature differences.

3.7 AquaClara Inventions

3.7.1 Plant layout

1. Compact layout with processes sharing common walls when possible
2. Walkways set at optimal elevation for observation and maintenance of processes
3. Open tanks used whenever possible to simplify maintenance
4. Building enclosure to protect the entire plant from UV and for security

3.7.2 Chemical dosing

1. Linear flow orifice meter to both measure the plant flow rate and to turn the entrance tank water surface into a flow sensor input for the chemical dosing system.
2. Gravity powered semi-automated dosing system that delivers a constant dose even when plant flow rate changes.
3. Slider on a calibrated scale for intuitive changes in chemical dose

3.7.3 Rapid mix

1. Simple orifice for hydraulic rapid mix

3.7.4 Flocculation

1. Obstacles between baffles to create a more uniform distribution of energy dissipation rate and a more efficient use of available energy
2. Plastic modules that can easily be removed from channels for maintenance
3. Compact vertical flow flocculators for low flow plants

3.7.5 Sedimentation

1. Four channel inlet/outlet system that enables
 - (a) dumping flocculated water that doesn't meet specifications
 - (b) taking one sedimentation tank offline by placing a pipe stub in the inlet and a cap on the outlet
 - (c) dumping settled water that doesn't meet specifications
2. Inlet manifold with flow diffusers that straighten the flow into a continuous line jet
3. Inlet manifold is offset from center to force jet to all go in a consistent direction through the jet reverser
4. Jet reverser that efficiently reverses the direction of the incoming water to be able to resuspend settled flocs that are sliding down the inclines
5. Zero settled sludge in the main part of the sedimentation tank
6. Hydraulically cleaned sedimentation tank with no moving parts
7. Floc Hopper that consolidates the floc slurry prior to draining.
8. Floc blanket that is stable due to loss to or gain from settled sludge

3.7.6 Filtration

1. Sand drain system to empty sand from filter hydraulically
2. Wing and orifice system to inject water into the filter bed
3. Stacked Rapid Sand Filtration system that has the same flow rate for filtration and for backwash
4. Uses settled water for backwash to eliminate need for pumps and clearwells and to eliminate failure mode of inadequate supply of filtered water for backwash.
5. Air valve control system to trigger mode change from backwash to filtration and from filtration to backwash

6. No valves needed on inlet and outlet pipes

INTRODUCTION DESIGN CHALLENGE

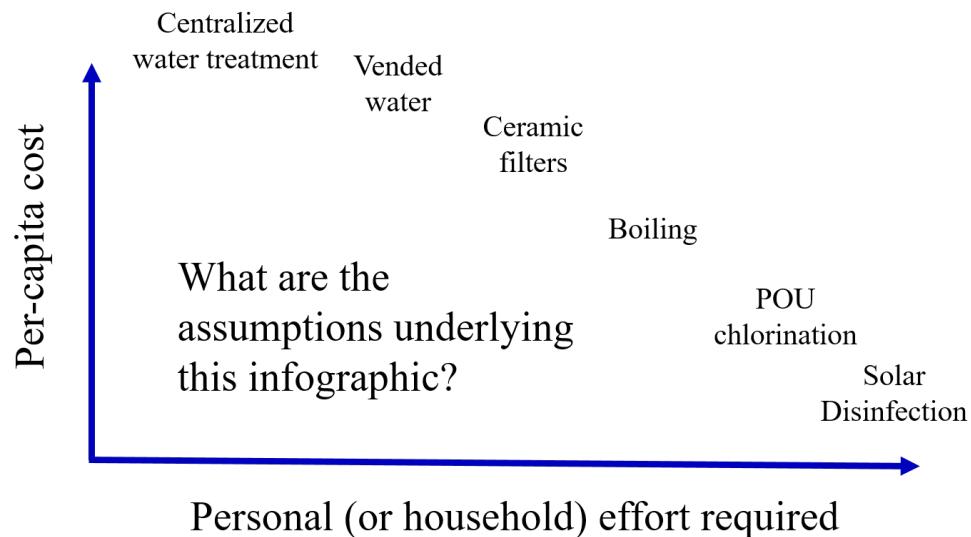


Fig. 4.1: Comparison of water treatment alternatives with a focus on household options.

4.1 1)

What is the cost of the chemical used in treating 1,000,000 liters of water using an AguaClara plant that uses a chemical coagulant (Poly aluminum chloride or PACl - 6 mg/L as aluminum) and chlorine (calcium hypochlorite - 1 mg/L as Cl_2)? The major operation and maintenance costs for an AguaClara surface water treatment plant are the chemicals and labor costs for the plant operators. You may assume the cost of the PACl is 1 USD/kg of aluminum and the cost of calcium hypochlorite is 1.5 USD per kg of equivalent Cl_2 .

Table 4.1: Overview of three AguaClara facilities that all began operation in 2015 and 2016

Town	Jesus de Otoro	Moroceli	San Matias
Year of Construction	2015	2014	2015
Current population	4,675	5,775	3,850
Design flow (L/s)	20	16	14
Current average Flow (L/s)	14	13	9
Current per capita consumption (L/d)	259	194	202
Capital Cost (USD)	\$233,453	\$200,883	\$191,248
Annual Chemical Cost (USD)	\$10,332	\$6,814	\$7,675
Chemical cost per ML	\$23	\$17	\$27
Annual Plant operator wages (USD)	\$5,412	\$27,245	\$8,118
Annual tariff income (USD)	\$30,529	\$51,660	\$34,440
Annual household water tariff (USD)	\$35.92	\$49.20	\$49.20

4.2 2)

Labor costs for AguaClara plants are highly variable, but a recent study (see Table 4.1) found a range of total annual wages for the plant operators. Use the data for Moroceli to estimate the labor cost per million liters produced. Note that high flow rate plants would have lower labor costs because a single operator can easily manage a much larger water treatment plant.

4.3 3)

Given the labor costs and chemical costs for the Moroceli AguaClara plant to produce 1,000,000 liters of water, how much water could be boiled for that same cost? You may assume that electrical heat is used to boil the water. Make reasonable assumptions as needed.

CHAPTER**FIVE**

PYTHON TUTORIAL

Table 5.1: Code resources.

Description	Theme	Example function	Function call
Python functions	low level programming functions	open a file	open(file, mode=r)
Pint universal constants	physical constants with units	Avogadros number	u.avogadro_number
Pint units	units that can be attached to numbers and numpy arrays	5 m/s	5*u.mm/u.s
aguaclara.core. physchem	pipeflow, orifices, viscosity of water, weirs, manifolds, Kozeny equation	total head loss in a straight pipe	headloss(FlowRate, Diam, Length, Nu, PipeRough, KMinor)
aguaclara.research. floc_model	AguaClara flocculation model, velocity gradients, Kolmogorov length scales	velocity gradient in a coiled tube	g_coil(FlowPlant, ID_Tube, RadiusCoil, Temp)
aguaclara.research. environmental_processes_analysis	carbonate system, CMFR and advection dispersion reactor analysis	extract data from ProCoDA generated Gran analysis file	Gran(data_file_path)
aguaclara.research. procoda_parser	Extracts data from multiple ProCoDA files based on the state and data column	extract a column of data from a ProCoDA data file	column_of_data(data_file_path, start, column)
SciPy	computing integrals numerically, solving differential equations, optimization, and sparse matrices	root finding	root(func, 0.3)
NumPy	Array manipulation and math functions	create an array with linearly spaced elements	np.linspace(start, stop, num)
PyPlot	Graphs!	Create beautiful graphs	see below

5.1 Import statements

```
import aquaclara
import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.core.constants as con
import aquaclara.research.environmental_processes_analysis as epa
import aquaclara.research.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy import constants, interpolate
```

Hint: If you are typing a function name and want to know what the options are for completing what you are typing, just hit the tab key for a menu of options.

Hint: If you want to see the source code associated with a function, you can do the following import inspect inspect.getsource(foo)

Where foo is the function that youd like to learn about.

5.2 Markdown

Markdown files allow you to mix code, beautiful Latex equations, nicely formatted text, figures, and tables. To create a markdown file in Atom simply create a new file (File menu) and give it a .md file extension. The .md file extension is critical so that atom knows how to interpret what you are typing!

Markdown allows you to create very nicely formatted text including Latex equations which will show up in the preview pane of Atom (type control - shift - E to display the preview pane of a Markdown file).

$$c = \sqrt{a^2 + b^2} \quad (5.1)$$

Markdown does not handle automatic numbering of equations, figures, and tables.

The Python Kernel remembers all definitions (functions and variables) as they are defined based on execution. Thus if you fail to execute a line of code, the parameters defined in that line wont be available. Similarly, if you define a parameter and then delete that line of code, that parameter remains defined until you restart Atom.

Before submitting a file for others to use, you need to verify that all of the dependencies are defined and that you didnt accidentally delete a definition that is required. You can do this by placing your cursor in a python code section and then clicking on the python 3idle in the toolbar at the bottom of the Atom window. Select restart Python 3 kernel. Then execute all of the python code in your document from top to bottom and make sure that all of the code performs as expected.

5.3 Transitioning From Matlab To Python

Indentation - When writing functions or using statements, Python recognizes code blocks from the way they are indented. A code block is a group of statements that, together, perform a task. A block begins with a header that is followed by one or more statements that are indented with respect to the header. The indentation indicates to the Python interpreter, and to programmers that are reading the code, that the indented statements and the preceding header form a code block.

Suppressing Statements - Unlike Matlab, you do not need a semi-colon to suppress a statement in Python;

Indexing - Matlab starts at index 1 whereas Python starts at index 0.

Functions - In Matlab, functions are written by invoking the keyword function, the return parameter(s), the equal to sign, the function name and the input parameters. A function is terminated with end.:

```
function
y = average(x)
if ~isvector(x)
    error('Input must be a vector') end
y = sum(x)/length(x);
end
```

In Python, functions can be written by using the keyword def, followed by the function name and then the input parameters in parenthesis followed by a colon. A function is terminated with return.:

```
def average(x):
if ~isvector(x):
    raise VocationError("Input must be a vector")
return sum(x)/length(x)
```

Statements - for loops and if statements do not require the keyword end in Python. The loop header in Matlab varies from that of Python. Check examples below:

Matlab code:

```
s = 10;
H = zeros(s);
for c = 1:s
    for r = 1:s
        H(r,c) = 1/(r+c-1);
    end
end
```

Printing - Use print() in Python instead of disp in Matlab.

Helpful Documents

- Numpy for Matlab Users
- Stepping from Matlab to Python
- Python for Matlab Users, UC Boulder

5.4 Useful hotkeys for executing code

Hydrogen hotkeys are listed under Packages->Hydrogen, and are also detailed here.

- Ctrl+Enter runs the currently selected line of code.
- Shift+Enter works as Ctrl+Enter and also moves the selection to the next line of code.
- Alt+Ctrl+Enter runs the current code block.
- Alt+Shift+Enter runs the current code block and moves the selection to the next block.

5.5 Units

Engineering requires calculations with units. Prior to modern computer languages engineers used paper and pencil, slide rules, calculators, and more recently spreadsheets to do calculations. All of these methods are prone to calculation errors because units aren't handled as an essential part of each value. Spreadsheets are especially notorious for calculation errors because unit conversions are buried in formulas that are hidden in the cells.

Operations on values with units follow very clear algebraic rules and thus units can be attached to numerical values and carried through math operations. This capability is implemented in Python using [Pint](#). The Pint package includes a host of units and prefixes (such as μ for 10^{-6}). As you master using Python and Pint you will say goodbye to mindless unit conversions forever!

Environmental engineers historically described surface loading rates for sedimentation tanks using units of gal/min per square foot. How fast is $\frac{1\text{gpm}}{\text{ft}^2}$ in $\frac{\text{mm}}{\text{s}}$?

```
V_surface_loading_rate = (1 * u.gal/(u.min * u.ft**2)).to(u.mm/u.s)
print('The surface loading rate is', V_surface_loading_rate)
aguaclara.core.units.set_sig_figs(3)
print('The surface loading rate is', V_surface_loading_rate)
```

The surface loading rate is 0.6791 millimeter / second

After reducing the number of significant digits to 3 we obtain: The surface loading rate is 0.679 millimeter / second

How long does it take to stop a car that is initially traveling at 60 mph if the coefficient of friction is 0.5?

```
v_0 = 60 * u.mile/u.hr
friction_coefficient = 0.5
deceleration = friction_coefficient * u.standard_gravity
t_deceleration = v_0/deceleration
print('The time to stop the car is', t_deceleration)
print('The time to stop the car is', t_deceleration.to_base_units())
```

The time to stop the car is 120 mile / hour / standard_gravity

We add the `.to_base_units()` directive to force pint to simplify the units.

The time to stop the car is 5.47 second

Many functions written in Python do not yet handle units and thus it is sometimes necessary to remove the units. Examples include graphs (although units might be coming to matplotlib), SciPy functions, and the NumPy functions used to populate arrays. For these cases you can strip the units off a number using the `.magnitude` method. Be careful to make sure you know what the units are before you remove them otherwise you may be confused by the results!

```
Q = 5 * u.gal/u.min
fill_time = 3*u.hr
Volume = Q * fill_time
print('The volume is', Volume)
print('The magnitude of the Volume is', Volume.magnitude)
print('The units of the flow are', Volume.units)
#force pint to display in a selected set of Units
print('The volume is', Volume.to(u.kL))
```

The volume is 15 gallon * hour / minute

The magnitude of the Volume is 15.0

The units of the flow are gallon * hour / minute

The volume is 3.41 kiloliter

It is useful to force pint to display the result in the units of your choice.

5.6 Arrays and units

Use NumPy arrays rather than Python lists to enable math with numbers and units. When creating arrays with units remember that

- Array elements dont have units!
- Arrays can have units.

Therefore always attach units to the array after the array has been created. This means that array elements should be dimensionless and thus arrays must be created using dimensionless values.

We can use NumPy linspace with a simple change to make it dimensionless. Usually linspace has start and stop elements that would logically have units: np.linspace(start, stop, num). But elements cant have units! We can make the inputs to linspace be dimensionless to create a dimensionless array and then multiplies it by the final value that includes the units to scale the array correctly. For evenly spaced arrays starting at the end of the first space we have either:

- np.linspace(start/stop, 1, num) * stop
- np.linspace(1 / num, 1, num) * stop

For evenly spaced arrays starting with zero we have:

- np.linspace(0, 1, num+1) * stop!

The print function cant currently handle arrays with units. The array can be printed nicely in two steps as shown below.

```
n_rows = 10
Flow = 20 * u.L/u.s
Flow_array = (np.linspace(1 / n_rows, 1, n_rows) * Flow)
print('The array of flow rates is',Flow_array.magnitude,Flow_array.units)

Flow_array = (np.linspace(1 / n_rows, 1, n_rows) * Flow).to(u.L/u.s)
print('The array of flow rates is',Flow_array.magnitude,Flow_array.units)

Flow_array = (np.linspace(0, 1, n_rows+1) * Flow).to(u.L/u.s)
print('The array of flow rates is',Flow_array.magnitude,Flow_array.units)
```

[2. 4. 6. 8. 10. 12. 14. 16. 18. 20.] liter / second

5.7 Plotting

We will use this `pyplot` coding style .

- `fig` is a `Figure` instance like a blank canvas
- `ax` is an `AxesSubplot` instance think of a frame for plotting in

Create a graph showing flow rate vs depth for two linear flow orifice meters that have a depth range of 20 cm and flow ranges of 50 and 70 L/s.

- `Legend location options`
- `Colors`
 - b, g, r, c, m, y, k, w

- blue, green, red, cyan, magenta, yellow, black, white
- Data markers (if you are plotting data and not a model or curve fit)

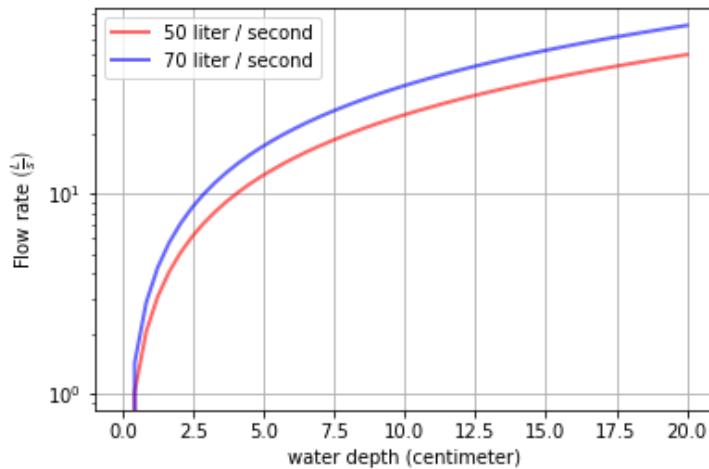


Fig. 5.1: The flow through an LFOM is directly proportional to the height of the water above the bottom of the first row of orifices.

Indexing is done by row and then by column. To call all of the elements in a row or column, use a colon. As you can see in the following example, indexing in python begins at zero. So `a[:, 1]` is calling all rows in the second column

5.8 Example problem

Calculate the number of moles of methane in a 20 L container at 15 psi above atmospheric pressure with a temperature of 30 C.

```
# First assign the values given in the problem to variables.
P = 15 * u.psi + 1 * u.atm
T = 30 * u.degC
V = 20 * u.L
# Use the equation PV=nRT and solve for n, the number of moles.
# The universal gas constant is available in pint.
nmolesmethane = (P*V/(u.R*T.to(u.kelvin))).to_base_units()
print(nmolesmethane)
print('There are ', nmolesmethane, ' of methane in the container.')
```

There are 1.625 mole of methane in the container.

5.9 Functions

When it becomes necessary to do the same calculation multiple times, it is useful to create a function to facilitate the calculation in the future.

- Function blocks begin with the keyword def followed by the function name and parentheses ().
- Any input parameters or arguments should be placed within these parentheses.

- The code block within every function starts with a colon (:) and is indented.
- The statement return [expression] exits a function and returns an expression to the user. A return statement with no arguments is the same as return None.
- (Optional) The first statement of a function can be the documentation string of the function or docstring, written with apostrophes .

Below is an example of a function that takes three inputs, pressure, volume, and temperature, and returns the number of moles.

```
# Creating a function is easy in Python
def nmoles(P, V, T):
    return (P*V / (u.R*T.to(u.kelvin))).to_base_units()
```

Try using the new function to solve the same problem as above. You can reuse the variables. You can use the new function call inside the print statement.

```
print('There are', nmoles(P,V,T), 'of methane in the container.')
```

```
There are 1.62 mol of methane in the container.
```

5.10 Pipe Database

The `pipes` has many useful functions concerning pipe sizing. It provides functions that calculate actual pipe inner and outer diameters given the nominal diameter of the pipe. Note that nominal diameter just means the diameter that it is called (hence the descriptor nominal) and thus a 1 inch nominal diameter pipe might not have any dimensions that are actually 1 inch!

```
import aquaclara.core.pipes as pipe
# The OD function in pipedatabase returns the outer diameter of a pipe given the
# nominal diameter, ND.
pipe.OD(6*u.inch)
```

6.625 inch

The `ND_SDR_available` function returns the nominal diameter of a pipe that has an inner diameter equal to or greater than the requested inner diameter `SDR`, standard diameter ratio . Below we find the smallest available pipe that has an inner diameter of at least 7 cm

```
IDmin = 7 * u.cm
SDR = 26
ND_my_pipe = pipe.ND_SDR_available(IDmin, SDR)
ND_my_pipe
```

3.0 inch

The actual inner diameter of this pipe is

```
ID_my_pipe = pipe.ID_SDR(ND_my_pipe, SDR)
print(ID_my_pipe.to(u.cm))
```

```
8.2 cm
```

We can display the available nominal pipe sizes that are in our database.

```
pipe.ND_all_available()
```

CHAPTER
SIX

PYTHON DESIGN CHALLENGE

6.1 1)

Calculate the minimum inner diameter of a PVC pipe that can carry a flow of at least 10 L/s for the town of Ojojona. The population is 4000 people. The water source is a dam with a surface elevation of 1500 m. The pipeline connects the reservoir to the discharge into a distribution tank at an elevation of 1440 m. The pipeline length is 2.5 km. The pipeline is made with PVC pipe with an SDR (standard diameter ratio) of 26.

The pipeline inlet at the dam is a square edge with a minor loss coefficient (K_e) of 0.5. The discharge at the top of the distribution tank results in a loss of all of the kinetic energy and thus the exit minor loss coefficient is 1. See the minor loss equation below.

$$h_e = K_e \frac{V^2}{2g}$$

The water temperature ranges from 10 to 30 Celsius. The roughness of a PVC pipe is approximately 0.1 mm. Use the fluids functions to calculate the minimum inner pipe diameter to carry this flow from the dam to the distribution tank.

Report the following

- critical design temperature
- kinematic viscosity (maximum viscosity will occur at the lowest temperature)
- the minimum inner pipe diameter (in mm).

Use complete sentences to report the results.

The critical water temperature for this design is 10 degC.
The kinematic viscosity of water is 1.3e-6 m²/s.
The minimum pipe inner diameter is 97 mm.

6.2 2)

Find the nominal diameter of a PVC pipe that is SDR 26. SDR means standard diameter ratio. The thickness of the pipe wall is 1/SDR of the outside diameter. The [pipes file](#) has a useful function that returns nominal diameter given SDR and inner diameter.

The nominal diameter of the pipeline is 4.0 in (1.0e+2 mm).

6.3 3)

What is the actual inner diameter of this pipe in mm? Compare this with the reported inner diameter for SDR-26 pipe to see if our pipe database is reporting the correct value.

The inner diameter of the pipe **is 106** mm.
Cresline reports the inner diameter **is 106** mm.

6.4 4)

What is the maximum flow rate that can be carried by this pipe at the coldest design temperature? Display the flow rate in L/s using the .to method.

The maximum flow rate at 10 celsius **is 13** l/s.

6.5 5)

What is the Reynolds number and friction factor for this maximum flow? Assign these values to variable names so you can plot them later on the Moody diagram.

The Reynolds number **and** friction factor **for** the pipeline flow are **1.2e+5 and 0.022** respectively.

6.6 6)

Check to see if the fluids functions are internally consistent by calculating the head loss given the flow rate that you calculated and comparing that head loss with the elevation difference. Display enough significant digits to see the difference in the two values. Note that the Moody diagram has an accuracy of about $\pm 5\%$ for smooth pipes and $\pm 10\%$ for rough pipes Moody, 1944 .

The head loss **is 60.5** m **and** that **is** close to the elevation difference of **60.0** m.

6.7 7)

How much more water (both volumetric and mass rate) will flow through the pipe at the maximum water temperature of 30 C? Take into account both the change in viscosity (changes the flow rate) and the change in density (changes the mass rate). Report the flow rates in L/s.

The increase **in** flow rate at 30 celsius **is 0.24** l/s.
The increase **in** mass rate at 30 celsius **is 0.19** kg/s.

6.8 8)

Why is the flow increase due to this temperature change so small given that viscosity actually changed significantly (see the calculation below)?

The viscosity ratio **for** the two temperatures was **0.62**.

The flow is turbulent and thus viscosity has little influence on the flow rate.

6.9 9)

Suppose an AquaClara plant is designed to be built up the hill from the distribution tank. The transmission line will need to be lengthened by 30 m and the elevation of the inlet to the entrance tank will be 1450 m. The rerouting will also require the addition of 3 elbows with a minor loss coefficient of 0.3 each. What is the new maximum flow from the water source?

The new maximum flow rate at **10** celsius **is** **12** l/s.

6.10 10)

How much less water will flow through the transmission line after the line is rerouted?

The reduction **in** flow **is** **1.3** l/s.

6.11 11)

There exists a function within physchem called `pc.fric(FlowRate, Diam, Nu, PipeRough)` that returns the friction factor for both laminar and turbulent flow. In this problem, you will be creating a new function which you shall call `f_of_Re()` that takes the Reynolds number and the dimensionless pipe roughness (ϵ/D) as inputs.

Recall that the format for defining a function is

```
def f_of_Re(input1, input2):
    f = buncha stuff
    return f
```

Since the equation for calculating the friction factor is different for laminar and turbulent flow (with the transition Reynolds number being defined within the physchem file), you will need to use an `if, else` statement for the two conditions.

6.12 12)

Create a beautiful Moody diagram. Include axes labels and show a legend that clearly describes each plot. The result should look like Fig. 7.6 below.

6.13 12a)

You will be creating a Moody diagram showing Reynolds number vs friction factor for multiple dimensionless pipe roughnesses. The first step to do this is to define the number of dimensionless pipe roughnesses you want to plot. We will plot 8 curves for the following values: 0, 0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03, 0.1. We will plot an additional

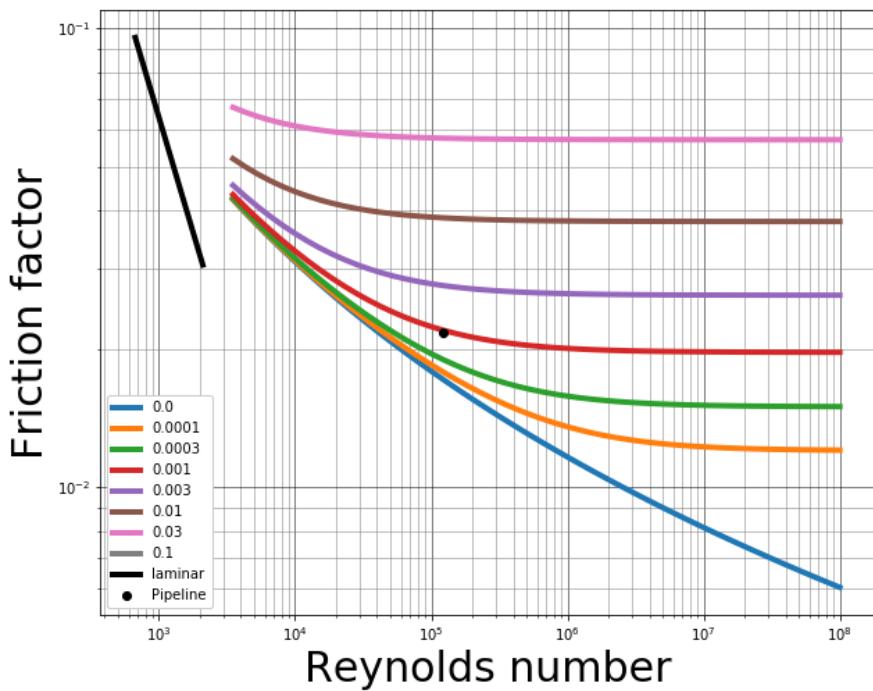


Fig. 6.1: Moody diagram describing fluid flow in straight pipes.

curve, which will be a straight line, for laminar flow, since it is not dependent on the pipe roughness value (see Fig. 7.6).

- Create an array for the dimensionless pipe roughness values, using `np.array([])`.
- Specify the amount of data points you want to plot for each curve. We will be using 50 points.

Because the Moody diagram is a log-log plot, we need to ensure that all 50 points on the diagram we are creating are equally spaced in log-space. Use the `np.logspace(input1, input2, input3)` function to create an array for turbulent Reynolds numbers and an array for laminar Reynolds numbers.

- `input1` is the exponent for the lower bound of the range. For example, if you want your lower bound to be 1000, your input should be `math.log10(1000)` which is equal to 3.
- `input2` is the exponent for the upper bound of the range. Format this input as you have formatted `input1`.
- `input3` is the number of data points you are using for each curve. We will be using 50 points.

12a) Deliverables

- Array of dimensionless pipe roughnesses. Call this array `eGraph`.
- Variable defining the amount of points on each pipe roughness curve * Two arrays created using `np.logspace` for turbulent and laminar Reynolds numbers, which will be the x-axis values for the Moody diagram

Note: The bounds for the laminar Reynolds numbers array should span between 670 and the predefined transition number used in Problem 11. The bounds for the turbulent Reynolds numbers array should span between 3,500 and 100,000,000. These ranges are chosen to make the curves fit well within the graph and to intentionally omit data in the transition range between laminar and turbulent flows.

6.14 12b)

Now you will create the y-axis values for turbulent flow (based on dimensionless pipe roughness) and laminar flow (not based on dimensionless pipe roughness). To do this, you will use the `fOfRe()` function you wrote in Problem 11 to find the friction factors.

Begin by creating an empty 2-dimensional array that will be populated by the turbulent-flow friction factors for each dimensionless pipe roughness. Use `np.zeros(number of rows, number of columns)`. The number of rows should be the number of dimensionless pipe roughness values (`len(eGraph)`), while the number of columns should be the number of data points per curve as defined above. Populating this array with friction factor values may require two `for` loops, one to iterate through rows and one to iterate through columns.

You will repeat this process to find the friction factors for laminar flow. The only difference between the turbulent and laminar friction flow arrays will be that the laminar array will only have one dimension since it does not affect by the dimensionless pipe roughness. Start by creating an empty 1-dimensional array and then use a single `for` loop.

12b) Deliverables

- One 1-D array containing friction factor values for laminar flow.
- One 2-D array containing friction factor values for each dimensionless pipe roughness for turbulent flow.

6.15 12c)

Now, we are ready to start making the Moody diagram! The plot formatting is included for you in the cell below. You will add to this cell the code that will actually plot the arrays you brought into existence in 12a) and 12b) with a legend. For the sake of your own sanity, please only add code where specified.

- First, plot your arrays. See the plots in the tutorial above for the syntax. Recall that each dimensionless pipe roughness is a separate row within the 2-D array you created. To plot these roughnesses as separate curves, use a `for` loop to iterate through the rows of your array. To plot all columns in a particular row, use the `[1, :]` call on an array, where 1 is the row you are calling.
- Plotting the laminar flow curve does not require a `for` loop because it is a 1-D array.
- Use a linewidth of 4 for all curves.
- Now plot the data point you calculated for the pipeline problem. Use the Reynolds number and friction factor obtained in Problem 5. Because this is a single point, it should be plotted as a circle instead of a line.
- You will need to make a legend for the graph using `ax.legend(stringarray, loc = 'best')`
- **The first input, `stringarray`, must be an array composed of strings instead of numbers. The array you created which contains the dimensionless pipe roughness values (`eGraph`) can be converted into a string array for your legend (`eGraph.astype('str')`). You will need to add Laminar and Pipeline as strings to the new `eGraph` string array. Perhaps you will find `np.append(basestring, ['string1', 'string2'])` to be useful.**

6.16 13)

What did you find most difficult about learning to use Python? Create a brief example as an extension to this tutorial to help students learn the topic that you found most difficult.

6.16.1 Final Pointer

Before submitting a file for others to use, you need to verify that all of the dependencies are defined and that you didn't accidentally delete a definition that is required. You can do this by placing your cursor in a python code section and then clicking on the python 3idle in the toolbar at the bottom of the Atom window. Select restart Python 3 kernel. Then execute all of the python code in your document from top to bottom and make sure that all of the code performs as expected.

CHAPTER
SEVEN

REVIEW: FLUID MECHANICS

This document is meant to be a refresher on fluid mechanics. It will only cover the topics in fluids mechanics that will be used heavily in the course.

If you wish to review fluid mechanics in (much) more detail, please refer to [this guide](#). Note that to view this link, you will need a Github account. If you wish to review from a legitimate textbook, you can find a pdf of good book by Frank White [here](#).

7.1 Important Terms and Equations

Terms:

1. *Laminar*
2. *Turbulent*
3. *Viscosity*
4. *Streamline*
5. *Control Volume*
6. *Head*
7. *Head loss*
8. *Driving head*
9. *Vena Contracta/Coefficient of Contraction*

Equations:

1. Continuity equation: (7.1)
2. Reynolds number (7.6)
3. Bernoulli equation (7.7)
4. Energy equation (7.9)
5. Darcy-Weisbach equation (7.10)
6. Swamee-Jain equation (7.13)
7. Hagen-Poiseuille equation (7.17)
8. Orifice equation (7.34)

7.2 Introductory Concepts

Before diving in to the rest of this document, there are a few important concepts to focus on which will be the foundation for building your understanding of fluid mechanics. One must walk before they can run, and similarly, the basics of fluid mechanics must be understood before moving on to the more fun (and exciting!) sections of this document.

7.2.1 Continuity Equation

Continuity is simply an application of mass balance to fluid mechanics. It states that the cross sectional area A that a fluid flows through multiplied by the fluids average flow velocity \bar{v} must equal the fluids flow rate Q :

$$Q = \bar{v}A \quad (7.1)$$

Note: The line above the v is called a bar, and represents an average. Any variable can have a bar. In this case, we are adding the bar to velocity v , turning it into average velocity \bar{v} . This variable is pronounced v bar.

In this course, we deal primarily with flow through pipes. For a circular pipe, $A = \pi r^2$. Substituting diameter in for radius, $r = \frac{D}{2}$, we get $A = \frac{\pi D^2}{4}$. You will often see this form of the continuity equation being used to relate the a pipes flow rate to its diameter and the velocity of the fluid flowing through it:

$$Q = \bar{v} \frac{\pi D^2}{4} \quad (7.2)$$

The continuity equation is also useful when flow is going from one geometry to another. In this case, the flow in one geometry must be the same as the flow in the other, $Q_1 = Q_2$, which yields the following equations:

$$\bar{v}_1 A_1 = \bar{v}_2 A_2 \quad (7.3)$$

$$\bar{v}_1 \frac{\pi D_1^2}{4} = \bar{v}_2 \frac{\pi D_2^2}{4} \quad (7.4)$$

Such that:

Q = fluid flow rate

\bar{v} = fluid average velocity

A = pipe area

r = pipe radius

D = pipe diameter

An example of changing flow geometries is when a change in pipe size occurs in a circular piping system, as is demonstrated below. The flow through pipe 1 must be the same as the flow through pipe 2.

7.2.2 Laminar and Turbulent Flow

Considering that this class deals with the flow of water through a water treatment plant, understanding the characteristics of the flow is very important. Thus, it is necessary to understand the most common characteristic of fluid flow: whether it is **laminar** or **turbulent**. Laminar flow is very smooth and highly ordered. Turbulent flow is chaotic, messy, and disordered. The best way to understand each flow and what it looks like is visually, like in the Wikipedia figure below or [in this video](#). Please ignore the part of the video after the image of the tap.

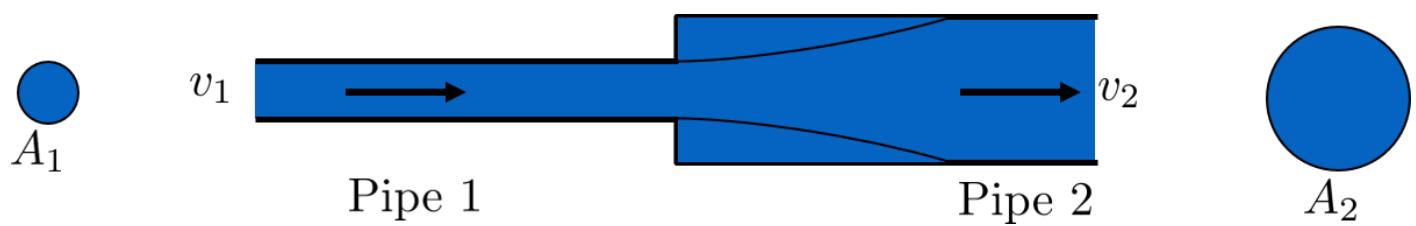


Fig. 7.1: Flow going from a small diameter pipe to a large one. The continuity principle states that the flow through each pipe must be the same.

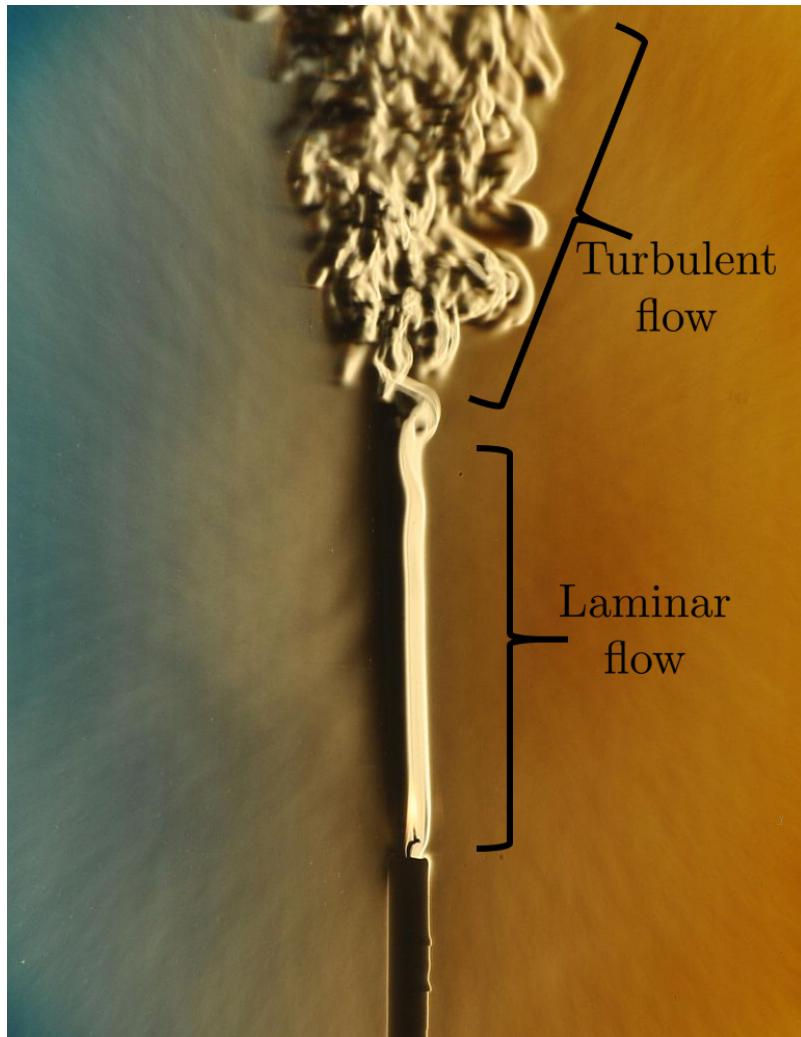


Fig. 7.2: This is a beautiful example of the difference between ordered and smooth laminar flow and chaotic turbulent flow.

A numeric way to determine whether flow is laminar or turbulent is by finding the Reynolds number, Re . The Reynolds number is a dimensionless parameter that compares inertia, represented by the average flow velocity \bar{v} times a length scale D to viscosity, represented by the kinematic viscosity ν . [Click here](#) for a brief video explanation of viscosity. If the Reynolds number is less than 2,100 the flow is considered laminar. If it is more than 2,100, it is considered turbulent.

$$\text{Re} = \frac{\text{inertia}}{\text{viscosity}} = \frac{\bar{v}D}{\nu} \quad (7.5)$$

The transition between laminar and turbulent flow is not yet well understood, which is why the concept of transitional flow is often simplified and neglected to make it possible to code for laminar or turbulent flow, which are better understood. We will assume that the transition occurs at $\text{Re} = 2100$. In aquaclara, this parameter shows us as `pc.RE_TRANSITION_PIPE`.

Fluid can flow through very many different geometries, like a pipe, a rectangular channel, or any other shape. To account for this, the characteristic length scale for the Reynolds number, which was written in the equation above as D , is quantified as the [hydraulic diameter](#), D_h when considering a general cross-sectional area. For circular pipes, which are the most common geometry you'll encounter in this class, the hydraulic diameter is simply the pipes diameter, $D_h = D$.

Here are other commonly used forms of the Reynolds number equation *for circular pipes*. They are the same as the one above, just with the substitutions $Q = \bar{v}\frac{\pi D^2}{4}$ and $\nu = \frac{\mu}{\rho}$

$$\text{Re} = \frac{\bar{v}D}{\nu} = \frac{4Q}{\pi D\nu} = \frac{\rho\bar{v}D}{\mu} \quad (7.6)$$

Such that:

Q = fluid flow rate in pipe

D = pipe diameter

\bar{v} = fluid velocity

ν = fluid kinematic viscosity

μ = fluid dynamic viscosity

See also:

Function in aquaclara: `pc.re_pipe(FlowRate, Diam, Nu)` Returns the Reynolds number *in a circular pipe*. Functions for finding the Reynolds number through other flow conduits and geometries can also be found in `physchem.py` within aquaclara.

Note: Definition of Flow Regimes: Laminar and turbulent flow are described as two different **flow regimes**. When there is a characteristic of flow and different categories of the characteristic, each category is referred to as a flow regime. For example, the Reynolds number describes a flow characteristic, and its categories, referred to as flow regimes, are laminar or turbulent.

7.2.3 Streamlines and Control Volumes

Both [streamlines](#) and [control volumes](#) are tools to compare different parts of a system. For this class, this system will always be hydraulic.

Imagine water flowing through a pipe. A streamline is the path that a particle would take if it could be placed in the fluid without changing the original flow of the fluid. A more technical definition is a line which is everywhere parallel to the local velocity vector. Computational tools, [dyes \(in water\)](#), or [smoke \(in air\)](#) can be used to visualize streamlines.

A **control volume** is just an imaginary 3-dimensional shape in space. Its boundaries can be placed anywhere by the person applying the control volume, and once set the boundaries remain fixed in space over time. These boundaries are usually chosen to compare two relevant surfaces to each other. These surfaces are called *Control Surfaces*. The entirety of a control volume is usually not shown, as it is often unnecessary. This is demonstrated in the following image:

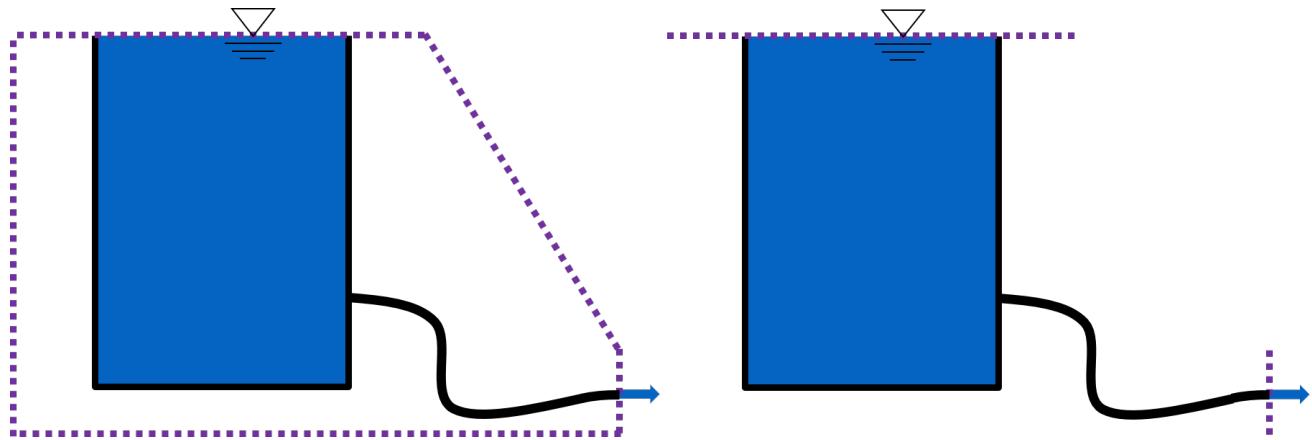


Fig. 7.3: While the image on the left indicates a complete control volume, control volumes are usually shortened to only include the relevant control surfaces, in which the control volume intersects the fluid. This is shown in the image on the right.

Important: Many images will be used over the course of this class to show hydraulic systems. A standardized system of lines will be used throughout them all to distinguish reference elevations from control volumes from streamlines. This system is described in the image below.

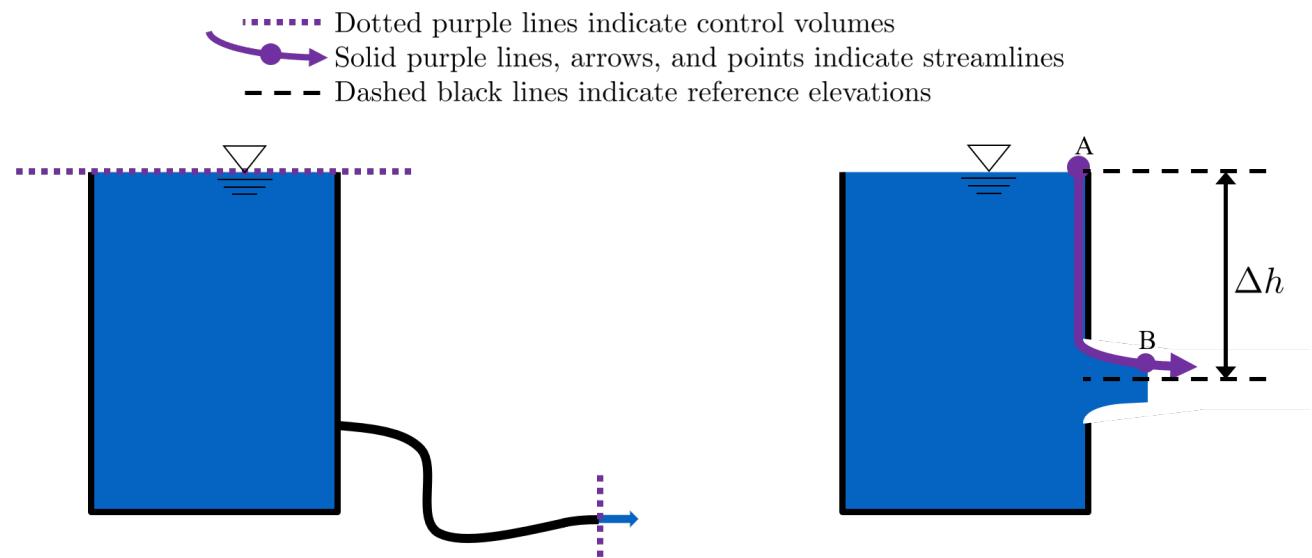


Fig. 7.4: On the right, a control volume is applied to a hydraulic system. On the left, a streamline is applied to a hydraulic system. A figure-convention for control volumes and streamlines will be very helpful throughout this course as there will be very, very many figures.

7.3 The Bernoulli and Energy Equations

As explained in almost every fluid mechanics class, the Bernoulli and energy equations are incredibly useful in understanding the transfer of the fluids energy throughout a streamline or through a control volume. The Bernoulli equation applies to two different points along one streamline, whereas the energy equation applies to fluid entering and exiting a control volume. The energy of a fluid has three forms: pressure, potential (deriving from elevation), and kinetic (deriving from velocity).

7.3.1 The Bernoulli Equation

These three forms of energy expressed above make up the Bernoulli equation:

$$\frac{p_1}{\rho g} + z_1 + \frac{v_1^2}{2g} = \frac{p_2}{\rho g} + z_2 + \frac{v_2^2}{2g} \quad (7.7)$$

Such that:

p = pressure

ρ = fluid density

g = acceleration due to gravity

z = elevation relative to a reference

v = fluid velocity

Notice that each term in this form of the Bernoulli equation has units of $[L]$, even though the terms represent the energy of the fluid, which has units of $\frac{[M] \cdot [L]^2}{[T]^2}$. When energy of the fluid is described in units of length, the term used is called **head** and referred to as h .

There are two important distinctions to keep in mind when using head to talk about a fluids energy. First is that head is dependent on the density of the fluid under consideration. Take mercury, for example, which is around 13.6 times more dense than water. 1 meter of mercury head is therefore equivalent to around 13.6 meters of water head. Second is that head is independent of the amount of fluid being considered, *as long as all the fluid is the same density*. Thus, raising 1 liter of water up by one meter and raising 100 liters of water up by one meter are both equivalent to giving the water 1 meter of water head, even though it requires 100 times more energy to raise the hundred liters than to raise the single liter. Since we are concerned mainly with water in this class, we will refer to water head simply as head.

Going back to the Bernoulli equation, the $\frac{p}{\rho g}$ term is called the pressure head, z is called the elevation head, and $\frac{v^2}{2g}$ is the velocity head. The following diagram shows these various forms of head via a 1 meter deep bucket (left) and a jet of water shooting out of the ground (right).

Assumption in using the Bernoulli equation

Though there are many assumptions needed to confirm that the Bernoulli equation can be used, the main one for the purpose of this class is that energy is not gained or lost throughout the streamline being considered. If we consider more precise fluid mechanics terminology, then friction by viscous forces must be negligible. What this means is that the fluid along the streamline being considered is not losing energy to viscosity. As a result, using the Bernoulli equation implies that energy cant be gained or lost. It can only be transferred between its three forms.

Example problems

Here is a simple worksheet with very straightforward example problems using the Bernoulli equation. Note that the solutions use the pressure-form of the Bernoulli equation. This just means that every term in the equation is multiplied by ρg , so the pressure term is just P . The form of the equation does not affect the solution to the problem it helps solved.

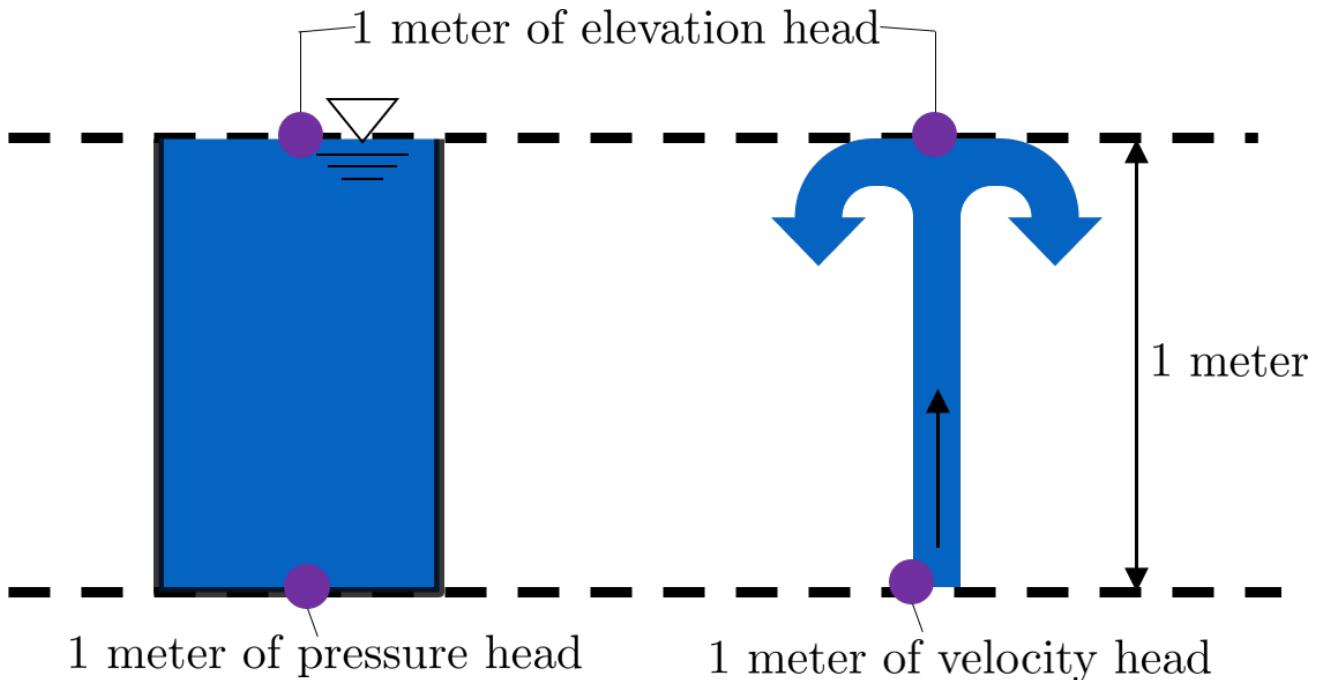


Fig. 7.5: The three forms of hydraulic head.

7.3.2 The Energy Equation

The assumption necessary to use the Bernoulli equation, which is stated above, represents the key difference between the Bernoulli equation and the energy equation for the purpose of this class. The energy equation accounts for the potential addition or loss of fluid energy within the control volume. (L)oss of energy is usually due to viscous friction resisting fluid flow, h_L , or the charging of a (T)urbine, h_T . The most common input of fluid energy into a system is usually caused by a (P)ump within the control volume, h_P .

$$\frac{p_1}{\rho g} + z_1 + \alpha_1 \frac{\bar{v}_1^2}{2g} + h_P = \frac{p_2}{\rho g} + z_2 + \alpha_2 \frac{\bar{v}_2^2}{2g} + h_T + h_L \quad (7.8)$$

You'll also notice the α term attached to the velocity head. This is a correction factor for kinetic energy, and will be neglected in this class; we assume that its value is 1. In the Bernoulli equation, the velocity of a streamline of the fluid is considered, v . The energy equation, however compares control surfaces instead of streamlines, and the velocities across a control surface may not all be the same. Hence, \bar{v} is used to represent the average velocity. Since AquaClara does not use pumps nor turbines, $h_P = h_T = 0$. With these simplifications, the energy equation can be written as follows:

$$\frac{p_1}{\rho g} + z_1 + \frac{\bar{v}_1^2}{2g} = \frac{p_2}{\rho g} + z_2 + \frac{\bar{v}_2^2}{2g} + h_L \quad (7.9)$$

This is the form of the energy equation that you will see over and over again in this book. To summarize, the main difference between the Bernoulli equation and the energy equation for the purposes of this class is energy loss. The energy equation accounts for the fluid's loss of energy over time while the Bernoulli equation does not. So how can the fluid lose energy?

7.4 Headloss

Head(L)oss, h_L is a term that is ubiquitous in both this class and fluid mechanics in general. Its definition is exactly as it sounds: it refers to the loss of energy of a fluid as it flows through space. There are two components to head loss:

major losses caused by (f)iction between the fluid the surface its flowing over, h_f , and minor losses caused by fluid-fluid internal friction resulting from flow (e)xpansions, h_e . These two components combine such that $h_L = h_f + h_e$.

7.4.1 Major Losses

These losses are the result of friction between the fluid and the surface over which the fluid is flowing. A force acting parallel to a surface is referred to as **shear**. It can therefore be said that major losses are the result of shear between the fluid and the surface its flowing over. To help in understanding major losses, consider the following example: imagine, as you have so often in physics class, pushing a large box across the ground. Friction is what resists your efforts to push the box. The farther you push the box, the more energy you expend pushing against friction. The same is true for water moving through a pipe, where water is analogous to the box you want to move, the pipe is similar to the floor that provides the friction, and the major losses of the water through the pipe is analogous to the energy **you** expend by pushing the box.

In this class, we will be dealing primarily with major losses in circular pipes, as opposed to channels or pipes with other geometries. Fortunately for us, Henry Darcy and Julius Weisbach came up with a handy equation to determine the major losses in a circular pipe *under both laminar and turbulent flow conditions*. Their equation is logically and unoriginally named the **Darcy-Weisbach equation**. It is shown below:

$$h_f = f \frac{L}{D} \frac{\bar{v}^2}{2g} \quad (7.10)$$

Substituting the continuity equation $Q = \bar{v}A$ in the form of $\bar{v}^2 = \frac{16Q^2}{\pi^2 D^4}$ gives another, equivalent form of Darcy-Weisbach which uses flow, Q , instead of velocity, \bar{v} :

$$h_f = f \frac{8}{g\pi^2} \frac{LQ^2}{D^5} \quad (7.11)$$

Such that:

h_f = major loss

f = Darcy friction factor

L = pipe length

Q = pipe flow rate

D = pipe diameter

See also:

Function in aquaclarar: `pc.headloss_fric(FlowRate, Diam, Length, Nu, PipeRough)` Returns only major losses. Works for both laminar and turbulent flow. PipeRough describes the pipe roughness ϵ described shortly below.

Darcy-Weisbach is wonderful because it applies to both laminar and turbulent flow regimes and contains relatively easy to measure variables. The one exception is the Darcy friction factor, f . This parameter is an approximation for the magnitude of friction between the pipe walls and the fluid, and its value changes depending on the whether or not the flow is laminar or turbulent, and varies with the Reynolds number in both flow regimes.

For laminar flow, the friction factor can be determined from the following equation:

$$f = \frac{64}{Re} \quad (7.12)$$

For turbulent flow, the friction factor is more difficult to determine. In this class, we will use the **Swamee-Jain equation**:

$$f = \frac{0.25}{\left[\log \left(\frac{\epsilon}{3.7D} + \frac{5.74}{Re^{0.9}} \right) \right]^2} \quad (7.13)$$

Such that:

ϵ = pipe roughness, [L]

D = pipe diameter, [L]

See also:

Function in aquaclara: `pc.fric(FlowRate, Diam, Nu, PipeRough)` Returns f for laminar or turbulent flow. For laminar flow, use zero for the PipeRough input.

The simplicity of the equation for f during laminar flow allows for substitutions to create a very useful, simplified equation for major losses during laminar flow. This simplification combines the Darcy-Weisbach equation, the equation for the Darcy friction factor during laminar flow, and the Reynolds number formula:

$$h_f = f \frac{8}{g\pi^2} \frac{LQ^2}{D^5} \quad (7.14)$$

$$f = \frac{64}{Re} \quad (7.15)$$

$$Re = \frac{4Q}{\pi D \nu} \quad (7.16)$$

To form the Hagen-Poiseuille equation for major losses during laminar flow, and *only* during laminar flow:

$$h_f = \frac{128\mu L Q}{\rho g \pi D^4} \quad (7.17)$$

$$h_f = \frac{32\nu L \bar{v}}{g D^2} \quad (7.18)$$

The significance of this equation lies in its relationship between h_f and Q . Hagen-Poiseuille shows that the terms are directly proportional ($h_f \propto Q$) during laminar flow, while Darcy-Weisbach shows that h_f grows with the square of Q during turbulent flow ($h_f \propto Q^2$). As you will soon see, minor losses, h_e , will grow with the square of Q in both laminar and turbulent flow. This has implications that will be discussed in a future chapter: *Flow Control and Measurement Design*.

In 1944, Lewis Ferry Moody plotted a ridiculous amount of experimental data, gathered by many people, on the Darcy-Weisbach friction factor to create what we now call the [Moody diagram](#). This diagram has makes it easy to find the friction factor f . f is plotted on the left-hand y-axis, relative pipe roughness $\frac{\epsilon}{D}$ is on the right-hand y-axis, and Reynolds number Re is on the x-axis. The Moody diagram is an alternative to computational methods for finding f .

7.4.2 Minor Losses

Unfortunately, there is no simple pushing a box across the ground example to explain minor losses. So instead, consider a [hydraulic jump](#). In the video, you can see lots of turbulence and eddies in the transition region between the fast, shallow flow and the slow, deep flow. The high amount of mixing of the water in the transition region of the hydraulic jump results in significant friction *between water and water*. This turbulent, eddy-induced, fluid-fluid friction results in minor losses, much like fluid-pipe friction results in major losses.

As occurs in a hydraulic jump, a flow expansion (from shallow flow to deep flow) creates the turbulent eddies that result in minor losses. This will be a recurring theme in throughout the course: **minor losses are caused by flow expansions**. Imagine a pipe fitting that connects a small diameter pipe to a large diameter one, as shown in Fig. 7.7 below. The flow must expand to fill up the entire large diameter pipe. This expansion creates turbulent eddies near the union between the small and large pipes, and these eddies result in minor losses. You may already know the equation for minor losses, but understanding where it comes from is very important for effective AquaClara plant design. For this reason, you are strongly recommended to read through its full derivation: [Review: Fluid Mechanics Derivations](#).

There are three forms of the minor loss equation that you will see in this class:

$$\text{First form : } h_e = \frac{(\bar{v}_{in} - \bar{v}_{out})^2}{2g} \quad (7.19)$$

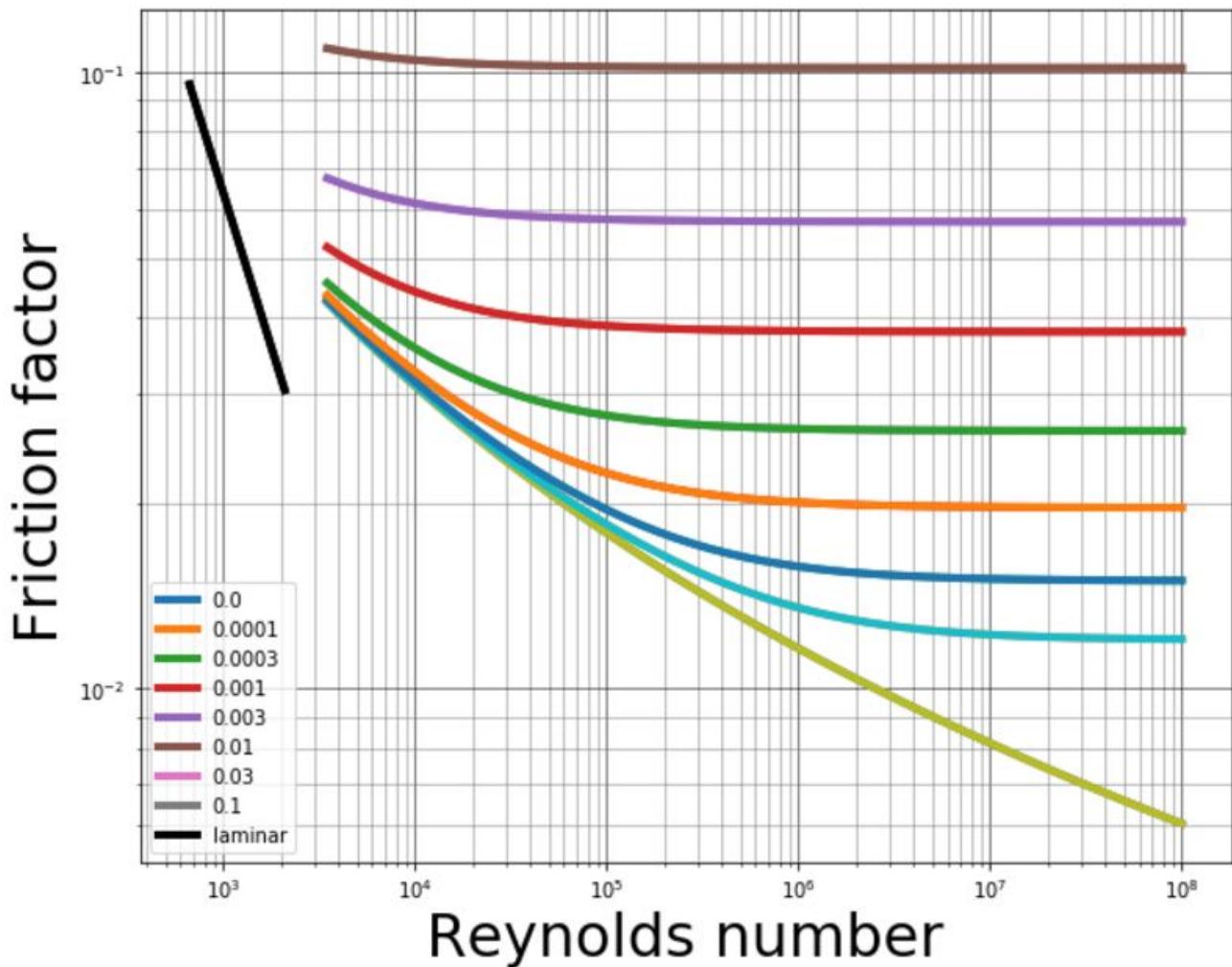


Fig. 7.6: This is the famous and famously useful Moody diagram.

$$\text{Second form : } h_e = \left(1 - \frac{A_{in}}{A_{out}}\right)^2 \frac{\bar{v}_{in}^2}{2g} = K'_e \frac{\bar{v}_{in}^2}{2g}, \quad \text{where} \quad K'_e = \left(1 - \frac{A_{in}}{A_{out}}\right)^2 \quad (7.20)$$

$$\text{Third form : } h_e = \left(\frac{A_{out}}{A_{in}} - 1\right)^2 \frac{\bar{v}_{out}^2}{2g} = K_e \frac{\bar{v}_{out}^2}{2g}, \quad \text{where} \quad K_e = \left(\frac{A_{out}}{A_{in}} - 1\right)^2 \quad (7.21)$$

Such that:

K'_e, K_e = minor loss coefficients, dimensionless

Note: You will most often see K'_e and K_e used without the e subscript, as K' and K .

See also:

Function in aquaclara: pc.headloss_exp_general(Vel, KMinor) Returns h_e . Can be either the second or third form due to user input of both velocity and minor loss coefficient. It is up to the user to use consistent \bar{v} and K_e .

See also:

Function in aquaclara: pc.headloss_exp(FlowRate, Diam, KMinor) Returns h_e . Uses third form, K_e .

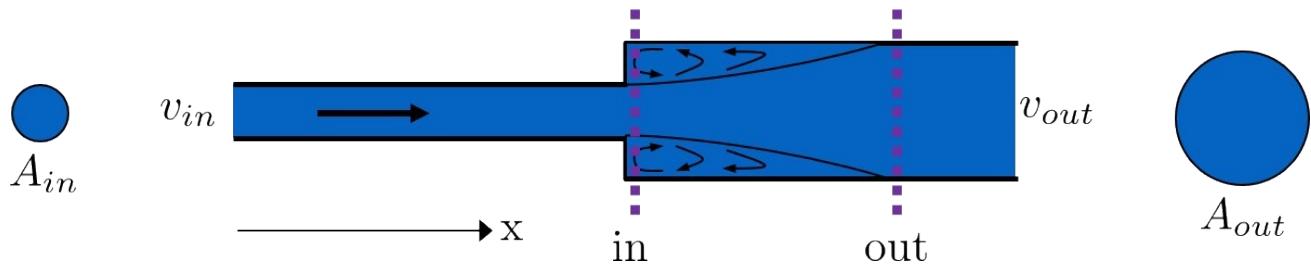


Fig. 7.7: The *in* and *out* subscripts in each of the three forms of the minor loss equation refer to this diagram that was used for the derivation.

The second and third forms are the ones which you are probably most familiar with. The distinction between them, however, is critical. First, consider the magnitudes of A_{in} and A_{out} . A_{in} can never be larger than A_{out} , because the flow is expanding. When flow expands, the cross-sectional area it flows through must increase. As a result, both $\frac{A_{out}}{A_{in}} > 1$ and $\frac{A_{in}}{A_{out}} < 1$ must always be true. This means that K' can never be greater than 1, while K technically has no upper limit.

If you have taken CEE 3310, you have seen tables of minor loss coefficients [like this one](#), and they almost all have coefficients greater than 1. This implies that these tables use the third form of the minor loss equation as we have defined it, where the velocity is \bar{v}_{out} . There is a good reason for using the third form over the second one: \bar{v}_{out} is far easier to determine than \bar{v}_{in} . Consider flow through a pipe elbow, as shown in the image below.

In order to find \bar{v}_{out} , we first need to know what (or where) is *out* and what is *in*. A simple way to distinguish the two surfaces is that *in* occurs when the flow is most contracted, and *out* occurs when the flow has fully expanded after that maximal contraction. Going on these guidelines, Control surface 2 (CS 2) in the figure above would be *in*, since it represents the most contracted flow in the elbow-pipe system. Therefore, CS 3 would be *out*, as it represents the flow having fully expanded after its compression at CS 2.

\bar{v}_{out} is easy to determine because it is the velocity of the fluid as it flows through the entire area of the pipe. Thus, \bar{v}_{out} can be found with the continuity equation, since the flow through the pipe and its diameter are easy to measure, $\bar{v}_{out} = \frac{4Q}{\pi D^2}$. On the other hand, \bar{v}_{in} is difficult to find, as the area of the contracted flow is dependent on the exact geometry of the elbow. This is why the third form of the minor loss equation, as we have defined it, is the most

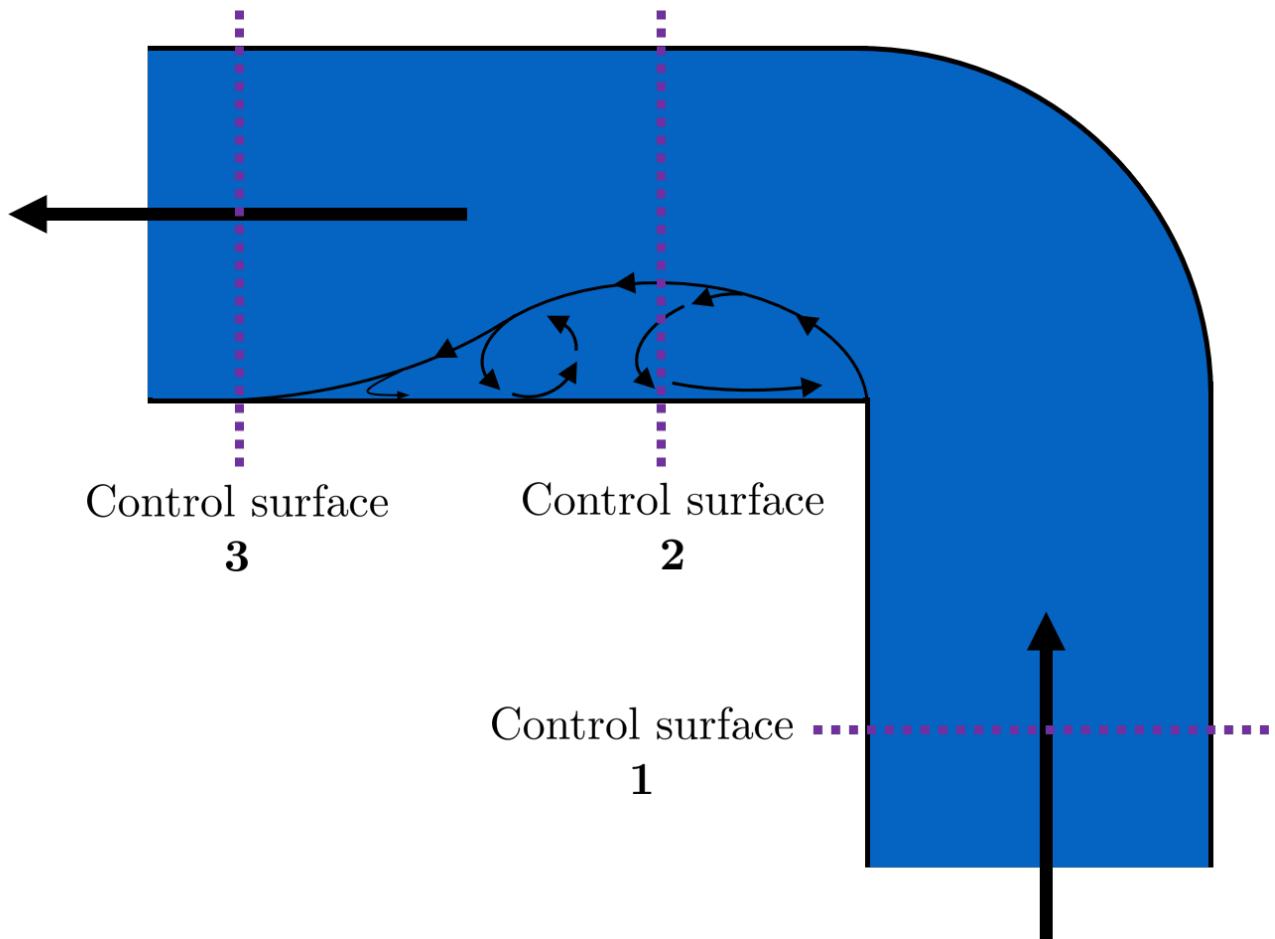


Fig. 7.8: Flow around a pipe elbow results in a minor loss. Control surface 1 can be abbreviated as CS 1

common:

$$h_e = K \frac{\bar{v}_{out}^2}{2g} = \left(\frac{A_{out}}{A_{in}} - 1 \right)^2 \frac{\bar{v}_{out}^2}{2g} \quad (7.22)$$

Note: When considering a hydraulic system within a control volume, there can be many sources of minor losses. Instead of saying $h_e = K_1 \frac{\bar{v}_{out}^2}{2g} + K_2 \frac{\bar{v}_{out}^2}{2g} + \dots$ we can simply lump all of the minor loss coefficients into one: $\sum K = K_1 + K_2 + \dots$. Thus, it is also common to see this form of the minor loss equation when finding the minor loss across control volumes: $\sum K \frac{\bar{v}_{out}^2}{2g}$.

7.4.3 Head Loss = Elevation Difference Trick

This trick, also called the control volume trick, or more colloquially, the head loss trick, is incredibly useful for simplifying hydraulic systems and is used all the time in this class.

Consider the following figure:

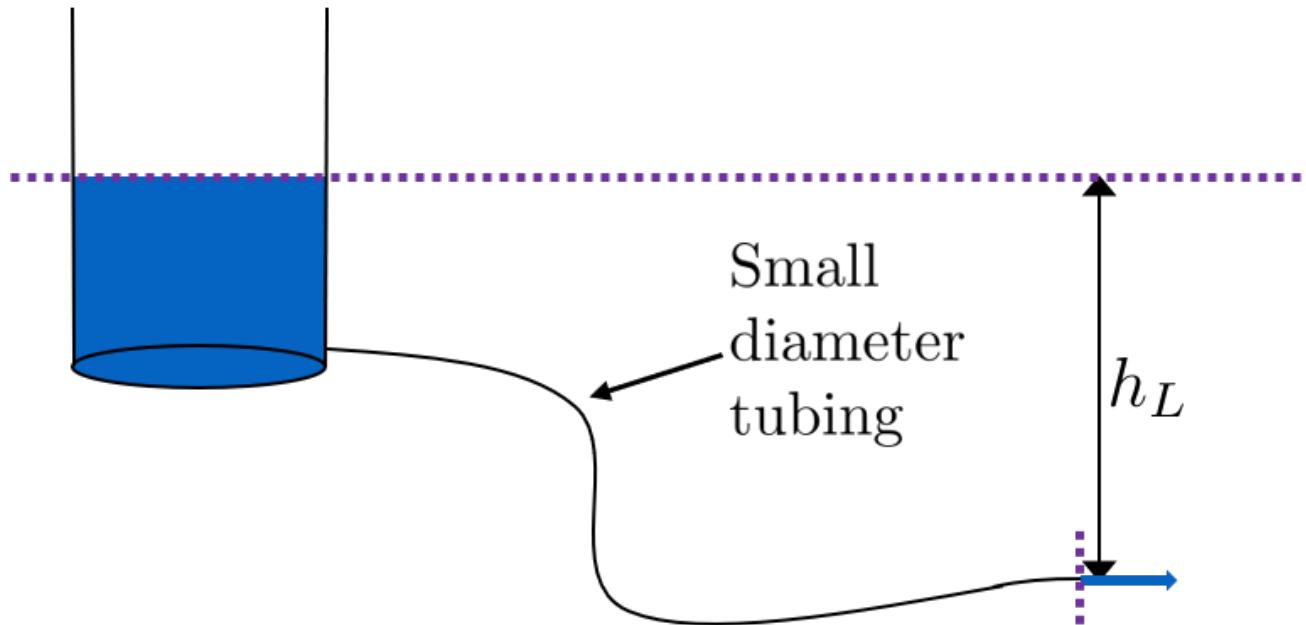


Fig. 7.9: A typical hydraulic system can be used to understand the head loss trick.

In systems like this, where an elevation difference is causing water to flow, the elevation difference is called the **driving head**. In the system above, the driving head is the elevation difference between the water level and the end of the tubing. Usually, driving head is written as Δz or Δh , though above it is labelled as h_L . Doesn't h_L refer to head loss though? Yes it does! Referring to Δh or Δz IS the head loss trick, and how it works is explained in the following paragraphs and equations.

The figure is technically violating the energy equation by saying that the elevation difference between the water in the tank and the end of the tube is h_L . It implies that all of the driving head, Δz , is lost to head loss. Since all of the energy is gone, there should not be water flowing out of the tubing. But there is. Lets apply the energy equation across the control surfaces shown in the figure. Pressures at both ends are atmospheric and the velocity of water at the top of tank is negligible.

$$\frac{p_f}{\rho g} + z_1 + \frac{\bar{v}_1^2}{2g} = \frac{p_f}{\rho g} + z_2 + \frac{\bar{v}_2^2}{2g} + h_L \quad (7.23)$$

We now get:

$$\Delta z = \frac{\bar{v}_2^2}{2g} + h_L \quad (7.24)$$

This equation contradicts the figure above, which says that $\Delta z = h_L$ and neglects $\frac{\bar{v}_2^2}{2g}$. The figure above is correct, however, if you apply the head loss trick. The trick incorporates the $\frac{\bar{v}_2^2}{2g}$ term *into* the h_L term as a minor loss. See the math below:

$$\Delta z = \frac{\bar{v}_2^2}{2g} + h_e + h_f \quad (7.25)$$

$$\Delta z = \frac{\bar{v}_2^2}{2g} + \left(\sum K \right) \frac{\bar{v}_2^2}{2g} + h_f \quad (7.26)$$

$$\Delta z = \left(1 + \sum K \right) \frac{\bar{v}_2^2}{2g} + h_f \quad (7.27)$$

$$\Delta z = \left(\sum K \right) \frac{\bar{v}_2^2}{2g} + h_f \quad (7.28)$$

This last step incorporated the kinetic energy term of the energy equation, $\frac{\bar{v}_2^2}{2g}$, into the minor loss equation by saying that its K is 1 and incorporating that 1 into $\sum K$. From here, we reverse our steps to get $\Delta z = h_L$, starting with $h_e = (\sum K) \frac{\bar{v}_2^2}{2g}$

$$\Delta z = h_e + h_f \quad (7.29)$$

$$\Delta z = h_L \quad (7.30)$$

By applying the head loss trick, you are considering the entire flow of the fluid out of a control volume as energy lost via minor losses. This is just an algebraic trick, the only thing to remember when applying this trick is that $\sum K$ will always be at least 1, even if there are no real minor losses in the system.

7.5 The Orifice Equation

This equation is one that you'll see and use again and again throughout this class. Understanding it now will be invaluable, as future concepts will use and build on this equation.

7.5.1 What is a Vena Contracta?

Before describing the equation, we must first understand the concept of a *vena contracta*. Refer to the figure below.

The flow contracts as the fluid moves past the gate. This happens because the fluid can't make a sharp turn as it tries to go around the gate, as indicated by the streamline in the figure. Instead, the most extreme streamline makes a gradual change in direction. As a result of this gradual turn, the flow contracts and the cross-sectional area the fluid is flowing decreases.

The term *vena contracta* describes the phenomenon of contracting flow due to streamlines being unable to make sharp turns. Π_{vc} is a dimensionless ratio comparing the flow area at the point of maximal contraction, $A_{downstream}$, and the flow area *before* the contraction, A_{gate} . In the figure above, the equation for the vena contracta coefficient would be:

$$\Pi_{vc} = \frac{A_{downstream}}{A_{gate}} \quad (7.31)$$

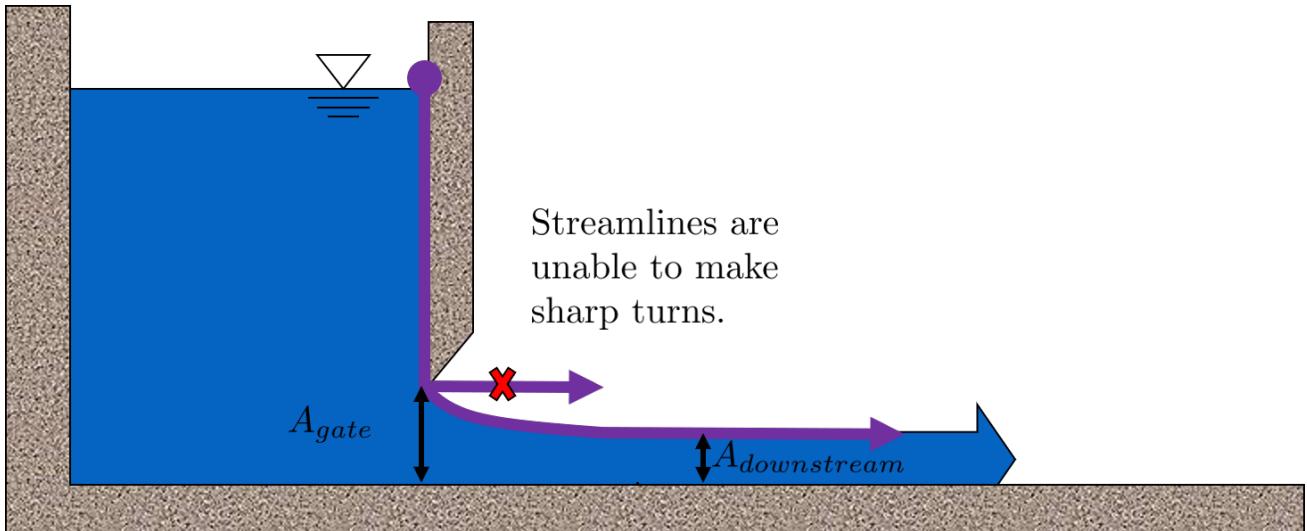


Fig. 7.10: This figure shows flow around a sluice gate. Since streamlines can't make sharp turns, the flow is forced to gradually curve and contract to an area smaller than the area of the gate.

When the most extreme turn a streamline must make is 90°, the value of the vena contracta coefficient is close to 0.62. This parameter value, 0.62, is in aquaclara as `pc.VC_ORIFICE_RATIO`. The vena contracta coefficient value is a function of the flow geometry. Since the ratio always puts the most contracted area over the least contracted area, Π_{vc} is always less than 1.

Important: A vena contracta coefficient is not a minor loss coefficient. Though the equations for the two both involve contracted and non-contracted areas, these coefficients are not the same. Minor losses coefficients imply energy loss, and vena contractas do not. Minor losses coefficients deal with flow expansions, and vena contractas deal with flow contractions. Confusing the two coefficients is a common mistake that this paragraph will hopefully help you to avoid.

Note: Note that what this class calls Π_{vc} is often referred to as a Coefficient of Contraction, C_c , in other engineering courses and settings.

7.5.2 Origin of the Orifice Equation

The orifice equation is derived from the Bernoulli equation as applied to the purple points in the following image:

At point 1, the pressure is atmospheric and the instantaneous velocity is negligible as the water level in the bucket drops slowly. At point 2, the pressure is also atmospheric. We define the difference in elevations between the two points, $z_1 - z_2$, to be Δh . With these simplifications ($p_1 = \bar{v}_1 = p_2 = 0$) and assumptions ($z_A - z_B = \Delta h$), the Bernoulli equation becomes:

$$\Delta h = \frac{\bar{v}_2^2}{2g} \quad (7.32)$$

Substituting the continuity equation $Q = \bar{v}A$ in the form of $\bar{v}_2^2 = \frac{Q^2}{A_{vc}^2}$, the vena contracta coefficient in the form of $A_{vc} = \Pi_{vc}A_{or}$ yields:

$$\Delta h = \frac{Q^2}{2g\Pi_{vc}^2 A_{or}^2} \quad (7.33)$$

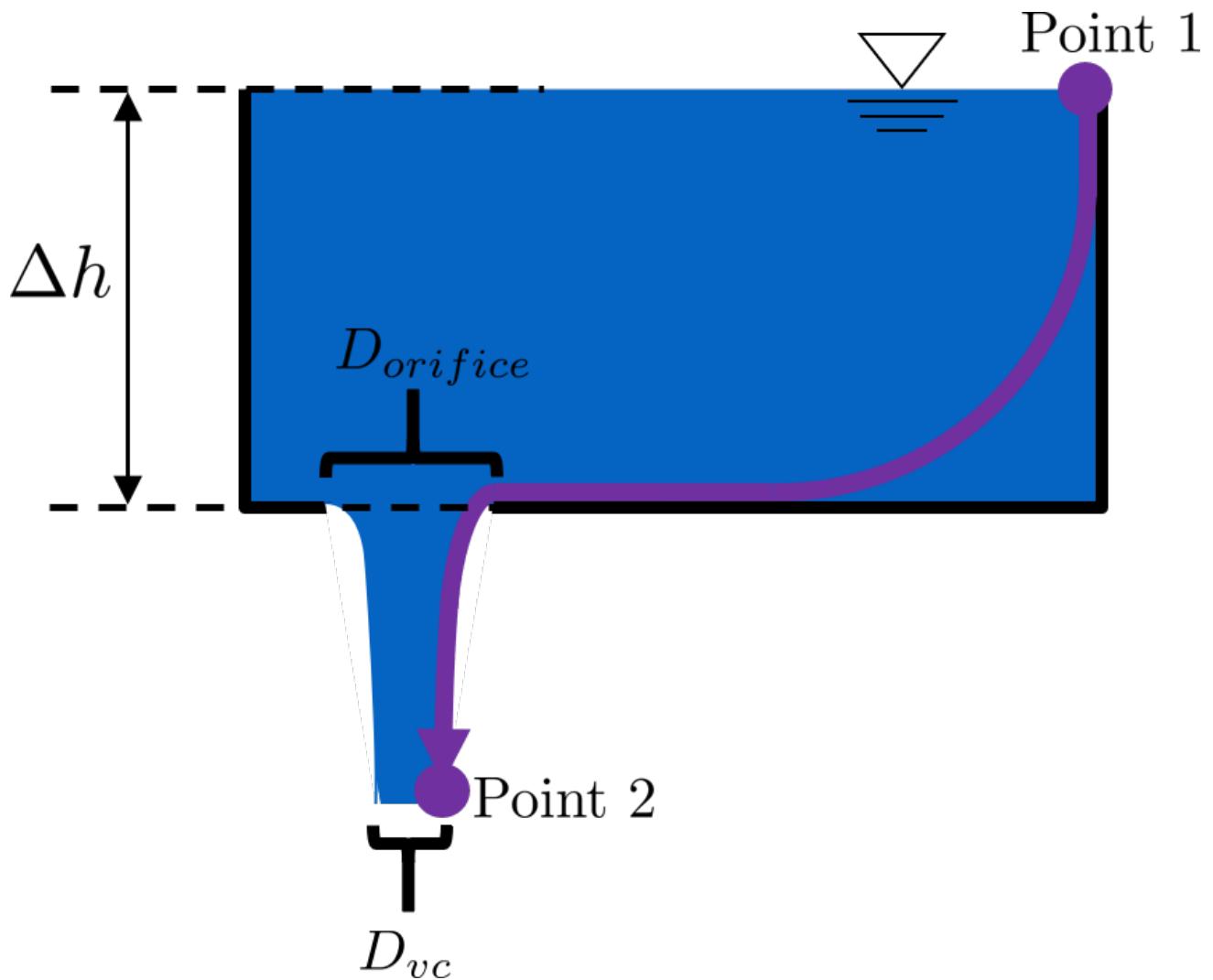


Fig. 7.11: Flow through a hole in the bottom of a bucket is a great example of the orifice equation.

Which, rearranged to solve for Q gives **The Orifice Equation:**

$$Q = \Pi_{vc} A_{or} \sqrt{2g\Delta h} \quad (7.34)$$

Such that:

$\Pi_{vc} = 0.62$ = vena contracta coefficient, as `pc.VC_ORIFICE_RATIO`

A_{or} = orifice area- NOT contracted flow area

Δh = elevation difference between orifice and water level

See also:

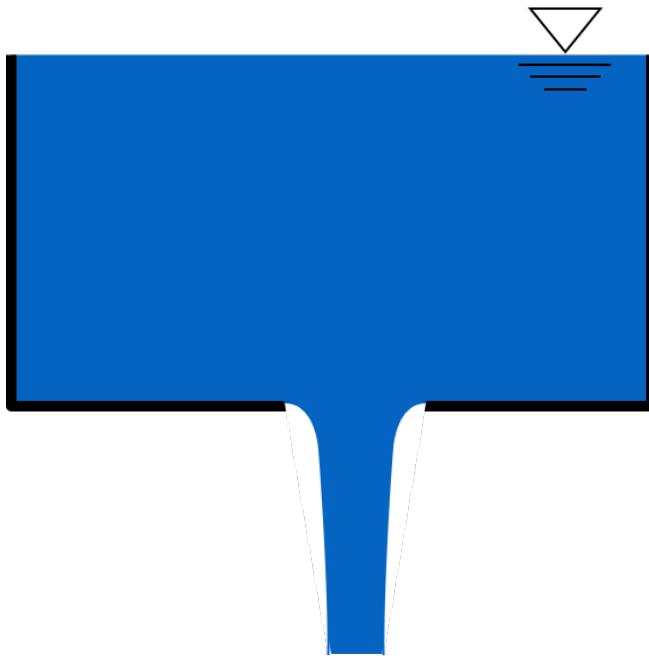
Equation in aquaclarar: `pc.flow_orifice(Diam, Height, RatioVCOrifice)` Returns flow through a horizontal orifice.

See also:

Equation in aquaclarar: `pc.flow_orifice_vert(Diam, Height, RatioVCOrifice)` Returns flow through a vertical orifice. The height parameter refers to height above the center of the orifice.

There are two configurations for an orifice in the tank holding a fluid: horizontal and vertical. These are both displayed in the figure below. The orifice equation written is for a horizontal orifice; the equation for flow through vertical orifice equation requires integration or the orifice equation across its height to return the correct flow. This is explored in the Flow Control and Measurement Examples section.

Horizontal orifice



Vertical orifice



Fig. 7.12: The descriptions vertical and horizontal **apply to the orientation of the orifices**, not to the orientation of the fluid coming out of the orifices.

7.6 Section Summary

1. Introductory Concepts:

- **Continuity** means that the mass of a fluid is conserved as it flows, and implies a constant density. The continuity equation has two purposes:

(a) Relating the average velocity of a fluid, \bar{v} , to its flow rate, Q , via the cross-sectional area, A , that it flows through. When the fluid is flowing in a pipe, we can simply this even further to relate the flow rate and velocity to the pipes diameter, D . The final equation below is only used for circular pipes, as it includes a pipe diameter.

$$Q = \bar{v}A = \bar{v}\frac{\pi D^2}{4} \quad (7.35)$$

- (a) Finding the average velocity or flow when the geometry of a fluids flow changes, as the mass of the fluid must be conserved when it transitions through flow geometries.

$$Q_1 = Q_2 \quad (7.36)$$

$$\bar{v}_1 A_1 = \bar{v}_2 A_2 \quad (7.37)$$

$$\bar{v}_1 \frac{\pi D_1^2}{4} = \bar{v}_2 \frac{\pi D_2^2}{4} \quad (7.38)$$

- **Laminar and Turbulent flow** describe the disorder and chaos of fluid flow. The **Reynolds number**, Re is used to distinguish laminar from turbulent flow. For $Re < 2100$, flow is considered laminar. For $Re > 2100$, flow is considered turbulent. The equations for the Reynolds number are below:

$$Re = \frac{\bar{v}D}{\nu} = \frac{4Q}{\pi D \nu} = \frac{\rho \bar{v} D}{\mu} \quad (7.39)$$

- **Control volumes vs Streamlines.** This section is quite short, a summary would simply repeat what the sections says. The section is its own summary; read it here: [Streamlines and Control Volumes](#)

2. Bernoulli vs Energy equations:

The Bernoulli equation assumes that energy is conserved throughout a streamline or control volume. The Energy equation assumes that there is energy loss, or head loss h_L . This head loss is composed of major losses, h_f , and minor losses, h_e .

Bernoulli equation:

$$\frac{p_1}{\rho g} + z_1 + \frac{\bar{v}_1^2}{2g} = \frac{p_2}{\rho g} + z_2 + \frac{\bar{v}_2^2}{2g} \quad (7.40)$$

Energy equation, simplified to remove pumps, turbines, and α factors:

$$\frac{p_1}{\rho g} + z_1 + \frac{\bar{v}_1^2}{2g} = \frac{p_2}{\rho g} + z_2 + \frac{\bar{v}_2^2}{2g} + h_L \quad (7.41)$$

3. Major losses:

Defined as the energy loss due to shear between the walls of the pipe/flow conduit and the fluid. The Darcy-Weisbach equation is used to find major losses in both laminar and turbulent flow regimes. The equation for finding the Darcy friction factor, f , changes depending on whether the flow is laminar or turbulent. The Moody diagram is a common graphical method for finding f . During laminar flow, the Hagen-Poiseuille equation, which is just a combination of Darcy-Weisbach, Reynolds number, and $f = \frac{64}{Re}$, can be used

Darcy-Weisbach equation:

$$h_f = f \frac{L}{D} \frac{\bar{v}^2}{2g} \quad (7.42)$$

For water treatment plant design we tend to use plant flow rate, Q , as our master variable and thus we have.

$$h_f = f \frac{8}{g\pi^2} \frac{LQ^2}{D^5} \quad (7.43)$$

f for laminar flow:

$$f = \frac{64}{Re} = \frac{16\pi D\nu}{Q} = \frac{64\nu}{\bar{v}D} \quad (7.44)$$

f for turbulent flow:

$$f = \frac{0.25}{\left[\log \left(\frac{\epsilon}{3.7D} + \frac{5.74}{Re^{0.9}} \right) \right]^2} \quad (7.45)$$

Hagen-Poiseuille equation for laminar flow:

$$h_f = \frac{32\mu L \bar{v}}{\rho g D^2} = \frac{128\mu Q}{\rho g \pi D^4} \quad (7.46)$$

4. **Minor losses:** Defined as the energy loss due to the generation of turbulent eddies when flow expands. Once more: minor losses are caused by flow expansions. There are three forms of the minor loss equation, two of which look the same but use different coefficients (K' vs K) and velocities (\bar{v}_{in} vs \bar{v}_{out}). *Make sure the coefficient you select is consistent with the velocity you use.* The third form, written in purple, is the most commonly used form of the minor loss equation.

$$\text{First form : } h_e = \frac{(\bar{v}_{in} - \bar{v}_{out})^2}{2g} \quad (7.47)$$

$$\text{Second form : } h_e = \left(1 - \frac{A_{in}}{A_{out}} \right)^2 \frac{\bar{v}_{in}^2}{2g} = K'_e \frac{\bar{v}_{in}^2}{2g}, \quad \text{where } K'_e = \left(1 - \frac{A_{in}}{A_{out}} \right)^2 \quad (7.48)$$

$$\text{Third form : } h_e = \left(\frac{A_{out}}{A_{in}} - 1 \right)^2 \frac{\bar{v}_{out}^2}{2g} = K_e \frac{\bar{v}_{out}^2}{2g}, \quad \text{where } K_e = \left(\frac{A_{out}}{A_{in}} - 1 \right)^2 \quad (7.49)$$

5. **Major and minor losses vary with flow:** While it is generally important to know how increasing or decreasing flow will affect head loss, it is even more important for this class to understand exactly how flow will affect head loss. As the table below shows, head loss will always be proportional to flow squared during turbulent flow. During laminar flow, however, the exponent on Q will be between 1 and 2 depending on the proportion of major to minor losses.

Table 7.1: Proportionality between head loss h_L and flow rate Q for different flow regimes and types of head loss.

$h_L \propto Q^?$	Major Losses	Minor Losses
Laminar	Q	Q^2
Turbulent	Q^2	Q^2

6. The **head loss trick**, also called the control volume trick, can be used to incorporate the kinetic energy out term of the energy equation, $\frac{\bar{v}_2^2}{2g}$, into head loss as a minor loss with $K = 1$, so the minor loss equation becomes $(1 + \sum K) \frac{\bar{v}^2}{2g}$. This is used to be able to say that $\Delta z = h_L$ and makes many equation simplifications possible in the future.

7. Orifice equation and vena contractas: The orifice equation is used to determine the flow out of an orifice given the elevation of water above the orifice. This equation introduces the concept of vena contracta, which describes flow contraction due to the inability of streamlines to make sharp turns. The equation shows that the flow out of an orifice is proportional to the square root of the driving head, $Q \propto \sqrt{\Delta h}$. Depending on the orientation of the orifice, vertical (like a hole in the side of a bucket) or horizontal (like a hole in the bottom of a bucket), a different equation in aquaclaras should be used.

The Orifice Equation:

$$Q = \Pi_{vc} A_{or} \sqrt{2g\Delta h} \quad (7.50)$$

REVIEW: FLUID MECHANICS DERIVATIONS

8.1 Minor Loss Equation

This section contains the derivation of the minor loss equation using the following figure as a reference. The derivation begins with a slightly simplified energy equation across the control volume shown. Our energy equation begins with h_P and h_T having been eliminated.

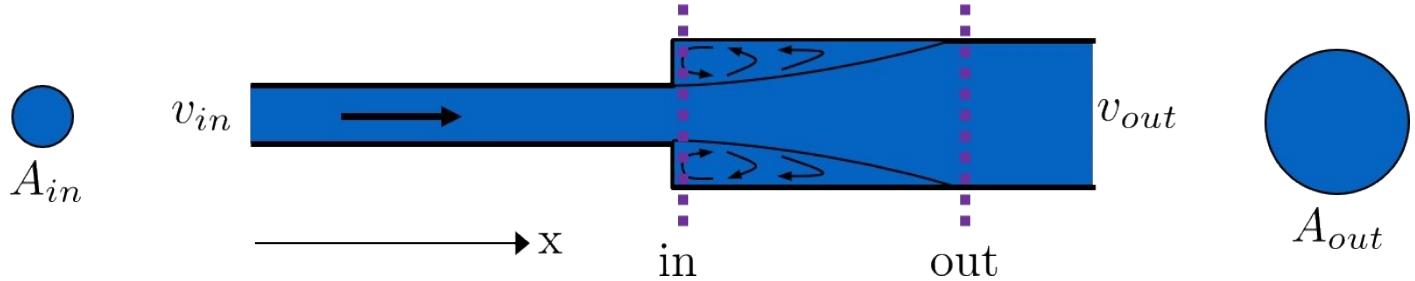


Fig. 8.1: This is the system we will use to derive the minor loss equation.

$$\frac{p_{in}}{\rho g} + z_{in} + \frac{\bar{v}_{in}^2}{2g} = \frac{p_{out}}{\rho g} + z_{out} + \frac{\bar{v}_{out}^2}{2g} + h_L \quad (8.1)$$

Since the elevations at the center of the *in* and *out* control surfaces are the same, we can eliminate z_{in} and z_{out} . As we are considering such a small length of pipe, we will neglect the major loss component of head loss. Thus, $h_L = h_e + h_f$. The following three equations are all the same, simply rearranged to solve for h_e .

$$\frac{p_{in}}{\rho g} + \frac{\bar{v}_{in}^2}{2g} = \frac{p_{out}}{\rho g} + \frac{\bar{v}_{out}^2}{2g} + h_e \quad (8.2)$$

$$\frac{p_{in} - p_{out}}{\rho g} = \frac{\bar{v}_{out}^2 - \bar{v}_{in}^2}{2g} + h_e \quad (8.3)$$

$$h_e = \frac{p_{in} - p_{out}}{\rho g} + \frac{\bar{v}_{in}^2 - \bar{v}_{out}^2}{2g} \quad (8.4)$$

This last equation has h_e as a function of four variables (p_{in} , p_{out} , v_{in} , and v_{out}); we would like it to be a function of only one. Thus, we will invoke conservation of momentum in the horizontal direction across our control volume to remove variables. The difference in momentum from the *in* point to the *out* point is driven by the pressure difference between each end of the control volume. We will be considering the pressure at the centroid of our control surfaces, and we will neglect shear along the pipe walls. After these assumptions, our momentum equation becomes the following:

$$M_{in, x} + M_{out, x} = F_{p_{in, x}} + F_{p_{out, x}} \quad (8.5)$$

Such that:

M_x = momentum flowing through the control volume in the x-direction

F_{px} = force due to pressure acting on the boundaries of the control volume in the x-direction

Recall that momentum is mass times velocity for solids, mv , with units of $\frac{[M][L]}{[T]}$. Since we consider water flowing through a pipe, there is not one singular mass or one singular velocity. Instead, there is a mass flow rate, or a mass per time indicated by $\dot{m} = \rho Q$, which has units of $\frac{[M]}{[T]}$. Therefore, the momentum for a fluid is $\rho Q \bar{v}$. Applying the continuity equation $Q = \bar{v}A$, we get to the following equation for the momentum of a fluid flowing through a pipe which we will use in this derivation, $M = \rho \bar{v}^2 A$. The pressure force is simply the pressure at the centroid of the flow multiplied by the area the pressure is acting upon, pA .

To ensure correct sign convention, we will make each side of the equation negative for reasons discussed shortly. Since $\bar{v}_{in} > \bar{v}_{out}$, the left hand side will be $M_{out} - M_{in}$ in order to be negative. The reduction in velocity from *in* to *out* causes an increase in pressure, therefore $p_{in} - p_{out}$ is negative. With these substitutions, the conservation of momentum equation becomes as follows:

$$M_{out} - M_{in} = p_{in} - p_{out} \quad (8.6)$$

$$\rho \bar{v}_{out}^2 A_{out} - \rho \bar{v}_{in}^2 A_{in} = p_{in} A_{out} - p_{out} A_{out} \quad (8.7)$$

Note that the area term attached to p_{in} is actually A_{out} instead of A_{in} , as one might think. This is because $A_{out} = A_{in}$. We chose our control volume to start a few millimeters into the larger pipe, which means that the cross-sectional area does not change over the course of the control volume.

Dividing both sides of the equation by $A_{out}\rho g$, we obtain the following equation, which contains the very same pressure term as our adjusted energy equation above, equation (8.4). This is why we chose a negative sign convention.

$$\frac{p_{in} - p_{out}}{\rho g} = \frac{\bar{v}_{out}^2 - \bar{v}_{in}^2 \frac{A_{in}}{A_{out}}}{g} \quad (8.8)$$

Now, we combine the momentum, continuity, and adjusted energy equations:

$$\text{Energy equation : } h_e = \frac{p_{in} - p_{out}}{\rho g} + \frac{\bar{v}_{in}^2 - \bar{v}_{out}^2}{2g} \quad (8.9)$$

$$\text{Momentum equation : } \frac{p_{in} - p_{out}}{\rho g} = \frac{\bar{v}_{out}^2 - \bar{v}_{in}^2 \frac{A_{in}}{A_{out}}}{g} \quad (8.10)$$

$$\text{Continuity equation : } \frac{A_{in}}{A_{out}} = \frac{\bar{v}_{out}}{\bar{v}_{in}} \quad (8.11)$$

To obtain an equation for minor losses with just two variables, \bar{v}_{in} and \bar{v}_{out} .

$$h_e = \frac{\bar{v}_{out}^2 - \bar{v}_{in}^2 \frac{\bar{v}_{out}}{\bar{v}_{in}}}{g} + \frac{\bar{v}_{in}^2 - \bar{v}_{out}^2}{2g} \quad (8.12)$$

Now we will combine the two terms. The numerator and denominator of the first term, $\frac{\bar{v}_{out}^2 - \bar{v}_{in}^2 \frac{\bar{v}_{out}}{\bar{v}_{in}}}{g}$ will be multiplied by 2 to become $\frac{2\bar{v}_{out}^2 - 2\bar{v}_{in}^2 \frac{\bar{v}_{out}}{\bar{v}_{in}}}{2g}$. The equation then looks like:

$$h_e = \frac{\bar{v}_{out}^2 - 2\bar{v}_{in}\bar{v}_{out} + \bar{v}_{in}^2}{2g} \quad (8.13)$$

8.1.1 Final Forms of the Minor Loss Equation

Factoring the numerator yields to the first final form of the minor loss equation:

$$\text{First form : } h_e = \frac{(\bar{v}_{in} - \bar{v}_{out})^2}{2g} \quad (8.14)$$

From here, the two other forms of the minor loss equation can be derived by solving for either \bar{v}_{in} or \bar{v}_{out} using the ubiquitous continuity equation $\bar{v}_{in}A_{in} = \bar{v}_{out}A_{out}$:

$$\text{Second form : } h_e = \left(1 - \frac{A_{in}}{A_{out}}\right)^2 \frac{\bar{v}_{in}^2}{2g} = K'_e \frac{\bar{v}_{in}^2}{2g}, \quad \text{where } K'_e = \left(1 - \frac{A_{in}}{A_{out}}\right)^2 \quad (8.15)$$

$$\text{Third form : } h_e = \left(\frac{A_{out}}{A_{in}} - 1\right)^2 \frac{\bar{v}_{out}^2}{2g} = K_e \frac{\bar{v}_{out}^2}{2g}, \quad \text{where } K_e = \left(\frac{A_{out}}{A_{in}} - 1\right)^2 \quad (8.16)$$

Note: You will often see K'_e and K_e used without the e subscript, they will appear as K' and K .

Being familiar with these three forms and how they are used will be of great help throughout the class. The third form is the one that is most commonly used.

FLOW CONTROL AND MEASUREMENT INTRODUCTION

9.1 Tank with a Valve

9.1.1 Flow Q and Water Level h as a Function of Time

Our first step is to see if we can get constant head out of a simple system. The most simple flow control system is a bucket or tank with a hole in it. This system is too coarse to provide constant head. One step above that is a bucket or tank with a valve. This is where we begin our search for constant head.

Using the setup of in the image below, we derive the following equation for flow Q through the valve as a function of time t . The derivation is found here: [for a Tank with a Valve](#). You are advised to read through it if you are at all confused about this equation.

$$\frac{Q}{Q_0} = 1 - \frac{1}{2} \frac{t}{t_{Design}} \frac{h_{Tank}}{h_0} \quad (9.1)$$

Such that:

$Q = Q(t)$ = flow of hypochlorite through valve at time t

Q_0 = flow of hypochlorite through valve at time $t = 0$

t = elapsed time

t_{Design} = time it *would* take for tank to empty if flow stayed constant at Q_0 , which it does not

h_{Tank} = elevation of water level with reference to tank bottom at time $t = 0$

h_0 = elevation of water level with reference to the valve at time $t = 0$

This equation has historically give students some trouble, and while its nuances are explained in the derivation, they will be quickly summarized here:

- t_{Design} is NOT the time it takes to drain the tank. It is the time that it *would* take to drain the tank *if* the flow rate at time $t = 0$, Q_0 , were the flow rate forever, which it is not. t_{Design} was used in the derivation to simplify the equation, which is why this potentially-confusing parameter exists. The actual time it takes to drain the tank lies somewhere between t_{Design} and $2t_{Design}$ and depends on the ratio $\frac{h_{Tank}}{h_0}$.
- h_{Tank} is not the same as h_0 . h_{Tank} is the height of water level in the tank with reference to the tank bottom. h_0 is the water level in the tank with reference to the valve. Neither change with time, they both refer to the water level at one instance in time, $t = 0$. Therefore, $h_0 \geq h_{Tank}$ is always true. If the tank is elevated far above the valve, then the $h_0 \gg h_{Tank}$. If the valve is at the same elevation as the bottom of the tank, then $h_0 = h_{Tank}$. Please refer to the figure above to clarify h_0 and h_{Tank} .

We can use the proportionality $Q \propto \sqrt{h}$, which applies to both minor losses and orifices to form a relationship between water level in the tank h and time t . This proportionality comes from rearranging the minor loss equation $h = K \frac{Q^2}{2gA^2}$ for Q instead of h . A table of proportionality between Q and h can be found in [Table 7.1](#)

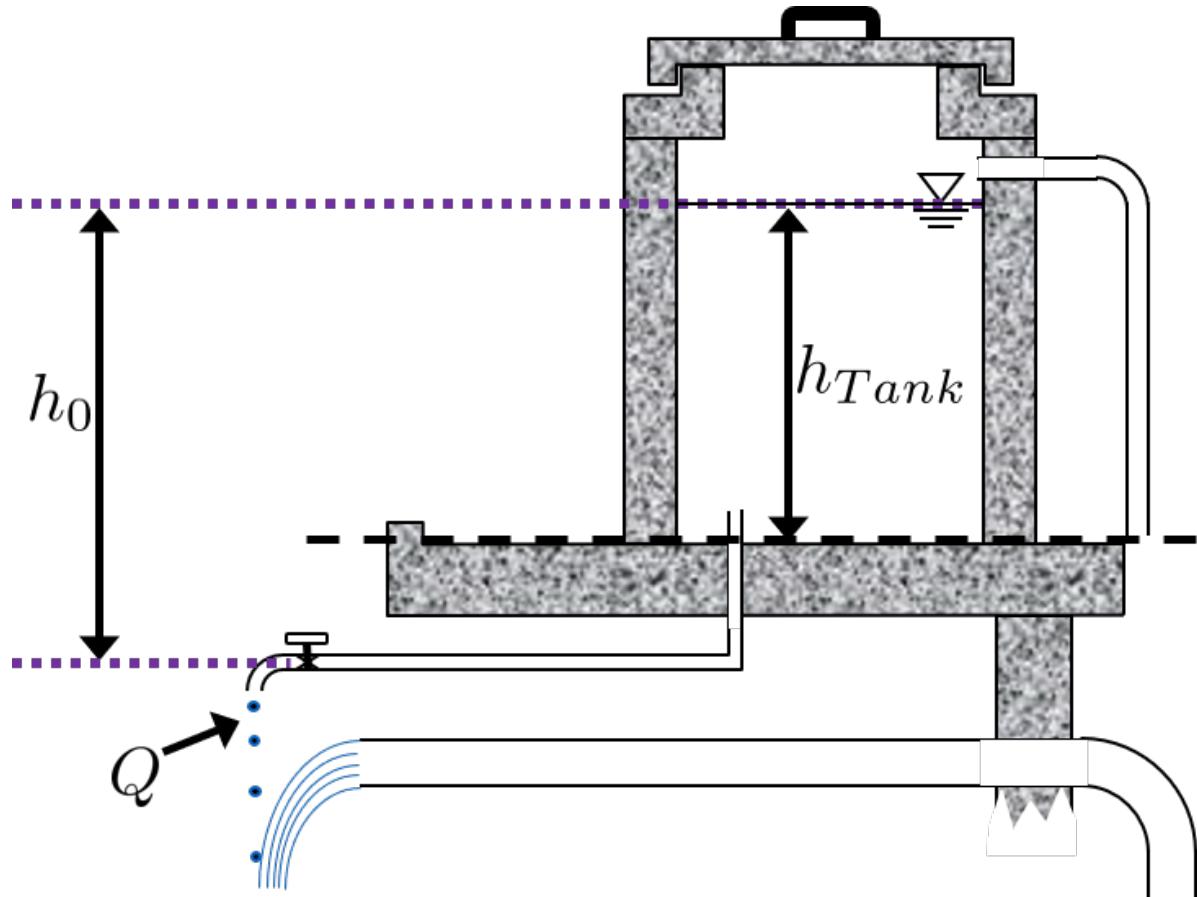


Fig. 9.1: This figure shows the variables that are defined in the equation above.

Using equation (9.1) and this proportionality relationship, we make the following plots. On the left, the valve is at the same elevation as the bottom of the tank, or $h_{Tank} = h_0$. Our attempt to get a continuous flow rate out of this system is to make $\frac{h_{Tank}}{h_0}$ very small by elevating the tank far above the valve. On the right, $\frac{h_{Tank}}{h_0} = \frac{1}{50}$. While the plot looks great and provides essentially constant head, elevating the tank by 50 times its height is not realistic. The tank with a valve is not a solution to the constant head problem.

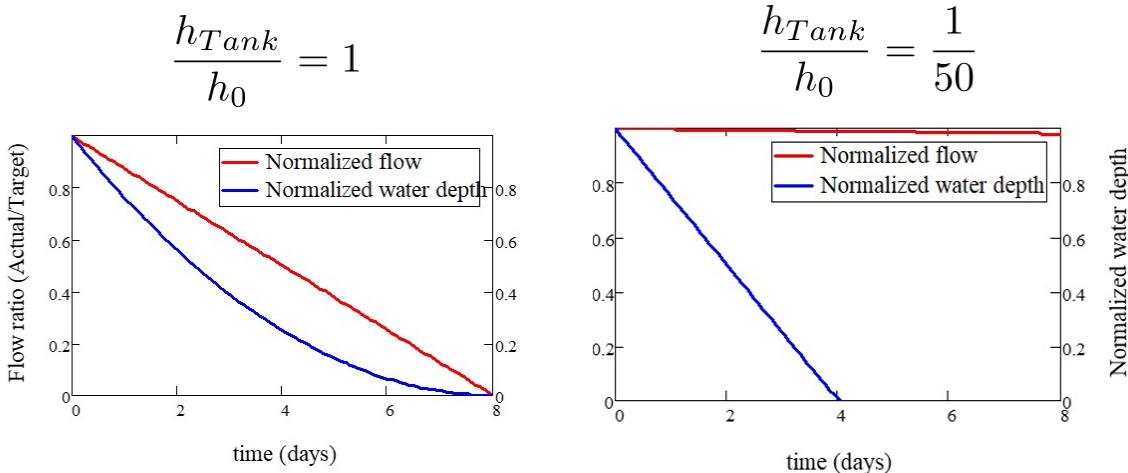


Fig. 9.2: These graphs show how manipulation of the variables in the $Q(t)$ expression can result in effectively constant head.

9.1.2 Drain System for a Tank

While the tank with a valve scenario is not a good constant head solution, we can use our understanding of the system to properly design drain systems for AquaClara reactors like flocculators and sedimentation tanks, since they are just tanks with valves. The derivation for the following equation is here, along with more details on AquaClaras pipe stub method for draining tanks: [and for Tank Drain Equation](#). The derived Tank Drain equation is as follows:

$$D_{Pipe} = \sqrt{\frac{8L_{Tank}W_{Tank}}{\pi t_{Drain}}} \left(\frac{H_{Tank} \sum K}{2g} \right)^{\frac{1}{4}} \quad (9.2)$$

The equation can also be rearranged to solve for the time it would take to drain a tank given its dimensions and a certain drain pipe size:

$$t_{Drain} = \frac{8L_{Tank}W_{Tank}}{\pi D_{Pipe}^2} \left(\frac{H_{Tank} \sum K}{2g} \right)^{\frac{1}{2}} \quad (9.3)$$

Such that:

D_{Pipe} = Diameter of the drain piping

$L_{Tank}, W_{Tank}, H_{Tank}$ = Tank dimensions

t_{Drain} = Time it takes to drain the tank

$\sum K$ = Sum of all the minor loss coefficients in the system

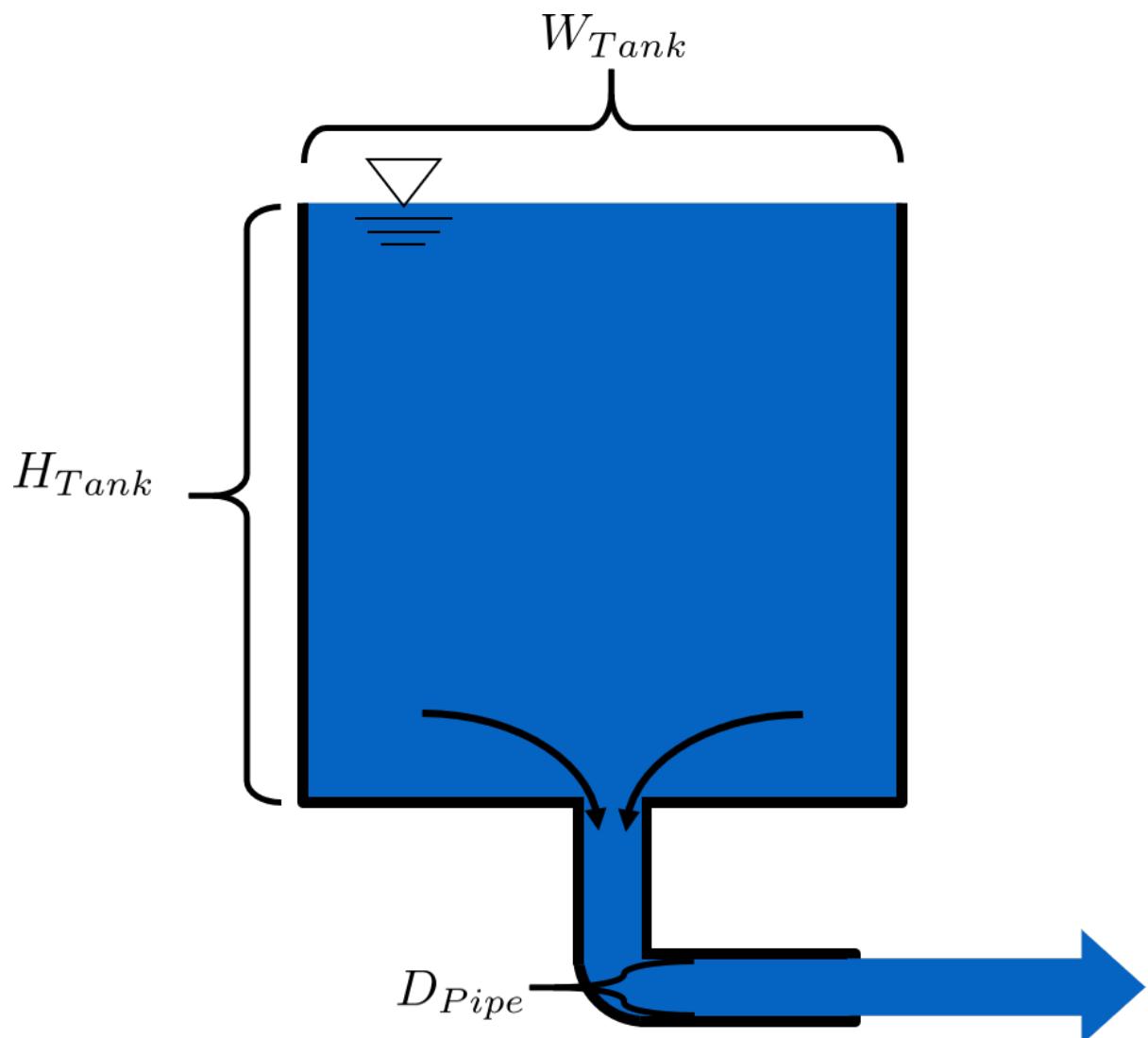


Fig. 9.3: Variables for draining a tank

FLOW CONTROL AND MEASUREMENT DESIGN

This section explores AguaClaras search for constant head in chemical dosing. The term **constant head** means that the driving head of a system, Δz or Δh , does not change over time, even as water flows through or out of the system. Constant head implies constant flow, since the energy driving the flow does not change.

The challenge of constant head in chemical dosing for water treatment plants is not *just* providing one continuous flow of chemicals; it is also varying that flow of chemicals as the flow rate through the plant changes, so that the concentration of chemicals in the raw water stays the same.

10.1 Important Terms and Equations

Terms:

1. Dose
2. Coagulant
3. Chlorination
4. Turbidity
5. Organic Matter
6. Constant Head Tank
7. Sutro weir

Equations:

1. Hagen-Poiseuille equation

10.2 AguaClara Flow Control and Measurement Technologies

Each technology or component for this section will have five subsections:

- **What it is**
- **What it does and why**
- **How it works**
- **Notes**

Before diving into the technologies, recall the purpose of the chemicals that we are seeking to constantly **dose**, and why it is important to keep a constant, specific dose. Also recall that dose means concentration of chemical *in the water we are trying to treat*, not in the stock tanks of the chemicals. **Coagulant** like alum, PAC, and some iron-based

chemicals are used to turn small particles into bigger particles, allowing them to be captured more easily. Waters with high [turbidity](#), indicative of a lot of particles like clay and bacteria, require more coagulant to treat effectively. Additionally, waters with a lot of [organic matter](#) require significantly more coagulant to treat. [Chlorine](#) is used to disinfect water that has already been fully treated. A proper and consistent chlorine dose is required, as too low of a dose creates a risk of reintroduction of pathogens in the distribution system and too high of a dose increases the risk of carcinogenic [disinfection byproduct](#) formation.

Important: This section will often refer to the proportionality between flow Q and head Δh (recall that $\Delta h = h_L$ after applying the head loss trick) by using the proportional to symbol, \propto . It is important to remember that it doesn't necessarily matter whether Q or h_L goes first, $Q \propto \sqrt{h_L}$ is equivalent to saying that $h_L \propto Q^2$.

10.2.1 Almost Linear Flow Controller

What it is

This device consists of a bottle of chemical solution, called the **Constant Head Tank** (CHT), a float valve to keep a solution in the CHT at a constant water level, a flexible tube starting at the bottom of the CHT, and many precisely placed and equally spaced holes in a pipe, as the image below shows. The holes in the pipe hold the other end of the tube that starts at the CHT.

Chemical solution, either coagulant or chlorine, is stored in a stock tank somewhere above the CHT. A different tube connects the stock tank to the float valve within the CHT.

What it does and why

This flow controller provides a constant flow of chemical solution to the water in the plant. When the end of the flexible tube is placed in a hole, the elevation difference between the water level in the bottle and the hole is set and does not change unless the tube is then placed in another hole. Thus, a constant flow is provided while the end of the tube is not moved.

As has been mentioned previously, the amount of chlorine and coagulant that must be added to the raw water changes depending on the flow rate of the plant; the change is necessary to keep the dose constant. More water flowing through the plant means more chlorine is necessary to maintain the dose of chlorine in the treated water. For coagulant, there are also other factors aside from plant flow rate that impact the required dose, including the turbidity and amount of organic matter in the water. The operator must be able to change the dose of both coagulant and chlorine quickly and easily, and they must be able to know the value of the new dose they set. The Almost Linear Flow Controller accomplishes this by having a large number of holes in the flow control pipe next to the CHT. This large number of holes gives the operator many options for adjusting the dose, and let them quickly change the flow of chemicals into the raw water by moving the end of the flexible tube from one hole to another.

How it works

The idea behind this flow controller is to have a linear relationship between Q and h_L (remember that h_L is equal to Δh when we apply the head loss trick), which can be written as $Q \propto h_L$. Here, Q is the flow of chemicals out of the flexible tube, and h_L is the elevation difference between the water level in the CHT and the end of the flexible tube.

As you remember from section 1.5, the summary of Fluids Review, $Q \propto \Delta h$, or $\Delta h \propto Q$ as it was written in the section summary, is only true for the combination of major losses and laminar flow, which makes applicable the Hagen-Poiseuille equation. Therefore, the flow must always be laminar in the flexible tube that goes between the CHT and the holes, and major losses must far exceed minor losses.

It is easy to design for laminar flow, but the Almost Linear Flow Controller was unable to make major losses far exceed minor losses. The bending in the flexible tube caused a lot of minor losses which changed in magnitude depending on exactly how the tube was bent. This made the flow controller almost linear, but that wasn't good enough.

Notes

- This flow controller is **no longer used by AquaClara**.
- The tube connecting the CHT to the outlet of chemicals must really belong and, more importantly, **straight** to form a linear relationship between driving head and flow. This was not true for the Almost Linear Flow Controller. When you read about the Linear Chemical Flow Controller (CDC), you will be learning about the replacement to the Almost Linear Flow Controllers replacement.

10.2.2 Linear Flow Orifice Meter (LFOM)

What it is

The LFOM is a weir shape cut into a pipe. It was meant to imitate the [Sutro Weir](#) while being far easier to build. The LFOM is a pipe with rows of holes, or orifices, drilled into it. There are progressively fewer holes per row as you move up the LFOM, as the shape is meant to resemble half a parabola on each side. The size of all holes is the same, and the amount of holes per row are precisely calculated. Water in the entrance tank flows into and down the LFOM, towards the rapid mix orifice and flocculator.

What it does and why

The LFOM does one thing and serves two purposes.

What it does:

The LFOM creates a linear relationship between water level in the entrance tank and the flow out of the entrance tank. *It does not control the flow through the plant.* If the LFOM were replaced with a hole in the bottom of the entrance tank, the same flow rate would go through the plant, the only difference being that the water level in the entrance tank would scale with flow squared $h \propto Q^2$ instead of $h \propto Q$. For example, if an LFOM has 10 rows of holes and has been designed for a plant whose maximum flow rate is 10 L/s, then the operator knows that the number of rows submerged in water is equal to the flow rate of the plant in L/s. So if the water were up to the third row of holes, there would be 3 L/s of water flowing through the plant.

Why it is useful:

1. Allows the operator to measure the flow through the plant quickly and easily, explained above.
2. Allows for the Linear Chemical Dose Controller, which will be explained next, to automatically adjust the flow of coagulant/chlorine into the plant as the plant flow rate changes. This means the operator would only need to adjust the flow of coagulant when there is a change in turbidity or organic matter.

How it works

This is best understood with examples. By shaping a weir differently, different relationships between Q and h are formed: * In the case of a [rectangular weir](#), $Q \propto h^{\frac{3}{2}}$ * In the case of a [v-notch weir](#), $Q \propto h^{\frac{5}{2}}$ * In the case of a [Sutro weir](#) and thus LFOM, $Q \propto h$.

Sutro Weir



LFOM

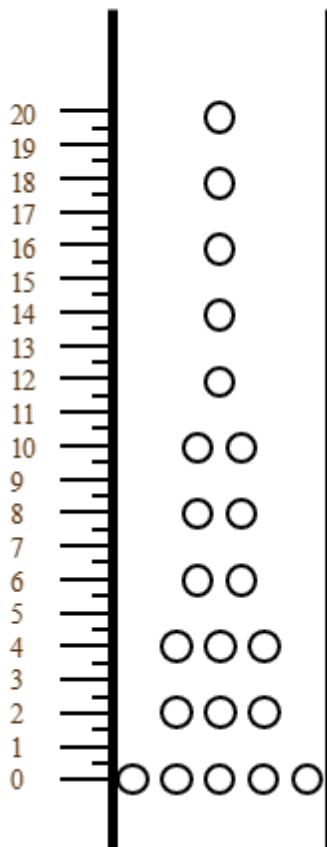


Fig. 10.1: On the left is a sutro weir. On the right is AguaClaras approximation of the sutro weirs geometry. This elegant innovation is called a linear flow orifice meter, or LFOM for short.

Notes

- The LFOM is not perfect. Before the water level reaches the second row of holes, the LFOM is simulating a rectangular weir, and thus $h \propto Q$. The Sutro weir also experiences this problem.
- If the water level exceeds the topmost row of the LFOMs orifices, the linearity also breaks down. The entire LFOM begins to act like an orifice, the exponent of Q in $h \propto Q$ becomes greater than 1. This is because the LFOM approaches orifice behavior, and for orifices, $h \propto Q^2$.

10.2.3 Linear Chemical Dose Controller (CDC)

Since the Linear Chemical Dose Controller has become the standard in AquaClara, it is often simply called the Chemical Dose Controller, **or CDC for short**. It can be confusing to describe with words, so be sure to flip through the slides in the Flow Control and Measurement powerpoint, as they contain very, very, helpful diagrams of the CDC.

What it is

The CDC brings together the LFOM and many improvements to the Almost Linear Flow Controller. Lets break it down, with the image below as a guide.

1. Start at the Constant Head Tank (CHT). This is the same set up as the Almost Linear Flow Controller. The stock tank feeds into the CHT, and the float valve makes sure that the water level in the constant head tank is always the same.
2. Now the tubes. These fix the linearity problems that were the main problem in the Almost Linear Flow Controller.
* The tube connected to the bottom of the CHT is large diameter to minimize any head loss through it.
 - The three thin, straight tubes are designed to generate a lot of major losses and to minimize any minor losses. This is to make sure that major losses far exceed any minor losses, which will ensure that the Hagen-Poiseuille equation is applicable and that flow will be directly proportional to the head, $Q \propto \Delta h$. Why are there 3 tubes?
 1. **3 short instead of 1 short** Removing 2 of the 3 tubes would mean 3 times the flow through the remaining tube. This means the velocity in the tube would be 3 times as fast. Since minor losses scale with v^2 and major losses only scale with v , this would increase the ratio of $\frac{\text{minor losses}}{\text{major losses}}$, which would break the linearity we're trying to achieve. It would also increase the total head loss through the system, resulting in a lower maximum flow rate than before.
 2. **1 long instead of 3 short** One tube whose length is equal to the three combined would be inconveniently long, and would suffer from the same problems as above. There would be even more head loss through the tube, since its length would be longer.
 - The large-diameter tube on the right of the three thin, straight tubes is where the chemicals flow out. The end of the tube is connected to both a slider and a drop tube. The drop tube allows for supercritical flow of the chemical leaving the dosing tubes; once the chemical enters the drop tube it falls freely and no longer affects the CDC system.
3. The slider rests on a lever. This lever is the critical part of the CDC, it connects the water level in the entrance tank, which is adjusted by the LFOM, to the difference in head between the CHT and the end of the dosing tube. This allows the flow of chemicals to automatically adjust to a change in the plant flow rate, maintaining a constant dose in the plant water. One end of the lever tracks the water level in the entrance tank by using a float. The counterweight on the other side of the lever is to make sure the float floats, since this float is usually made of PVC, which is more dense than water.
4. The slider itself controls the dose of chemicals. For any given plant flow rate, the slider can be adjusted to increase or decrease the amount of chemical flowing through the plant.

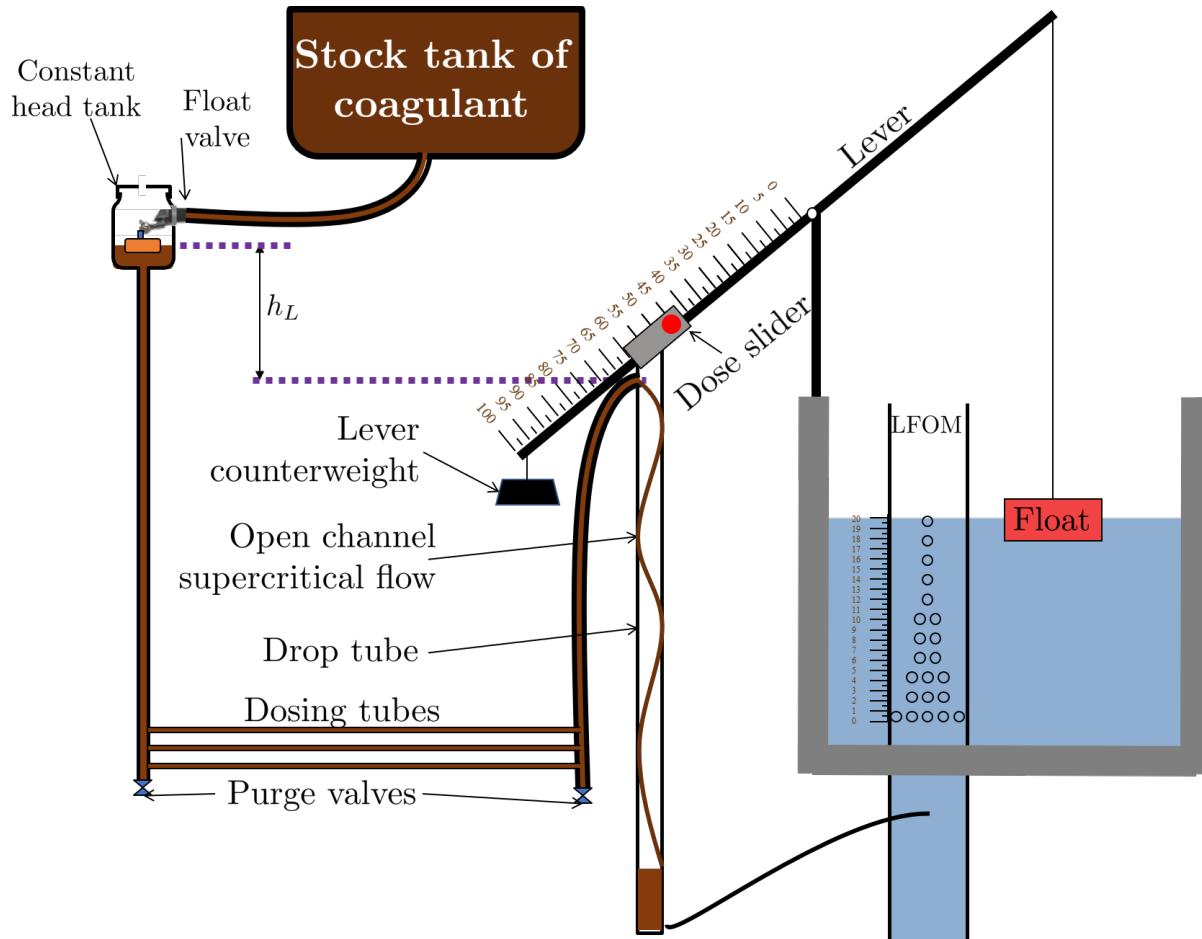


Fig. 10.2: This is the setup of the chemical dose controller.

What it does and why

The CDC makes it easy and accurate to dose chemicals. The flow of chemicals automatically adjusts to changes in the plant flow rate to keep a constant dose, set by the operator. When a turbidity event occurs, the operator can change the dose of coagulant by moving the coagulant slider *lower* on the lever to increase the dose. The slider has labelled marks so the operator can record the dose accurately.

How it works

A lot of design has gone into the CDC. The design equations and their derivations that the following steps are based on can be found here: *Design Equations for the Linear Chemical Dose Controller (CDC)*, and you are very, very strongly encouraged to read them.

The CDC can be designed manually using the equations from the derivation linked above or via aguaclara, using the equations found in `cdc_functions.py`. Either way, the design algorithm is roughly the same:

1. Calculate the maximum flow rate, $Q_{Max, Tube}$, through each available dosing tube diameter D that keeps error due to minor losses below 10% of total head loss. Recall that tubing diameter is an array, as there are many diameters available at hardware stores and suppliers. This means that for each step, there will be as many solutions as there are reasonable diameters available.

$$Q_{Max, Tube} = \frac{\pi D^2}{4} \sqrt{\frac{2h_L g \Pi_{Error}}{\sum K}} \quad (10.1)$$

2. Calculate how much flow of chemical needs to pass through the CDC at maximum plant flow and maximum chemical dose. This depends on the concentration of chemicals in the stock tank.

$$Q_{Max, CDC} = \frac{Q_{Plant} \cdot C_{Dose, Max}}{C_{StockTank}} \quad (10.2)$$

3. Calculate the number of dosing tubes required if the tubes flow at maximum capacity (round up)

$$n_{Tubes} = \text{ceil} \left(\frac{Q_{Max, CDC}}{Q_{Max, Tube}} \right) \quad (10.3)$$

4. Calculate the length of dosing tube(s) that correspond to each available tube diameter.

$$L_{Min} = \left(\frac{gh_L \pi D^4}{128\nu Q_{Max}} - \frac{Q_{Max}}{16\pi\nu} \sum K \right) \quad (10.4)$$

5. Select a tube length from your array of solutions. Pick the longest dosing tube that you can, keeping in mind that the tube(s) must be able to fit in the plant and cant be longer than the length of the plant wall it will be placed along.

6. Finally, select the dosing tube diameter and flow rate corresponding to the selected tube length.

Notes

Nothing in life is perfect, and the CDC is no exception. It has a few causes of inaccuracy which go beyond non-zero minor losses: * Float valves are not perfect. There will still be minor fluctuations of the fluid level in the CHT which will result in imperfect dosing. * Surface tension may resist the flow of chemicals from the dosing tube into the drop tube during low flows. Since the CDC design does not consider surface tension, this is a potential source of error. * The lever and everything attached to it are not weightless. Changing the dose of coagulant or chlorine means moving the slider along the lever. Since the slider and tubes attached to it (drop tube, dosing tube) have mass, moving the slider means that the torque of the lever is altered. This means that the depth that the float is submerged is changed, which affects Δh of the system. This can be remedied by making the floats diameter as large as possible, which makes these fluctuations small. This problem can not be avoided entirely.

10.3 Section Summary

1. Tank with a valve:

$$\frac{Q}{Q_0} = 1 - \frac{1}{2} \frac{t}{t_{Design}} \frac{h_{Tank}}{h_0} \quad (10.5)$$

This equation describes flow Q as a function of time t of a fluid leaving a tank through a valve. Attempting to get this tank with a valve system to yield constant head means raising the tank far, far above the valve that controls the flow. This is unreasonable when designing a flow control system for constant dosing, but can be used to design systems to drain a tank. See the section above for a description of the variables in the equation.

2. **LFOM:** The LFOM makes the water level in the entrance tank linear with respect to the flow out of the entrance tank. This is useful in measuring the flow and is a critical component in AquaClaras chemical dosing system. The LFOM *measures* the flow through the plant, it does not *control* the flow through the plant.
3. **The Linear Chemical Dose Controller (CDC)** combines the:
 - * linear relationship between water level and flow in the entrance tank caused by the LFOM,
 - * linear relationship between elevation difference and flow caused by the Hagen-Poiseuille equation, which is only valid for major losses under laminar flow, and
 - * a lever to link the two linear relationships

To keep the chemical dose constant by automatically adjusting the addition of coagulant and chlorine as the plant flow rate varies. Two sliders on the lever allows the operator to change the dose of coagulant and chlorine independently of the plant flow rate.

FLOW CONTROL AND MEASUREMENT DERIVATIONS

11.1 $Q(t)$ for a Tank with a Valve

This document contains the derivation of the flow through a tank-with-a-valve over time, $Q(t)$. Our reference will be a simple hypochlorinator, shown in the following image. In the image, a hypochlorite solution is slowly dripping and mixing with piped source water, thereby disinfecting it. The valve is almost closed to make sure that the hypochlorite solution drips instead of flows. At the end of this document is an image which shows the variables in the final equation.

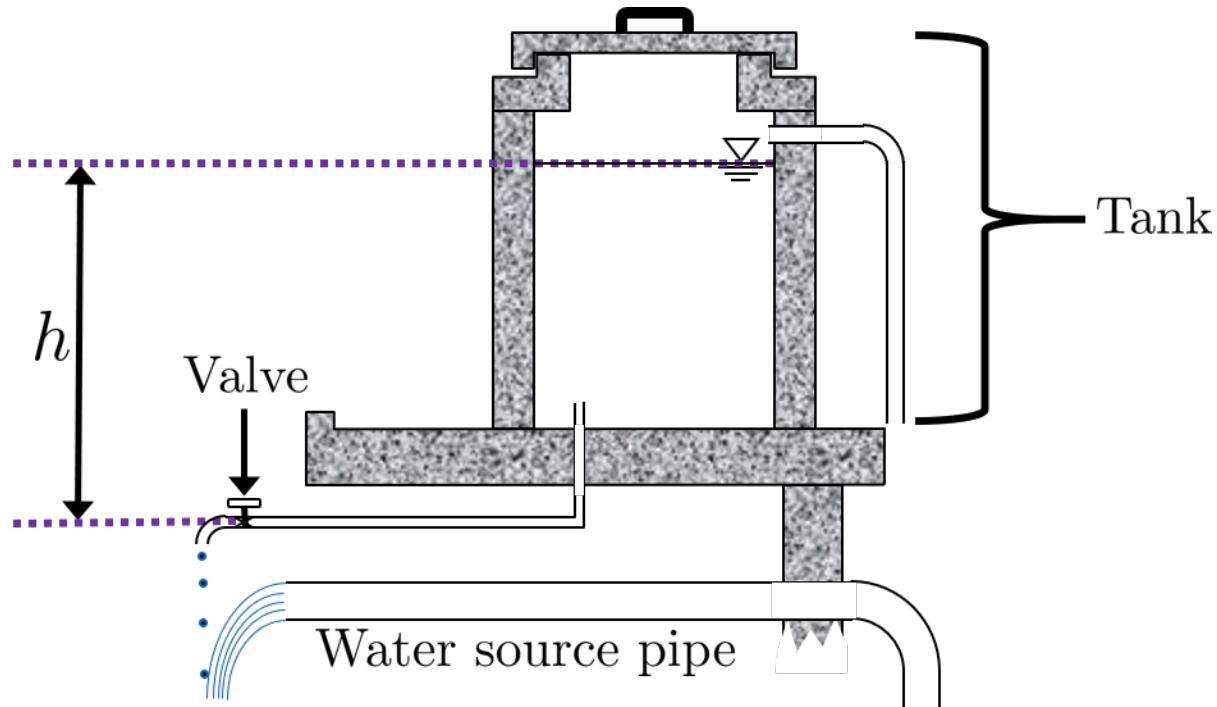


Fig. 11.1: This is a common setup for chlorinating water before distributing it to a nearby community.

This derivation begins by finding two equations for flow, Q , through the hypochlorinator and setting them equal to each other. First, the rate of change of the volume of hypochlorite solution in the tank is equivalent to the flow out of the hypochlorinator. Since the volume of hypochlorite solution in the tank is equal to the tanks cross-sectional area times its height, we get the following equation:

$$Q = -\frac{dV}{dt} = -\frac{A_{\text{Tank}} dh}{dt} \quad (11.1)$$

Such that:

$\frac{dV}{dt}$ = rate of change in volume of solution in the tank

$\frac{dh}{dt}$ = rate of change in height of water (hypochlorite solution) level with time

Our other equation for flow is the head loss equation. Since major losses are negligible for a short pipe-low flow rate system, we only need to consider minor losses. The only real minor loss in this system occurs in the almost-closed valve that is dripping the hypochlorite solution. However, we will also use the head loss trick. Therefore, the total driving head of the system h is equal to the minor losses:

$$h = h_e = \left(\sum K \right) \frac{Q^2}{2gA_{Valve}^2} \quad (11.2)$$

Bear in mind that this is the second form of the minor loss equation as described in [this derivation](#). Rearranging the minor loss equation to solve for Q , it looks like this:

$$Q = A_{Valve} \sqrt{\frac{2h_e g}{\sum K}} \quad (11.3)$$

Now we can set both equations for Q equal to each other and move them both to one side:

$$A_{Tank} \frac{dh}{dt} + A_{Valve} \sqrt{\frac{2gh}{\sum K}} = 0 \quad (11.4)$$

From here, calculus and equation substitution dominate the derivation. Separating the variables of the equation immediately above, we get the following integral:

$$\frac{-A_{Tank}}{A_{Valve} \sqrt{\frac{2g}{\sum K}}} \int_{h_0}^h \frac{dh}{\sqrt{h}} = \int_0^t dt \quad (11.5)$$

Which, when integrated, yields:

$$\frac{-A_{Tank}}{A_{Valve} \sqrt{\frac{2g}{\sum K}}} \cdot 2 \left(\sqrt{h} - \sqrt{h_0} \right) = t \quad (11.6)$$

And solved for \sqrt{h} returns:

$$\sqrt{h} = \sqrt{h_0} - t \frac{A_{Valve}}{2A_{tank}} \sqrt{\frac{2g}{\sum K}} \quad (11.7)$$

At this point, the steps and equation substitutions may begin to seem unintuitive. Do not worry if you do not understand why *exactly* a particular substitution is occurring. Since we determined above that $h_e = h$, our equation above for \sqrt{h} is also an equation for $\sqrt{h_e}$. As such, we will plug the equation above back into the minor loss equation solved for Q from above, $Q = A_{Valve} \sqrt{\frac{2h_e g}{\sum K}}$, to produce:

$$Q = A_{Valve} \sqrt{\frac{2g}{\sum K}} \left(\sqrt{h_0} - t \frac{A_{Valve}}{2A_{tank}} \sqrt{\frac{2g}{\sum K}} \right) \quad (11.8)$$

Now we can focus on getting rid of the variables A_{Valve} , $\sum K$, and A_{tank} . By using the minor loss equation once more, we can remove both A_{Valve} and $\sum K$. Consider the initial state of the system, when the hypochlorinator is set up and starts dropping its first few drops of hypochlorite solution into the water. The initial flow rate, Q_0 , and elevation difference between the water level and the valve, h_0 , can be input into the minor loss equation, which can then be solved for A_{Valve} :

$$A_{Valve} = \frac{Q_0}{\sqrt{\frac{2h_0 g}{\sum K}}} \quad (11.9)$$

Plugging this equation for A_{Valve} into the equation for Q just above, we get the following two equations, in which the second equation is a simplified version of the first:

$$Q = Q_0 \frac{1}{\sqrt{h_0}} \left(\sqrt{h_0} - \frac{Q_0 t}{2A_{Tank}\sqrt{h_0}} \right) \quad (11.10)$$

$$\frac{Q}{Q_0} = 1 - \frac{tQ_0}{2A_{Tank}h_0} \quad (11.11)$$

This next step will eliminate A_{Tank} . However, it requires some clever manipulation that has a tendency to cause some confusion. We will define a new parameter, t_{Design} , which represents the time it would take to empty the tank if the initial flow rate through the valve, Q_0 , stays constant in time. Of course, the flow Q through the valve does not stay constant in time, which is why this derivation document exists. But imagining this hypothetical t_{Design} parameter allows us to form the following equation:

$$Q_0 t_{Design} = A_{Tank} h_{Tank} \quad (11.12)$$

This equation describes draining all the hypochlorite solution from the tank. The volume of the solution, $A_{Tank} h_{Tank}$, is drained in t_{Design} . Rearranged, the equation becomes:

$$\frac{Q_0}{A_{Tank}} = \frac{h_{Tank}}{t_{Design}} \quad (11.13)$$

Such that:

h_{Tank} = elevation of water level in the tank with reference to tank bottom at the initial state, $t = 0$

Here lies another common source of confusion. h_{Tank} is not the same as h_0 . h_{Tank} is the height of water level in the tank with reference to the tank bottom. h_0 is the water level in the tank with reference to the valve. Therefore, $h_0 \geq h_{Tank}$ is true if the valve is located at or below the bottom of the tank. If the tank is elevated far above the valve, then the $h_0 \gg h_{Tank}$. If the valve is at the same elevation as the bottom of the tank, then $h_0 = h_{Tank}$. Please refer to the following image to clarify h_0 and h_{Tank} . Also note that both h_{Tank} and h_0 are not variables, they are constants which are defined by the initial state of the hypochlorinator, when the solution just begins to flow.

Finally, our fabricated equivalence, $\frac{Q_0}{A_{Tank}} = \frac{h_{Tank}}{t_{Design}}$ can be plugged into $\frac{Q}{Q_0} = 1 - \frac{tQ_0}{2A_{Tank}h_0}$ to create the highly useful equation for flow rate as a function of time for a drip hypochlorinator:

$$\frac{Q}{Q_0} = 1 - \frac{1}{2} \frac{t}{t_{Design}} \frac{h_{Tank}}{h_0} \quad (11.14)$$

Which can be slightly rearranged to yield:

$$Q(t) = Q_0 \left(1 - \frac{1}{2} \frac{t}{t_{Design}} \frac{h_{Tank}}{h_0} \right) \quad (11.15)$$

Such that:

$Q = Q(t)$ = flow of hypochlorite through valve at time t

t = elapsed time

t_{Design} = time it would take for tank to empty if flow stayed constant at Q_0 , which it does not

h_{Tank} = elevation of water level with reference to tank bottom

h_0 = elevation of water level with reference to the valve

How does this tank with a valve scenario differ from the hole in a bucket scenario? some might ask. If you are interested, you may go through the derivation on your own using the orifice equation instead of the minor loss equation for the first step. If you do so you'll find that the equation remains almost the same, the only difference being that the $\frac{h_{Tank}}{h_0}$ term drops out for an orifice, as $h_{Tank} = h_0$. The big difference in the systems lies with the flexibility of having a valve. It can be tightened or loosened to change the flow rate, whereas changing the size of an orifice multiple times in a row is not recommended and is usually irreversible.

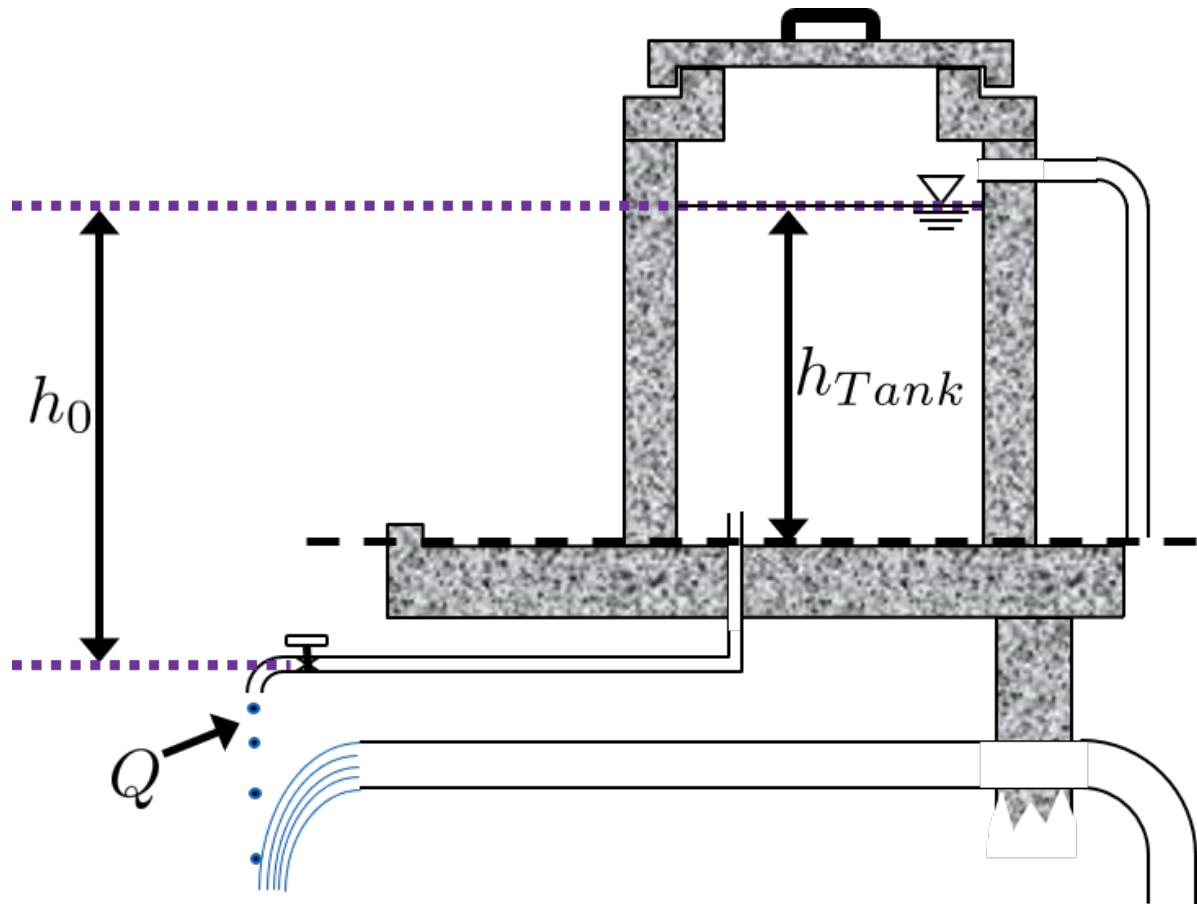


Fig. 11.2: Q_0 = initial flow rate of hypochlorite solution at time $t = 0$, t_{Design} = time it would take to drain the tank if flow was held constant at Q_0

11.2 $D(t)$ and $t(D)$ for Tank Drain Equation

This document contains the derivation of D_{Pipe} , which is the pipe diameter necessary to install in a drain system to entirely drain a tank in time t_{Drain} .

First, it is necessary to understand how AquaClara tank drains work and what they look like. Many tanks, including the flocculator and entrance tank, have a hole in their bottoms which are fitted with [pipe couplings](#). During normal operation, these couplings have pipe stubs in them, and the pipe stubs are tall enough to go above the water level in the tank and not allow water to flow into the drain. When the pipe stub is removed, the water begins to flow out of the drain, as the image below indicates. The drain pipe consists of pipe and one elbow, shown in the image.

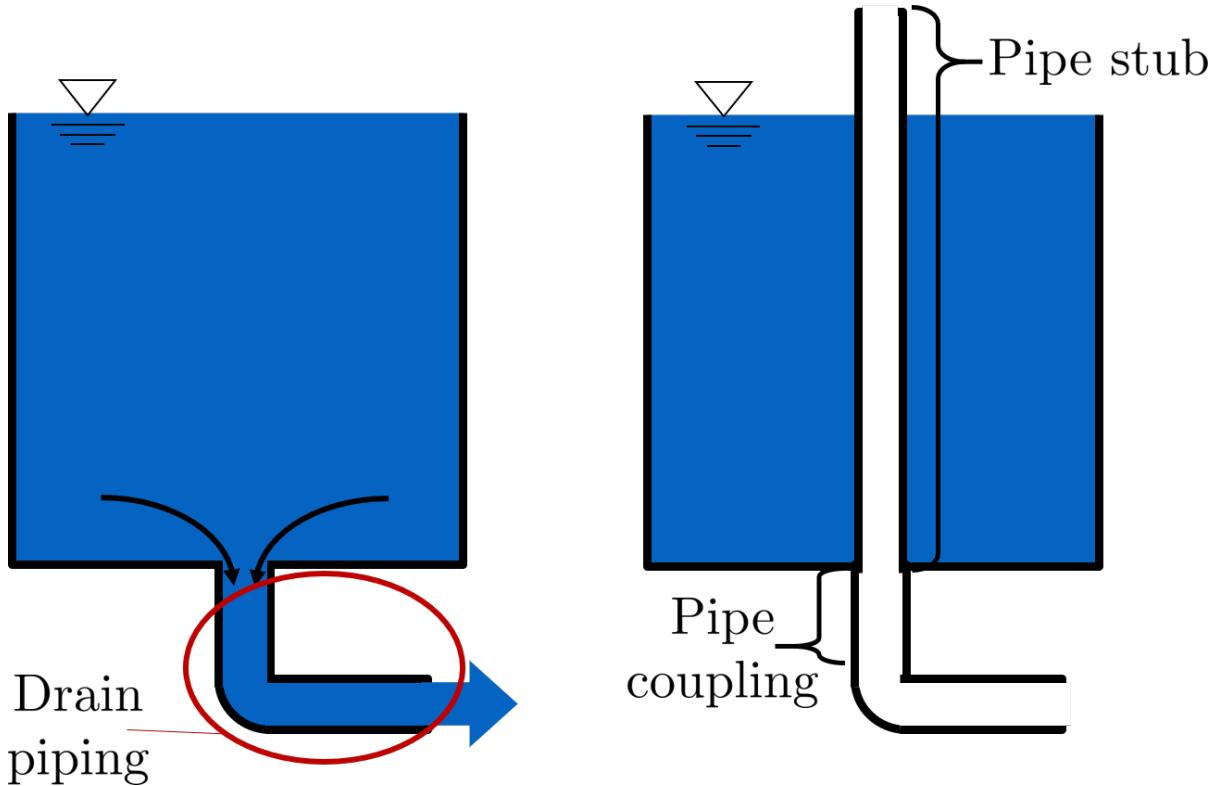


Fig. 11.3: This is AquaClaras alternatives to having valves.

While AquaClara sedimentation tanks use valves instead of pipe to begin the process of draining, the actual drain piping system is the same, pipe and an elbow. The equation that will soon be derived applies to both pipe stub and valve drains.

We will start the derivation from the following equation, which is found in an intermediate step from the [Q\(t\) for a Tank with a Valve](#). While this system does not have a valve, it has other sources of minor loss and therefore the equation is still valid.

$$\sqrt{h} = \sqrt{h_0} - t \frac{A_{Valve}}{2A_{Tank}} \sqrt{\frac{2g}{K}} \quad (11.16)$$

We need to make some adjustments to this equation before proceeding, to make it applicable for this new drain-system scenario. First, we want to assume that the tank has fully drained. Thus, $t = t_{Drain}$ and $h = 0$. Next, we recall that the tank drain is not actually a valve, but just pipe and an elbow, so $A_{Valve} = A_{Pipe}$. Additionally, there can be multiple points of minor loss in the drain system: the entrance from the tank into the drain pipe, the elbow, and potentially the exit of the water out of the drain pipe. When considering a sedimentation tank, the open valve required

to begin drainage also has a minor loss associated with it. Therefore, it is necessary to substitute $\sum K$ for K . With these substitutions, the equation becomes:

$$\sqrt{h_0} = t_{Drain} \frac{A_{Pipe}}{2A_{Tank}} \sqrt{\frac{2g}{\sum K}} \quad (11.17)$$

Now, with the knowledge that $A_{Pipe} = \frac{\pi D_{Pipe}^2}{4}$ and rearranging to solve for D_{Pipe} , we obtain the following equation:

$$D_{Pipe} = \sqrt{\frac{8A_{Tank}}{\pi t_{Drain}}} \sqrt{\frac{h_0 \sum K}{2g}} \quad (11.18)$$

To get the equation in terms of easily measureable tank parameters, we substitute $L_{Tank}W_{Tank}$ for A_{Tank} . To maintain consistency in variable names, we substitute H_{Tank} for h_0 .

Note: By saying that $h_0 = H_{Tank}$, we are making the assumption that the pipe drain is at the same elevation as the bottom of the tank. The pipe drain is actually a little lower than the bottom of the tank, but that would make the tank drain faster than t_{Drain} , which is preferred. Therefore, we are designing a slight safety factor when we say that $h_0 = H_{Tank}$.

Finally, we arrive at the equation for drain pipe sizing:

$$D_{Pipe} = \sqrt{\frac{8L_{Tank}W_{Tank}}{\pi t_{Drain}}} \left(\frac{H_{Tank} \sum K}{2g} \right)^{\frac{1}{4}} \quad (11.19)$$

We can also easily rearrange to find the time required to drain a tank given a drain diameter:

$$t_{Drain} = \frac{8L_{Tank}W_{Tank}}{\pi D_{Pipe}^2} \sqrt{\frac{H_{Tank} \sum K}{2g}} \quad (11.20)$$

Such that the variables are as the appear in the image below.

11.3 Design Equations for the Linear Chemical Dose Controller (CDC)

This document will include the equation derivations required to design a CDC system. The most important restriction in this design process is maintaining linearity between head h and flow Q , which is the entire purpose of the CDC. Recall that major losses under laminar flow scale with Q and minor losses scale with Q^2 . Since it is impossible to remove minor losses from the system entirely, we will simply try to make minor losses very small compared to major losses. The CDC does this by including dosing tube(s), which are long, straight tubes designed to generate a lot of major losses. There can be one tube or multiple, depending on the design conditions.

We will use the head loss trick that was introduced in the Fluids Review section. Therefore, the elevation difference between the water level in the constant head tank (CHT) and the end of the tube connected to the slider, Δh , is equal to the head loss between the two points, h_L . Thus, $\Delta h = h_L = h_e + h_f$.

11.3.1 CDC Design Equation Derivation

Important: When designing the CDC, there are a few parameters which are picked and set initially, before applying any equations. These parameters are:

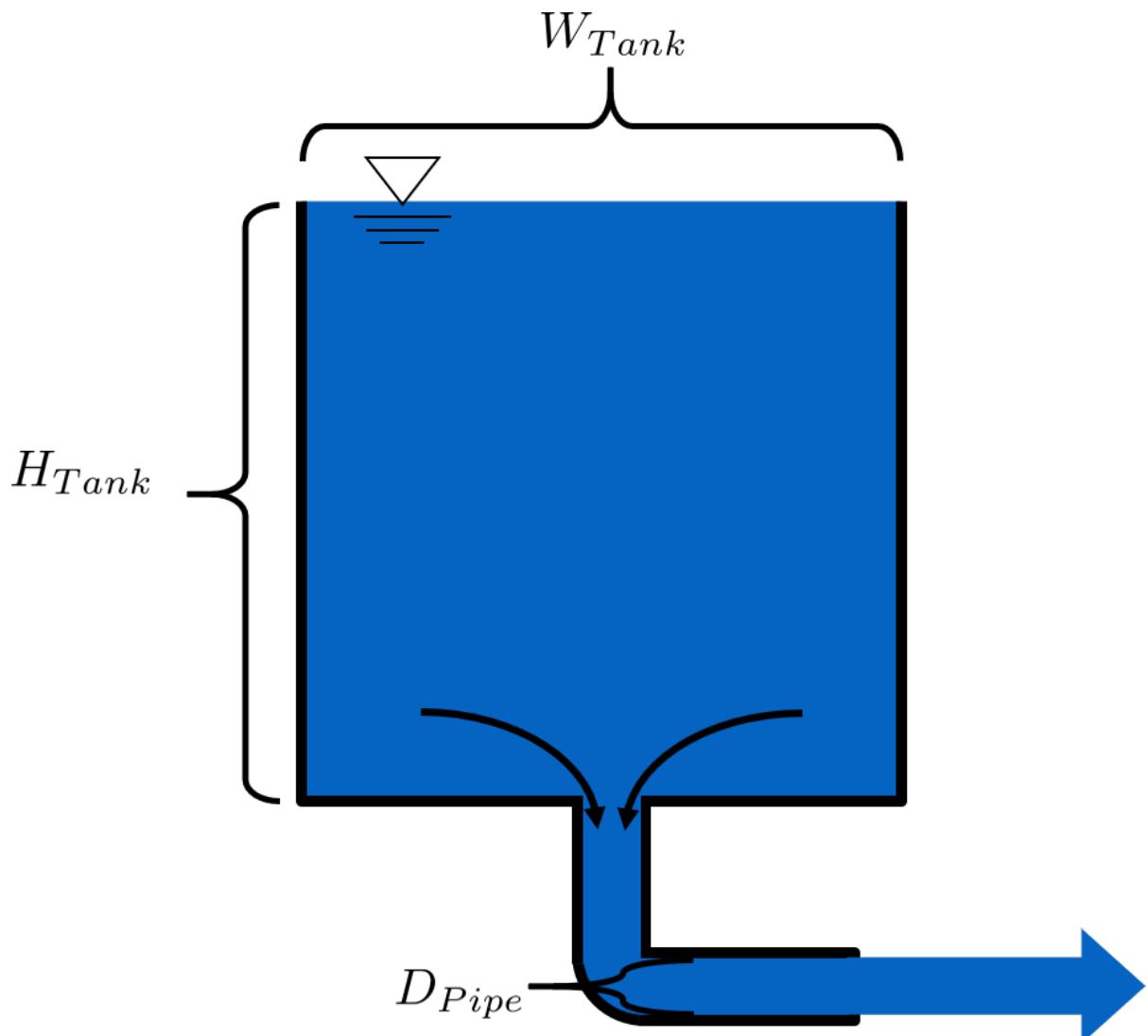


Fig. 11.4: L_{Tank} is the length of the tank which goes the page. K is the aggregate minor loss coefficient of the drain system.

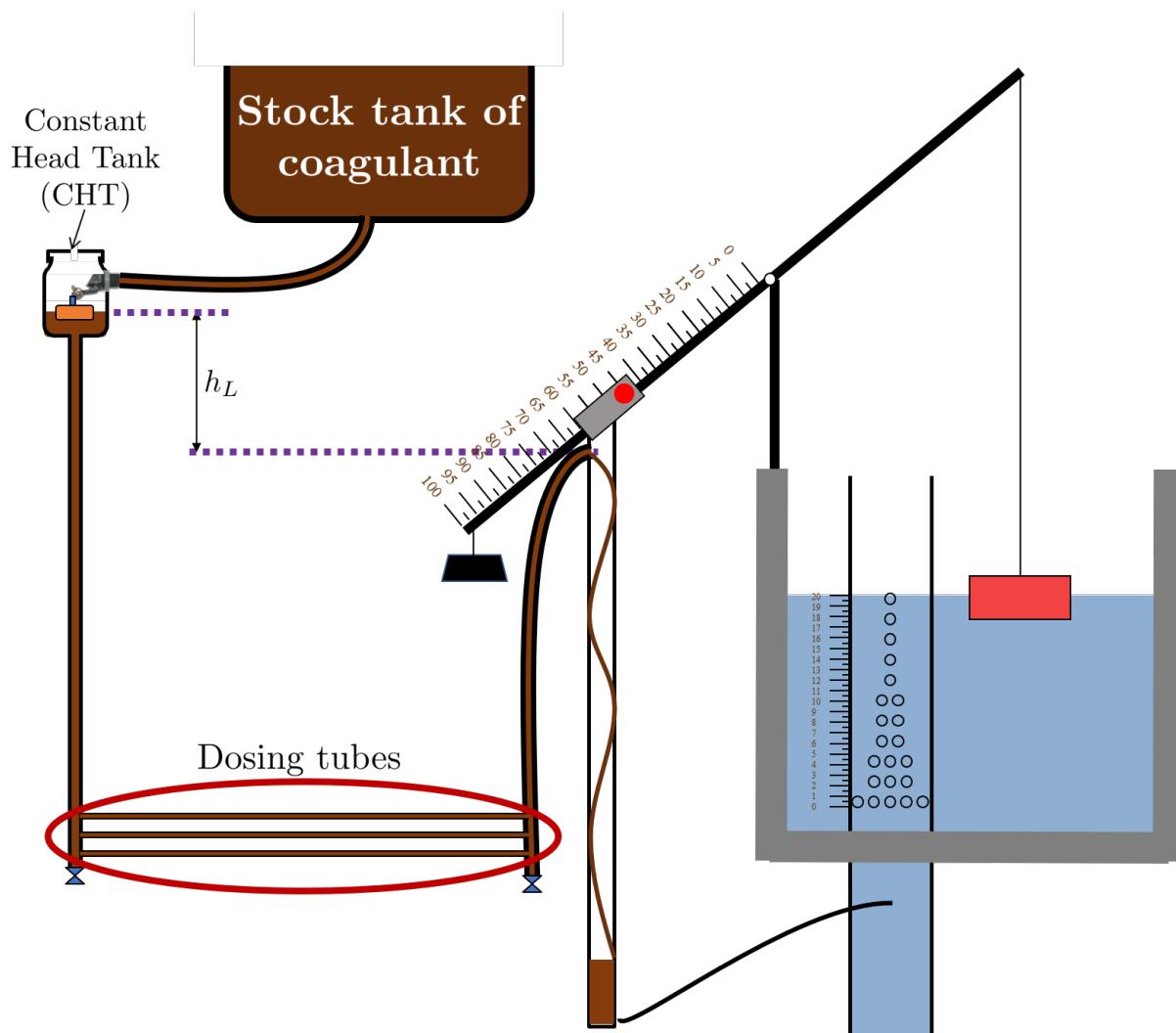


Fig. 11.5: Visual representation of CDC.

1. D = tube diameter. only certain tubing diameters are manufactured (like $\frac{x}{16}$ inch), so an array of available tube diameters is set initially.
2. $\sum K$ = sum of minor loss coefficients for the whole system. This is also set initially, it is usually 2.
3. $h_{L_{Max}}$ = maximum elevation difference between CHT water level and outlet of solution. This parameter is usually 20 cm.

We begin by defining the head loss through the system h_L , which is equivalent to defining the driving head Δh . Major losses will be coded as red.

$$h_f = \frac{128\nu L Q}{g\pi D^4} \quad (11.21)$$

Such that:

ν = kinematic viscosity of the solution going through the dosing tube(s). This is either coagulant or chlorine

Q = flow rate through the dosing tube(s)

L = length of the dosing tube(s)

Note: Tube(s) is used because there may be 1 or more dosing tubes depending on the particular design.

Minor losses are equal to:

$$h_e = \frac{8Q^2}{g\pi^2 D^4} \sum K \quad (11.22)$$

Therefore, the total head loss is a function of flow, and is shown in the following equation.

$$h_L(Q) = \frac{128\nu L Q}{g\pi D^4} + \frac{8Q^2}{g\pi^2 D^4} \sum K \quad (11.23)$$

Blue will be used to reference *actual* head loss from now on. This is the same equation as above.

$$h_L(Q) = \left(\frac{128\nu L}{g\pi D^4} + \frac{8Q}{g\pi^2 D^4} \sum K \right) Q \quad (11.24)$$

This equation is not linear with respect to flow. We can make it linear by turning the variable Q in the $\frac{8Q}{g\pi^2 D^4} \sum K$ term into a constant. To do this, we pick a maximum flow rate of coagulant/chlorine through the dose controller, Q_{Max} , and put that into the term in place of Q . The term becomes $\frac{8Q_{Max}}{g\pi^2 D^4} \sum K$, and our linearized model of head loss, coded as green, becomes:

$$h_{L_{linear}}(Q) = \left(\frac{128\nu L}{g\pi D^4} + \frac{8Q_{Max}}{g\pi^2 D^4} \sum K \right) Q \quad (11.25)$$

Here is a plot of the three colored equations above. Our goal is to minimize the minor losses in the system; to bring the red and blue curves as close as possible to the green one.

Designing for the error constraint, Π_{Error}

Important: The first step in the design is to make sure that major losses far exceed minor losses. This will result in an equation for the maximum velocity that can go through the dosing tube(s), \bar{v}_{Max} .

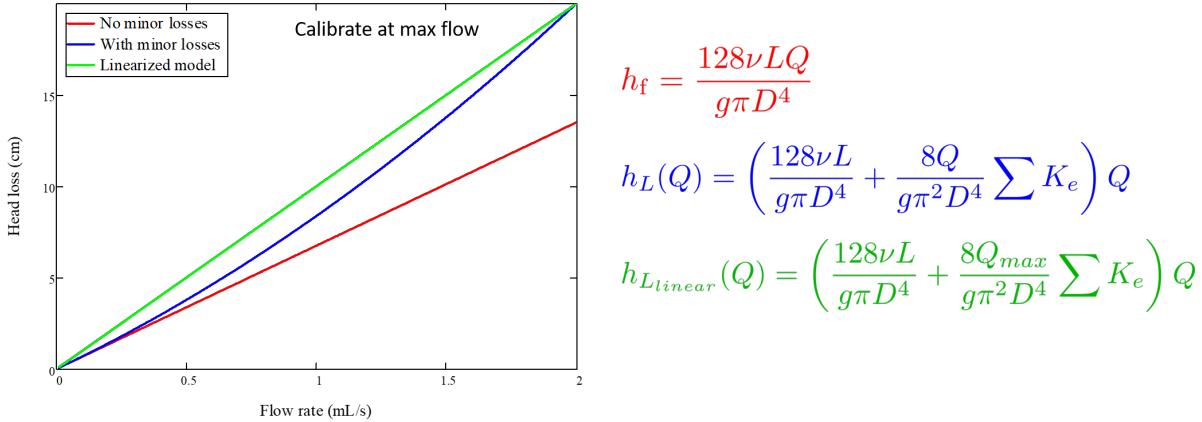


Fig. 11.6: MathCAD generated graph for linearity error analysis. TODO: make this in python

Minor losses will never be 0, so how much error in our linearity are we willing to accept? Lets define a new parameter, Π_{Error} , as the maximum amount of error we are willing to accept. We are ok with 10% error or less, so $\Pi_{Error} = 0.1$.

$$\Pi_{Error} = \frac{h_{L_{linear}} - h_L}{h_{L_{linear}}} = 1 - \frac{h_L}{h_{L_{linear}}} \quad (11.26)$$

$$1 - \Pi_{Error} = \frac{h_L}{h_{L_{linear}}} \quad (11.27)$$

Now we plug $h_L(Q)$ and $h_{L_{linear}}$ back into the equation for $1 - \Pi_{Error}$ and take the limit as $Q \rightarrow 0$, as that is when the relative difference between actual head loss and our linear model for head loss is the greatest.

$$1 - \Pi_{Error} = \frac{\left(\frac{128\nu L}{g\pi D^4} + \frac{8Q}{g\pi^2 D^4} \sum K \right) Q}{\left(\frac{128\nu L}{g\pi D^4} + \frac{8Q_{Max}}{g\pi^2 D^4} \sum K \right) Q} = \frac{\left(\frac{128\nu L}{g\pi D^4} \right)}{\left(\frac{128\nu L}{g\pi D^4} + \frac{8Q_{Max}}{g\pi^2 D^4} \sum K \right)} \quad (11.28)$$

The next steps are algebraic rearrangements to solve for L . This L describes the *minimum* length of dosing tube necessary to meet our error constraint at *maximum* flow. Thus, we will refer to it as $L_{Min, \Pi_{Error}}$.

$$(1 - \Pi_{Error}) \frac{128\nu L}{g\pi D^4} + (1 - \Pi_{Error}) \frac{8Q_{Max}}{g\pi^2 D^4} \sum K = \frac{128\nu L}{g\pi D^4} \quad (11.29)$$

$$-\Pi_{Error} \frac{128\nu L}{g\pi D^4} + (1 - \Pi_{Error}) \frac{8Q_{Max}}{g\pi^2 D^4} \sum K = 0 \quad (11.30)$$

$$L = \left(\frac{1 - \Pi_{Error}}{\Pi_{Error}} \right) \frac{Q_{Max}}{16\nu\pi} \sum K \quad (11.31)$$

$$L_{Min, \Pi_{Error}} = L = \left(\frac{1 - \Pi_{Error}}{\Pi_{Error}} \right) \frac{Q_{Max}}{16\nu\pi} \sum K \quad (11.32)$$

Note that this equation is independent of head loss.

Unfortunately, both $L_{Min, \Pi_{Error}}$ and Q_{Max} are unknowns. We can plug this equation for $L_{Min, \Pi_{Error}}$ back into the head loss equation at maximum flow, which is $h_{L_{Max}} = \left(\frac{128\nu L Q_{Max}}{g\pi D^4} + \frac{8Q_{Max}^2}{g\pi^2 D^4} \sum K \right)$ and rearrange for Q_{Max} to get:

$$Q_{Max} = \frac{\pi D^2}{4} \sqrt{\frac{2h_{L_{Max}} g \Pi_{Error}}{\sum K}} \quad (11.33)$$

See also:

Function in aquaclarar `cdc.max_linear_flow(Diam, HeadlossCDC, Ratio_Error, KMinor)` Returns the maximum flow Q_{Max} that can go through a dosing tube will making sure that linearity between head loss and flow is conserved.

From this equation for Q_{Max} , we can get to our first design equation, \bar{v}_{Max} by using the continuity equation $\bar{v}_{Max} = \frac{Q_{Max}}{\frac{\pi D^2}{4}}$

$$\bar{v}_{Max} = \sqrt{\frac{2h_L g \Pi_{Error}}{\sum K}} \quad (11.34)$$

Designing for the proper amount of head loss, $h_{L_{Max}}$

Important: The second step in the design is to make sure that the maximum head loss corresponds to the maximum flow of chemicals. This will result in an equation for the length of the dosing tube(s), L_{Min} .

We previously derived an equation for the minimum length of the dosing tube(s), $L_{Min, \Pi_{Error}}$, which was the minimum length needed to ensure that our linearity constraint was met. This equation is shown again below, in red:

$$L_{Min, \Pi_{Error}} = \left(\frac{1 - \Pi_{Error}}{\Pi_{Error}} \right) \frac{Q_{Max}}{16\nu\pi} \sum K \quad (11.35)$$

This equation does not, however, account for getting to the proper amount of head loss. If we were to use this equation to design the dosing tubes, we might not end up with the proper amount of flow Q_{Max} at the maximum head loss $h_{L_{Max}}$. So we need to double check to make sure that we get our desired head loss.

First, consider the head loss at maximum flow that was used to get the equation for Q_{Max} :

$$h_{L_{Max}} = \left(\frac{128\nu L Q_{Max}}{g\pi D^4} + \frac{8Q_{Max}^2}{g\pi^2 D^4} \sum K \right) \quad (11.36)$$

Now that we know all of the parameters in this equation except for L , we can solve the equation for L . This the *shortest* tube that generates our required head loss, $h_{L_{Max}}$.

$$L_{Min, headloss} = L = \left(\frac{gh_{L_{Max}}\pi D^4}{128\nu Q_{Max}} - \frac{Q_{Max}}{16\pi\nu} \sum K \right) \quad (11.37)$$

See also:

Function in aquaclarar: `cdc._length_cdc_tube_array(FlowPlant, ConcDoseMax, ConcStock, DiamTubeAvail, HeadlossCDC, temp, en_chem, KMinor)` Returns L_{Min} , takes in the flow rate input of *plant design flow rate*.

See also:

Function in aquaclarar: `cdc._len_tube(Flow, Diam, HeadLoss, conc_chem, temp, en_chem, KMinor)` Returns L_{Min} , takes in the flow rate input of *max flow rate through the dosing tube(s)*.

If you decrease the max flow Q_{Max} and hold $h_{L_{Max}}$ constant, $L_{Min, headloss}$ becomes larger. This means that a CDC system for a plant of 40 $\frac{L}{s}$ must be different than one for a plant of 20 $\frac{L}{s}$. If we want to maintain the same head loss at maximum flow in both plants, then the dosing tube(s) will need to be a lot longer for the 20 $\frac{L}{s}$ plant.

To visualize the distinction between $L_{Min, \Pi_{Error}}$ and $L_{Min, headloss}$, see the following plot. $L_{Min, headloss}$ is discontinuous because it takes in the smallest allowable tube diameter as an input. As the chemical flow rate through the

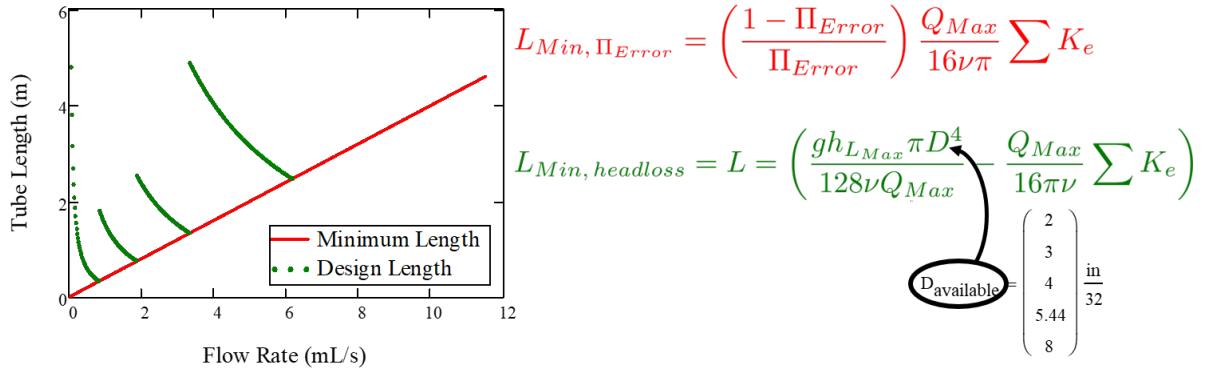


Fig. 11.7: CDC length modeling in MathCAD.

dosing tube(s) decreases, the dosing tube diameter does as well. Whenever you see a jump in the green points, that means the tubing diameter has changed.

As you can see, the head loss constraint is more limiting than the linearity constraint when designing for tube length. Therefore, the design equation for tube length is the one which accounts for head loss. This is the second and final design equation for designing the CDC:

$$L_{Min} = L_{Min, headloss} = \left(\frac{gh_{L_{Max}}\pi D^4}{128\nu Q_{Max}} - \frac{Q_{Max}}{16\pi\nu} \sum K \right) \quad (11.38)$$

The equations for \bar{v}_{Max} and L_{Min} are the only ones you **need** to manually design a CDC.

CDC Dosing Tube(s) Diameter D_{Min} Plots

Below are equations which also govern the CDC and greatly aid in understanding the physics behind it, but are not strictly necessary in design.

By rearranging $Q_{Max} = \frac{\pi D^2}{4} \sqrt{\frac{2h_L g \Pi_{Error}}{\sum K}}$, we can solve for D to get the *minimum* diameter we can use assuming the shortest tube possible that meets the error constraint, $L_{Min, \Pi_{Error}}$. If we use a diameter smaller than $D_{Min, \Pi_{Error}}$, we will not be able to simultaneously reach Q_{Max} and meet the error constraint Π_{Error} .

$$D_{Min, \Pi_{Error}} = \left[\frac{8Q_{Max}^2 \sum K}{\Pi_{Error} h_L g \pi^2} \right]^{\frac{1}{4}} \quad (11.39)$$

We can also find the minimum diameter needed to guarantee laminar flow, which is another critical condition in the CDC design. We can do this by combining the equation for Reynolds number at the maximum Re for laminar flow, $Re_{Max} = 2100$ with the continuity equation at maximum flow:

$$Re_{Max} = \frac{\bar{v}_{Max} D}{\nu} \quad (11.40)$$

$$\bar{v}_{Max} = \frac{4Q_{Max}}{\pi D^2} \quad (11.41)$$

To get:

$$D_{Min, Laminar} = \frac{4Q_{Max}}{\pi \nu Re_{Max}} \quad (11.42)$$

Combined with the discrete amount of tubing sizes (shown in dark green), we can create a graph of the three diameter constraints:

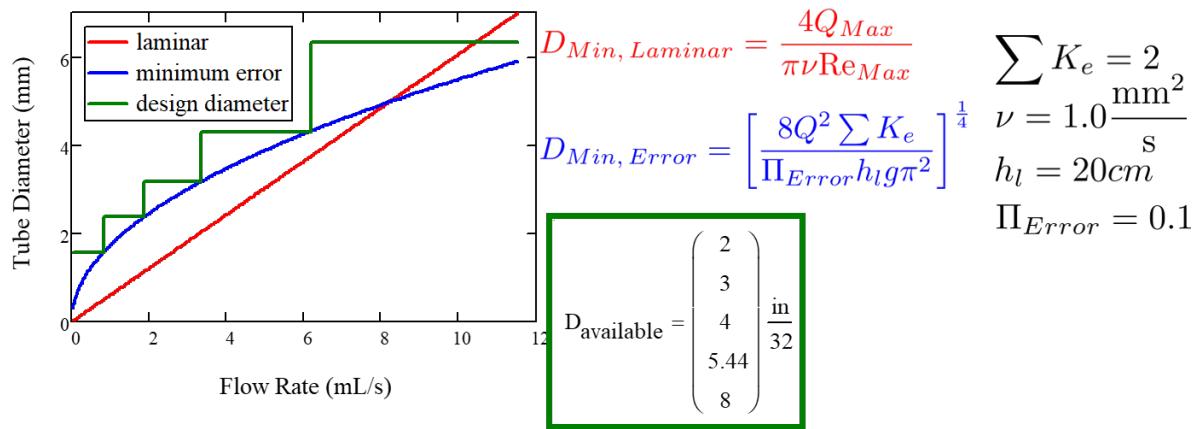


Fig. 11.8: CDC diameter modeling in MathCAD.

FLOW CONTROL AND MEASUREMENT SOLUTION

12.1 A brief programming guide

1. Do not use a numerical or iterative solution when an analytical solution is easily available.
2. Whenever a function has the potential to be used multiple times, create a function call that includes the parameters that could potentially change.
3. Do not break dependency. That means that if I change an input parameter at the top of your worksheet, that I should get the correct answers for the new parameter for all related calculations in the worksheet.
4. Always use dimensions (units). All calculations involving physical units must include those units.
5. Document your design process with comments.
6. Do not redefine your variables in subsequent problems. This loses valuable digits of precision on your numbers and can cause a lot of trouble and frustration.
7. For everyone's sake, use logical and reasonable variable names. [Here is AguaClara variable naming convention](#)

12.2 A brief Design Challenge guide

1. Read the Problem statement in its entirety before beginning a problem. If you don't immediately know what to do, read it again, thoroughly. If you are getting stuck, read it a third time. If you have a good understanding of what the problem is asking and are still having trouble, TAs can help through email or office hours.
2. If you decide to email a TA, make sure the other two are CCed. This minimizes the time you will have to wait until one responds.
3. When in doubt, Kernel -> Restart & Run All
4. Play around! Print arrays, test inputs, and ask yourself if your answers are reasonable. Should flow have units of km mg/s?

```
from aguaclara.core import constants as con
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.core.utility as ut
import numpy as np
import matplotlib.pyplot as plt
```

12.3 Vertical orifice equation

12.3.1 1)

Find the vena contract (VC) coefficient ratio for an orifice in the expert_inputs and print the result in a sentence. Please display 2 significant figures.

```
print('The vena contracta coefficient for an orifice is ',con.VC_ORIFICE_RATIO,'.')
```

The vena contracta coefficient for an orifice is 0.63.

12.3.2 2)

The simple orifice equation $Q = \Pi_{vc} A_{or} \sqrt{2g\Delta h}$ that we normally use is not applicable for vertically oriented orifices that are partially or barely submerged. The [USGS published a great solution](#) for flow through partially submerged vertically oriented orifices. AquaClara uses a general solution for a vertically oriented orifice, which is available in the physchem file as `pc.flow_orifice_vert`. That function handles vertically oriented orifices even if they are only partially submerged.

The vertical orifice equation is based on the concept that the velocity through the orifice at any point is equal to $\sqrt{2gh}$, where h is the local depth of submergence. The total flow can be obtained by integration of that velocity over the submerged area of the orifice.

For this question, you will create a well formatted graph with two curves to display flow rate through a 5 cm diameter orifice oriented **vertically and horizontally**.

We want to be able to describe the height of the water in the orifice as relative to the orifice diameter size. The relationship between velocity and orifice diameter is true for orifices of any size, so it is valuable to create a nondimensional model that can be understood for all diameters. The flow rate that you will use for this question is as a function of the normalized depth of water from 1 diameter below the center of the orifice to 2 diameters above the center of the orifice.

The steps for making the graph are as follows:

- Use `np.linspace` to generate an array of 100 dimensionless water surface elevations. The surface elevations should be normalized (also referred to as nondimensionalized) by the diameter of the orifice, and should range from -1 to 2 orifice diameters.
- Create a second array for water elevation (with units) by multiplying the normalized water elevation array by the orifice diameter.
- Create two arrays of flow rates through the orifice: one for the horizontal orifice orientation and one for the vertical orifice orientation. Use the two orifice equations `pc.flow_orifice` and `pc.flow_orifice_vert` in the physchem file, with orifice diameter and the dimensional water elevation array you created as inputs.
- Plot the curves for vertical and horizontal orifice flow in L/s as a function of the normalized height of water.
- Label the graph with flow rate in L/s as the y-axis and with normalized water elevation above the center of the orifice as the x-axis.
- Include a legend for the two curves.

```
WaterElevationNormalized = np.linspace(-1,2,100)
DiamOrifice = 5*u.cm
WaterElevation = WaterElevationNormalized*DiamOrifice

#Here we initialize some empty numpy arrays of the same size and type as
WaterElevationNormalized
```

(continues on next page)

(continued from previous page)

```

HorizontalOrificeFlows = np.empty_like(WaterElevationNormalized)
VerticalOrificeFlows = np.empty_like(WaterElevationNormalized)

#Here we need to populate the arrays we created above.
# Our graphing library, pyplot, cannot handle units, so we need to remove
# them from each element as we insert it into the array.
# While the code looks somewhat ugly, this is the best place to remove units;
# we are removing them after all calculations have been made to minimize the
# severity of floating-point errors.
for i in range(len(WaterElevation)):
    HorizontalOrificeFlows[i] = (pc.flow_orifice(DiamOrifice, WaterElevation[i], con.VC_
    ↵ORIFICE_RATIO).to(u.L/u.s).magnitude)
    VerticalOrificeFlows[i] = (pc.flow_orifice_vert(DiamOrifice, WaterElevation[i], con.
    ↵VC_ORIFICE_RATIO).to(u.L/u.s).magnitude)

fig, ax = plt.subplots()
ax.plot(WaterElevationNormalized, HorizontalOrificeFlows, 'r-',_
    ↵WaterElevationNormalized, VerticalOrificeFlows, 'b-')

ax.set(xlabel='Normalized height of water above center of the orifice')
ax.set(ylabel='Flow rate through the orifice (L/s)')
ax.legend(['Horizontal Orientation', 'Vertical orientation'])
ax.grid(True)
fig.savefig('Flow_Control_and_Measurement/Images/Horizontal_vs_Vertical_Orifice_
    ↵Orientation')
plt.show()

```

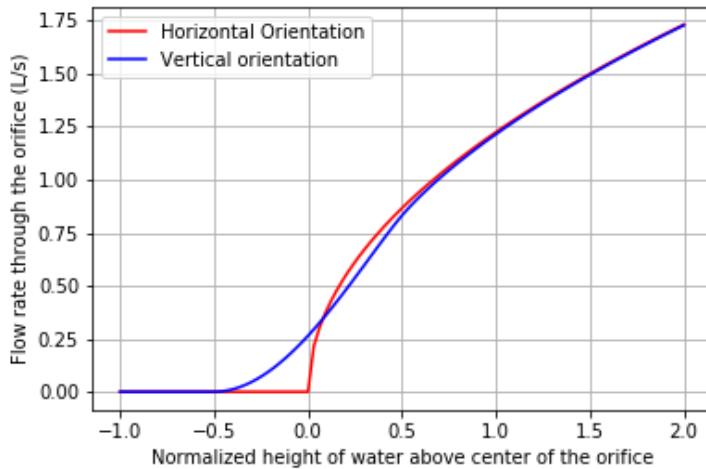


Fig. 12.1: Horizontal vs. Vertical Orifice Orientation

12.3.3 3)

Write a paragraph about what the graph means by explaining the following two items:

- Explain why the vertical orifice equation predicts more flow when the water level is below the center of the orifice and predicts less flow when the water level is above the center of the orifice. It might help to draw a picture of what the equations are describing to understand what is happening here!
- Explain how the horizontal orifice equation function from physchem.py predicts the flow rate for submergence depths that are negative. You will need to find the function and look at the code.

Explanation

The vertical orifice has the lower part of the orifice partially submerged before the horizontal orifice has any part submerged. This explains why the vertical orifice has more flow than the horizontal orifice between -0.5 and 0.

The horizontal orifice has higher flow rates between 0 and 0.5 because it is fully submerged when the vertical orifice is still not fully submerged.

At the elevation where the vertical orifice is first fully submerged the flow rate through the vertical orifice is less than the flow rate through the horizontal orifice. This is a result of the nonlinear relationships between depth of submergence and velocity.

The difference between the two equations becomes negligible for submergence greater than 1 diameter.

For negative depths of submergence the horizontal orifice function uses an if statement to set the flow rate through the orifice equal to zero.

12.4 Linear Flow Orifice Meter (LFOM)

A linear flow orifice meter is used in AquaClara plants to measure the plant flow rate and to provide a linear relationship between flow rate and the depth of water in the entrance tank. Below, we use the LFOM code to obtain a design for a linear flow orifice meter. Your task will be to test this design using the orifice equations to see if it is correct. We have a custom [‘LFOM class<https://github.com/AquaClara/aguaclara/blob/master/aguaclara/design/lfom.py>’](https://github.com/AquaClara/aguaclara/blob/master/aguaclara/design/lfom.py) that defines the LFOM properties. This code will be updated soon based on the code below. In the meantime, the code below can be used for calculations.

The following questions are all answered in one big block of code to make it easy to change values and then see the resulting graph.

12.4.1 4)

Create a function that calculates the flow rate through the LFOM as a function of only water elevation using the vertical orifice function. Use the arrays for LFOM key parameters, given above as `my_LFOM.orifice_diameter`, `my_LFOM.n_orifices_per_row`, and `my_LFOM.height_orifices`.

- Create an array for depth of submergence for each row of orifices at a given a height of water in the LFOM. This array is dependent on the water elevation (which should be your function input) and the height of the LFOM orifices (which is from the LFOM key parameters). Use this submergence depth array as the height input to your vertical orifice function. The array should be created within your function.
- To calculate the flow rate through the LFOM, multiply the calculated flow for each row of orifices by the number of orifices in that row (`my_LFOM.n_orifices_per_row`) to get an array of flows through each row of orifices. Note: the vertical orifice function will report zero flow for any orifices that aren't submerged, so you can send the whole array of depth of submergence for each row of orifices.
- At the end of your function, sum flows from each row of the LFOM and return that value with the correct units.
- Add a comment under the function definition to explain what the function does (see any of the aguaclara design files for examples of descriptive comments).

Calculate the total flow through the LFOM using the vertical orifice equation for the case when the water level is at the maximum water level for the LFOM, `HeadlossLfom`. You are checking to make sure that the LFOM produces the correct target flow (given as `Flow`) at the maximum height. Does it?

We want to compare the actual flow rate through the LFOM to the expected flow rate through the elevation as a function of water depth. Create a graph of the normalized actual and expected flow rates, using the following steps:

- Create an 100-unit long array of water depths using `np.linspace`. Note: the expected flow rate at elevation zero is zero, which makes the normalized flow rate undefined for zero elevation. An undefined normalized flow will not run and Python will report an error. You can solve this by beginning your water depth array at a very small (nonzero) elevation. You can end your water depth array at the maximum water depth. Recall that an array of elevations should have units of length.
- Create an array of normalized actual flow rates at each water depth; use the function you created in Problem 4 and a `for` loop (the function you created in Problem 4 probably can't handle an array of depths as input, so you need the `for` loop to cycle through each depth value to make your array of flows).
 - Start by creating an empty array for actual flow rates that is the same shape as the 100-unit water depth array you just created.
 - In your `for` loop, normalize the actual flow rates by using the following relationship: normalized actual flow rate = (actual flow rate)/[(water depth * target flow rate)/maximum water level]
- Plot a straight horizontal line at $y = 1$, which is your normalized expected flow value if the LFOM were perfect.

```
"""Build an Linear Flow Orifice Meter"""
import aquaclara.core.constants as con
import aquaclara.core.physchem as pc
import aquaclara.core.pipes as pipe
import aquaclara.core.utility as ut
import aquaclara.core.drills as drills
from aquaclara.core.units import unit_registry as u
import numpy as np
import math

class LFOM:

    def __init__(self, q=20*u.L/u.s, hl=20*u.cm, safety_factor=1.2, sdr=26,
                 drill_bits=drills.DRILL_BITS_D_IMPERIAL, s_orifice=0.5*u.cm):

        self.q = q
        self.hl = hl
        self.safety_factor = safety_factor
        self.sdr = sdr
        self.drill_bits = drill_bits
        self.s_orifice = s_orifice

    def width_stout(self, z):
        """Return the width of a Stout weir at elevation z. More info
        here. <https://confluence.cornell.edu/display/AGUACLARA/LFOM+sutrot+weir+research>
        """
        w_per_flow = 2 / ((2 * pc.gravity * z) ** (1 / 2) *
                           con.VC_ORIFICE_RATIO * np.pi * self.hl)
        return w_per_flow

    @property
    def n_rows(self):
```

(continues on next page)

(continued from previous page)

```

"""This equation states that the open area corresponding to one row
can be set equal to two orifices of diameter=row height. If there
are more than two orifices per row at the top of the LFOM then there
are more orifices than are convenient to drill and more than
necessary for good accuracy. Thus this relationship can be used to
increase the spacing between the rows and thus increase the diameter
of the orifices. This spacing function also sets the lower depth on
the high flow rate LFOM with no accurate flows below a depth equal
to the first row height.
But it might be better to always set then number of rows to 10.
The challenge is to figure out a reasonable system of constraints that
reliably returns a valid solution.
"""

N_estimated = (self.hl * np.pi / (2 * self.width_stout(self.hl) * self.q)).
    to(u.dimensionless)
variablerow = min(10, max(4, math.trunc(N_estimated.magnitude)))
return variablerow

@property
def b_rows(self):
    """The distance center to center between each row of orifices.
    Message how long it took to load everything (minus packages)"""
    return self.hl / self.n_rows

@property
def vel_critical(self):
    """The average vertical velocity of the water inside the LFOM pipe
    at the very bottom of the bottom row of orifices. The speed of
    falling water is 0.841 m/s for all linear flow orifice meters of
    height 20 cm, independent of total plant flow rate. """
    return (4 / (3 * math.pi) * (2 * pc.gravity * self.hl) ** (1 / 2)).to(u.m/u.s)

@property
def area_pipe_min(self):
    """The minimum cross-sectional area of the LFOM pipe that assures
    a safety factor."""
    return (self.safety_factor * self.q / self.vel_critical).to(u.cm**2)

@property
def nom_diam_pipe(self):
    """The nominal diameter of the LFOM pipe"""
    ID = pc.diam_circle(self.area_pipe_min)
    return pipe.ND_SDR_available(ID, self.sdr)

@property
def area_top_orifice(self):
    """Estimate the orifice area corresponding to the top row of orifices.
    Another solution method is to use integration to solve this problem.
    Here we use the width of the stout weir in the center of the top row
    to estimate the area of the top orifice
    """
    # Calculate the center of the top row:
    z = self.hl - 0.5 * self.b_rows
    # Multiply the stout weir width by the height of one row.
    return self.width_stout(z) * self.q * self.b_rows

@property

```

(continues on next page)

(continued from previous page)

```

def d_orifice_max(self):
    """Determine the maximum orifice diameter."""
    return pc.diam_circle(self.area_top_orifice)

@property
def orifice_diameter(self):
    """The actual orifice diameter. We don't let the diameter extend
    beyond its row space."""
    maxdrill = min(self.b_rows, self.d_orifice_max)
    return ut.floor_nearest(maxdrill, self.drill_bits)

@property
def drillbit_area(self):
    """The area of the actual drill bit."""
    return pc.area_circle(self.orifice_diameter)

@property
def n_orifices_per_row_max(self):
    """A bound on the number of orifices allowed in each row.
    The distance between consecutive orifices must be enough to retain
    structural integrity of the pipe."""
    c = math.pi * pipe.ID_SDR(self.nom_diam_pipe, self.sdr)
    b = self.orifice_diameter + self.s_orifice

    return math.floor(c/b)

@property
def flow_ramp(self):
    """An equally spaced array representing flow at each row."""
    return np.linspace(1 / self.n_rows, 1, self.n_rows)*self.q

@property
def height_orifices(self):
    """Calculates the height of the center of each row of orifices.
    The bottom of the bottom row orifices is at the zero elevation
    point of the LFOM so that the flow goes to zero when the water height
    is at zero."""
    return (np.linspace(0, self.n_rows-1, self.n_rows))*self.b_rows + 0.5 * self.orifice_diameter

def flow_actual(self, Row_Index_Submerged, N_LFOM_Orifices):
    """Calculates the flow for a given number of submerged rows of orifices
    harray is the distance from the water level to the center of the
    orifices when the water is at the max level.
    Parameters
    -----
    Row_Index_Submerged: int
        The index of the submerged row. All rows below and including this
        index are submerged.
    N_LFOM_Orifices: [int]
        The number of orifices at each row.
    Returns
    -----
    The flow through all of the orifices that are submerged.

```

(continues on next page)

(continued from previous page)

```

"""
flow = 0
for i in range(Row_Index_Submerged + 1):
    flow = flow + (N_LFOM_Orifices[i] * (
        pc.flow_orifice_vert(self.orifice_diameter,
                              self.b_rows*(Row_Index_Submerged + 1)
                              - self.height_orifices[i],
                              con.VC_ORIFICE_RATIO)))
return flow

@property
def n_orifices_per_row(self):
    """Calculate number of orifices at each level given an orifice
    diameter."""
    """
    # H is distance from the bottom of the next row of orifices to the
    # center of the current row of orifices
    H = self.b_rows - 0.5*self.orifice_diameter
    flow_per_orifice = pc.flow_orifice_vert(self.orifice_diameter, H, con.VC_
    ↪ORIFICE_RATIO)
    n = np.zeros(self.n_rows)
    for i in range(self.n_rows):
        # calculate the ideal number of orifices at the current row without
        # constraining to an integer
        flow_needed = self.flow_ramp[i] - self.flow_actual(i, n)
        n_orifices_real = (flow_needed / flow_per_orifice).to(u.dimensionless)
        # constrain number of orifices to be less than the max per row and
        # greater or equal to 0
        n[i] = min((max(0, round(n_orifices_real))), self.n_orifices_per_row_max)
    return n

@property
def error_per_row(self):
    """This function calculates the error of the design based on the
    differences between the predicted flow rate
    and the actual flow rate through the LFOM."""
    FLOW_lfom_error = np.zeros(self.n_rows)
    for i in range(self.n_rows):
        actual_flow = self.flow_actual(i, self.n_orifices_per_row)
        FLOW_lfom_error[i] = (((actual_flow - self.flow_ramp[i]) / self.flow_
    ↪ramp[i]).to(u.dimensionless)).magnitude
    return FLOW_lfom_error

#Now we design a 60 L/s LFOM.
LFOM_hl = 20 * u.cm
LFOM_flow = 60 * u.L/u.s

my_LFOM = LFOM(q=LFOM_flow, hl=LFOM_hl, drill_bits = drills.DRILL_BITS_D_IMPERIAL)
orifice_diameter = my_LFOM.orifice_diameter
n_orifices_per_row = my_LFOM.n_orifices_per_row
height_orifices = my_LFOM.height_orifices
print('The velocity of the water exiting the bottom of the LFOM is', my_LFOM.vel_
    ↪critical)
print('The flow rate of the LFOM is', my_LFOM.q)
print('The nominal diameter of the LFOM is', my_LFOM.nom_diam_pipe)
print('The total height range from zero flow to max flow is', my_LFOM.hl)

```

(continues on next page)

(continued from previous page)

```

print('The number of rows of orifices is',my_LFOM.n_rows)
print('The vertical spacing of the rows of orifices is',my_LFOM.b_rows)
print('The orifice diameter is',orifice_diameter.to(u.cm))
print('The number of orifices in each row is',n_orifices_per_row)
print('The height of the center of the orifices measured from the LFOM datum, the_'
      ↪bottom of the bottom row of orifices, is',height_orifices.magnitude,height_orifices.
      ↪units)
print('The error is',my_LFOM.error_per_row)

def flow_lfom_vert(height, orifice_diameter, n_orifices_per_row, height_orifices):
    "Returns the flow through the LFOM as a function of height above the datum"
    Flow = np.empty_like(n_orifices_per_row)
    for i in range(len(n_orifices_per_row)):
        Flow[i] = ((pc.flow_orifice_vert(orifice_diameter, height - height_
    ↪orifices[i], con.VC_ORIFICE_RATIO)*n_orifices_per_row[i]).to(u.L/u.s)).magnitude
    return (sum(Flow))*u.L/u.s

flow_lfom_vert(LFOM_hl, orifice_diameter, n_orifices_per_row, height_orifices)

FlowMaxVert = flow_lfom_vert(LFOM_hl, orifice_diameter, n_orifices_per_row, height_
    ↪orifices)
print('The flow at a depth of ',LFOM_hl,' is ',FlowMaxVert,'.')

#Create an array of water depths
graph_points=50
HeightGraph = np.linspace(0.001, LFOM_hl.to(u.cm), graph_points) * u.cm

#Create an array that is empty
FlowGraph = np.empty_like(HeightGraph)
# or FlowGraph = []
for i in range(len(HeightGraph)):
    FlowGraph[i] = (flow_lfom_vert((HeightGraph[i]), orifice_diameter, n_orifices_per_
    ↪row, height_orifices)/((HeightGraph[i]) / LFOM_hl * LFOM_flow)).magnitude

fig, ax = plt.subplots()
ax.plot(HeightGraph, FlowGraph, 'r-', HeightGraph, np.ones(graph_points), 'c-')
ax.set(xlabel='Water Depth (cm)')
ax.set(ylabel='Normalized flow rate')
ax.legend(['Vertical Orifice', 'Target'], loc='best')
ax.grid(True)
fig.savefig('Flow_Control_and_Measurement/Images/Normalized_Flow_Rate_vs_Water_Depth')
plt.show()

```

12.4.2 7)

Play with the value for the plant flow rate, LFOM_flow, and try a bunch of different flows over the range 1 to 100 L/s. The LFOM isn't accurate for the first couple of rows.

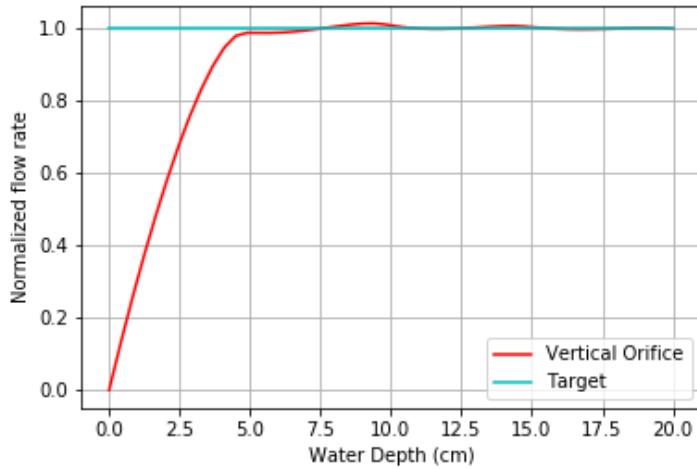


Fig. 12.2: Normalized Flow Rate vs. Water Depth

12.4.3 8)

DO you observe any failure modes where the design produces very inaccurate flow measurements? If so, then create an issue!

12.4.4 9)

Explain why all LFOMs perform poorly when the water depth is in the first row of orifices.

The relationship between head and flow is nonlinear for a single row of orifices. Thus it is impossible for the LFOM to be accurate when there is only one row of orifices.

12.5 Laminar Flow Based Flow Controller

You will design (by completing the following questions) a laminar flow controller for chlorine feed for a plant design flow rate of 50 L/s.

For the following steps do NOT use the aquaclaracdc code. Instead, create the functions that you need to solve this problem. At the end, we will compare your solution to the aquaclaracdc solution.

You may assume that the chlorine stock solution kinematic viscosity is approximately the same as water. The dose controller is to have a maximum head loss of 20 cm through the dosing tubes. We will start with commercially available liquid bleach (equivalent to 51.4 gm/L of chlorine gas), which we will use in our chemical stock tanks without dilution. Our goal is to provide a constant chlorine dose of 2 mg/L to the water entering the storage tank. We will be following the guidelines given below.

1. Calculate the maximum flow rate through each available dosing tube diameter that keeps error due to minor losses below 10%.
2. Calculate the total chemical flow rate that would be required by the treatment system for the maximum chemical dose and the maximum allowable stock concentration.
3. Calculate the number of dosing tubes required if the tubes flow at maximum capacity (round up).

4. Calculate the length of the dosing tubes that correspond to each available tube diameter.
5. Select the longest dosing tube that is shorter than the maximum tube length allowable based on geometric constraints.
6. Select the dosing tube diameter, flow rate, and stock concentration corresponding to the selected tube length.

```
FlowPlant = 50*u.L/u.s
T = u.Quantity(20,u.degC)
NuBleach = pc.viscosity_kinematic(T)
HeadlossDosingTubeMax = 20*(u.cm)
StockCl2 = 51.4*(u.gram/u.L)
DoseCl2 = 2*(u.mg/u.L)
RatioError = 0.1
Kminor = 2
```

12.5.1 11)

At the given water treatment plant design flow rate, what is the required flow of bleach (the chlorine stock solution)?

```
FlowStockClMax = (FlowPlant * DoseCl2 / StockCl2).to(u.mL/u.s)
print('The required flow of bleach is', FlowStockClMax)
```

The required flow of bleach is 1.95 ml/s

12.5.2 12)

How many liters of liquid bleach are required in one day? (you can simply change the units on the flow rate!)

```
print('The daily required flow of bleach is', FlowStockClMax.to(u.L/u.day))
```

The daily required flow of bleach is 168.09 l/day

12.5.3 13)

Our next big goal is to choose a tubing size for the dosing tube (or tubes). This requires multiple steps. Begin by first creating a numpy array of tubing sizes between 1/16 and 5/16 with a 1/16 interval. Your list should contain 5 elements. Does np.linspace work here? What about np.arange? Remember to always attach the units to the entire array and not to array elements!

```
DiamTubeArray = (np.arange(1,6,1)) * 1 / 16 * u.inch
print(DiamTubeArray.magnitude, DiamTubeArray.units)
```

[0.0625 0.125 0.1875 0.25 0.3125] inch

12.5.4 14)

What is the maximum average velocity in a dosing tube based on the constraint that minor losses must be small? This means that the minor losses account for RatioError fraction of the total losses (10% when RatioError is 0.1). Note that this velocity is independent of the tube diameter.

```
VelTubeMax = (((RatioError * 2 * HeadlossDosingTubeMax * pc.gravity) / KMinor)**(1/  
              2)).to(u.meter/u.s)  
print('The maximum average velocity in a dosing tube is', VelTubeMax)
```

The maximum average velocity in a dosing tube is 0.443 m/s

12.5.5 15)

What is the head loss due to minor losses in the tube when the tube is flowing at maximum capacity? Solve for this value algebraically by substituting your equation for the velocity in the tube into the minor loss equation and then calculate the value.

```
HeadlossMinorMax = RatioError * HeadlossDosingTubeMax  
print('The head loss due to minor losses when the tube is at maximum capacity is',  
      HeadlossMinorMax)
```

The head loss due to minor losses when the tube is at maximum capacity is 2.0 cm

12.5.6 16)

Create an array of the maximum flow rates corresponding to the array of tubing diameters. The flow rates must meet the error constraint.

—First, create a function that uses diameter and velocity as inputs to return flow rate. Note that “pc.area_circle(diam)” returns $\pi d^2/4$.
(12.1)

```
def flow_cdc_max(diam, VelTubeMax):  
    Flow = pc.area_circle(diam) * (VelTubeMax)  
    return Flow  
  
FlowMaxArray = flow_cdc_max(DiamTubeArray, VelTubeMax).to(u.mL/u.s)  
print(FlowMaxArray)
```

[0.87658228 3.5063291 7.88924048 14.02531641 21.91455688] milliliter / second

12.5.7 17)

Find the minimum number of tubes for each of the available tube diameters that would be required to deliver the maximum flow of bleach.

```
NDosingTubes = np.ceil(FlowStockClMax / FlowMaxArray)  
print('The number of tubes of each diameter is', NDosingTubes)
```

The number of tubes of each diameter is [3. 1. 1. 1. 1.] dimensionless

12.5.8 18)

Create an array of the maximum flow rate per tube for each of the available tubing diameters, given the number of tubes that would be used. This will be the flow through each dosing tube at the maximum flow of bleach.

```
FlowDosingTubeArray = FlowStockC1Max / NDosingTubes
print('The flow rate per tube is', FlowDosingTubeArray)
print(FlowStockC1Max)
```

The flow rate per tube is [0.64850843 1.94552529 1.94552529 1.94552529 1.94552529] milliliter / second
1.9455252918287937 milliliter / second

12.5.9 19)

We now know the target flow in the dosing tubes, the diameter of the tubes, and the target head loss through the tubes. Thus, we can solve for the length of the tube that will deliver that target flow. Write a function to find the length of each tube that could handle the entire flow. Your function should use the following equation:

$$L = \frac{gh_L\pi D^4}{128\nu Q_{Max}} - \frac{Q_{Max}}{16\pi\nu} \sum K_e \quad (12.2)$$

Call your function to return the length of tubing required for each tube size.

```
def length_tube(flow_max, diam, headloss_max, nu, k_minor):
    "Returns the length of tube necessary to handle the maximum flow."
    L = (((pc.gravity * headloss_max * np.pi * diam**4) /
          (128 * nu * flow_max)) -
         ((k_minor * flow_max) /
          (16 * np.pi * nu)))
    return L

LengthDosingTube = length_tube(FlowDosingTubeArray, DiamTubeArray,
                               HeadlossDosingTubeMax, NuBleach, KMinor).to(u.m)

print('The length of each dosing tube would be', LengthDosingTube)
```

The length of each dosing tube would be [0.44406171 2.42832361 12.60675229 40.01021413 97.79237081] meter

12.5.10 20)

Which option do you think is best? You can simply set the array index to your choice and then display your solution by using that index value on your arrays for number of tubes, flow rates, tube diameters, and length of tubes.

```
MYPICK = 1
print('The number of dosing tubes I will need is', NDosingTubes[MYPICK])
print('The flow through each tube is', FlowDosingTubeArray[MYPICK])
print('The inner diameter of the tube is', DiamTubeArray[MYPICK])
print('The length of each tube is', LengthDosingTube[MYPICK].to(u.m))
```

The number of dosing tubes I will need is 1 The flow through each tube is 1.95 ml/s The inner diameter of the tube is 0.125 inch The length of each tube is 2.43 m

12.5.11 21)

What physical constraints might you use to select the best solution? How did you make your selection in Problem 19?

The ideal solution will have - a reasonable number of tubes and thus one possibility is to select the smallest diameter of tubing that uses a single tube. However, this wont work for plants with high flow rates of chemicals. - tubes that are short enough to mount in the water treatment plant

12.5.12 22)

AguaClara has coded these dosing tube size functions in the CDC Functions (cdc_functions). Find the function calls for the length, diameter, and number of dosing tubes and use those functions to calculate the values for the problem that you solved above. Compare your answers. Your answers should agree!

```
from aguaclara.design import cdc as cdc
pc.viscosity_kinematic(22*u.degC)

#The cdc functions file has an enumerated list for different checmials to obtain the_
#correct kinematic viscosity.
# 0 is Alum, 1 is PACl, 2 is water or chlorine

EnChem = 2

#see the
#viscosity_kinematic_chem(conc_chem, temp, en_chem):
#to see how EnChem is used.
```

```
# The maximum tube length constraint might be based on the length of the available_
#wall where the
# dosing tube will be mounted. You might change this depending on which solution
# you picked in step 20. Here the wall length is LengthTubeMax.

LengthTubeMax = 5*u.m

LengthTubeCheck = cdc.len_cdc_tube(FlowPlant, DoseC12, StockC12,
                                    DiamTubeArray, HeadlossDosingTubeMax,
                                    LengthTubeMax, T, EnChem, KMinor)

print('The length of the CDC tube is ', LengthTubeCheck)

DiamTubeCheck = cdc.diam_cdc_tube(FlowPlant, DoseC12, StockC12,
                                   DiamTubeArray, HeadlossDosingTubeMax,
                                   LengthTubeMax, T, EnChem, KMinor)

print('The diameter of the CDC tube is', DiamTubeCheck.to(u.inch))

NTube = cdc.n_cdc_tube(FlowPlant, DoseC12, StockC12,
                       DiamTubeArray, HeadlossDosingTubeMax,
                       LengthTubeMax, T, EnChem, KMinor)

print('The number of CDC tubes is ', NTube)
```

The length of the CDC tube is 2.43 m The diameter of the CDC tube is 0.125 in The number of CDC tubes is 1.0

CHAPTER
THIRTEEN

FLOW CONTROL AND MEASUREMENT DESIGN CHALLENGE

Researchers in the AguaClara laboratory collected the following head loss data through a 1/8 diameter tube that was 2 m long using water at 22°C. The data is in a comma separated data (.csv) file named `Head_loss_vs_Flow_dosing_tube_data.csv`. Use the pandas read csv function (`pd.read_csv('filename.csv')`) to read the data file. Display the data so you can see how it is formatted.

```
import pandas as pd
import numpy as np
url = 'https://raw.githubusercontent.com/AguaClara/CEE4520/master/DC%20Solutions/
    ↵Markdown/Head_loss_vs_Flow_dosing_tube_data.csv'
head_loss_data = pd.read_csv(url)
head_loss_data
```

13.1 1)

Using the data table above, assign the head loss and flow rate data to separate 1-D arrays. Attach the correct units. `np.array` can extract the data by simply inputting the text string of the column header. Here is example code to create the first array:

```
HL_data=np.array(head_loss_data['Head loss (m)'])*u.m
```

13.2 2)

Calculate and report the maximum and minimum Reynolds number for this data set. Use the tube and temperature parameters specified above. Use the `min` and `max` functions which take arrays as their inputs.

The Reynolds number varied from 2.9e+2 to 1.0e+3.

13.3 3)

You will now create a graph of head loss vs flow for the tube mentioned in the previous problems. This graph will have two sets of data: the real data contained within the csv file and some theoretical data. The theoretical data is what we would expect the head loss through the tube to be in an ideal world for any given flow. When calculating the theoretical head loss, assume that minor losses are negligible. Plot the data from the csv file as individual data points and the theoretical head loss as a continuous curve. Make the y-axis have units of cm and the x-axis have units of mL/s.

A few hints.

- To find the theoretical head loss, you will first need to create an array of different flow values. While you could use the values in the csv file that you extracted in Problem 14, we would instead like you to create an array of 50 equally-spaced flow values. These values shall be between the minimum and maximum flows in the csv file.
- You can use the `np.linspace(input1, input2, input3)` function to create this set of equally-spaced flows. Inputs for `np.linspace` are the same as they were for `np.logspace`, which was used in Problem 12a). Linspace does not work with units; you will need to remove the units (using `.magnitude`) from the inputs to `np.logspace` and then reattach the correct units of flow after creating the array.
- The `pc.headloss_fric` function can handle arrays as inputs, so that makes it easy to produce the theoretical head loss array once you have finished your equally-spaced flow array.
- When using plotting, make sure to convert the flow and head loss data to the desired units.

13.4 4)

The theoretical model doesn't fit the data very well. We assumed that major losses dominated. But that assumption was wrong. So let's try a more sophisticated approach where we fit minor losses to the data. Below we demonstrate the use of the `scipy curve_fit` method to fit the minor loss coefficient given this data set. In this example, `Q_data` is the flow rate array for the csv file from problem 13. You should re-name this variable below to whatever you titled this variable.

```
from scipy.optimize import curve_fit

# Define a new function that calculates head loss given the flow rate
# and the parameter that we want to use curve fitting to estimate
# Define the other known values inside the function because we won't be passing those
# parameters to the function.

def HL_curvefit(FlowRate, KMinor):
    # The tubing is smooth AND pipe roughness isn't significant for laminar flow.
    PipeRough = 0*u.mm
    L_tube = 2*u.m
    T_data = u.Quantity(22,u.degC)
    nu_data = pc.viscosity_kinematic(T_data)
    D_tube = 1/8*u.inch
    # pass all of the parameters to the head loss function and then strip the units so
    # the curve fitting function can handle the data.
    return (pc.headloss(FlowRate, D_tube, L_tube, nu_data, PipeRough, KMinor)).
    magnitude

# The curve fit function will need bounds on the unknown parameters to find a real
# solution.
# The bounds for K minor are 0 and 20.

# The curve fit function returns a list that includes the optimal parameters and the
# covariance.

popt, pcov = curve_fit(HL_curvefit, Q_data, HL_data, bounds=[[0],[20]])

K_minor_fit = popt[0]

# Plot the raw data
plt.plot(Q_data.to(u.mL/u.s), HL_data.to(u.cm), 'o', label='data')
```

(continues on next page)

(continued from previous page)

```

# Plot the curve fit equation.
plt.plot(Q_data.to(u.mL/u.s), ((HL_curvefit(Q_data, *popt))*u.m).to(u.cm), 'r-',_
         label='fit')
plt.xlabel('Flow rate (mL/s)')
plt.ylabel('Head loss (cm)')
plt.legend()
plt.show()

#Calculate the root mean square error to estimate the goodness of fit of the model to_
#the data
RMSE_Kminor = (np.sqrt(np.var(np.subtract((HL_curvefit(Q_data, *popt)), HL_data.-
                                           magnitude)))*u.m).to(u.cm)
print('The root mean square error for the model fit when adjusting the minor loss_'
      'coefficient was', RMSE_Kminor)

```

The root mean square error for the model fit when adjusting the minor loss coefficient was 0.39 cm

13.5 5)

Repeat the analysis above, but this time assume that the minor loss coefficient is zero and that diameter is the unknown parameter. The bounds specified in the line beginning with `popt`, `pcov` should be changed from the previous question (which had bounds from 0 to 20) to the new bounds of 0.001 to 0.01.

Hint: You only need to change the name of the defined function (perhaps `HL_curvefit2?`) and adjust its inputs/values.

The root mean square error for the model fit when adjusting the diameter was 0.47 cm

13.6 6)

Changes to which of the two parameters, minor loss coefficient or tube diameter, results in a better fit to the data?

13.7 7)

Create a design for a `chemical dose controller` using `aguaclara`. Then take that design and calculate the maximum coagulant dose that it will deliver.

13.8 8)

An AguaClara plant will be upgraded from 20 L/s to 30 L/s by adding two sedimentation tanks, increasing the head loss through the flocculator, and adding an additional StaRS filter. Give the current design specs for the CDC. Propose the most cost effective modification to the CDC to handle the additional flow.

13.9 9)

Determine if the LFOM diameter will need to be increased for a 20 L/s to 30 L/s plant upgrade.

CHAPTER
FOURTEEN

RAPID MIX INTRODUCTION

This chapter is currently home for the prerequisites of successful flocculation. Those prerequisites include:

- ensuring that the pH is in the correct range for coagulant nanoparticle formation after the coagulant has been added to the raw water.
- increasing the coagulant dose to account for the coagulant interactions with dissolved species that effectively cover some of the coagulant nanoparticle surfaces.
- increasing the coagulant dose to account for the available surface area of suspended particles to achieve
- If there are multiple treatment trains, mixing the coagulant with the raw water so that parallel treatment trains receive the same concentration of coagulant (perhaps the conventional role of rapid mix)
- transporting the coagulant nanoparticles to attach to suspended particles

Rapid mix is the term commonly used to describe the processes that occur between the coagulant addition to the raw water and the flocculation process. The processes that occur are not well understood and thus design guidelines have been empirical.

In summary, little is known about rapid mix, much less any sensitivity to scale. However, the models and data reviewed suggest the need to be on the lookout for certain effects. From what is presently known, it can be speculated that since coagulant precipitation is sensitive to both micro- and macro-mixing, scale-up must consider not only energy dissipation rate, but also the reaction injection point and the contacting method. - [Mixing in Coagulation and Flocculation 1991 page 292](#).

Although the processes have not been well characterized, the energy that is invested for rapid mix processes is significant. In many cases the amount of energy used isn't practical for gravity powered water treatment plants. The high energy consumption of rapid mix units has led some municipal water treatment plant operators to experiment with turning off rapid mix units. They have found that at least under some conditions there is no indication that the energy used in rapid mix improved plant performance. Thus there is a need to understand the physical and chemical processes that occur when a concentrated liquid coagulant is added to water.

Rapid mix sets the stage for aggregation of both suspended particles and dissolved substances. Particle and dissolve substance aggregation is mediated by coagulant nanoparticles. The nanoparticles attach to raw water particles as well as to some dissolved species. After the nanoparticles have been mixed with the raw water and have attached to raw water particles the next process, flocculation, can begin. *Flocculation* is the process of producing collisions between particles to create flocs (aggregates of particles).

Coagulant nanoparticle application includes multiple steps that must occur before the raw water particles can begin to aggregate. The sticky nanoparticles can be aluminum (Al^{+3}) or iron (Fe^{+3}) based and in either case the nanoparticles are formed from precipitated hydroxide species ($Al(OH)_3$ or $Fe(OH)_3$). The series of events that are contained in the broad designation of rapid mix are:

1. Liquid coagulant stock solution with a low pH is injected into the raw water
2. Fluid Mixing: Turbulent eddies randomize the fluids (but don't blend them)

- (a) Large scale eddies mix the coagulant with the raw water by creating large fluid deformations. This stretching and turning of the raw water and coagulant is analogous to shuffling a deck of cards. The cards are randomized, but the cards maintain their identity. The original liquids retain their chemical composition. This step must be completed before any flow splitting for parallel treatment trains.
- (b) Turbulent eddies disintegrate into smaller and smaller eddies.
- (c) At a very small scale (Inner viscous length scale) viscosity becomes significant and the kinetic energy of the eddies begins to be converted to heat by viscosity.
3. The coagulant is blended with the raw water by molecular diffusion
 4. The higher pH of the raw water causes the coagulant to begin to precipitate as $Al_{12}AlO_4(OH)_{24}(H_2O)_{12}^{7+}$, an aluminum, Al, nanoparticle.
 5. The precipitating Al_{13} molecules aggregates with other nearby Al_{13} molecules to form aluminum hydroxide nanoparticles. It is also possible that the nanoparticles are already formed in the coagulant stock suspension. Polyaluminum chloride stock solutions turn white in about a year at room temperature and this suggests that nanoparticles form in the stock solution.
 6. The Al nanoparticles attach to other dissolved species and suspended particles.
 7. Molecular diffusion causes some dissolved species and Al nanoparticles to aggregate.
 8. Fluid shear and molecular diffusion cause Al nanoparticles with attached formerly dissolved species to collide with inorganic particles (such as clay) and organic particles (such as viruses, bacteria, and protozoans).

These multiple steps cover a wide range of length scales and it is not clear at the onset which processes might be the rate limiting steps. We will develop time scale estimates for several of these steps to help identify which processes will likely require the most attention to design. Many of these transport processes are presumed to occur in parallel. Fig. 14.1 shows the range of length scales.

14.1 Chemistry of Coagulant Nanoparticles

Aluminum based coagulants are commonly used in drinking water treatment plants. Less frequently iron based coagulants are

- Aluminum (1.61) and Oxygen (3.44) is 1.83
- Iron (1.83) and Oxygen (3.44) is 1.61
- Hydrogen (2.20) and Oxygen (3.44) is 1.24

Thus both aluminum and iron coagulants are more polar than water and it is possibly that it is their strong polarity that enables them to displace water that is bound to particles surfaces and then form bonds with that surface. In order to displace water molecules that are bound to the particles surfaces, the coagulants must have stronger bonds to particles surfaces than the polar water molecules and thus it seems likely that coagulants must be more polar than water.

14.1.1 pH Effects of Adding Coagulant

The coagulants used for drinking water treatment are acidic and thus result in a lowering of the pH of the treated water. The optimal pH for aluminum coagulant nanoparticle formation is between pH of 6.5 and 8.5. This is also the pH range set by the EPA secondary standards for drinking water. Although many water sources are within this pH range, there are some waters with more extreme values of pH. The aluminum and iron based coagulants are also acidic and in some waters the pH may drop below the ideal range when adding the coagulant. When the pH is outside the acceptable range it is necessary to adjust the pH by adding either a base or an acid.

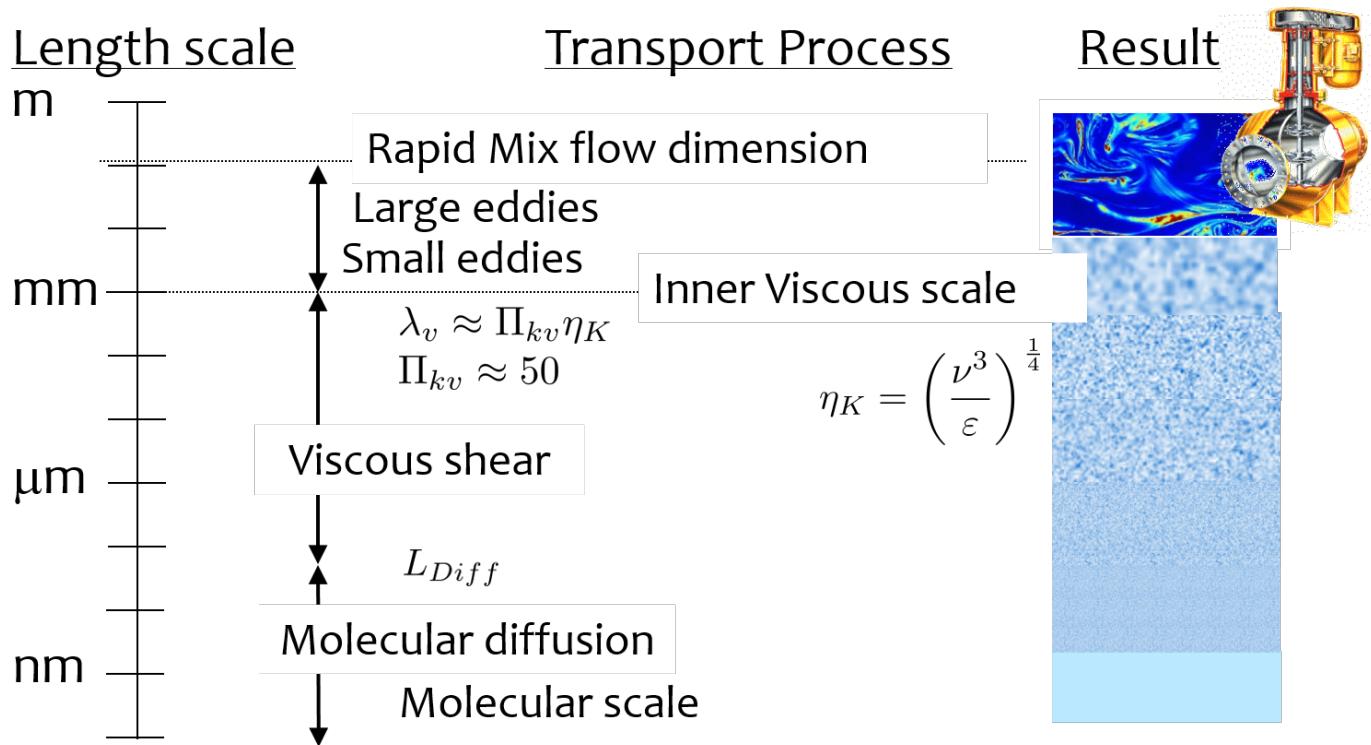
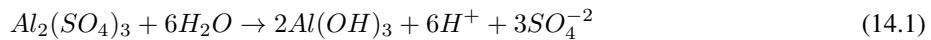


Fig. 14.1: Transport of coagulant nanoparticles occurs over length scales ranging from meter to a fraction of a nanometer.

When aluminum sulfate (alum) to water it dissociates and then precipitates as $Al(OH)_3$. In the process protons, H^+ are released and thus the pH ($-\log[H^+]$) decreases.



The release of these protons reduces the acid neutralizing capacity, ANC, (also known as alkalinity) of the water. ANC is traditionally measured with units of mg/L of $CaCO_3$ rather than eq/L. Table 14.1 shows the relationship between ANC measured as mg/L of $CaCO_3$ and alum ($Al_2(SO_4)_3 \cdot 14H_2O$)

Table 14.1: Reduction in ANC caused by addition of alum.

	Alum	Calcium Carbonate
Molecular Formula	$Al_2(SO_4)_3 \cdot 14H_2O$	$CaCO_3$
Molecular mass	600 g/mole	100 g/mole
eq/mole	6	2
Molecular mass/eq	100 g/eq	50 g/eq
Simple guide	1 mg/L alum consumes	0.5 mg/L calcium carbonate ANC

Low ANC waters (See section on *Buffering Capacity of Natural Waters*.) could have their ANC increased by addition of a base. A simpler approach is often to use a different coagulant that is less acidic.

Polyaluminum chloride (PACl) is another aluminum based coagulant that performs similarly to alum. PACl is manufactured by slowly titrating an acidic solution containing dissolved aluminum with a base (in the chemical plant) to produce a meta-stable and soluble polymeric aluminum. PACl consumes less alkalinity (ANC) because it is partially neutralized by the titration with a base. In addition, the aluminum mass fraction of PACl is higher than in alum because there are no attached water molecules. The mass of PACl required for flocculation is less than for alum due largely to

the higher aluminum fraction. The lower mass of PACl required is an economic benefit when shipping is a significant cost of the coagulant.

There are many different molecular formulas given for PACl. The molecular formulas are either from the perspective of the chemical supplier or represent the composition of the PACl nanoparticles. The molecular formula from the chemical supplier represents the extent of neutralization and hence the replacement of chloride with hydroxide. In that case the PACl molecular formula is:



The extent of the PACl titration with base is defined as basicity. Basicity is the ratio of hydroxyl equivalents to aluminum equivalents. Basicity of 1 would mean that the PACl does not produce any protons when it dissolves in water. Basicity of 0 means it produces 3 protons per Al (like alum). The equation for basicity is:

$$\text{Basicity} = \left(\frac{m}{3n} \right) \quad (14.3)$$

The lowest basicity commercial PACl formulations are about 10%. Most PACls are in the medium to high basicity range (50-70%). The highest stable basicity (83%) is aluminum chlorohydrate ($Al_2(OH)_5Cl$) that is useful for treating water with very low ANC.

The ANC of the aluminum coagulant can be obtained from the number of protons it releases:

$$ANC_{Al} = 3(\text{Basicity} - 1)[Al] = \left(\frac{m}{n} - 3 \right) [Al] = \Pi_{Al}[Al] \quad (14.4)$$

where:

$\Pi_{Al} = \left(\frac{m}{n} - 3 \right)$ is ANC per mole of aluminum for the given coagulant

Thus the ANC of alum (with 0 hydroxides) is $-3[Al]$. The method of calculating the ANC_{Al} will be used to calculate the amount of base that must be added to achieve a target pH.

The molecular formula of the PACl nanoparticles is always some combination of Al, O, and H. One common molecular structure is a Keggin structure with 13 aluminum atoms. This molecule has a tetrahedral Al atom in the center of the cluster coordinated to 4 oxygen atoms. The molecular formula of the Keggin structured aluminum molecule is



The attached water molecules are sometimes omitted. Coagulant nanoparticles are presumably created by aggregation of these Keggin structure molecules.

14.1.2 Buffering Capacity of Natural Waters

When acid is added to a water containing bicarbonate, HCO_3^- , one of the potential reactions is for a proton to combine with HCO_3^- to form carbonic acid, H_2CO_3 . If a base is added to water the reaction will proceed in the opposite direction. Carbonic acid, H_2CO_3 , is chemical indistinguishable from dissolved carbon dioxide, CO_{2aq} , and the total of carbonic acid and dissolved carbon dioxide is represented as $H_2CO_3^*$. The reaction of bicarbonate to form carbonic acid removes one proton from solution and thus the concentration of protons doesn't increase as fast as we might have first expected as acid is added to the water.

The reactions of carbonate species with protons provides pH buffering capacity that must be considered when calculating the effect of acid or base addition. Since carbonates are the dominant buffering agents in natural waters it is essential to account for their influence on pH.

The effect of acid or base addition to a water containing carbonates (or other weak acids and bases) can be modeled using the equation for [Acid Neutralizing Capacity](#).

14.1.3 pH Range for Precipitation of Coagulant Nanoparticles

A critical property of coagulants is that in order to act as an adhesive between particles they must be solid phase at neutral pH. Both Al(III) and Fe(III) have low solubility at neutral pH and thus meet this requirement. The pH region of low solubility sets the range of pH where flocculation is effective. Fig. 14.2 shows the solubility of aluminum as a function of pH.

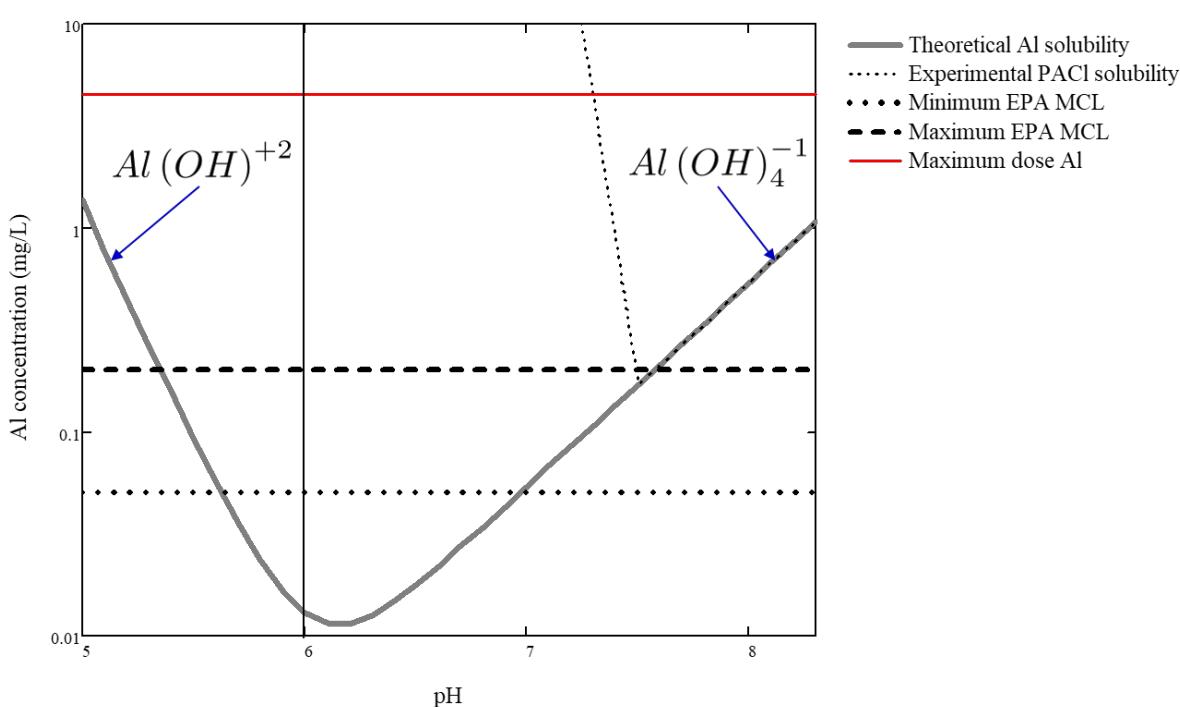


Fig. 14.2: Solubility of aluminum as a function of pH. Figure adapted from Pernitsky and Edzwald.

Research is needed to quantify flocculation performance in continuous flow floc/floc blanket/plate settler systems as a function of pH.

The aluminum concentration range used for flocculation ranges from approximately 0.4 - 10 mg/L and is strongly influenced by the concentration of dissolved organic matter and the concentration of suspended solids. The flocculation and floc blanket capacity to produce collisions between suspended particles also influences the required aluminum concentration.

14.1.4 pH Adjustment in Water Treatment Plants

In drinking water treatment plant operation it is sometimes necessary to add a base (or acid) to increase (or decrease) the pH of the raw water. The added coagulant tends to reduce the pH. The carbonate system is most important in understanding how the base will adjust the pH because the reaction between carbonic acid and bicarbonate occurs around pH 6.3, the pK (negative log of the dissociation constant is the pH where that reaction is centered) for that reaction. Carbon dioxide exchange with the atmosphere is insignificant in drinking water treatment unit processes unless there is an aeration stage. Thus we can use the ANC equation for the case with no CO_2 exchange with the atmosphere.

In the section, *pH Adjustment* we evaluate the case where we add a base that will increase the ANC of the raw water and it might also increase the total carbonate concentration. We calculate how much of that base to add to reach a target pH.

14.2 Fluid Mixing

Fluid mixing is the process by which large scale eddies distribute packets of the coagulant stock throughout the raw water. The term Rapid mix is probably best used to describe this process. Traditional methods of achieving this fluid mixing include various methods of generating intense turbulence. Fluid mixing is able to rapidly blend the coagulant with the raw water in a matter of a few seconds. The equations describing the fluid mixing process are presented in the section on *Estimates of time required for mixing processes*.



Fig. 14.3: Backmix: a mechanical rapid mixer that has a relatively long residence time in a completely mixed flow reactor.

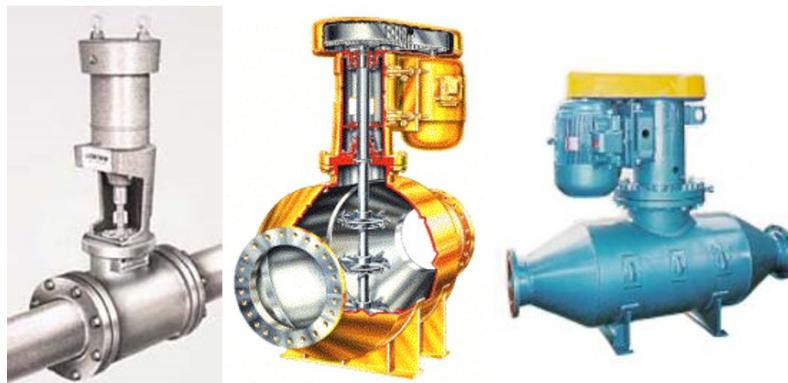


Fig. 14.4: Inline: a mechanical rapid mixer that has a short residence time in a completely mixed flow reactor that is often built into a pipe.

The hydraulic jump in Fig. 14.5 uses a flow expansion to generate mixing in an open channel and that suggests that a flow expansion could also be used to generate mixing in a closed conduit. AquaClara rapid mix units consist of an orifice in the bottom of the Linear Flow Orifice Meter (*Linear Flow Orifice Meter (LFOM)*) where the water enters the flocculator (see Fig. 14.6). However, given that fluid mixing is so easy to attain it is unclear if the energy used in the rapid mix orifice is necessary.

14.2.1 Conventional Mechanical Rapid Mix

14.2.2 Maximum Velocity Gradients



Fig. 14.5: Hydraulic jump: a hydraulic rapid mixer uses the flow expansion downstream from supercritical open channel flow.

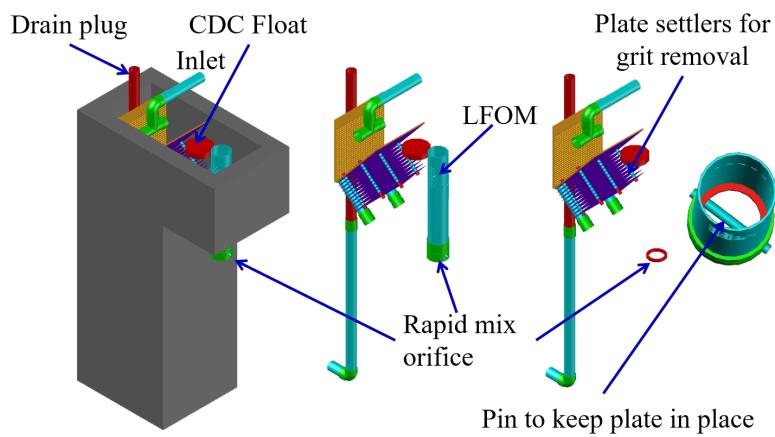


Fig. 14.6: The orifice creates a high velocity jet that generates mixing as it expands in the contact chamber prior to flocculation.

```

import numpy as np
import matplotlib.pyplot as plt
from aquaclaracore.units import unit_registry as u
import aquaclaracore.physchem as pc

Mix_HRT = np.array([0.5, 15, 25, 35, 85])*u.s
Mix_G = np.array([4000, 1500, 950, 850, 750])/u.s
Mix_CP = np.multiply(Mix_HRT, np.sqrt(Mix_G))
Mix_Gt = np.multiply(Mix_HRT, Mix_G)
Mix_EDR = (Mix_G**2*pc.viscosity_kinematic(Temperature))

fig, ax = plt.subplots()
ax.plot(Mix_G.to(1/u.s), Mix_HRT.to(u.s), 'o')
ax.yaxis.set_major_formatter(FormatStrFormatter('%.f'))
ax.xaxis.set_major_formatter(FormatStrFormatter('%.f'))
ax.set(xlabel='Velocity gradient (Hz)', ylabel='Residence time (s)')
fig.savefig(imagepath+'Mechanical_RM_Gt')
plt.show()

```

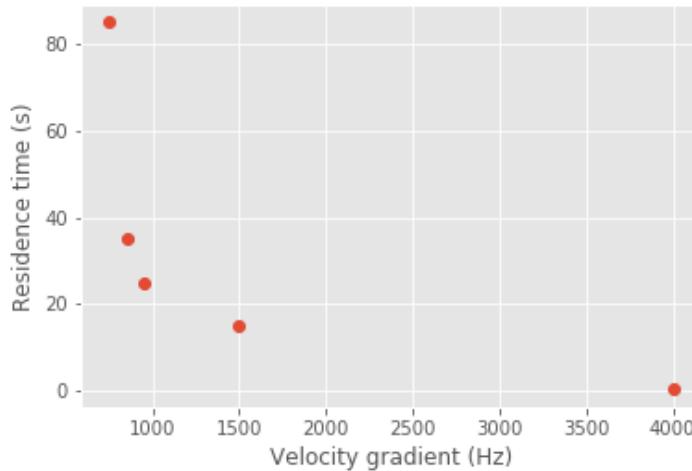


Fig. 14.7: Mechanical rapid mix units use a wide range of velocity gradients and residence times.

Conventional rapid mix units use mechanical or potential energy to generate intense turbulence to begin the mixing process. Conventional design is based on the use of \bar{G} (an average velocity gradient) as a design parameter. We don't yet know what the design objective is for rapid mix and thus it isn't clear which parameters matter. We hypothesize that both velocity gradients that cause deformation of the fluid and time for molecular diffusion are required to ultimately transport coagulant nanoparticles to the surfaces of clay particles.

The velocity gradient can be obtained from the rate at which mechanical energy is being dissipated and converted to heat by viscosity.

$$\varepsilon = G^2 \nu \quad (14.6)$$

where ε is the energy dissipation rate, G is the velocity gradient, and ν is the kinematic viscosity of water. We can estimate the power input required to create a target energy dissipation rate for a conventional design by noting that power is simply the energy dissipation rate times the mass of water in the rapid mix unit.

$$P = \bar{\varepsilon} V \rho \quad (14.7)$$

$$P = \bar{G}^2 \nu V \rho \quad (14.8)$$

We can relate reactor volume to a hydraulic residence time, θ , and volumetric flow rate, Q .

$$P = \rho \bar{G}^2 \nu Q \theta \quad (14.9)$$

This equation is perfectly useful for estimating electrical motor sizing requirements for mechanical rapid mix units. For gravity powered hydraulic rapid mix units it would be more intuitive to use the change in water surface elevation, Δh instead of power input.

$$P = \rho g Q \Delta h \quad (14.10)$$

Combining the two equations we obtain.

$$\Delta h = \frac{\bar{G}^2 \nu \theta}{g} \quad (14.11)$$

Table 14.2: Typical values for conventional rapid mix residence time and average velocity gradients

Residence Time (s)	Velocity gradient G (1/s)	Energy dissipation rate (W/kg)	Equivalent height (m)
0.5	4000	16	0.8
10 - 20	1500	2.25	2.3 - 4.6
20 - 30	950	0.9	1.8 - 2.8
30 - 40	850	0.72	2.2 - 2.9
40 - 130	750	0.56	2.3 - 7.5

From Environmental Engineering: A Design Approach by Sincero and Sincero. 1996. page 267.

Rotating propellers can either be installed in open tanks or enclosed in pipes. From a mixing and fluids perspective it doesn't make any difference whether the tank is open to the atmosphere or not. The parameters of interest are the rate of fluid deformation and the residence time in the mixing zone.

14.2.3 Mixing time

The time required for mixing in a turbulent environment is a function of the rate that kinetic energy is being dissipated as heat (the energy dissipation rate) and the length scale of the eddies. Given that turbulent energy is passed from large eddies to smaller and smaller eddies, the amount of energy that is being transferred at any given length scale is independent of scale. The result (see equation (16.49)) is that the time required for mixing is dominated by the time required for the largest eddies to turn over (Fig. 14.8).

The eddy turnover times are longest for the largest eddies and this analysis suggests that it only takes a few seconds for turbulent eddies to mix from the scale of the flow down to the inner viscous length scale.

The large scale mixing time is critical for the design of water treatment plants for the case where the flow is split into multiple treatment trains after coagulant addition. In this case it is critical that the coagulant be mixed equally between all of the treatment trains and thus the mixing times shown in the previous graph represent a minimum time between where the coagulant is added and where the flow is divided into the parallel treatment trains.

It is likely this process of mixing from the scale of the flow down to the inner viscous length scale is commonly referred to as rapid mix. Here we showed that this mixing is indeed rapid and is really only a concern in the case where the coagulant injection point is very close to the location where the flow is split into multiple treatment trains.

Fluid deformation dominated by viscous shear and molecular diffusion finish the process of blending the coagulant nanoparticles with the water. We show in Fluid_Deformation_by_Shear that the time required by fluid deformation and molecular diffusion to finish the blending process is approximately equal to $1/G$ where G is the velocity gradient.

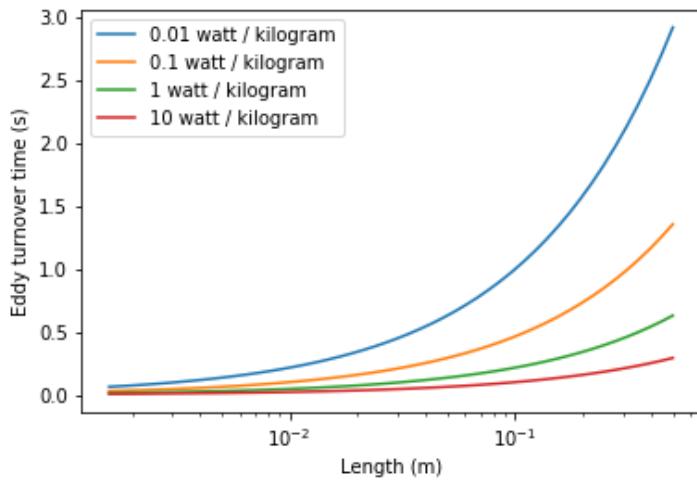


Fig. 14.8: Eddy turnover times as a function of length scale for a range of energy dissipation rates.

Given that velocity gradients in rapid mix units are typically greater than a thousand Hz the time required to finish the blending is approximately 1 ms.

Thus the time required for mixing the coagulant nanoparticles with the fluid typically only requires a few seconds and will be accomplished whether or not the rapid mix unit is turned on. The turbulent eddies from the water flowing a the channel or pipe between the coagulant injection point and the flocculator in most cases will be sufficient to achieve the fluid mixing. However, the step of the *coagulant nanoparticles attaching to the suspended particles* may be aided by the high energy of the rapid mix unit.

14.3 Coagulant Nanoparticle Interactions

Coagulant nanoparticles are sticky and can attach to suspended particles as well as to each other. Some dissolved substances also adsorb to coagulant nanoparticles. The development of models to describe these interactions has been impeded by the charge neutralization hypothesis that failed to account for the size of the coagulant nanoparticles and by the complexity of modeling all of these competing processes. Although the model describing removal of dissolved organic matter is still nascent, it is possible that a simplified approach that separates fast and slow processes will enable a sequential model.

Interactions between the various suspended and dissolved substances (see Fig. 14.9) can occur simultaneously as soon as the coagulant is blended with the raw water. The rates of these interactions are controlled by the transport processes of fluid deformation and molecular diffusion. Molecular diffusion is fastest for small particles and fluid deformation is most effective for larger particles. Thus the fastest process is hypothesized to be the diffusion of low mass molecules to the coagulant nanoparticles. Transport of the coagulant nanoparticles to attach to suspended solids is expected to be a slower process. Transport of suspended particles to collide with other suspended particles (flocculation) is even slower.

14.3.1 Dissolved Organic Matter

Dissolved organic matter (DOM) includes humic substances, fulvic acids, and other organic molecules. The distinction between dissolved and particulate organic matter is somewhat arbitrary and often 450 nm is used as the transition. The dissolved organic matter could also be referred to as macromolecules or as nanoparticles.

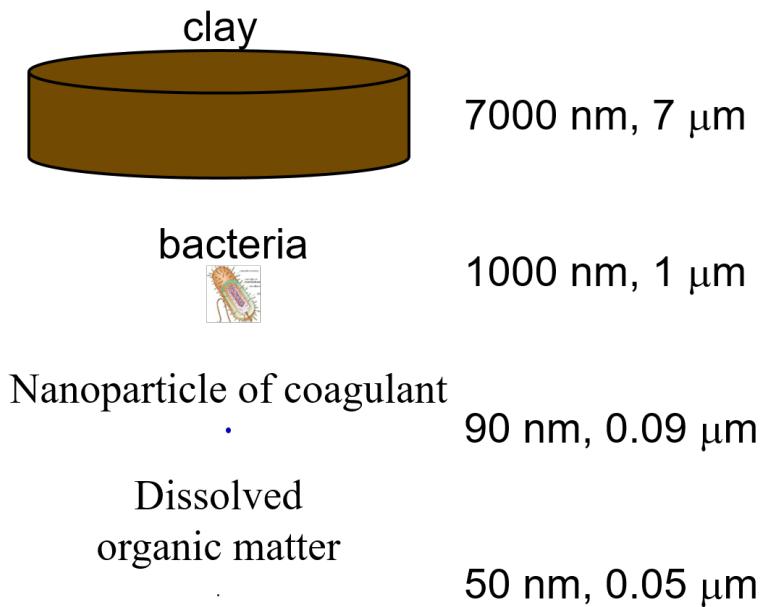


Fig. 14.9: The size range of particles and nanoparticles that are important in drinking water treatment ranges from approximately a nanometer (for example arsenic $HAsO_4^{2-}$) to thousands of nanometers for clay and protozoa.

Because of its small size the DOM has a large surface per unit mass. Water that contains high DOM concentrations requires much higher coagulant dosages to achieve effective flocculation. Removal of DOM is a high priority for drinking water treatment plants because DOM both interferes with disinfection processes and produces disinfection by products. A significant fraction of DOM can be removed by coagulant nanoparticles.

14.3.2 Suspended Solids

Suspended solids include both organic and inorganic particles. Organic particles of concern include virus, bacteria, and protozoa. Inorganic particles include clay and other minerals. Naturally occurring suspended solids tend to have negative surface charge at neutral pH. The negative surface charge effectively prevents particle aggregation and thus these particles can remain suspended for a very long time.

14.3.3 Pathogens

Virus particles readily attach to coagulant nanoparticles (see [Effects of Floc-Virus Association on Chlorine Disinfection Efficiency by Shinichiro Ohgaki and Prasang Mongkonsiri](#)) and this attachment makes it possible to efficiently remove virus particles by flocculation followed by sedimentation. Bacteria (cite Yolanda Brook paper when it is published) and protozoans (need reference) are also removed by flocculation by coagulant nanoparticles.

14.3.4 Rate Estimates for Coagulant Nanoparticle Transport to Suspended Solids

Coagulant nanoparticles require significant time to attach to the surfaces of suspended solids. The time required is estimated in [Diffusion and Shear Transport Coagulant Nanoparticles to Clay](#). It is quite possible that this stage of the rapid mix/flocculation process has been overlooked in the past. Transport of the nanoparticles to the suspended solids is accomplished by a combination of fluid deformation and diffusion.

14.4 Energy Dissipation Rate, Velocity Gradient, and Mixing

In addition to the general fluids review ([Review: Fluid Mechanics](#)), there are a few extra fluid dynamics concepts that are important:

- Turbulence
- Viscosity
- Shear
- Velocity Gradients (G), which serve as a measure of fluid deformation
- Energy Dissipation Rate (EDR, ε)

Knowledge of these concepts and how they interact is critical to understand rapid mix, flocculation, filtration, and disinfection. These concepts and their interactions first become relevant in rapid mix, the step in which the coagulant gets added to the raw water.

The two concepts that were not covered in the previous chapter, [Review: Fluid Mechanics](#), are velocity gradient G and energy dissipation rate ε . While these will be very thoroughly described over the course of this introduction, a brief and simple explanation is included to help get the ball rolling.

14.4.1 Understanding G and ε

G , or velocity gradient, is a measure of fluid deformation. It is defined by how quickly one point of water along one streamline moves in comparison to another point on another streamline (v_A compared to v_B , for example), taking into account the distance between the streamlines, Δh . A visual example of a velocity gradient is shown in the image below:

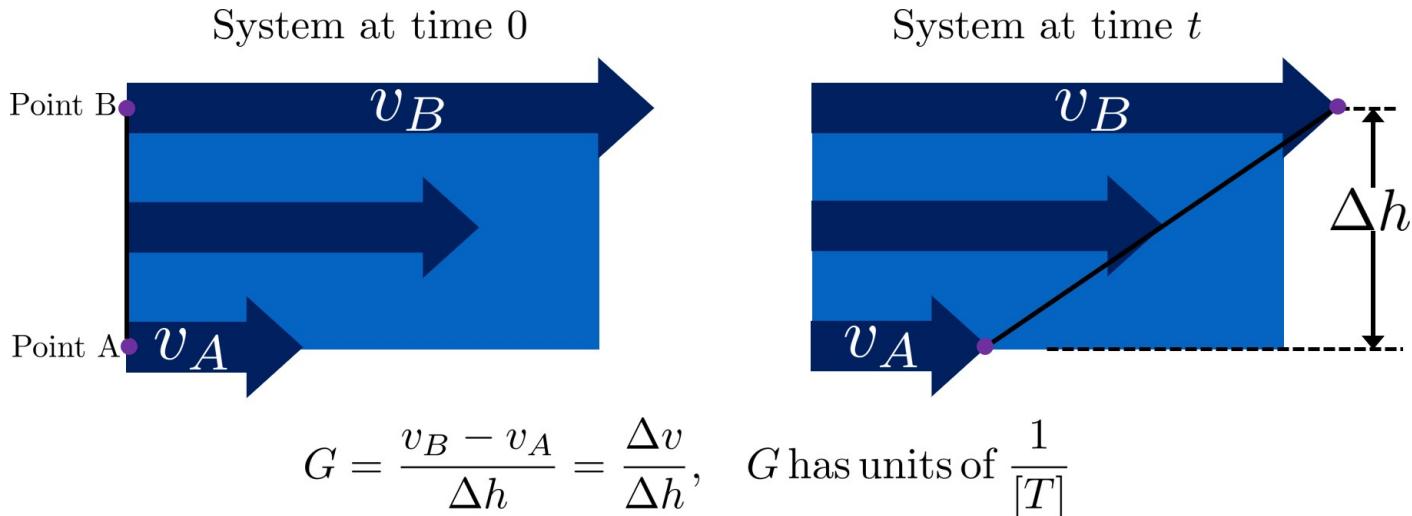


Fig. 14.10: Velocity gradients cause relative velocities of fluid elements. Those relative velocities form the basis of particle collisions that are essential for the flocculation process.

Note on terminology: Fluid deformation is equivalent to velocity gradient, and the two terms can be used interchangeably. They are different ways of thinking about the same concept. Thus, G is the measure of both terms.

ε , or energy dissipation rate, is the rate that the kinetic energy of the fluid is being converted to heat. EDR is a very useful concept because the last step of converting kinetic energy into heat is accomplished by viscosity (ν). This kinetic energy being dissipated by viscosity is the energy associated with velocity gradients (G). Thus, through EDR there is a direct connection between ν and G . This connection will be further covered later on in this introduction.

As mentioned above, EDR and velocity gradients play an important role in mixing and in causing suspended particles to collide with each other, both of which are important topics in flocculation. Their use is not limited to flocculation, they are also helpful in understanding failure modes of plate settlers and terminal head loss of sand filters.

We will begin by defining the concept of energy dissipation rate for a control volume. In a control volume that does not include pumps, turbines or other external energy sources or sinks, the mechanical energy lost is indicated by a change in elevation and quantified as gh_L . That mechanical energy is lost in the time that the fluid is in the control volume, θ .

$$\bar{\varepsilon}\theta = gh_L \quad (14.12)$$

This equation simply states that the average rate of energy dissipation times the time over which that dissipation occurs is equal to the total lost mechanical energy. The dimensions of ε are:

$$\varepsilon = \frac{[m^3]}{[s^3]} = \frac{W}{kg} \quad (14.13)$$

These dimensions can be understood as a velocity squared per time, otherwise known as a rate of kinetic energy loss (recall that kinetic energy is $Ke = \frac{\bar{v}^2}{2g}$, or $Ke \propto \bar{v}^2$), or as power per unit mass, which would be $\frac{W}{kg}$.

Velocity gradients are central to flocculation because they cause the deformation of the fluid, and this results in particle collisions. Consider a real-world example via the image below: if everyone on a sidewalk is walking in the same direction at exactly the same velocity, then there will never be any collisions between people (top). If, however, people at one side of the sidewalk stand still and people walk progressively faster as a function of how far they are away from the zero velocity side of the sidewalk, then there will be many collisions between the pedestrians (see Fig. 14.11). Indeed, the rate of collisions is proportional to the velocity gradient.

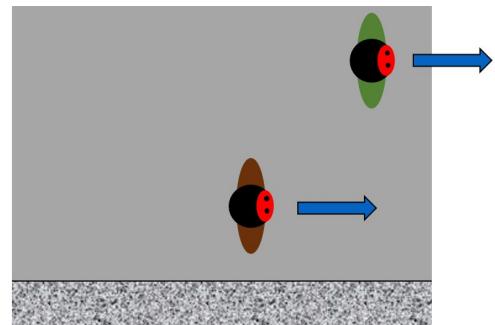
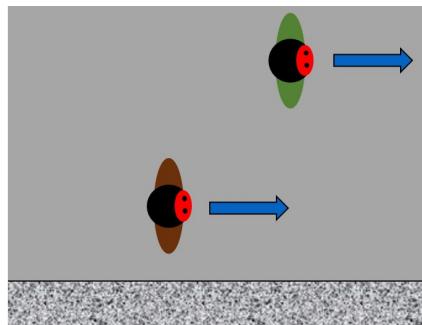
14.5 Common Flow Geometries that Dissipate Energy

Water treatment plants at research and municipal scales deploy a wide range of flow geometries. The following list includes the flow geometries that are commonly used for mixing processes.

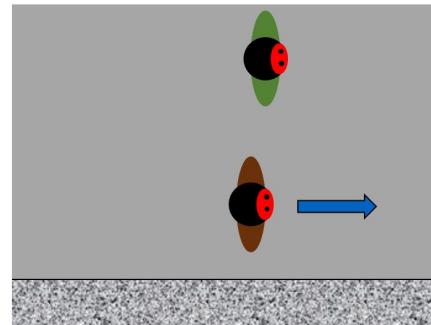
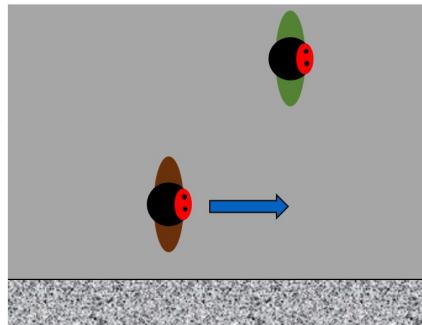
- Straight pipe (wall shear) - [uncommon, but included for completeness]
- Coiled tube (wall shear and expansions) - [research scale mixing]
- Series of expansions (expansions) - [hydraulic flocculators]
- Mechanical mixing - [mechanical rapid mix and flocculators]
- Between flat plates (wall shear) - [plate settlers]
- Round jet - (expansion) - [hydraulic rapid mix]
- Plane jet - (expansion) - [inlet into sedimentation tank]
- Behind a flat plate - (expansion) - [mechanical mixers]

The following tables can serve as a convenient reference to the equations describing head loss, energy dissipation rates, and velocity gradients in various flow geometries that are commonly encountered in water treatment plants. The *Equations for and in Varying Flow Geometries* are available as a reference.

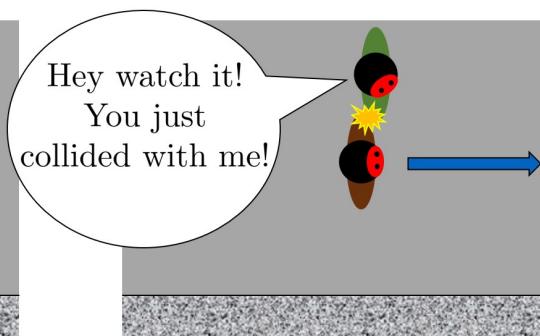
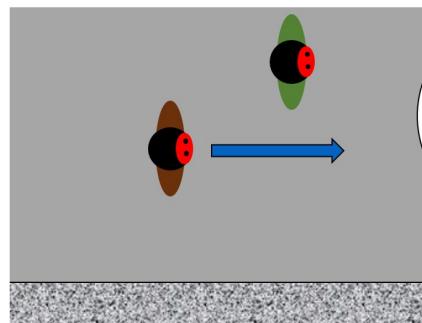
Everyone walks at the same pace. This models a velocity gradient of $G = 0$



Everyone walks at different rates depending on where along the sidewalk they are. There **ISN'T** a big difference in speed between people on opposite ends of the sidewalk. This models a **LOW G**.



Everyone walks at different rates depending on where along the sidewalk they are. There **IS** a big difference in speed between people on opposite ends of the sidewalk. This models a **HIGH G**.



Pedestrians walking on a sidewalk serve as a model for velocity gradients

Fig. 14.11: Pedestrians walking on a sidewalk serve as a model for velocity gradients.

Table 14.3: Table of equations for control volume averaged values of head loss, energy dissipation rate, and the Camp-Stein velocity gradient.

Geometry	h_L	Energy dissipation rate	$G_{CS}(barv)$	$G_{CS}(Q)$
Straight pipe	$h_f = f \frac{L}{D} \frac{\bar{v}^2}{2g}$	$\bar{\varepsilon} = \frac{f}{2} \frac{\bar{v}^3}{D}$	$G_{CS} = \left(\frac{f}{2\nu} \frac{\bar{v}^3}{D} \right)^{\frac{1}{2}}$	$G_{CS} = \left(\frac{32f}{\pi^3 \nu} \frac{Q^3}{D^7} \right)^{\frac{1}{2}}$
Straight pipe laminar	$h_f = \frac{32\nu L \bar{v}}{g D^2}$	$\bar{\varepsilon} = 32\nu \left(\frac{\bar{v}}{D} \right)^2$	$G_{CS} = 4\sqrt{2} \frac{\bar{v}}{D}$	$G_{CS} = \frac{16\sqrt{2}}{\pi} \frac{Q}{D^3}$
Parallel plates laminar	$h_f = 12 \frac{\nu L \bar{v}}{g S^2}$	$\bar{\varepsilon} = 12\nu \left(\frac{\bar{v}}{S} \right)^2$	$G_{CS} = 2\sqrt{3} \frac{\bar{v}}{S}$	•
Coiled tube laminar	$h_{L_{coil}} = \frac{32\nu L \bar{v}}{g D^2} \left[1 + 0.033 (\log_{10} De)^4 \right]^2$	$\bar{\varepsilon} = \frac{32\nu L \bar{v}}{g D^2} \left[1 + 0.033 (\log_{10} De)^4 \right]^2$	$G_{CS_{coil}} = \frac{4\sqrt{2} \frac{\bar{v}}{D} \left[1 + 0.033 (\log_{10} De)^4 \right]^{\frac{1}{2}}}{(\log_{10} De)^4}$	•
Porous media	$h_f = f_\phi \frac{L}{D_{sand}} \frac{v_a^2 (1-\phi)}{2g \phi^3}$	$\bar{\varepsilon} = \frac{f_\phi}{2} \frac{v_a^3}{D_{sand}} \frac{(1-\phi)}{\phi^4}$	$G_{CS} = \left(\frac{f_\phi}{2\nu} \frac{v_a^3}{D_{sand}} \frac{(1-\phi)}{\phi^4} \right)^{\frac{1}{2}}$	•
Expansions	$h_e = K \frac{\bar{v}_{out}^2}{2g}$	$\bar{\varepsilon} = K \frac{\bar{v}_{out}^3}{2H}$	$G_{CS} = \bar{v}_{out} \sqrt{\frac{K \bar{v}_{out}}{2H \nu}}$	•

The equations used to convert between columns in the table above are:

$$\bar{\varepsilon} = \frac{gh_L}{\theta} \quad G_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad \bar{v} = \frac{4Q}{\pi D} \quad (14.14)$$

Note that the velocity gradient is independent of viscosity (and hence temperature) for laminar flow. This is because the total amount of fluid deformation is simply based on geometry. The no slip condition, the diameter, and the length of the flow passage set the total fluid deformation. Of course, if temperature decreases and viscosity increases the amount of energy required to push the fluid through the flow passage will increase (head loss is proportional to viscosity for laminar flow).

For turbulent flow and for flow expansions the amount of fluid deformation decreases as the viscosity increases and the total energy required to send the flow through the reactor is almost independent of the viscosity. The almost is because for wall shear under turbulent conditions there is a small effect of viscosity that is buried inside the friction factor.

Table 14.4: Equations for maximum (wall) energy dissipation rates and wall velocity gradients.

Geometry	Energy dissipation rate at the wall	Velocity gradient at the wall
Straight pipe	$\varepsilon_{wall} = \frac{1}{\nu} \left(f \frac{\bar{v}^2}{8} \right)^2$	$G_{wall} = f \frac{\bar{v}^2}{8\nu}$
Straight pipe laminar	$\varepsilon_{wall} = \left(\frac{8\bar{v}}{D} \right)^2 \nu$	$G_{wall} = \frac{8\bar{v}}{D}$
parallel plates laminar	$\varepsilon_{wall} = 36 \left(\frac{\bar{v}}{S} \right)^2 \nu$	$G_{wall} = \frac{6\bar{v}}{S}$
Coiled pipe laminar	•	$G_{CS_{wallcoil}} = f \left[1 + 0.033 (\log_{10} De)^4 \right] \frac{\bar{v}^2}{8\nu}$

Table 14.5: Equations for maximum energy dissipation rates and velocity gradients for flow expansions.

Geometry	$P_{i_{Jet}}$	Maximum energy dissipation rate	Maximum velocity gradient
Round jet	0.08	$\varepsilon_{Max} = \Pi_{JetRound} \frac{\bar{v}_{jet}^3}{D_{jet}}$	$G_{Max} = \bar{v}_{jet} \sqrt{\frac{\Pi_{JetRound} \bar{v}_{jet}}{\nu D_{jet}}}$
Plane jet	0.0124	$\varepsilon_{Max} = \Pi_{JetPlane} \frac{\bar{v}_{jet}^3}{S_{jet}}$	$G_{Max} = \bar{v}_{jet} \sqrt{\frac{\Pi_{JetPlane} \bar{v}_{jet}}{\nu S_{jet}}}$
Behind a flat plate	0.04	$\varepsilon_{Max} = \Pi_{Plate} \frac{\bar{v}^3}{W_{Plate}}$	$G_{Max} = \bar{v} \sqrt{\frac{\Pi_{Plate} \bar{v}}{\nu W_{Plate}}}$

For mechanical mixing where an impeller or other stirring device is adding shaft work to a control volume we have

$$\bar{\varepsilon} = \frac{P}{m} = \frac{P}{\rho V} \quad (14.15)$$

where

P = power input into the control volume

m = mass of fluid in the control volume

V = volume of the control volume

ρ = density of the fluid

The Camp-Stein velocity gradient for a mechanically mixed reactor is

$$G_{CS} = \sqrt{\frac{P}{\rho V}} \quad (14.16)$$

**CHAPTER
FIFTEEN**

RAPID MIX DESIGN

As of 2018 the design for AguaClara rapid mix units has been based on the goal of achieving a target energy dissipation rate. This in turn was based on the assumption that it was important to rapidly mix the coagulant with the water, perhaps to minimize the self-aggregation of coagulant nanoparticles. We dont yet have any experimental evidence that rapid mixing is important and it is quite likely that the energy dissipation rate found in the hydraulic flocculator is sufficient to provide the required mixing.

The design requirements for fluid mixing of the coagulant is an area that needs research. If the goal is to achieve a velocity gradient for a number of seconds, then this design will be the same as that developed in the flocculator section. Until we have clear guidance on the goal of rapid mix we will not provide a detailed design here.

CHAPTER
SIXTEEN

RAPID MIX DERIVATIONS

16.1 Carbonate reactions, buffering, and pH

Carbonates provide the majority of the buffering for drinking water as long as the pH is close to neutral. These equations provide a basis to calculate how much base or acid must be added to a natural water to achieve a target pH.

16.1.1 Carbonic Acid and Bicarbonate



Where:

K_1 is the dissociation constant defined below.

$$K_1 = \frac{[H^+] [HCO_3^-]}{[H_2CO_3^*]} \quad (16.2)$$

Where the [] indicates concentration in mole/L. We will use the p function, $p(x) = -\log_{10}(x)$, to define the dissociation constant.

$$pK_1 = 6.3 \quad (16.3)$$

At the point of equal concentrations of bicarbonate and carbonic acid the dissociation constant, K_1 , is equal to the hydrogen ion concentration, H^+ . Thus we have equal concentrations at $pK_1 = pH$. This reaction is centered at pH = 6.3 and thus there is maximum buffering due to this reaction at pH = 6.3.

16.1.2 Bicarbonate and Carbonate



$$K_2 = \frac{[H^+] [CO_3^{2-}]}{[HCO_3^-]} \quad (16.5)$$

$$pK_2 = 10.3 \quad (16.6)$$

Thus the carbonate system also provides buffering around pH 10.3.

16.1.3 Total Concentration of Carbonates

The total concentration of carbonate species is given by

$$C_T = [H_2CO_3^*] + [HCO_3^-] + [CO_3^{2-}] \quad (16.7)$$

Where: C_T is the total concentration of carbonates.

The total concentration of carbonates, C_T , is useful because it is conservative (in a closed system) even though the individual species concentrations change as pH changes.

16.1.4 Alpha Notation

The alpha notation is used to show the concentration dependence on pH and to make the equations simpler.

$$[H_2CO_3^*] = \alpha_0 C_T \quad (16.8)$$

$$[HCO_3^-] = \alpha_1 C_T \quad (16.9)$$

$$[CO_3^{2-}] = \alpha_2 C_T \quad (16.10)$$

The alphas sum to 1 because each α is the fraction of the carbonates corresponding to that species. The alphas are each a function of the proton concentration and the dissociation constants of the carbonate reactions.

$$\alpha_0 = \frac{1}{1 + \frac{K_1}{[H^+]} + \frac{K_1 K_2}{[H^+]^2}} \quad (16.11)$$

$$\alpha_0 = \frac{1}{1 + \frac{K_1}{[H^+]} \left(1 + \frac{K_2}{[H^+]}\right)} \quad (16.12)$$

$$\alpha_1 = \frac{1}{\frac{[H^+]}{K_1} + 1 + \frac{K_2}{[H^+]}} \quad (16.13)$$

$$\alpha_2 = \frac{1}{\frac{[H^+]^2}{K_1 K_2} + \frac{[H^+]}{K_2} + 1} \quad (16.14)$$

$$\alpha_2 = \frac{1}{1 + \frac{[H^+]}{K_2} \left(1 + \frac{[H^+]}{K_1}\right)} \quad (16.15)$$

16.1.5 Acid Neutralizing Capacity (ANC) or Alkalinity

Acid neutralizing capacity or alkalinity is the ability of a water sample to react with and neutralize an input of acid. The units of ANC are equivalents (or protons) per liter. Bicarbonate, HCO_3^- , can react with one proton, H^+ , and thus each mole of HCO_3^- provides one equivalent per liter of ANC. The other terms in the equation have similar explanations.

$$ANC = [HCO_3^-] + 2[CO_3^{2-}] + [OH^-] - [H^+] \quad (16.16)$$

Note that carbonic acid and dissolved carbon dioxide are not in the ANC equation because they have no ability to neutralize protons.

We can write the ANC equation using alpha notation

$$ANC = C_T(\alpha_1 + 2\alpha_2) + \frac{K_w}{[H^+]} - [H^+] \quad (16.17)$$

For completeness we include acid neutralizing capacity for the case where the system is in equilibrium with atmospheric carbon dioxide, CO_2 .

$$ANC_{atm\ equilibrium} = \frac{PCO_2 K_H}{\alpha_0} (\alpha_1 + 2\alpha_2) + \frac{K_w}{[H^+]} - [H^+] \quad (16.18)$$

16.1.6 pH Adjustment

The final ANC, ANC_1 , after base addition and aluminum coagulant addition is given by

$$ANC_1 = ANC_0 + \Pi_{base} C_B + \Pi_{Al} C_{Al} \quad (16.19)$$

where:

ANC_0 is the initial acid neutralizing capacity of the water sample.

ANC_1 is the final acid neutralizing capacity of the mixture after the base and aluminum coagulant is added.

C_B is concentration of base in mole/liter

Π_{base} is ANC per mole of base

C_{Al} is the concentration of coagulant in mole of aluminum/liter

Π_{Al} is ANC per mole of aluminum

The final carbonate concentration is given by

$$C_{T_1} = C_{T_0} + \Pi_{CO_3^{-2}} C_B \quad (16.20)$$

where:

C_{T_1} is the final total carbonate concentration of the mixture after the base is added.

$\Pi_{CO_3^{-2}}$ is mole of carbonate per mole of base (0 for $NaOH$ and 1 for Na_2CO_3)

Substituting these values into the ANC equation we obtain

$$ANC_0 + \Pi_{base} C_B + \Pi_{Al} C_{Al} = (C_{T_0} + \Pi_{CO_3^{-2}} C_B)(\alpha_1 + 2\alpha_2) + \frac{K_w}{[H^+]} - [H^+] \quad (16.21)$$

Now we solve for C_B , the concentration of base that must be added to reach a target pH.

$$(\Pi_{base} - \Pi_{CO_3^{-2}}(\alpha_1 + 2\alpha_2))C_B = C_{T_0}(\alpha_1 + 2\alpha_2) + \frac{K_w}{[H^+]} - [H^+] - ANC_0 - \Pi_{Al} C_{Al} \quad (16.22)$$

$$C_B = \frac{C_{T_0}(\alpha_1 + 2\alpha_2) + \frac{K_w}{[H^+]} - [H^+] - ANC_0 - \Pi_{Al} C_{Al}}{\Pi_{base} - \Pi_{CO_3^{-2}}(\alpha_1 + 2\alpha_2)} \quad (16.23)$$

Note that the equations above can also be used for the case where acid is added to reduce the pH. In that case Π_{base} will have a negative value.

An example using this equation to find the required amount of base addition is given in *Example: pH Adjustment*.

16.2 Equations for ε and G in Varying Flow Geometries

Estimation of velocity gradients for various flow geometries is the basis for the design of rapid mix, flocculators, and plate settlers. Thus, our goal is to define the velocity gradients consistently across a range of possible flow regimes. There are three approaches to calculating the average velocity gradient within a control volume. 1) Use the

Navier Stokes equations and solve for the spatially averaged velocity gradient. 1) Use Computational Fluid Dynamics (CFD) to solve for the spatially averaged velocity gradient. 1) Use the total mechanical energy loss in the control volume to calculate the energy dissipation rate. Estimate the velocity gradient directly from the energy dissipation rate, $G_{CS} = \sqrt{\frac{\varepsilon}{\nu}}$, as defined by Camp and Stein in 1943 (Camp, T. R., and Stein, P. C. (1943) Velocity Gradients and Hydraulic Work in Fluid Motion, J. Boston Soc. Civil Eng., 30, 203–221.).

The first approach would be ideal but is difficult in practice because Navier Stokes solutions are only available for limited geometries and laminar flow. CFD could be used but is difficult to use as a general engineering design approach given the large number of geometries that are used in drinking water treatment plants. For these reasons we will use the control volume approach to estimate the average velocity gradient. This method incorrectly assumes that the energy dissipation rate is completely uniform in the control volume and hence the velocity gradient is also uniform. This method results in an over estimation of the velocity gradient. The Camp-Stein estimate of G_{CS} is based on a control volume where the velocity gradient is uniform. Consider a layer of fluid of depth H and apply a velocity, v at the top of the fluid. The velocity gradient, G , is thus $\frac{v}{H}$ everywhere in the fluid. The force required to move the top of the fluid at velocity v can be obtained from the required shear, τ . From Newtons Law of Friction we have

$$\tau = \mu \frac{v}{H} = \mu G = \nu \rho G \quad (16.24)$$

Where τ is the force required per unit plan view area. The power per unit area required to move the fluid at velocity v is τv . The mass per unit area is ρH . Thus the energy dissipation rate or the power per mass is

$$\varepsilon = \frac{P}{m} = \frac{\tau v}{\rho H} = \frac{\nu \rho G v}{\rho H} = \nu G^2 \quad (16.25)$$

This equation has no approximations, but has one very important assumption. We derived this equation for a control volume where the velocity gradient was **uniform**. The reactors and control volumes that we will be using as we design water treatment plants will **not** have uniform velocity gradients. Indeed, several of the water treatment processes will be turbulent and thus the velocity gradients in the fluid will vary in both space and time. Even in laminar flow in a pipe the velocity gradient is far from uniform with high velocity gradients at the wall and zero velocity gradient at the center of the pipe.

We'd like to know if we can apply the previous equation

$$\varepsilon = \nu G^2 \quad (16.26)$$

to the case where the energy dissipation rate and velocity gradients are nonuniform by simply introducing average values of both quantities.

$$\bar{\varepsilon} \stackrel{?}{=} \nu \bar{G}^2 \quad (16.27)$$

We will test this option with a simple case. Consider a hypothetical reactor (case 2) that is 4 times as large in plan view area as the uniform velocity gradient case explored above (case 1). In addition, assume that 3/4 of the reactor has a velocity gradient of zero. The average energy dissipation rate for case 1 is

$$\bar{\varepsilon}_1 = \frac{P_1}{m_1} = \nu \bar{G}_1^2 \quad (16.28)$$

The average energy dissipation rate for case 2 is

$$\bar{\varepsilon}_2 = \frac{P_1}{4m_1} = \frac{\bar{\varepsilon}_1}{4} \quad (16.29)$$

This makes sense because we are putting in the same amount of energy into a control volume that is 4 times bigger.

Now we calculate the velocity gradients. As previously determined,

$$\bar{G}_1 = \sqrt{\frac{\bar{\varepsilon}_1}{\nu}} \quad (16.30)$$

The average velocity gradient in the second control volume is simply the volume weighted average

$$\bar{G}_2 = \bar{G}_1 \frac{1}{4} + 0 \frac{3}{4} \quad (16.31)$$

where 1/4 of the case 2 control volume has the same velocity gradient as the case 1 control volume and 3/4 of the control volume has a velocity gradient of 0. The Camp Stein method would suggest that \bar{G}_2 is equal to

$$\bar{G}_2 \stackrel{?}{=} \sqrt{\frac{\bar{\varepsilon}_2}{\nu}} = \sqrt{\frac{\bar{\varepsilon}_1}{4\nu}} \quad (16.32)$$

Now we check to see if the Camp Stein method of estimating the average velocity gradient, \bar{G} , is correct.

$$\bar{G}_2 = \frac{\bar{G}_1}{4} \neq \sqrt{\frac{\bar{\varepsilon}_1}{4\nu}} = \frac{\bar{G}_1}{2} \quad (16.33)$$

Given that the energy dissipation rate is proportional to the square of the velocity gradient the mean of the energy dissipation rate is **not** proportional to the mean of the velocity gradient. Thus the Camp Stein method of calculating the average velocity gradient is not correct except in the case of uniform velocity gradient. The Camp Stein equation is dimensionally correct and could be corrected by adding a dimensionless constant Π_{CS} that is a function of the energy dissipation rate distribution within the control volume.

$$\bar{G} = \Pi_{CS} \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad (16.34)$$

where Π_{CS} is 1 for a uniform velocity gradient and is less than one for non uniform velocity gradients. We can think Π_{CS} as a measure of the efficiency of using energy to deform the fluid. We can calculate Π_{CS} for cases where we have either a Navier Stokes or a computation fluid dynamics estimate of \bar{G} .

The conventional approach to design of flocculators uses the Camp Stein definition of

$$G_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad (16.35)$$

where G_{CS} is **not** the average velocity gradient, but is larger than the average velocity gradient by a factor of Π_{CS} . Thus we have

$$G_{CS} = \Pi_{CS} \bar{G} \quad (16.36)$$

Use of the Camp Stein velocity gradient in design of mixing units and flocculators results in an error when applying results from one reactor to another. If the energy dissipation rate distribution within the reactors is different, then Π_{CS} will be different for the two reactors and the actual average velocity gradient, \bar{G} will be different for the two reactors.

Given that energy is used more efficiently to produce velocity gradients if the velocity gradients are uniform, our goal is to design mixing and flocculation units that have relatively uniform velocity gradients. If all of our reactors at both research scale and municipal scale have similar values of Π_{CS} , then we can use the Camp Stein definition of G_{CS} and not introduce any significant errors. It will not be reasonable, however, to expect similar performance based on similar values of G_{CS} if one reactor has relatively uniform energy dissipation rates and the other reactor has zones with very high energy dissipation rates and zones with very low energy dissipation rates.

We will demonstrate later that mechanically mixed reactors typically have a much wider range of energy dissipation rates than do well designed hydraulically mixed reactors. Thus comparisons between mechanically mixed and hydraulically mixed reactors must account for differences in Π_{CS} .

We will use the Camp Stein definition $G_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}}$ as the design parameter of convenience in this textbook.

16.3 Estimates of time required for mixing processes

16.3.1 Turbulent Large Scale Eddies

The first step in mixing is at the scale of the largest eddies. The largest eddies are limited in size by the smallest dimension normal to the direction of flow. Thus in a pipe the dimension of the largest eddies is set by the pipe diameter. In an open channel the dimension of the largest eddies is usually the water depth although it could be the width of the channel for the case of a narrow, deep channel.

Eddy turnover time, t_{eddy} , is the time it takes for the eddy to travel a distance equal to its length-scale. Thus the eddy turnover time provides a good estimate of the time required for mixing to occur at the length scale of the eddy. We assume that the energy of the large eddy is dissipated into smaller length scales in the time t_{eddy} :

$$t_{eddy} \approx \frac{L_{eddy}}{v_{eddy}} \quad (16.37)$$

The rate of energy loss to smaller scales is

$$\bar{\varepsilon} \approx \frac{v_{eddy}^2}{t_{eddy}} \quad (16.38)$$

Combining the two equations

$$\bar{\varepsilon} \approx \frac{v_{eddy}^3}{L_{eddy}} \quad (16.39)$$

We can use this equation to estimate the eddy velocity given an energy dissipation rate.

$$v_{eddy} \approx (\bar{\varepsilon} L_{eddy})^{\frac{1}{3}} \quad (16.40)$$

Now we can solve for the eddy turnover time which is a measure of the mixing time at the eddy scale.

$$t_{eddy} \approx \frac{L_{eddy}}{(\bar{\varepsilon} L_{eddy})^{\frac{1}{3}}} \approx \left(\frac{L_{eddy}^2}{\bar{\varepsilon}} \right)^{\frac{1}{3}} \quad (16.41)$$

This provides a simple insight that the time required for an eddy to turn over scales with the size of the eddy raised to the $2/3$ power. Thus large eddies take more time to turn over than do small eddies. Thus if we calculate the time required for large scale mixing using the dimension of the eddies, it will provide a reasonable estimate of the total time for mixing because mixing at all smaller scales requires much less time. A notable exception to this is the case of mixing in rivers. Rivers are usually shallow and wide. The largest eddies in a river are limited by the depth of the river. Mixing over the width of the river takes much longer than vertical mixing because multiple eddies are required to transport a substance from one side of the river to the other.

We can use the eddy velocity to estimate how long it will take for an eddy to cross the smallest dimension of flow. Eddy velocity is $v_{eddy} \approx (\bar{\varepsilon} L_{eddy})^{\frac{1}{3}}$. The \approx indicates that this relationship is the same order of magnitude.

Chemical injection into the center of a pipe is common in drinking water treatment plants. We can develop equations to estimate the distance required for full mixing with the fluid in the pipe. In a pipe we have

$$v_{eddy} \approx (\bar{\varepsilon} D)^{\frac{1}{3}} \quad (16.42)$$

For a long straight pipe $\bar{\varepsilon} = \frac{f \bar{v}^3}{2 D}$ (Equation (16.54)) and thus we can obtain the ratio between mean velocity and the velocity of the large scale eddies.

$$v_{eddy} \approx \left(\frac{f \bar{v}^3}{2 D} D \right)^{\frac{1}{3}} \quad (16.43)$$

$$\frac{v_{eddy}}{\bar{v}} \approx \left(\frac{f}{2}\right)^{\frac{1}{3}} \quad (16.44)$$

Given a friction factor of 0.02, the eddy velocity is approximately 20% of the mean velocity. We can use this ratio to estimate how many pipe diameters downstream from an injection point will the coagulant be mixed across the diameter of the pipe.

$$: label : mixing_{pipe diameters}$$

$$N_{D_{pipe}} \approx \frac{\bar{v}}{v_{eddy}} \approx \left(\frac{2}{f}\right)^{\frac{1}{3}} \quad (16.45)$$

Where $N_{D_{pipe}}$ is the distance in number of pipe diameters downstream of the injection point where complete mixing will have occurred. This estimate is a minimum distance and a factor of safety of 2 or more would reasonably be applied. In addition it is best practice to inject the coagulant in the center of the pipe. Injecting the coagulant at the side of the pipe will require considerably greater distance downstream for mixing across the pipe.

```
print((0.02/2)**(1/3))
```

16.3.2 Inner Viscous Length Scale

The smallest scale at which inertia containing eddies causes mixing is set by the final damping of inertia by viscosity. Turbulence occurs when fluid inertia is too large to be damped by viscosity. The ratio of inertia to viscosity is given by the Reynolds number, Re:

$$Re = \frac{\bar{v}D}{\nu} \quad (16.46)$$

Flows with high Reynolds numbers are turbulent (inertia dominated) and with low Reynolds are laminar (viscosity dominated). The transition Reynolds number is a function of the flow geometry and the velocity and length scale that are used to characterize the flow. In all turbulent flows there is a length scale at which inertia finally loses to viscosity. The scale where viscosity wins is some multiple of the Kolmogorov length scale, which is defined as:

$$\eta_K = \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \quad (16.47)$$

where η_K is the Kolmogorov length scale. At the Kolmogorov length scale viscosity completely dampens the inertia of the eddies and effectively kills the turbulence.

The length scale at which most of the kinetic energy contained in the small eddies is dissipated by viscosity is the inner viscous length scale, λ_v , which according to Dimotakis (2000) is about 50 times larger than η_K . Thus we have

$$\lambda_v = \Pi_{K\nu} \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \quad (16.48)$$

where $\Pi_{K\nu} = 50$

At length scales larger than the inner viscous length scale, λ_v , the dominant transport mechanism is by turbulent eddies. At length scales smaller than λ_v the dominant transport mechanism is fluid deformation due to shear. If the flow regime is completely laminar such as in a small diameter tube flocculator, then the dominant transport mechanism is fluid deformation due to shear at length scales all the way up to the diameter of the tubing.

The dividing line between eddy transport and fluid deformation controlled by viscosity can be calculated as a function of the energy dissipation rate using (16.48).

```
""" importing """

from aquaclaracore.units import unit_registry as u
import aquaclaracore.utility as ut
import numpy as np
import matplotlib.pyplot as plt
import aquaclararesearch.floc_model as fm

from matplotlib.ticker import FormatStrFormatter
imagepath = 'Rapid_Mix/Images/'
EDR_array = np.logspace(0, 4, num=50) * u.mW/u.kg
Temperature = 20*u.degC
def Inner_viscous(EDR, Temperature):
    return fm.RATIO_KOLMOGOROV * fm.eta_kolmogorov(EDR, Temperature)

fig, ax = plt.subplots()
ax.semilogx(EDR_array.to(u.mW/u.kg), Inner_viscous(EDR_array, Temperature).to(u.mm))
ax.yaxis.set_major_formatter(FormatStrFormatter('%.f'))
ax.xaxis.set_major_formatter(FormatStrFormatter('%.f'))
ax.set(xlabel='Energy dissipation rate (W/kg)', ylabel='Inner viscous length scale ↴ (mm)')
ax.text(30, 6, 'Eddies cause mixing', fontsize=12, rotation=-30)
ax.text(1, 5, 'Shear and diffusion cause mixing', fontsize=12, rotation=-30)
fig.savefig(imagepath+'Inner_viscous_vs_EDR')
plt.show()
```

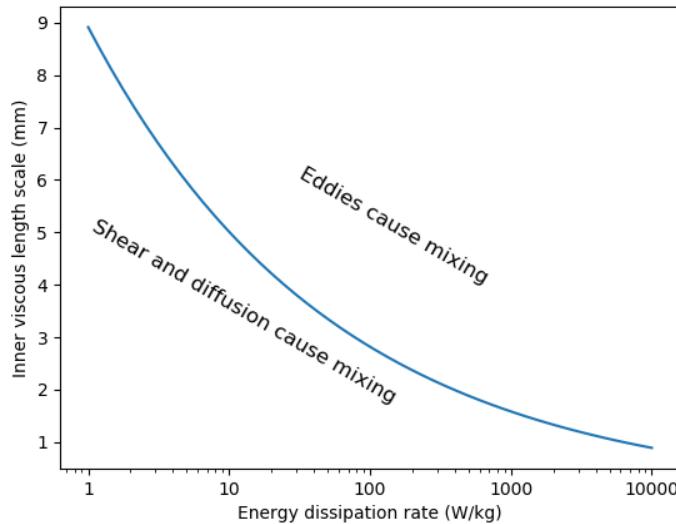


Fig. 16.1: Eddies can cause fluid mixing down to the scale of a few millimeters for energy dissipation rates used in rapid mix units and flocculators.

16.3.3 Turbulent Mixing Time as a Function of Scale

We are searching for the rate limiting step in the mixing process as we transition from the scale of the flow down to the scale of the coagulant nanoparticles. We can estimate the time required for eddies to mix at their length scales by assuming that the eddies pass all of their energy to smaller scales in the time it takes for an

eddy to travel the distance equal to the length scale of the eddy. This time is known as the ‘**eddy turnover time**’ [‘\[http://ceeserver.cee.cornell.edu/eac20/cee637/handouts/TURBFLOW_1.pdf\]\(http://ceeserver.cee.cornell.edu/eac20/cee637/handouts/TURBFLOW_1.pdf\)’](http://ceeserver.cee.cornell.edu/eac20/cee637/handouts/TURBFLOW_1.pdf), t_{eddy} . The derivation for the equation below is found here.

$$t_{eddy} \approx \left(\frac{L_{eddy}^2}{\bar{\varepsilon}} \right)^{\frac{1}{3}} \quad (16.49)$$

We can plot the eddy turnover time as a function of scale from the inner viscous length scale up to the scale of the flow. We will discover whether large scale mixing by eddies is faster or slower than small scale mixing by eddies.

```
from aquaclara.core.units import unit_registry as u
import aquaclara.core.utility as ut
import numpy as np
import matplotlib.pyplot as plt
import aquaclara.research.floc_model as fm
EDR_graph = np.array([0.01, 0.1, 1, 10]) * u.W/u.kg
"""Use the highest EDR to estimate the smallest length scale"""
Inner_viscous_graph = Inner_viscous(EDR_graph[2], Temperature)
Inner_viscous_graph
L_flow = 0.5*u.m
L_scale = np.logspace(np.log10(Inner_viscous_graph.magnitude), np.log10(L_flow.
    ↪magnitude), 50)
L_scale
imagepath = 'Rapid_Mix/Images/'
fig, ax = plt.subplots()
for i in range(len(EDR_graph)):
    ax.semilogx(L_scale, ((L_scale**2/EDR_graph[i])***(1/3)).to_base_units())
ax.legend(EDR_graph)

#ax.yaxis.set_major_formatter(FormatStrFormatter('%.f'))
#ax.xaxis.set_major_formatter(FormatStrFormatter('%.f'))
ax.set(xlabel='Length (m)', ylabel='Eddy turnover time (s)')
fig.savefig(imagepath+'Eddy_turnover_time')
plt.show()
```

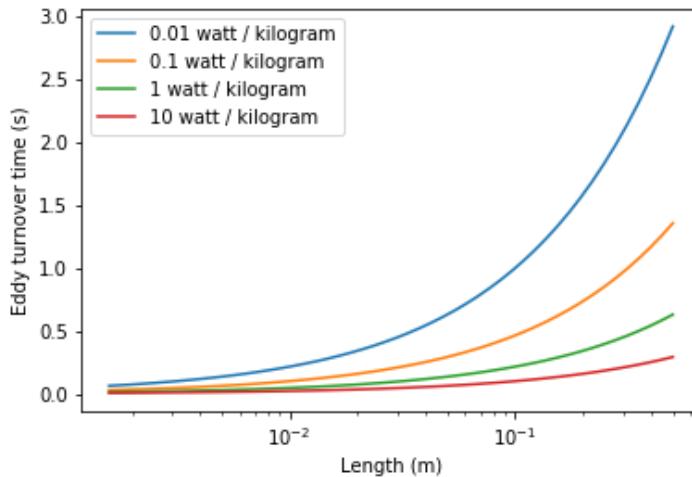


Fig. 16.2: Small eddies turn in less time than large eddies. This is why the mixing at the largest scale dominates the mixing time.

16.3.4 Shear-Diffusion Transport

After the first few seconds in which mixing occurs from the length scale of the flow down to the inner viscous length scale the next step in the transport process is blending of the coagulant uniformly with the raw water. At the end of the turbulent transport the coagulant stock has been stretched out into thin bands throughout the raw water, but the two fluids are not actually blended together by turbulence. The blending is accomplished by a combination of fluid deformation controlled by viscous shear and then by molecular diffusion.

16.3.5 Fluid Deformation by Shear

The time scale for fluid deformation is $1/G$ where G is the velocity gradient. This simple relationship is because the velocity of fluid deformation is proportional to the length scale and thus the time to travel any given distance in a linear velocity gradient is always the same. Velocity gradients in conventional mechanized rapid mix units are order 1000 Hz and thus the time for fluid deformation to blur concentration gradients is approximately 1 ms. This confirms the idea that blending the coagulant with the raw water is actually a very fast process with the slowest phase being the transport by turbulent eddies at the scale of reactor.

The full time required for fluid deformation to achieve blending down to the scale where molecular diffusion takes over is likely a multiple of $1/G$ where the multiple is determined by the number of different directions that the fluid must be sheared in to reach close to uniform blending. However, even multiplying $1/G$ by a factor of 10 still results in very rapid mixing.

16.3.6 Einsteins Diffusion Equation

The final step of mixing is mediated by molecular diffusion. We can estimate the length scale at which fluid shear and diffusion provide transport at the same rate. Einsteins diffusion equation is

$$D_{Diffusion} = \frac{k_B T}{3\pi\mu d_P} \quad (16.50)$$

where k_B is the Boltzmann constant and d_P is the diameter of the particle that is diffusion in a fluid with viscosity ν and density ρ . The diffusion coefficient $D_{Diffusion}$ has dimensions of $\frac{[L^2]}{[T]}$ and can be understood as the velocity of the particle multiplied by the length of the mean free path. From dimensional analysis the time for diffusion to blur a concentration gradient over a length scale, $L_{Diffusion}$ is

$$t_{Diffusion} \approx \frac{L_{Diffusion}^2}{D_{Diffusion}} \quad (16.51)$$

The shear time scale is $1/G$ and thus we can solve for the length scale at which diffusion and shear have equivalent transport rates.

$$1/G \approx t_{Diffusion} \approx \frac{L_{Diffusion}^2}{D_{Diffusion}} \quad (16.52)$$

Substitute Einsteins diffusion equation and solve for the length scale that transitions between shear and diffusion transport.

$$L_{Diffusion}^{Shear} \approx \sqrt{\frac{k_B T}{3G\pi\mu d_P}} \quad (16.53)$$

```

import aquaclara.core.physchem as pc
import aquaclara.research.floc_model as fm
def L_Shear_Diffusion(G, Temperature, d_particle):
    return np.sqrt((u.boltzmann_constant*Temperature/
(3 * G * np.pi *pc.viscosity_dynamic(Temperature)* d_particle)).to_base_units())

G = np.arange(10,5000)*u.Hz
d_particle = fm.PACl.Diameter*u.m
Temperature=20*u.degC
x = (L_Shear_Diffusion(G, Temperature, d_particle)).to(u.nm)
imagepath = 'Rapid_Mix/Images/'
fig, ax = plt.subplots()
ax.semilogx(G,x)
ax.set(xlabel='Velocity gradient (Hz)', ylabel='Length scale (nm)')
fig.savefig(imagepath+'Shear_diffusion_length_scale')
plt.show()

```

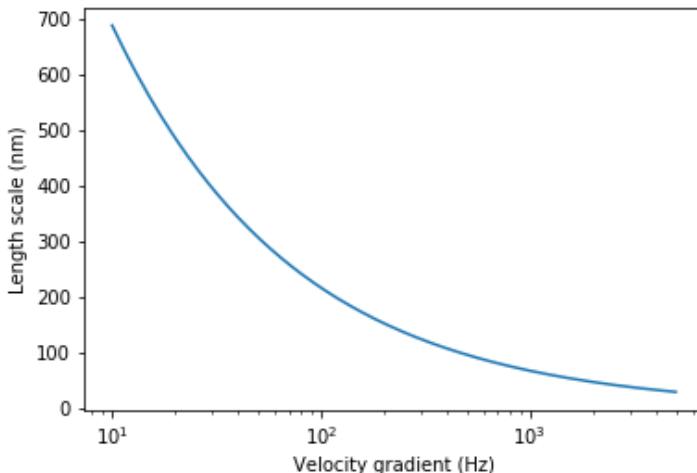


Fig. 16.3: The length scale at which diffusion becomes the dominant transport mechanism for coagulant nanoparticles as a function of the velocity gradient. The time scale for the final diffusion to achieve blending of the nanoparticles with the water is simply $1/G$.

Molecular diffusion finishes the blending process by transporting the coagulant nanoparticles the last few hundred nanometers. The entire mixing process from the coagulant injection point to uniform blending with the raw water takes only a few seconds.

We have demonstrated that all of the steps for mixing of the coagulant nanoparticles with the raw water are very fast. Compare

1. Molecular diffusion causes some dissolved species and Al nanoparticles to aggregate. 1. Fluid shear and molecular diffusion cause Al nanoparticles with attached formerly dissolved species to collide with inorganic particles (such as clay) and organic particles (such as viruses, bacteria, and protozoans).

The time scale for the fluid shear and molecular diffusion to cause coagulant nanoparticles to collide with particles is estimated in [Diffusion and Shear Transport Coagulant Nanoparticles to Clay](#).

Below are the derivations for the equations that appear in Table 14.3 containing equations for G , ε , and h_L .

16.3.7 Straight pipe (wall shear)

The average energy dissipation rate, $\bar{\varepsilon}$, in a control volume with residence time θ is

$$\bar{\varepsilon} = \frac{gh_L}{\theta} \quad (16.54)$$

The residence time can be expressed as a function of length and average velocity.

$$\theta = \frac{L}{\bar{v}} \quad (16.55)$$

For straight pipe flow the only head loss is due to wall shear and thus we have the Darcy Weisbach equation.

$$h_f = f \frac{L}{D} \frac{\bar{v}^2}{2g} \quad (16.56)$$

Combining the 3 previous equations we obtain the energy dissipation rate for pipe flow

$$\bar{\varepsilon} = \frac{f}{2} \frac{\bar{v}^3}{D} \quad (16.57)$$

The average velocity gradient was defined by Camp and Stein as

$$G_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad (16.58)$$

where this approximation neglects the fact that square root of an average is not the same as the average of the square roots.

$$G_{CS} = \left(\frac{f}{2\nu} \frac{\bar{v}^3}{D} \right)^{\frac{1}{2}} \quad (16.59)$$

or in terms of flow rate, we have:

$$G_{CS} = \left(\frac{32f}{\pi^3 \nu} \frac{Q^3}{D^7} \right)^{\frac{1}{2}} \quad (16.60)$$

16.3.8 Straight Pipe Laminar

Laboratory scale apparatus is often limited to laminar flow where viscosity effects dominate. The equations describing laminar flow conditions always include viscosity. For the case of laminar flow in a straight pipe, we have:

$$f = \frac{64}{Re} \quad (16.61)$$

Reynolds number is defined as

$$Re = \frac{\bar{v}D}{\nu} \quad (16.62)$$

The Darcy Weisbach head loss equation simplifies to the Hagen–Poiseuille equation for the case of laminar flow.

$$h_f = \frac{32\nu L \bar{v}}{g D^2} \quad (16.63)$$

and thus the energy dissipation rate in a straight pipe under conditions of laminar flow is

$$\bar{\varepsilon} = 32\nu \left(\frac{\bar{v}}{D} \right)^2 \quad (16.64)$$

The Camp-Stein velocity gradient in a long straight laminar flow tube is thus

$$G_{CS}^2 = 32 \left(\frac{\bar{v}}{D} \right)^2 \quad (16.65)$$

$$G_{CS} = 4\sqrt{2} \frac{\bar{v}}{D} \quad (16.66)$$

Our estimate of G_{CS} based on \bar{v} is an overestimate because it assumes that the energy dissipation is completely uniform through the control volume. The true spatial average velocity gradient, \bar{G} , for laminar flow in a pipe is (Gregory, 1981),

$$\bar{G} = \frac{8}{3} \frac{\bar{v}}{D} \quad (16.67)$$

Our estimate of G_{CS} for the case of laminar flow in a pipe is too high by a factor of $\frac{3}{\sqrt{2}}$.

As a function of flow rate we have

$$\bar{v} = \frac{Q}{A} = \frac{4Q}{\pi D^2} \quad (16.68)$$

$$G_{CS} = \frac{16\sqrt{2}}{\pi} \frac{Q}{D^3} \quad (16.69)$$

16.3.9 Parallel Plates Laminar

Flow between parallel plates occurs in plate settlers in the sedimentation tank. We will derive the velocity gradient at the wall using the Navier Stokes equation.

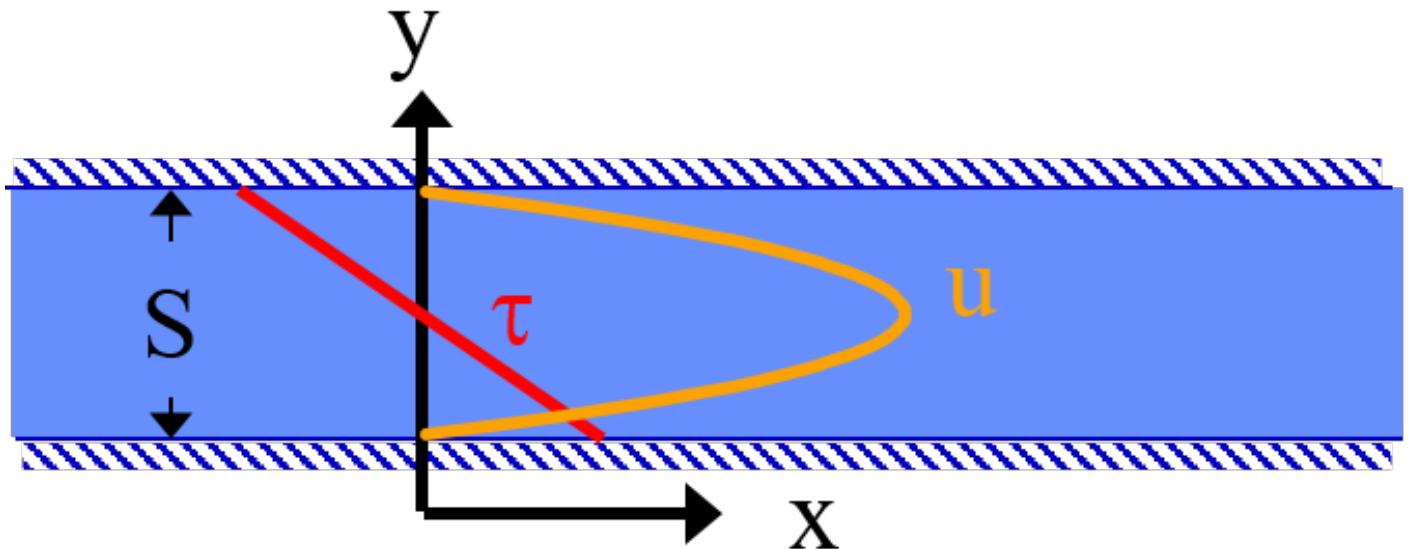


Fig. 16.4: A fluid flowing from left to right due to a pressure gradient results in wall shear on the parallel plates. This flow profile is for the case when $\frac{dp}{dx}$ is negative.

We start with the Navier-Stokes equation written for flow in the x direction.

$$\frac{y^2}{2} \frac{dp}{dx} + Ay + B = \mu u \quad (16.70)$$

where u is the velocity in the x direction.

Apply the no slip condition at bottom plate.

$$u = 0 \quad \text{at} \quad y = 0 \quad (16.71)$$

Thus the constant $B = 0$.

Apply the no slip condition at top plate.

$$u = 0 \quad \text{at} \quad y = S \quad (16.72)$$

Thus the constant $A = -\frac{S}{2} \frac{dp}{dx}$

Substitute the values for constants A and B into the original equation.

$$\frac{y^2}{2} \frac{dp}{dx} - \frac{S}{2} \frac{dp}{dx} y = \mu u \quad (16.73)$$

Simply the equation to obtain

$$u = \frac{y(y-S)}{2\mu} \frac{dp}{dx} \quad (16.74)$$

We need a relationship between average velocity and $\frac{dp}{dx}$. We can obtain this by integrating from 0 to S .

$$\bar{v} = \frac{q}{S} = \frac{1}{S} \int_0^S u dy = \frac{1}{S} \int_0^S \left(\frac{y^2 - Sy}{2\mu} \left(\frac{dp}{dx} \right) \right) dy \quad (16.75)$$

$$\bar{v} = -\frac{S^2}{12\mu} \frac{dp}{dx} \quad (16.76)$$

Solving for $\frac{dp}{dx}$

$$\frac{dp}{dx} = -\frac{12\mu\bar{v}}{S^2} \quad (16.77)$$

From the Navier Stokes equation after integrating once we get

$$\mu \left(\frac{du}{dy} \right) = y \frac{dp}{dx} + A \quad (16.78)$$

Substituting our boundary condition, $A = -\frac{S}{2} \frac{dp}{dx}$ we obtain

$$\frac{du}{dy}_{y=0} = -\frac{S}{2\mu} \frac{dp}{dx} \quad (16.79)$$

Substituting the result for $\frac{dp}{dx}$ we obtain

$$\frac{du}{dy}_{y=0} = \frac{6\bar{v}}{S} \quad (16.80)$$

Therefore in velocity gradient notation we have

$$G_{wall} = \frac{6\bar{v}}{S} \quad (16.81)$$

The energy dissipation rate at the wall

$$\varepsilon_{wall} = G_{wall}^2 \nu \quad (16.82)$$

$$\varepsilon_{wall} = \left(\frac{6\bar{v}}{S} \right)^2 \nu \quad (16.83)$$

Head loss due to shear on the plates is obtained from a force balance on a control volume between two parallel plates as shown in Fig. 16.4.

A force balance on a control volume gives

$$2\tau LW = -\Delta PWS \quad (16.84)$$

$$\Delta P = -\frac{2\tau L}{S} \quad (16.85)$$

The equation relating shear and velocity gradient is

$$\tau = \nu\rho \frac{du}{dy} = \nu\rho G \quad (16.86)$$

The velocity gradient at the wall is

$$G_{wall} = \frac{6\bar{v}}{S} \quad (16.87)$$

$$\tau = \nu\rho \frac{6\bar{v}}{S} \quad (16.88)$$

Substituting into the force balance equation

$$\Delta P = -\frac{2\nu\rho 6\bar{v}L}{S^2} \quad (16.89)$$

The head loss for horizontal flow at uniform velocity simplifies too

$$h_f = \frac{-\Delta P}{\rho g} \quad (16.90)$$

$$h_f = 12 \frac{\nu \bar{v} L}{g S^2} \quad (16.91)$$

The average energy dissipation rate is

$$\bar{\varepsilon} = \frac{gh_L}{\theta} \quad (16.92)$$

$$\bar{\varepsilon} = 12\nu \left(\frac{\bar{v}}{S} \right)^2 \quad (16.93)$$

The Camp-Stein velocity gradient for laminar flow between parallel plates is

$$G_{CS} = 2\sqrt{3} \frac{\bar{v}}{S} \quad (16.94)$$

16.3.10 Coiled tubes (laminar flow)

Coiled tubes are used as flocculators at laboratory scale. The one shown below is a doubled coil. A single coil would only go around one cylinder

‘ <https://confluence.cornell.edu/display/AGUACLARA/Laminar+Tube+Floc?preview=/10422268/258146480/ReportLaminarTubeFlocSpring2014.pdf>’ __

The ratio of the coiled to straight friction factors is given by Mishra and Gupta

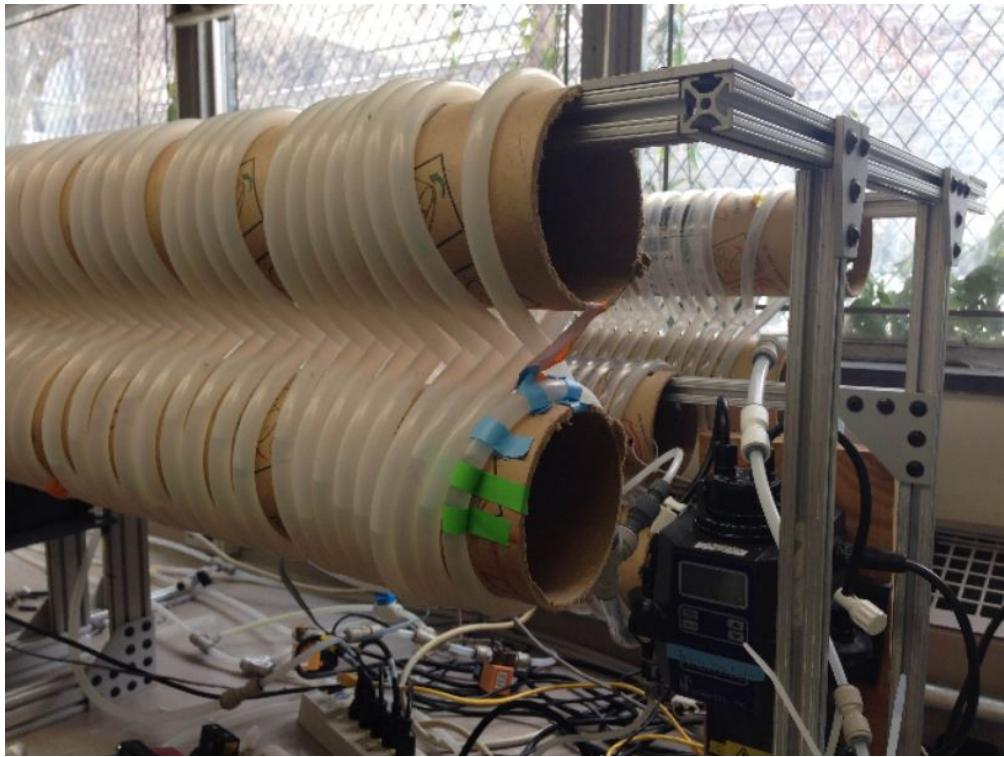


Fig. 16.5: The double coiled laminar flow flocculator creates secondary currents that oscillate in direction. This may be helpful in creating much more mixing than would occur in a straight laminar flow pipe.

The Dean number is defined as:

$$De = Re \left(\frac{D}{D_c} \right)^{\frac{1}{2}} \quad (16.95)$$

where D is the inner diameter of the tube and D_c is the diameter of the coil. Note that the tubing coils are actually helixes and that for the tubing diameters and coil diameters used for flocculators that the helix doesn't significantly change the radius of curvature.

$$\frac{f_{coil}}{f} = 1 + 0.033 (\log_{10} De)^4 \quad (16.96)$$

$$h_{L_{coil}} = h_f \left[1 + 0.033 (\log_{10} De)^4 \right] \quad (16.97)$$

where $h_f = \frac{32\nu L \bar{v}}{g D^2}$. Note that we switch from major losses to total head loss here because the head loss from flowing around the coil is no longer simply due to shear on the wall.

$$h_{L_{coil}} = \frac{32\nu L \bar{v}}{g D^2} \left[1 + 0.033 (\log_{10} De)^4 \right] \quad (16.98)$$

The average energy dissipation rate is

$$\bar{\varepsilon} = 32\nu \left(\frac{\bar{v}}{D} \right)^2 \left[1 + 0.033 (\log_{10} De)^4 \right] \quad (16.99)$$

The average velocity gradient is proportional to the square root of the head loss and thus we obtain

$$G_{CS_{coil}} = G_{CS} \left[1 + 0.033 (\log_{10} De)^4 \right]^{\frac{1}{2}} \quad (16.100)$$

where $G_{CS} = 4\sqrt{2}\frac{\bar{v}}{D}$ for laminar flow in a straight pipe.

$$G_{CS_{coil}} = 4\sqrt{2}\frac{\bar{v}}{D} \left[1 + 0.033 (\log_{10} De)^4 \right]^{\frac{1}{2}} \quad (16.101)$$

16.3.11 Flow in porous media

The Ergun equation applies to both laminar and turbulent flow through packed beds.

$$h_f = \frac{L}{2gD_{sand}} \left(300 \frac{\nu v_a (1 - \phi)^2}{D_{sand} \phi^3} + 3.5 \frac{v_a^2 (1 - \phi)}{\phi^3} \right) \quad (16.102)$$

The Erdun equation can be rewritten to be consistent with the format of the Darcy-Weisbach equation. To do this we add a factor of 2 to the definition of the Erdun friction factor and will divide by the factor of 2 in a subsequent equation. The modified Erdun friction factor is

$$f_\phi = \frac{300}{Re_\phi} + 3.5 \quad (16.103)$$

where the Re_ϕ includes the porosity effect on the velocity and is given by

$$Re_\phi = \frac{v_a D_{sand}}{\nu (1 - \phi)} \quad (16.104)$$

I added a factor of two into the denominator to be consistent with the Darcy-Weisbach equation to obtain the modified Ergun equation.

$$h_f = f_\phi \frac{L}{D_{sand}} \frac{v_a^2}{2g} \frac{(1 - \phi)}{\phi^3} \quad (16.105)$$

The residence time in a porous media bed is given by

$$\theta = \frac{L\phi}{v_a} \quad (16.106)$$

Now we can calculate the average energy dissipation rate in porous media.

$$\bar{\varepsilon} = \frac{gh_f}{\theta} \quad (16.107)$$

Substituting the equations for head loss and residence times we obtain

$$\bar{\varepsilon} = \frac{f_\phi}{2} \frac{v_a^3}{D_{sand}} \frac{(1 - \phi)}{\phi^4} \quad (16.108)$$

The Camp Stein velocity gradient for flow in porous media is

$$G_{CS} = \left(\frac{f_\phi}{2\nu} \frac{v_a^3}{D_{sand}} \frac{(1 - \phi)}{\phi^4} \right)^{\frac{1}{2}} \quad (16.109)$$

16.3.12 Expansions

The average energy dissipation rate for a flow expansion really only has meaning if there is a defined control volume where the mechanical energy is lost. Hydraulic flocculators provide such a case because the same flow expansion is repeated and thus the mechanical energy loss can be assumed to happen in the volume associated with one flow expansion. In this case we have

$$h_e = K \frac{\bar{v}_{out}^2}{2g} \quad (16.110)$$

In this equation K represents the fraction of the kinetic energy that is dissipated.

If we define the length of the control volume (in the direction of flow) as H then the residence time is

$$\theta = \frac{H}{\bar{v}} \quad (16.111)$$

$$\bar{\varepsilon} = \frac{gh_e}{\theta} \quad (16.112)$$

Combining the previous equations we obtain

$$\bar{\varepsilon} = K \frac{\bar{v}_{out}^3}{2H} \quad (16.113)$$

$$G_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad (16.114)$$

$$G_{CS} = \bar{v}_{out} \sqrt{\frac{K \bar{v}_{out}}{2H\nu}} \quad (16.115)$$

16.4 Maximum velocity gradients

16.4.1 Straight pipe (major losses)

The maximum velocity gradient in pipe flow occurs at the wall. This is true for both laminar and turbulent flow. In either case a force balance on a control volume of pipe gives us the wall shear and the wall shear can then be used to estimate the velocity gradient at the wall.

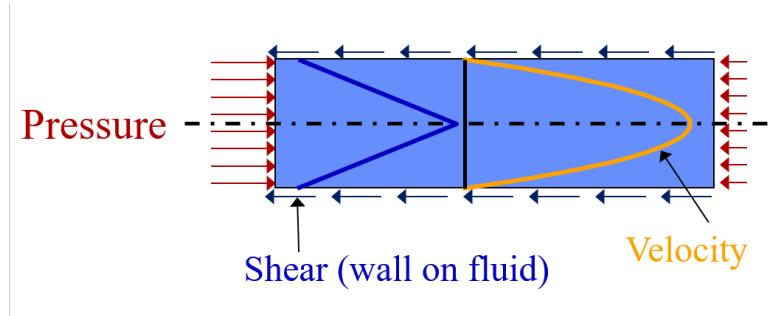


Fig. 16.6: A fluid flowing from left to right due to a pressure gradient results in wall shear.

A force balance for the case of steady flow in a round pipe requires that sum of the forces in the x direction must equal zero. Given a pipe with diameter, D , and length, L , we obtain

$$(P_{in} - P_{out}) \frac{\pi D^2}{4} = \tau_{wall} \pi D L \quad (16.116)$$

$$-\Delta P \frac{D}{4} = \tau_{wall} L \quad (16.117)$$

For this control volume the energy equation simplifies to

$$-\Delta P = \rho g h_f \quad (16.118)$$

The relationship between shear and velocity gradient is

$$\tau_{wall} = \mu \frac{du}{dy}_{wall} = \nu \rho G_{wall} \quad (16.119)$$

Combining the energy equation, the force balance, and the relationship between shear and velocity gradient we obtain

$$\rho g h_f \frac{D}{4} = \nu \rho G_{wall} L \quad (16.120)$$

$$G_{wall} = \frac{g h_f D}{4 \nu L} \quad (16.121)$$

This equation is valid for both laminar flow. For turbulent flow it is necessary to make the approximation that wall shear perpendicular to the direction of flow is insignificant in increasing the magnitude of the wall shear. We can substitute the Darcy Weisbach equation for head loss to obtain

$$G_{wall} = f \frac{\bar{v}^2}{8\nu} \quad (16.122)$$

The energy dissipation rate at the wall is

$$\varepsilon_{wall} = G_{wall}^2 \nu \quad (16.123)$$

$$\varepsilon_{wall} = \frac{1}{\nu} \left(f \frac{\bar{v}^2}{8} \right)^2 \quad (16.124)$$

For laminar flow we can substitute $f = \frac{64}{Re}$ and the definition of the Reynolds number to obtain

$$G_{wall} = \frac{8\bar{v}}{D} \quad (16.125)$$

This equation is useful for finding the velocity gradient at the wall of a tube settler.

The energy dissipation rate at the wall is

$$\varepsilon_{wall} = G_{wall}^2 \nu \quad (16.126)$$

$$\varepsilon_{wall} = \left(\frac{8\bar{v}}{D} \right)^2 \nu \quad (16.127)$$

16.4.2 Coiled tubes (laminar flow)

The shear on the wall of a coiled tube is not uniform. The outside of the curve has a higher velocity gradient than the inside of the curve and there are secondary currents that results in wall shear that is not purely in the locally defined upstream direction. We do not have a precise equation for the wall shear. The best we can do currently is define an average wall shear in the locally defined direction of flow by combining $G_{CS_{wall_{coil}}} = f_{coil} \frac{\bar{v}^2}{8\nu}$ and $f_{coil} = f \left[1 + 0.033 (\log_{10} De)^4 \right]$ to obtain

$$G_{CS_{wall_{coil}}} = f \left[1 + 0.033 (\log_{10} De)^4 \right] \frac{\bar{v}^2}{8\nu} \quad (16.128)$$

16.4.3 Expansions

Flow expansions are used intentionally or unavoidable in multiple locations in hydraulically optimized water treatment plants. Rapid mix and hydraulic flocculation use flow expansions to generate fluid mixing and collisions between particles.

16.4.4 Round Jet

Baldyga, et al. 1995

$$\varepsilon_{Centerline} = \frac{50D_{Jet}^3 \bar{v}_{Jet}^3}{(x - 2D_{Jet})^4} \quad (16.129)$$

$$\varepsilon_{Max} = \frac{\left(\frac{50}{(5)^4}\right) \bar{v}_{Jet}^3}{D_{Jet}} \quad (16.130)$$

$$\varepsilon_{Max} = \Pi_{JetRound} \frac{\bar{v}_{Jet}^3}{D_{Jet}} \quad (16.131)$$

$$\Pi_{JetRound} = 0.08 \quad (16.132)$$

The maximum velocity gradient in a jet is thus

$$G_{Max} = \bar{v}_{Jet} \sqrt{\frac{\Pi_{JetRound} \bar{v}_{Jet}}{\nu D_{Jet}}} \quad (16.133)$$

Below we plot the Baldyga et al. equation for the energy dissipation rate as a function of distance from the discharge location for the case of a round jet that is discharging into a large tank.

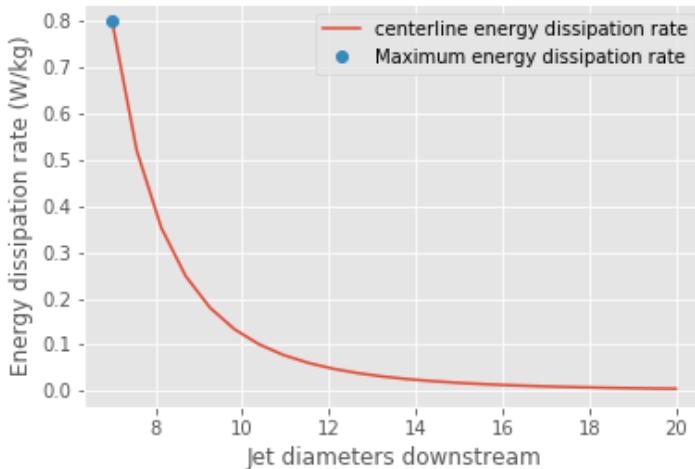


Fig. 16.7: The centerline energy dissipation rate downstream from a round jet. The distance downstream is measured in units of jet diameters. The energy dissipation rate between the jet and 7 jet diameters is developing as the shear between the stationary fluid and the jet propagates toward the center of the jet and turbulence is generated.

16.4.5 Plane Jet

Plane jets occur in hydraulic flocculators and in the sedimentation tank inlet jet system. We havent been able to find a literature estimate of the maximum energy dissipation rate in a plane jet. Original measurements of a plane turbulent jet have been made by Heskestad in 1965 and it may be possible to use that data to get a better estimate of \$:raw-latex:\$ $\Pi_{JetPlane}$ \$ from that source.

$$\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}} = \frac{\varepsilon_{Max}}{\bar{\varepsilon}} \quad (16.134)$$

$$\varepsilon_{Max} = \Pi_{JetPlane} \frac{\bar{v}_{Jet}^3}{S_{Jet}} \quad (16.135)$$

The maximum velocity gradient is thus

$$G_{Max} = \bar{v}_{Jet} \sqrt{\frac{\Pi_{JetPlane} \bar{v}_{Jet}}{\nu S_{Jet}}} \quad (16.136)$$

$$\bar{v} = \frac{Q}{SW} \quad (16.137)$$

$$\bar{v}_{Jet} = \frac{\bar{v}}{\Pi_{VCBaffle}} \quad (16.138)$$

$$S_{Jet} = S\Pi_{VCBaffle} \quad (16.139)$$

The average hydraulic residence time for the fluid between two baffles is

$$\theta_B = \frac{H}{\bar{v}} \quad (16.140)$$

where H is the depth of water. Substituting into the equation for ε_{Max} to get the equation in terms of the average velocity \bar{v} and flow dimension S

$$\varepsilon_{Max} = \frac{\Pi_{JetPlane}}{S\Pi_{VCBaffle}} \left(\frac{\bar{v}}{\Pi_{VCBaffle}} \right)^3 \quad (16.141)$$

From the control volume analysis the average energy dissipation rate is

$$\bar{\varepsilon} = K \frac{\bar{v}^2}{2} \frac{1}{\theta_B} = \frac{K}{2} \frac{\bar{v}^3}{H_e} \quad (16.142)$$

where K is the minor loss coefficient for flow around the end of a baffle with a 180° turn.

Substitute the values for $\bar{\varepsilon}$ and ε_{Max} to obtain the ratio, $\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}}$

$$\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}} = \frac{\Pi_{JetPlane}}{\Pi_{VCBaffle}^4} \frac{2H_e}{KS} \quad (16.143)$$

$\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}}$ has a value of 2 for $H_e/S < 5$ (CFD analysis and Haarhoff, 2001) The transition value for H_e/S is at 5 (from CFD analysis, our weakest assumption).

We also have that $\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}}$ has a value of $\frac{\Pi_{JetPlane}}{\Pi_{VCBaffle}^4} \frac{2H_e}{KS}$ for $H_e/S > 5$. Thus we can solve for $\Pi_{JetPlane}$ at $H_e/S = 5$

$$\Pi_{JetPlane} = \left(\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}} \Pi_{VCBaffle}^4 \frac{K}{2} \frac{S}{H_e} \right) \quad (16.144)$$

$$\Pi_{JetPlane} = 0.0124 \quad (16.145)$$

```
import aquaclara.core.constants as con
VC_BAFFLE_RATIO=con.VC_ORIFICE_RATIO**2
K_MINOR_FLOC_BAFFLE = (1/VC_BAFFLE_RATIO - 1)**2
Ratio_Jet_Plane = 2*con.VC_ORIFICE_RATIO**8 * K_MINOR_FLOC_BAFFLE/2/5
Ratio_Jet_Plane

VC_BAFFLE_RATIO**4*K_MINOR_FLOC_BAFFLE/Ratio_Jet_Plane
```

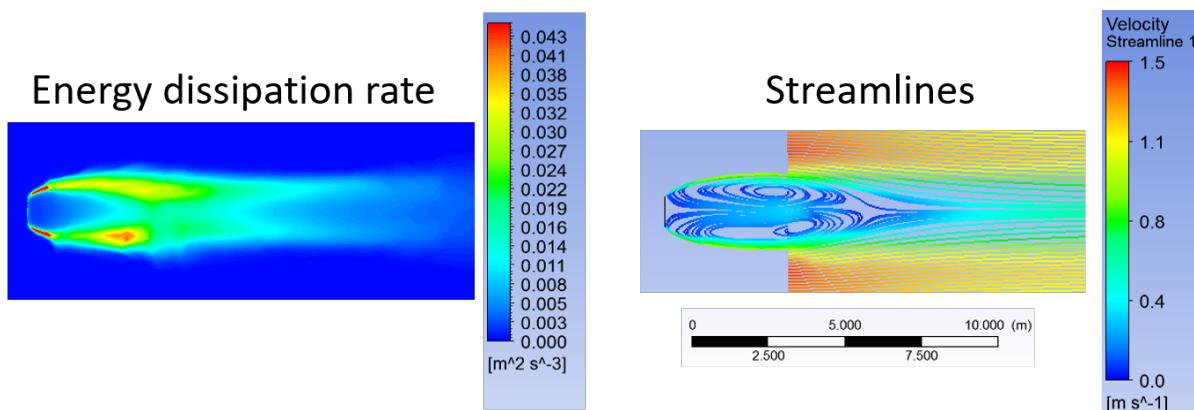


Fig. 16.8: The energy dissipation rate and streamlines for a 1 m wide plate in two dimensional flow with an approach velocity of 1m/s. The maximum energy dissipation rate was approximately $0.04W/kg$.

16.4.6 Behind a flat plate

A flat plate normal to the direction of flow could be used in a hydraulic flocculator. In vertical flow flocculators it would create a space where flocs can settle and thus it is not a recommended design.

The impellers used in mechanical flocculators could be modeled as a rotating flat plate. The energy dissipation rate in the wake behind the flat plate is often quite high in mechanical flocculators and this may be responsible for breaking previously formed flocs.

Ariane Walker-Horn modeled the flat plate using Fluent in 2015.

$$\varepsilon_{Max} = \Pi_{Plate} \frac{\bar{v}^3}{W_{Plate}} \quad (16.146)$$

The maximum velocity gradient is thus

$$G_{Max} = \bar{v} \sqrt{\frac{\Pi_{Plate} \bar{v}}{\nu W_{Plate}}} \quad (16.147)$$

$$\Pi_{Plate} = \frac{(\varepsilon_{Max} W_{Plate})}{\bar{v}^3} \quad (16.148)$$

```
"""CFD analysis setup used by Ariane Walker-Horn in 2015"""
EDR_Max = 0.04*u.W/u.kg
v = 1*u.m/u.s
W = 1*u.m
Ratio_Jet_Plate = (EDR_Max * W/v**3).to_base_units()
print(Ratio_Jet_Plate)
```

The flat plate Π_{Plate} has a value of 0.04.

CHAPTER
SEVENTEEN

RAPID MIX APPENDIX C: EXAMPLES

17.1 Example: pH Adjustment

Find the required dose of several bases to raise the pH at the Manzaragua Water Treatment Plant. The Mazaragua AguaClara plant consists of two 1 L/s plants operating in parallel. The plant is located in the municipality of Guinope, the department of El Paraiso, Honduras.



Fig. 17.1: Manzaragua water treatment plant using two of the AguaClara 1 L/s plants in parallel.

The plant performed very poorly from the first day of operation. The first attempted fix was to double the flocculator residence time by increasing the number of flocculator pipes (3 inch diameter by 1.5 m long) from 12 to 24. This improved performance, but the plant continued to perform poorly. A raw water sample was analyzed on May 30, 2018 and the following results were obtained.

 SANTAC Agua Para... Para Vida		SERVICIO AUTONOMO NACIONAL DE ACUEDUCTOS Y ALCANTARILLADOS LABORATORIO CONTROL DE CALIDAD - DIVISION METROPOLITANA Col. Villa Los Laureles, 1,5 km carretera al Seminario Mayor, Comayagüela, MDC Tel/fax: 2227-4498			
INFORME DE RESULTADOS FISICOQUIMICOS					
No de Informe:	548	RTL-33-01	No de Solicitud:	220	
DATOS DEL CLIENTE					
Nombre	Agua para El Pueblo				
Proyecto					
Dirección	San Rafael, fte a Rest. Pekin				
Teléfono/fax:	2232-6558 9544-1949				
Correo Electrónico					
DATOS DEL MUESTREO					
Fuente	Entrada Planta Manzanares				
Localidad	Guinope El Paraíso				
Tomada por	Antonio Elvir				
Fecha/Hora	30/05/2018 3:00pm				
Tipo de Muestra	AC	AT	INV	AR	PZ
Entregada por	Antonio Elvir 05/06/2018 1:45pm				
Datos de Campo	T° FQ/MB	Cl	pH	ODis	Otros
ANALISIS FISICOS					
Parámetro	Método		Norma	Resultado	*Ue
Turbiedad (NTU)	Parte 2130B		5	71.00	
Color (UC)	Parte 2120B		15	150.00	
Temperatura (°C)	Parte 2550B		18 - 30		
Olor			Inodoro		
ANALISIS QUIMICOS					
Parámetro	Método		*Norma (mg/L)	Resultado	*Ue
pH	Parte 4500H+B		6,5 - 8,5	5,91	
Cloro residual	Colorimétrico con ortotolidina		0,5 - 1,0		
Conductividad	Parte 2510B		400 μ s/cm	69,15	
Alcalinidad total	Parte 2320B		-	24,50	
Bicarbonatos	Parte 2320B		-	24,50	
Carbonatos	Parte 2320B			0,00	
Hidróxidos	Parte 2320B				
Acidez					
Dureza total	Parte 2340C		400	15,68	
Dureza de Calcio	Parte 3500 Ca B		-	9,80	
Dureza de Magnesio	Parte 3500 Mg B		-	5,88	
Calcio	Por calculo a partir de 3500 Ca B		100	3,92	
Magnesio	Por calculo a partir de 3500 Mg B		30	1,43	
Sulfato	Parte 4500-SO ₄ E		250	8,61	
*o-fosfatos	Parte 365,3 (EPA)		-		
Cloruros	Parte 4500Cl C		250	5,14	
Hierro total	Parte 3500-Fe B		0,3	0,27	
Manganoso	MERCK 14770		0,5		
Aluminio	Parte 3500-Al-B		0,3		
Fósforo	Por calculo a partir de 365,3 (EPA)		-		
Flúor	Parte 4500 FD		0,7 - 1,5		
*Nitratos	Parte 352,1 (EPA)		50		
*Nitritos	Parte 354,1 (EPA)		0,1 - 3,0		
152	Nitrógeno amoniacal	Parte 350,2 (EPA)	0,5	1,39	
	Materia Orgánica	Volumétrico con KMnO ₄ DE-07-43	7,50		
	Oxígeno disuelto	Parte 4500 O C	6 - 8		
	DBO ₅	Parte 5210B	50		
	PO ₄	Parte 5220D	200		

Table 17.1: Manzaráqua water quality analysis

Parameter	Units	Standard	Results
Turbidity	NTU	5	71
Color	color units	15	150
pH	pH	6.5 - 8.5	5.91
Conductivity	$\mu\text{s}/\text{cm}$	400	69.15
Alkalinity	mg/L as CaCO_3	•	24.5
Bicarbonates	mg/L as CaCO_3	•	24.5
Carbonates	mg/L as CaCO_3	•	0
Hardness	mg/L as CaCO_3	400	15.68

This water has high color which suggests a high concentration of dissolved organic matter. The pH is a clear problem because the pH is too low for the coagulant nanoparticles to precipitate. As the water sample pH of 5.91 a significant fraction of the coagulant will remain soluble.

Our goal is to determine how much base will need to be added to raise the pH. We do not have data on the *optimal* pH for treating high color water with PACl and so we will use pH 7 as the target.

At circumneutral pH (pH close to 7) the buffering capacity of the water is dominated by carbonate chemistry and specifically by the equilibrium between H_2CO_3^* and HCO_3^- . We will use the acid neutralizing capacity (reported as calcium carbonate alkalinity) and the pH from the sample analysis to estimate the total concentration of carbonates. We will not use the sample analysis carbonate concentrations because they can not be precisely correct.

We will find the amount of base that must be added using (16.23).

Table 17.2: ANC and carbonate values for several bases and acids

Base/Acid	Pi_{ANC}	$Pi_{\text{CO}_3^{-2}}$
Na_2CO_3 or CaCO_3	2	1
NaHCO_3	1	1
NaOH	1	0
HCl or HNO_3	-1	0
H_2SO_4	-2	0

For $\text{Na}_2\text{CO}_3 * \Pi_{ANC} = 2$ because we are adding CO_3^{-2} which is multiplied by two in the ANC equation because CO_3^{-2} can react with two protons. $* \Pi_{\text{CO}_3^{-2}} = 1$ because there is one mole of CO_3 per mole of Na_2CO_3 .

Below is the code used to calculate the required base addition.

```
from aguaclara.core.units import unit_registry as u
import aguaclara.research.environmental_processes_analysis as epa
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

"""define molecular weights"""
m_Ca = 40.078*u.g/u.mol
m_C = 12.011*u.g/u.mol
m_O = 15.999*u.g/u.mol
m_Na = 22.99*u.g/u.mol
m_H = 1.008*u.g/u.mol
```

(continues on next page)

(continued from previous page)

```

m_CaCO3 = m_Ca+m_C+3*m_O
m_Na2CO3 = 2*m_Na+m_C+3*m_O
m_NaHCO3 = m_Na+m_H+m_C+3*m_O
m_NaOH = m_Na+m_O+m_H

"""Raw water characteristics"""
pH_0 = 5.91
ANC_0 = (24.5 * u.mg/u.L/m_CaCO3).to(u.mmol/u.L)
ANC_0

def total_carbonates_closed(pH, ANC):
    """This function calculates total carbonates for a closed system given pH and ANC

    Parameters
    -----
    pH : float
        pH of the sample
    ANC: float
        acid neutralizing capacity of the sample
    Returns
    -----
    The total carbonates of the sample
    Examples
    -----
    >>> total_carbonates_closed(1*u.mmol/u.L, 8)
    1.017 mole/liter
    """
    return (ANC - epa.Kw/epa.invpH(pH) + epa.invpH(pH)) / (epa.alpha1_carbonate(pH) +
    ↪2 * epa.alpha2_carbonate(pH))

CT_0 = total_carbonates_closed(pH_0, ANC_0)

""" calculate the amount of base that must be added to reach a target pH"""

def pH_adjust(pH_0, ANC_0, Pi_base, Pi_CO3, Pi_Al, C_Al, pH_target):
    """This function calculates the required base (or acid) to adjust the pH to a
    target value. The buffering capacity is assumed to be completely due to carbonate
    species. The initial carbonate concentration is calculated based on the initial pH
    and the initial ANC.

    Parameters
    -----
    pH_0: float
        pH of the sample
    ANC_0: float
        acid neutralizing capacity (Alkalinity) of the sample in eq/L.
    Pi_base: float
        equivalents of ANC per mole of base (or acid)
    Pi_CO3: float
        mole of carbonate per mole of base (or acid)
    Pi_Al : float
        equivalents of ANC per mole of aluminum coagulant
    C_Al
        concentration of aluminum coagulant in moles/L
    pH_target: float

```

(continues on next page)

(continued from previous page)

```


pH goal



Returns



-----



The required concentration of base (or acid) in millimoles/L



Examples



-----



```
>>> pH_adjust(5.91,0.2*u.mmol/u.L,1,1,0,0,7)
2.2892822041250924 millimole/liter
>>> pH_adjust(7,0.2*u.mmol/u.L,1,1,0,0,0,0,7)
0.0 millimole/liter
>>> pH_adjust(7,0*u.mmol/u.L,1,0,-3,1*u.mmol/u.L,7)
3.0 millimole/liter
"""
CT_0 = total_carbonates_closed(pH_0,ANC_0)
B_num = CT_0 * (epa.alpha1_carbonate(pH_target) + 2 * epa.alpha2_carbonate(pH_
 ↪target)) + epa.Kw/epa.invpH(pH_target) - epa.invpH(pH_target) - ANC_0 - Pi_Al*C_Al
B_den = Pi_base - Pi_CO3*(epa.alpha1_carbonate(pH_target) + 2 * epa.alpha2_
 ↪carbonate(pH_target))
return (B_num/B_den).to(u.mmol/u.L)

"""target pH"""
pH_target = 7

Pi_base_Na2CO3 = 2
Pi_CO3_Na2CO3 = 1

Pi_base_NaHCO3 = 1
Pi_CO3_NaHCO3 = 1

Pi_base_NaOH = 1
Pi_CO3_NaOH = 0

C_Na2CO3 = pH_adjust(pH_0,ANC_0,Pi_base_Na2CO3,Pi_CO3_Na2CO3,0,0,pH_target)

C_NaHCO3 = pH_adjust(pH_0,ANC_0,Pi_base_NaHCO3,Pi_CO3_NaHCO3,0,0,pH_target)
C_NaOH = pH_adjust(pH_0,ANC_0,Pi_base_NaOH,Pi_CO3_NaOH,0,0,pH_target)

"""Display results in a pandas table"""
base = ["NaOH","NaHCO3","Na2CO3"]
myindex = "[mmoles/L],[mg/L]"
row1 = [C_Na2CO3.magnitude,C_NaHCO3.magnitude,C_NaOH.magnitude]
row2 = [(C_Na2CO3*m_Na2CO3).to(u.mg/u.L).magnitude,(C_NaHCO3*m_NaHCO3).to(u.mg/u.L) .
 ↪magnitude,(C_NaOH*m_NaOH).to(u.mg/u.L).magnitude]
df = pd.DataFrame([row1,row2],index=myindex,columns=base)
print(df.round(2))

"""Graph the base concentration required as a function of the target pH"""
pH_graph = np.linspace(6,7,50)
C_Na2CO3 = pH_adjust(pH_0,ANC_0,Pi_base_Na2CO3,Pi_CO3_Na2CO3,0,0,pH_graph)
C_NaHCO3 = pH_adjust(pH_0,ANC_0,Pi_base_NaHCO3,Pi_CO3_NaHCO3,0,0,pH_graph)
C_NaOH = pH_adjust(pH_0,ANC_0,Pi_base_NaOH,Pi_CO3_NaOH,0,0,pH_graph)

fig, ax = plt.subplots()

ax.plot(pH_graph,C_NaHCO3)
ax.plot(pH_graph,C_Na2CO3)

```


```

(continues on next page)

(continued from previous page)

```

ax.plot (pH_graph,C_NaOH)
imagepath = 'Rapid_Mix/Images/'
ax.set(xlabel='pH target', ylabel='Base concentration (mmole/L)')
ax.legend(["sodium bicarbonate","sodium carbonate","sodium hydroxide"])
fig.savefig(imagepath+'mole_base_for_target_pH')
plt.show()

fig, ax = plt.subplots()
ax.plot (pH_graph, (C_Na2CO3*m_Na2CO3).to(u.mg/u.L))
ax.plot (pH_graph, (C_NaOH*m_NaOH).to(u.mg/u.L))
ax.set(xlabel='pH target', ylabel='Base concentration (mg/L)')
ax.legend(["sodium carbonate","sodium hydroxide"])
fig.savefig(imagepath+'mg_base_for_target_pH')
plt.show()

```

The analysis reveals that the choice of base matters. The most efficient (on a mass or mole basis) base is *NaOH* because it doesn't add any carbonates that don't fully react with the hydrogen ions. The decision about which base to use will be influenced by economics, operator safety, and by whether additional carbonate buffering simplifies plant operation with changing raw water quality.

Table 17.3: Calcium base.

Chemical name	common name	Chemcal formula
calcium carbonate	limestone or chalk	<i>CaCO</i> ₃
calcium hydroxide	slaked lime or hydrated lime	<i>Ca(OH)</i> ₂
calcium oxide	quicklime	<i>CaO</i>

The calcium bases are relatively inexpensive and have the disadvantage of lower solubility than sodium bases. Calcium carbonate has a low solubility, carbon dioxide is present in the atmosphere, and thus calcium carbonate precipitation limits the concentration that can be used for chemical feeds.

The required dose for each of the bases is summarized below.

Table 17.4: Dose of each base required to change the pH of the Manzaraqua water to 7.

units	<i>NaOH</i>	<i>NaHCO</i> ₃	<i>Na</i> ₂ <i>CO</i> ₃
[mmoles/L]	0.45	2.8	0.53
[mg/L]	47.21	235.0	21.19

17.2 LFOM and coagulant injection sizing

A water treatment plant that is treating 120 L/s of water injects the coagulant into the middle of the pipe that delivers the raw water to the plant and then splits the flow into 2 parallel treatment trains for subsequent flocculation. The pipe is PVC 24 inch nominal pipe diameter SDR 26. The water temperature is 0°C. Estimate the minimum distance between the injection point and the flow split.

We will use a *linear flow orifice meter* with 20 cm of head loss. The first step is to determine the diameter of the LFOM.

.todo:: This example needs to be updated once the LFOM OO code is fixed.

```
""" importing """
```

(continues on next page)

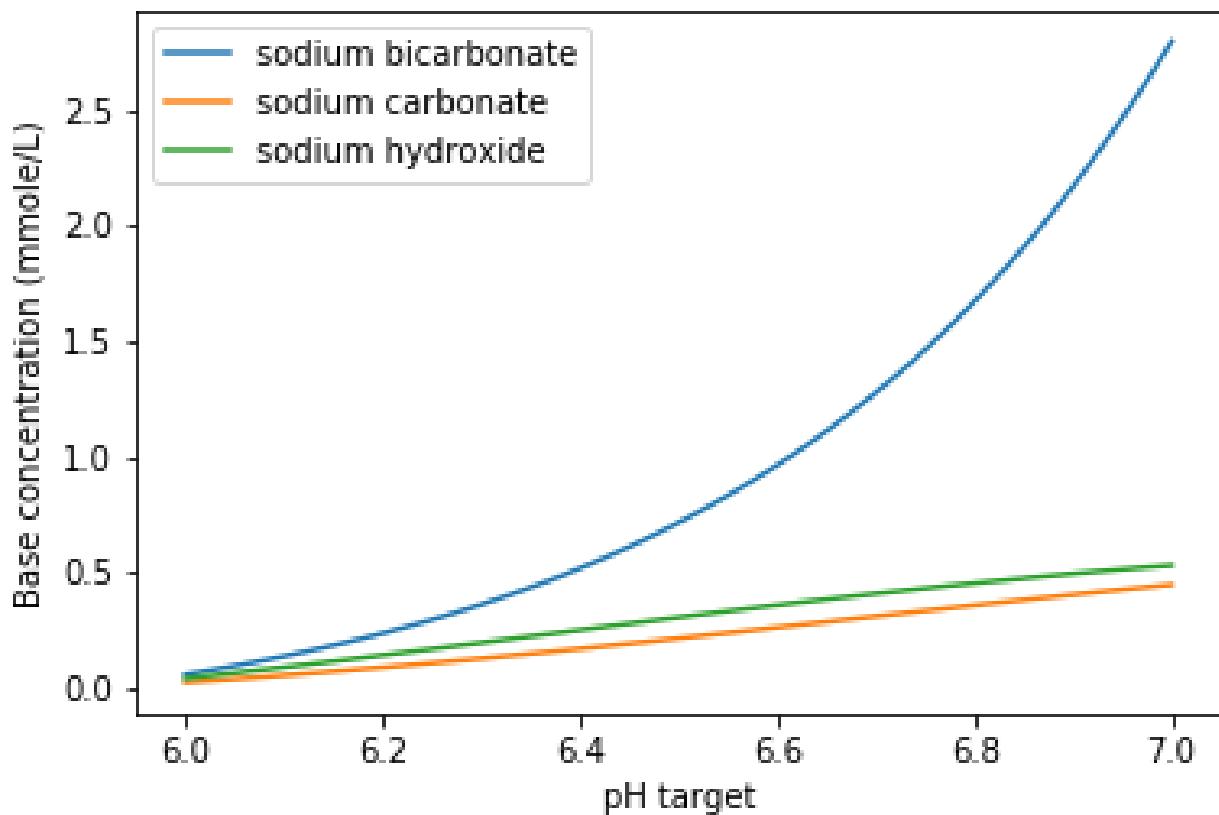


Fig. 17.3: Dose of three bases (in mole/L) required to achieve a target pH for the Manzaráguá water. Carbonates provide more buffering and less change in the pH compared with $NaOH$.

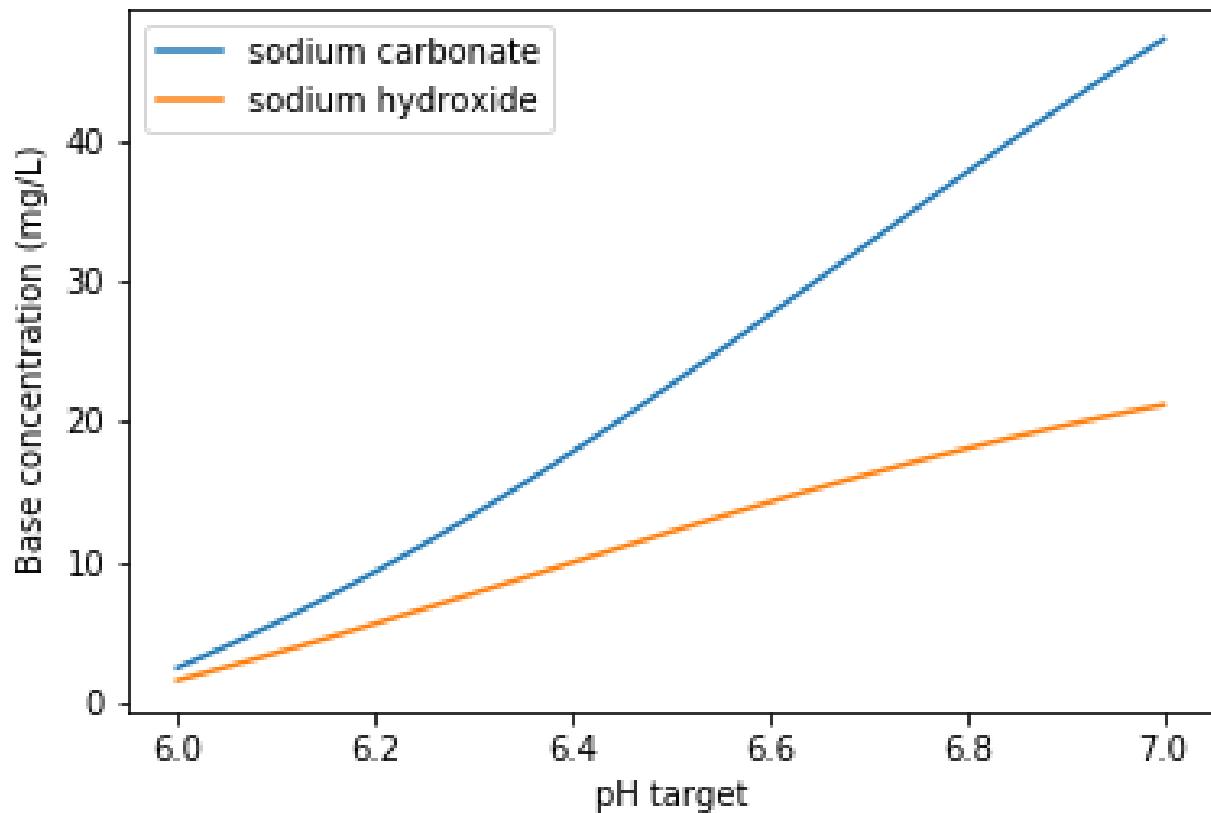


Fig. 17.4: Dose of two bases (in mg/L) required to achieve a target pH for the Manzaráguá water. Carbonates provide more buffering and less change in the pH compared with $NaOH$.

(continued from previous page)

```

from aquaclara.core.units import unit_registry as u
import aquaclara.research.environmental_processes_analysis as epa
import numpy as np
import matplotlib.pyplot as plt
import aquaclara.research.floc_model as fm

#from matplotlib.ticker import FormatStrFormatter
imagepath = 'AquaClara Water Treatment Plant Design/Rapid Mix/Images/'

Q_plant = 120 * u.L/u.s
HL_LFOM = 20 * u.cm
Pi_LFOM_safety = 1.2
SDR_LFOM = 26

from aquaclara.design import lfom as lfom
ND_LFOM = lfom.nom_diam_lfom_pipe(Q_plant, HL_LFOM)
print(ND_LFOM, '(', ND_LFOM.to(u.cm), ')')

L_flow = pipe.ID_SDR(ND_LFOM, SDR_LFOM)
L_flow

```

The LFOM requires a 24 inch diameter pipe.

17.3 Example problem: Energy dissipation rate in a straight pipe

Solution scheme 1) Calculate the friction factor 1) Use mixing_pipe_diameters to estimate the mixing length in pipe diameters 1) Convert to pipe length in meters.

```

import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.core.materials as mat
T_water=0*u.degC
Pipe_roughness = mat.PVC_PIPE_ROUGH
Pipe_roughness
Nu_water = pc.viscosity_kinematic(T_water)
Q_pipe = 120 * u.L/u.s
ND_pipe = 24*u.inch
SDR_pipe = 26
ID_pipe = pipe.ID_SDR(ND_pipe, SDR_pipe)
f_pipe = pc.fric(Q_pipe, ID_pipe, Nu_water, Pipe_roughness)
N_pipe_diameters = (2/f_pipe)**(1/3)
N_pipe_diameters
"""The minimum length for mixing is thus"""
L_mixing = ID_pipe*N_pipe_diameters
print('The minimum distance required for mixing across the diameter of the pipe is ',
      L_mixing.to_base_units())
v_lfom = (Q_plant/pc.area_circle(pipe.ID_SDR(ND_LFOM, SDR_LFOM))).to_base_units()
print(v_lfom)
t_mixing = (L_mixing/v_lfom).to(u.s)
t_mixing

```

The previous analysis provides a minimum distance for sufficient mixing so that equal mass flux of coagulant will end up in both treatment trains. This assumes that the coagulant was injected in the pipe centerline. Injection at the wall of the pipe is a poor practice and would require many more pipe diameters because it takes significant time for the coagulant to be mixed out of the slower fluid at the wall. The time required for mixing at the scale of the flow in

the plant is thus accomplished in a few seconds. This ends up being the fastest part of the transport of the coagulant nanoparticles on their way to attachment to the clay particles. Next we will determine a typical flow rate of coagulant. **Aluminum** concentrations for polyaluminum chloride (PACl) typically range from 1 to 10 mg/L. The maximum PACl stock solution concentration is about 70 g/L as **Al**.

```
C_PACl_stock = 70 * u.g/u.L
C_PACl_dose_max = 10 * u.mg/u.L
Q_PACl_max = (Q_plant*C_PACl_dose_max/C_PACl_stock).to(u.mL/u.s)
print(Q_PACl_max)
```

We can estimate the diameter of the injection port by setting the kinetic energy loss where the coagulant is injected into the main flow to be large enough to exceed the pressure fluctuations downstream of the LFOM. The amount of energy we invest in injecting the coagulant into the raw water is a compromise between having to raise the entire chemical feed system including the stock tanks to increase the potential energy and a goal of not having pressure fluctuations inside the LFOM pipe cause flow oscillations in the chemical dosing tube. Thus our goal is to have the kinetic energy at the injection point be large compared with the expected pressure fluctuations in the LFOM. Given that the head loss through the LFOM is often 20 cm, we expect the pressure fluctuations from turbulence to be a small fraction of that head loss. Thus we set the kinetic energy to be equivalent to 2 cm.

```
HL_Coag_injection = 2 * u.cm
v_Coag_injection = ((2 * u.gravity * HL_Coag_injection)**0.5).to(u.m/u.s)
print(v_Coag_injection)
D_Coag_injection_min = pc.diam_circle(Q_PACl_max/v_Coag_injection)
print(D_Coag_injection_min.to(u.mm))
```

CHAPTER
EIGHTEEN

RAPID MIX THEORY AND FUTURE WORK

Our understanding of the coagulant nanoparticle attachment to suspended particles and dissolved species is currently quite speculative. This is an area that requires substantial research and modeling because has the potential to significantly influence the design of flocculators. We have anecdotal evidence that the process of transporting coagulant nanoparticles to suspended particle surfaces may be a slow, rate-limiting process, especially when coupled with high rate flocculators. Dissolved organic matter may slow the rate of coagulant nanoparticle transport by effectively increasing the size of the coagulant nanoparticles and thus reducing the diffusion rate.

Developing a fundamental understanding of the mixing and transport processes that occur between coagulant addition and flocculation is a very high priority for the AguaClara program.

18.1 Diffusion and Shear Transport Coagulant Nanoparticles to Clay

The time required for shear and diffusion to transport coagulant nanoparticles to clay has previously been assumed to be a rapid process.

Our analysis suggests that this critical step may require significant time especially given our effort to reduce the time allotted for flocculation.

- Turbulent eddies, viscous shear, and diffusion blends the coagulant with the raw water sufficiently (*in a few seconds*) so that the coagulant precipitates and forms nanoparticles.
- Dissolved organic molecules diffuse to the coagulant nanoparticles and adhere to the nanoparticle surface.
- The coagulant nanoparticles are transported to suspended particle surfaces by a combination of diffusion and fluid shear.

The following is a very preliminary estimate of the time required for attachment of the nanoparticles to the clay particles. This analysis includes multiple simplifying assumptions and there is a reasonable possibility that some of those assumptions are wrong. However, the core assumption that coagulant nanoparticles are transported to clay particles by a combination of fluid deformation (shear) and molecular diffusion is reasonable.

The following analysis is similar to the collision analysis that was developed in the AguaClara flocculation model. The core idea is that viscous shear dominated fluid deformation causes relative motion between the coagulant nanoparticles and the clay particles (or other suspended particles) with which they eventually will collide. The collision between these particles that are very different in size is difficult to achieve because there is a viscous layer of fluid around the clay particles that the clay particle drags with it as it rotates in deforming fluid. Diffusion helps transport the coagulant nanoparticles through this viscous layer.

The volume of the suspension that is cleared of nanoparticles is proportional to a collision area defined by a ring around the clay particle with width of the diameter of the nanoparticle diffusion band. This diffusion band is the length scale over which diffusion is able to transport coagulant particles to the clay surface during the time that the nanoparticles are sliding past the clay particle.

If we go backwards in time, the ring of fluid around the clay particle would deform into two separate highly distorted volumes of fluid, one located to the left of the clay particle and one to the right (see figure Fig. 18.1). This volume of fluid is assumed to receive nanoparticles that are transported in by fluid deformation.

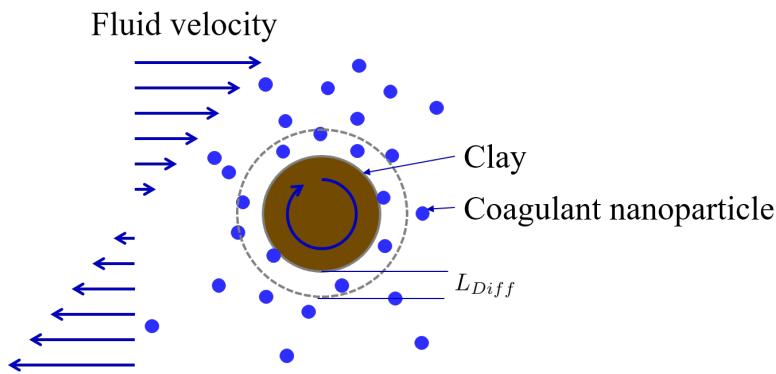


Fig. 18.1: Fluid deformation moves coagulant nanoparticles close to clay particles and diffusion helps transport the nanoparticles the last nanometers toward a successful collision.

18.1.1 Diffusion band thickness

The time required for shear to transport all of the fluid past the clay so that diffusion can transport the coagulant nanoparticles to the clay surface is significant. The diffusion coefficient for coagulant nanoparticles is given by

$$D_{Diffusion} = \frac{k_B T}{3\pi \mu d_{CN}} \quad (18.1)$$

where d_{cn} is the diameter of the coagulant nanoparticles. The length scale over which diffusion is occurring can be estimated from the diffusion coefficient and the time allotted.

$$L_{Diff} \approx \sqrt{D_{Diffusion} t_{Diffusion}} \quad (18.2)$$

The time for coagulant nanoparticles to diffuse through the boundary layer around the clay particle is equal to the distance they travel around the clay particle divided by their velocity. The distance they travel scales with d_{Clay} and their average velocity scales with the thickness of the diffusion layer/2 * the velocity gradient.

$$t_{Diffusion} = \frac{2d_{Clay}}{L_{Diff} G} \quad (18.3)$$

We can eliminate the diffusion time in equation (18.2) using equation (18.3).

$$L_{Diff} \approx \left(\frac{2k_B T d_{Clay}}{3\pi \mu d_{CN} G} \right)^{\frac{1}{3}} \quad (18.4)$$

This diffusion layer thickness is the length scale over which diffusion becomes the dominant transport mechanism for coagulant nanoparticles. Lets estimate the thickness of this diffusion layer.

```
import aquaclaracore.physchem as pc
from aquaclaracore.units import unit_registry as u
import aquaclararesearch.floc_model as fm
import numpy as np
import matplotlib.pyplot as plt
```

(continues on next page)

(continued from previous page)

```

"""I needed to attach units to material properties due to a bug in floc_model. This
→will need to be fixed when floc_model is updated."""
imagepath = 'Rapid_Mix/Images/'
T_graph = np.linspace(0, 30, 4)*u.degC
G = np.arange(50, 5000, 50)*u.Hz

def L_Diff(Temperature, G):
    return (((2*u.boltzmann_constant*Temperature * fm.Clay.Diameter*u.m) / (3 * np.pi * pc.
    →viscosity_dynamic(Temperature) * (fm.PACl.Diameter*u.m)*G))***(1/3)).to_base_units()

fig, ax = plt.subplots()
for i in range(len(T_graph)):
    ax.semilogx(G, L_Diff(T_graph[i], G).to(u.nm))

ax.legend([*T_graph])
"""* is used to unpack T_graph so that units are preserved when adding another legend
→item."""

ax.set(xlabel='Velocity gradient (Hz)', ylabel='Diffusion band thickness ($nm$)')
fig.savefig(imagepath+'Diffusion_band_thickness')
plt.show()

```

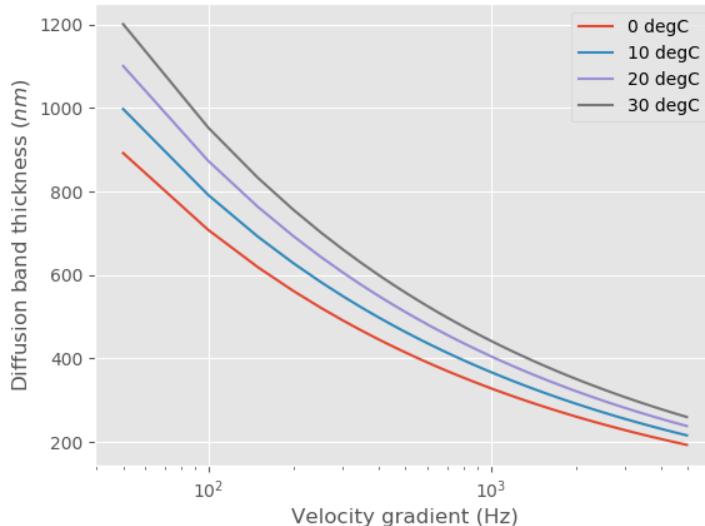


Fig. 18.2: Molecular diffusion band thickness as a function of velocity gradient. This length scale marks the transition between transport by fluid deformation and by diffusion.

Diffusion transports the coagulant nanoparticles a relatively short distance, a fraction of a μm .

We need to calculate the rate at which coagulant nanoparticles attach to the clay particles. The long range transport is assumed to be the rate limiting step. The volume cleared is proportional to the area of this ring with the ring thickness equal to the molecular diffusion band thickness. Here we assume that the $L_{Diff_{CN}} \ll d_{Clay}$

$$V_{Cleared} \propto \pi d_{Clay} L_{Diff_{CN}} \quad (18.5)$$

The volume cleared is proportional to time

$$V_{\text{Cleared}} \propto t \quad (18.6)$$

The volume cleared is proportional to the relative velocity between clay and nanoparticles. This relative velocity is in the viscous layer of fluid in the ring surrounding the clay particle.

$$V_{\text{Cleared}} \propto v_r \quad (18.7)$$

We use dimensional analysis to get a relative velocity for the long range transport controlled by shear. The relative velocity between coagulant nanoparticles and clay particle that they will eventually collide with is assumed to be proportional to the average distance between clay particles. This assumption is both critical for the following derivation and is suspect. It is critical because if we were to assume that the relative velocity caused by shear is proportional to the nanoparticle diameter, the clay diameter, or the diffusion length scale, then the velocity would be extremely small and the time to clear the volume of fluid associated with one clay particle would take a very long time. However, wishing for a speedy process doesn't justify incorrect scaling. The relative velocity is assumed to be the velocity at which coagulant nanoparticles are transported into the two separate fluid volumes that will deform into the ring around the clay particle in the next few seconds.

The assumption that the relative velocity scales with the average distance between clay particles leads to the following steps. The first step is just a proposed functional relationship. We could also have jumped to the assumption that the relative velocity is a function of the length scale and the velocity gradient.

$$v_r = f(\varepsilon, \nu, \Lambda_{\text{Clay}}) \quad (18.8)$$

In a uniform shear environment the velocity gradient is linear. Thus the relative velocity must be proportional to the length scale.

$$v_r = \Lambda_{\text{Clay}} f(\varepsilon, \nu) \quad (18.9)$$

The only way to for ε and ν to produce dimensions of time is to combine to create $1/G$.

$$v_r \approx \Lambda_{\text{Clay}} G \quad (18.10)$$

Combining the three equations for V_{Cleared} and the equation for v_r we obtain the volume cleared as a function of time.

$$V_{\text{Cleared}} \approx \pi d_{\text{Clay}} L_{\text{Diff}_{CN}} \Lambda_{\text{Clay}} G t_c \quad (18.11)$$

18.2 Collision Rates

The time for all of the fluid to have had one opportunity for a collision occurs when $V_{\text{Cleared}} = V_{\text{Occupied}} = \Lambda_{\text{Clay}}^3$

$$t_c = \frac{\Lambda_{\text{Clay}}^3}{\pi d_{\text{Clay}} L_{\text{Diff}_{CN}} \Lambda_{\text{Clay}} G} \quad (18.12)$$

The successful collision rate (the attachment rate) is given by

$$\frac{dN_c}{dt} = \frac{1}{t_c} \quad (18.13)$$

Substitute the equation for t_c .

$$dN_c = \pi d_{\text{Clay}} L_{\text{Diff}_{CN}} \Lambda_{\text{Clay}}^{-2} G dt \quad (18.14)$$

18.2.1 Collision Rate and Particle Removal

A fraction of the remaining coagulant nanoparticles are removed during the time required for one sweep past the clay particle.

$$\frac{dn_{CN}}{-k n_{CN}} = dN_c \quad (18.15)$$

$$\frac{dn_{CN}}{-k n_{CN}} = \pi d_{Clay} L_{Diff_{CN}} \Lambda_{Clay}^{-2} G dt \quad (18.16)$$

18.2.2 Integrate the coagulant transport model

Integrate from the initial coagulant nanoparticle concentration to the concentration at time t.

$$\int_{n_{CN0}}^{n_{CN}} n_{CN}^{-1} dn_{CN} = -\pi d_{Clay} L_{Diff_{CN}} \Lambda_{Clay}^{-2} G k \int_0^t dt \quad (18.17)$$

Use pC notation to be consistent with how we describe removal efficiency of other contaminants.

$$2.3pC_{CN} = \pi d_{Clay} L_{Diff_{CN}} \Lambda_{Clay}^{-2} G kt \quad (18.18)$$

Solve for the time required to reach a target efficiency of application of coagulant nanoparticles to clay.

$$t_{coagulant, application} = \frac{2.3pC_{CN} \Lambda_{Clay}^2}{\pi G k d_{Clay} L_{Diff_{CN}}} \quad (18.19)$$

Coagulant nanoparticle application

If we assume that we are willing to invest a certain amount of energy in the process, then we can estimate the time required to achieve a target coagulant nanoparticle application efficiency. The velocity gradient in the reactor where the coagulant is attaching to the clay particles is related to the head loss or drop in water level, Δh , through the reactor.

$$\Delta h = \frac{G^2 \nu \theta}{g} \quad (18.20)$$

Replace θ with $t_{coagulant, application}$.

$$\Delta h = \frac{G^2 \nu}{g} \frac{2.3pC_{CN} \Lambda_{Clay}^2}{\pi G k d_{Clay} L_{Diff_{CN}}} \quad (18.21)$$

$$L_{Diff} \approx \left(\frac{2k_B T d_{Clay}}{3\pi \mu d_{CN} G} \right)^{\frac{1}{3}} \quad (18.22)$$

$$\Delta h = \frac{G^2 \nu}{g} \frac{2.3pC_{CN} \Lambda_{Clay}^2}{\pi G k d_{Clay}} \left(\frac{3\pi \mu d_{CN} G}{2k_B T d_{Clay}} \right)^{\frac{1}{3}} \quad (18.23)$$

Solve for the velocity gradient.

$$\Delta h = \frac{G^{\frac{4}{3}} \nu}{g} \frac{2.3pC_{CN} \Lambda_{Clay}^2}{\pi k d_{Clay}} \left(\frac{3\pi \mu d_{CN}}{2k_B T d_{Clay}} \right)^{\frac{1}{3}} \quad (18.24)$$

$$G_{coagulant, application} = d_{Clay} \left(\frac{\pi k g \Delta h}{2.3pC_{CN} \Lambda_{Clay}^2 \nu} \right)^{\frac{3}{4}} \left(\frac{2k_B T}{3\pi \mu d_{CN}} \right)^{\frac{1}{4}} \quad (18.25)$$

Using the equation for L_{Diff} above, we can solve for the time required to reach a target efficiency of application of coagulant nanoparticles to clay:

$$t_{coagulant, application} = \frac{2.3pC_{CN} \Lambda_{Clay}^2}{\pi Gk d_{Clay} L_{Diff_{CN}}} \quad (18.26)$$

The time required for the coagulant to be transported to clay surfaces is strongly dependent on the turbidity as indicated by the average spacing of clay particles, Λ_{Clay} . As turbidity increases the spacing between clay particles decreases and the time required for shear to transport coagulant nanoparticles to the clay decreases. Increasing the shear also results in faster transport of the coagulant nanoparticles to clay surfaces. The times required are strongly influenced by the size of the coagulant nanoparticles because larger nanoparticles diffuse more slowly.

Below we estimate the time required to achieve 80% attachment of nanoparticles in a 10 NTU clay suspension in a reactor with G of 100 Hz.

```

import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.research.floc_model as fm
import numpy as np
import matplotlib.pyplot as plt
"""I needed to attach units to material properties due to a bug in floc_model. This
→will need to be fixed when floc_model is updated."""

imagepath = 'Rapid_Mix/Images/'

# conventional mechanical design values below
Mix_HRT = np.array([0.5, 15, 25, 35, 85])*u.s
Mix_G = np.array([4000, 1500, 950, 850, 750])/u.s
Mix_CP = np.multiply(Mix_HRT, np.sqrt(Mix_G))
Mix_Gt = np.multiply(Mix_HRT, Mix_G)
Mix_EDR = (Mix_G**2*pc.viscosity_kinematic(Temperature))

def Nano_coag_attach_time(pC_CN, C_clay, G, Temperature):
    """We assume that 70% of nanoparticles attach in the average time for one collision.
    """
    k_nano = 1-np.exp(-1)
    num=2.3*pC_CN*(fm.sep_dist_clay(C_clay,fm.Clay))**2
    den = np.pi * G* k_nano * fm.Clay.Diameter*u.m * L_Diff(Temperature,G)
    return (num/den).to_base_units()

C_Al = 2 * u.mg/u.L
C_clay = 10 * u.NTU
pC_CN = -np.log10(1-0.8)
"""apply 80% of the coagulant nanoparticles to the clay"""

G = np.arange(50,5000,10)*u.Hz

fig, ax = plt.subplots()

for i in range(len(T_graph)):
    ax.semilogx(G,Nano_coag_attach_time(pC_CN,C_clay,G,T_graph[i]))

ax.semilogx(Mix_G.to(1/u.s),Mix_HRT.to(u.s),'o')
ax.legend([*T_graph, "Conventional rapid mix"])
"""* is used to unpack T_graph so that units are preserved when adding another legend
→item."""

```

(continues on next page)

(continued from previous page)

```
ax.set(xlabel='Velocity gradient (Hz)', ylabel='Nanoparticle attachment time (s)')
fig.savefig(imagepath+'Coag_attach_time')
plt.show()
```

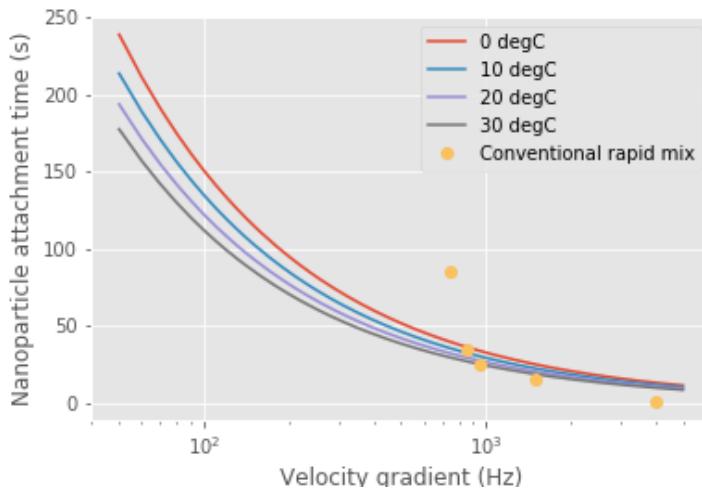


Fig. 18.3: An estimate of the time required for 80% of the coagulant nanoparticles to attach to clay particles given a raw water turbidity of 10 NTU.

18.2.3 Energy Tradeoff for Coagulant Transport

$$\Delta h = \frac{G^2 \nu \theta}{g} \quad (18.27)$$

```
import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.research.floc_model as fm
import numpy as np
import matplotlib.pyplot as plt
Nano_attach_time = Nano_coag_attach_time(pC_CN, C_clay, G, Temperature)

def HL_coag_attach(pC_CN, C_clay, G, Temperature):
    return (G**2 * pc.viscosity_kinematic(Temperature) * Nano_attach_time / u.gravity).to(u.cm)

fig, ax = plt.subplots()

for i in range(len(T_graph)):
    ax.loglog(G, HL_coag_attach(pC_CN, C_clay, G, T_graph[i]))

ax.legend(T_graph)

ax.set(xlabel='Velocity gradient (Hz)', ylabel='Head loss (cm)')
fig.savefig(imagepath+'Coag_attach_head_loss')
plt.show()
```

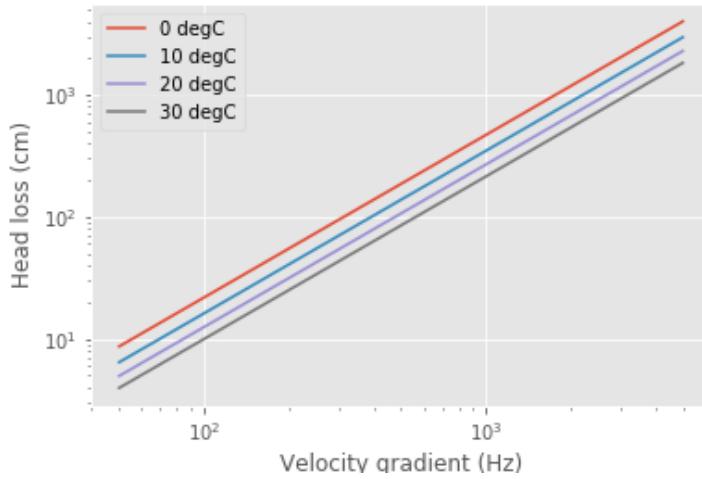


Fig. 18.4: The total energy required to attach coagulant nanoparticles to raw water inorganic particles increases rapidly with the velocity gradient used in the rapid mix process.

There is an economic tradeoff between reactor volume and energy input. The reactor volume results in a higher capital cost and the energy input requires both higher operating costs and higher capital costs. This provides an opportunity to optimize rapid mix design once we have a confirmed model characterizing the process.

The total potential energy used to operate an AquaClara plant is approximately 2 m. This represents the difference in elevation between where the raw water enters the plant and where the filtered water exits the plant. If we assume that the rapid mix energy budget is a fraction of that total and thus for subsequent analysis we will assume somewhat arbitrarily that the energy available to attach the coagulant nanoparticles to the raw water particles is 50 cm.

We solve the coagulant transport model, $t_{coagulant, application} = \frac{2.3pC_{CN} \Lambda_{Clay}^2}{\pi Gk d_{Clay} L_{Diff_{CN}}}$, for G given a head loss.

$$G_{coagulant, application} = d_{Clay} \left(\frac{\pi k g \Delta h}{2.3pC_{CN} \Lambda_{Clay}^2 \nu} \right)^{\frac{3}{4}} \left(\frac{2k_B T}{3\pi \mu d_{CN}} \right)^{\frac{1}{4}} \quad (18.28)$$

```
import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.research.floc_model as fm
import numpy as np
import matplotlib.pyplot as plt
"""find G for target head loss"""
HL_nano_transport = np.linspace(10, 100, 10)*u.cm
def G_max_head_loss(pC_CN, C_clay, HL_nano_transport, Temperature):
    k_nano = 1-np.exp(-1)
    num = u.gravity * HL_nano_transport * np.pi * k_nano
    den= 2.3 * pC_CN * (fm.sep_dist_clay(C_clay,fm.Clay))**2 * pc.viscosity_
    ↵kinematic(Temperature)
    num2 = 2 * u.boltzmann_constant * Temperature
    den2 = 3 * np.pi * pc.viscosity_dynamic(Temperature) * (fm.PACl.Diameter*u.m)
    return fm.Clay.Diameter*u.m*((((num/den)**(3) * (num2/den2)).to_base_units())**(.1/
    ↵4))

"""Note the use of to_base_units BEFORE raising to the fractional power. This_
    ↵prevents a rounding error in the unit exponent."""

```

(continues on next page)

(continued from previous page)

```
G_max = G_max_head_loss(pC_CN,C_clay,50*u.cm,Temperature)
print(G_max)

"""The time required?"""
Nano_attach_time = Nano_coag_attach_time(pC_CN,C_clay,G_max,Temperature)
print(Nano_attach_time)
print(G_max*Nano_attach_time)
```

According to the analysis above, the maximum velocity gradient that can be used to achieve 80% coagulant nanoparticle attachment using 50 cm of head loss is 283 Hz. This requires a residence time of 61 seconds. These model results must be experimentally verified and it is very likely that the model will need to be modified.

The analysis of the time required for shear and diffusion to transport the coagulant nanoparticles the last few millimeters suggests that it is the last step of diffusion to the clay particles that requires the most time. Indeed, the time required for coagulant nanoparticle attachment to raw water particles is comparable to the time that will be required for the next step in the process, flocculation.

18.3 Coagulant Attachment Mechanism

We do not yet understand the origin of the bonds that form between coagulant nanoparticles, between a coagulant nanoparticle and a suspended particle, and between coagulant nanoparticles and dissolved organic molecules. Historically the role of the coagulant was assumed to be to reduce the repulsive force between particles so that the particles could get close enough for Van der Waals forces to hold the particles together. That neglected the fact that Van der Waals forces were already acting between the water molecules and the suspended particle surfaces. In order for the water molecules to be pushed out of the way it is necessary for the coagulant nanoparticles to have stronger bonds with the suspended particles than the bonds between water molecules and the suspended particles.

The water molecules are subject to Brownian motion and thus it is possible that they are frequently vibrated free from the Van der Waals attractive forces. The coagulant nanoparticles are much larger, less affected by Brownian motion, and thus less likely to be vibrated. The fractal nature of the coagulant nanoparticles may also make it possible for multiple well aligned connections between the two surfaces. The fractal tentacles of the coagulant nanoparticle can align as needed to enable many strong bonding connections to the clay surface.

RAPID MIX MECHANICAL SOLUTION

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.core.utility as ut
import numpy as np
import matplotlib.pyplot as plt
```

Numpy, fortunately, understands how to do matrix/array operations index by index. So if you have two length 5 arrays that you want to multiply index by index, you can simply multiply them! This will allow you to focus more on the rest of the code rather than frustrating indexing errors.

```
array3 = array1*array2
```

This is more elegant and easier to troubleshoot than:

```
array3 = np.empty_like(array1) for i in range (0,len(array1)): array3[i] = array1[i]*array2[i]
```

Numpy can handle addition, subtraction, multiplication, and division between arrays.

This design challenge contains two parts: - In the first section, we will explore the energy requirements and operating costs for conventional mechanized rapid mix. This provides an opportunity to learn about the consequences of failing to design for sustainability. Rapid mix is both poorly understood and potentially very important in determining the successful operation of a water treatment plant. - In the second section, we will explore the energy dissipation rate in a free jet.

19.1 Part 1: Temperature effects, energy use, and operating costs

19.1.1 1)

Create a graph of the kinematic viscosity of water as a function of temperature. Use the pc.viscosity_kinematic function. 1. Create an array of temperatures (0 to 35) using np.linspace to create a dimensionless array. Then, attach appropriate temperature dimensions to the array by multiplying by the units. You can do this in one line! 1. Pass the entire array of temperatures to the kinematic viscosity function in aguaclara.core.pc to obtain an array of corresponding viscosities. 1. Plot viscosity (mm²/s) as a function of temperature (Celsius).

```
GraphTempArray = np.linspace(0,35)*u.degC
GraphKinematicViscosity=pc.viscosity_kinematic(GraphTempArray)
fig, ax = plt.subplots()
ax.plot(GraphTempArray, GraphKinematicViscosity.to(u.mm**2/u.s), 'r-')
ax.set(xlabel='Temperature (degrees Celcius)')
ax.set(ylabel='Kinematic viscosity (mm^2/s)')
```

(continues on next page)

(continued from previous page)

```
fig.savefig('Rapid_Mix/Images/Kinematic_vs_T')
plt.show()
```

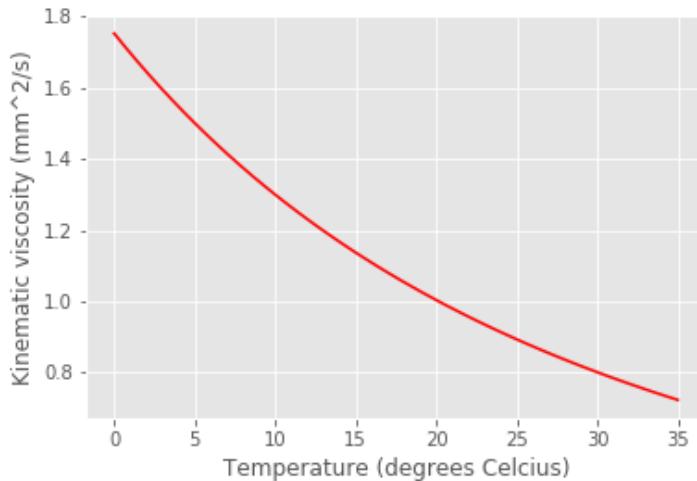


Fig. 19.1: Kinematic viscosity decreases rapidly with temperature. This will influence the amount of fluid deformation obtained given an input of energy. It will also influence the flow rate in the AquaClara chemical dosing system.

We will design a mechanized rapid mix unit and then analyze the energy costs of mixing for a conventional water treatment plant. Below is a table of conventional design parameters.

Residence Time (s)	Velocity Gradient (G) (1/s)	Equivalent Height (m)
0.5	4000	0.8
10 - 20	1500	2.3 - 4.6
20 - 30	950	1.8 - 2.8
30 - 40	850	2.2 - 2.9
40 - 130	750	2.3 - 7.5

19.1.2 2)

Define a function for the power required to run a rapid mix unit. The rapid mix shaft power function should have flow, velocity gradient, residence time (which can be described with t or θ , as in the equation below), and temperature as its only inputs.

$$P = G^2 Q \theta \nu \rho \quad (19.1)$$

Using your function, plot the shaft power (kW) required for a conventional rapid mix unit that has a residence time of $t = 15\text{s}$, a velocity gradient of $G = 1500/\text{s}$, and a flow rate of 50 L/s as a function of temperature (Celsius).

```
FlowPlant = 50*(u.L/u.s)
t = 15 * u.s
G = 1500 * 1/(u.s)

def rapid_mix_shaft_power(Flow, VelocityGradient, time, Temp):
    power = ((VelocityGradient**2) * Flow * time * pc.viscosity_kinematic(Temp) * pc.
    ↪density_water(Temp))
```

(continues on next page)

(continued from previous page)

```

return power

ShaftPower = rapid_mix_shaft_power(FlowPlant, G, t, GraphTempArray)

fig, ax = plt.subplots()
ax.plot(GraphTempArray, ShaftPower.to(u.kW), 'r-')
ax.set(xlabel='Temperature (degrees Celcius)')
ax.set(ylabel='shaft power required (kW)')
fig.savefig('Rapid_Mix/Images/Shafit_Power')
plt.show()

```

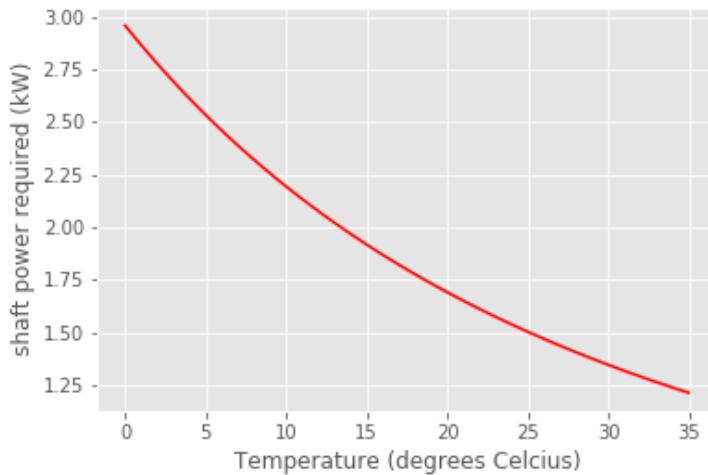


Fig. 19.2: Rapid mix shaft power required as a function of temperature. Significantly more power is required at low temperatures to overcome the effects of viscosity. This does assume that the goal for rapid mix is to achieve a target

19.1.3 3)

How would the shaft power for a conventional design be selected? You must consider the fact that temperature changes throughout the year, and temperature impacts the shaft power requirements as defined in your function above. Presumably, you should use the lowest water temperature that you expect to have to treat to make your design because low temperatures require the highest shaft power input (as shown in your graph above) and we want to design for the most conservative option.

Imagine you are designing a rapid mix unit for a water treatment plant in upstate NY. Water freezes in the winter, so the power requirements to run a rapid mix must be based on TempDesign = 0 °C.

What is the required shaft power **in horsepower** for your water treatment plant in upstate NY? Calculate the shaft power using the design temperature, and take other necessary parameters from Problem 2.

You can [find pints unit registry here..](#) We recommend bookmarking this page!

```

TempDesign = u.Quantity(0, u.degC)

RapidMixShaftPower = rapid_mix_shaft_power(FlowPlant, G, t, TempDesign).to(u.hp)
print('The required shaft power is', RapidMixShaftPower)

```

The required shaft power is 4.0 hp

19.1.4 4)

What is the equivalent height in meters that this much shaft power could elevate the water in a pump, assuming the pump is 100% efficient? Dimensional analysis of units is useful for this question. How do you get from units of power to units of length, and what constants will get you there?

You may assume 100% efficiency in conversion of shaft power to increased potential energy. For comparison, the difference in water level between the entrance tank and flocculator for an AquaClara plant is less than 40 cm, and 20 cm of that elevation drop is used for flow measurement in the LFOM.

```
ElevDrop = (RapidMixShaftPower/ (FlowPlant * pc.density_water(TempDesign) * pc.  
    ↪gravity)).to(u.m)  
print('The equivalent height is', ElevDrop)
```

The equivalent height is 6.034 m

19.1.5 5a)

Our next task is to select a motor that can provide the required shaft power, as calculated in Problem 3.

- First, bring in the `motor_efficiency.xlsx` file provided for this design challenge. You imported a similar file in Problem 13 of DC Python Tutorial 2, but this time you will use `pd.read_excel('filename.xlsx')` since the file is .xlsx instead of .csv.
- Display the extracted excel data, `motor_efficiency.xlsx`.
- Extract the column of available horsepower and the column of premium efficiencies from the excel file and place them within their own 1D numpy arrays.
- Apply appropriate units to the two extracted numpy arrays.

NOTE: The variable to which you assign the extracted excel document is a special type of data structure called a `dataframe`. You can call a column of data from a `dataframe` by indexing by the columns title in the original file, like `dataframe['Horsepower']`. Make sure the column header is input as a string!

```
#We use a pandas dataframe (df) to hold the data from the excel sheet.  
MotorEfficiencydf = pd.read_excel('motor_efficiency.xlsx')  
  
print(MotorEfficiencydf)  
  
MotorHpArray = np.array(MotorEfficiencydf['Horsepower'])*u.horsepower  
MotorEfficiencyArray = np.array(MotorEfficiencydf['Premium Efficiency'])/100  
  
MotorHpArray  
  
Horsepower  Premium Efficiency  
0          0.25            64  
1          0.3333          68  
2          0.5             71  
3          0.75           75.5  
4          1              82.5  
5          1.5            82.5  
6          2              84  
7          3              86  
8          5            89.6
```

(continues on next page)

(continued from previous page)

9	7.5	90
10	10	91.1
11	15	91.7
12	25	93
13	50	94.1
14	100	95
15	250	95.8

19.1.6 5b)

It is common in engineering design to have target design value that must be rounded up to the next available manufactured value. In the Flow Control and Measurement Design Challenge, we used this method to take a minimum pipe size required to deliver a target flow rate and then we rounded up to the nearest commonly available pipe size. We wrote a function, `ceil_nearest` in `aguaclara.core.utility`, that we used to select pipe sizes. Use that function to find a motor that is large enough to drive the rapid mix impeller ([Whats an impeller?](#)).

`ut.ceil_nearest` takes two inputs: the first is the value you are looking to match/exceed. In the pipe sizing example, this would be the smallest pipe that can handle the required flow. The second input is the array through which you are looking to compare the first input. If we continue with the pipe size example, this would be the array of available pipe sizes.

```
MotorDesign = ut.ceil_nearest(RapidMixShaftPower, MotorHpArray)
print('The rapid mix motor has', MotorDesign)
```

The rapid mix motor has 5 hp

19.1.7 6a)

Our next goal is to find the efficiency of this motor.

One way to do this is to find the index (row number) of this motor horsepower in the original dataframe (or in your created array) and then use that index to find its corresponding efficiency. Use the `np.where` function to find the index of the motor horsepower. The `where` function will return an array, so you will need to use indexes on the returned array to extract the index of the motor. If you display the returned results, it will help you figure out how to use indexes to get the desired element.

The input in `np.where` (`input`) is number you're looking to match == array you're looking to match it to. Example:
`np.where(variable == array)`

It is likely you will get the following output: `(array([8], dtype=int64),)`

How to make sense of this? The parentheses around the entire output specify an array (note the comma before the final parentheses, specifying a blank second index). So you can index with square brackets to call the value within the array (Perhaps try index 0?). You should then get another array. How do you extract the value within this new array?

```
MotorIndex=(np.where(MotorEfficiencydf['Horsepower'] == MotorDesign.magnitude))[0][0]
MotorIndex
```

8

19.1.8 6b)

You will now use the index of the motor horsepower that you just found to extract the efficiency of the motor. This can be done by calling the index on the array of efficiencies you created or by calling on data from the original dataframe.

Dataframes have a method called `get_value` that can return an element in a dataframe given a row index and a column heading. This is done for you as an example.

For this problem, **extract the efficiency of the motor by calling the index on the array you created in Problem 5a. Return the motor efficiency as a fraction rather than as a percent.**

Note: it would have been much easier to simply define a variable and type in the efficiency. However, if we did that, it would have broken dependency; the notebook wouldn't update correctly when you change the flow rate. Our goal is to create designs that scale correctly when the flow rate is changed.

```
#Example for how to use .get_value to return an element by calling on row index and
#column heading
MotorEfficiency=(MotorEfficiencydf.get_value(MotorIndex, 'Premium Efficiency',
#takeable=False))/100

#-----Your code below-----
#-----


print(MotorEfficiencyArray[MotorIndex])

print('The motor efficiency is ',MotorEfficiency,'.')
```

0.896 The motor efficiency is 0.896.

You might think that the rapid mix unit will take less electrical power when the water is warmer. But that isn't the case because the Reynolds number for the rapid mix propeller is quite high and thus the drag coefficient is independent of Re . This means that the torque required to spin the propeller doesn't change as the viscosity of the water changes. It would be possible to run the propeller slower when the water is warmer because the required energy dissipation rate is lower, but that would require a variable speed drive. You could add a variable speed motor controller to take advantage of this. However, the bigger problem is that we don't yet have a good model explaining what rapid mix does.

19.1.9 7)

Now, we want to find a 3 phase, 1800 rpm (approximately!) electric motor that is totally enclosed and fan cooled (TEFC), and that has enough power to operate your rapid mixer at [McMaster Carr](#). You can find the link to the CAD drawing if you pretend you are going to select the motor for purchase. This part of the design challenge breaks dependency.

Once you find a sufficient motor, **display the CAD drawing of the motor and create a hyperlink to the motor specifications.**

Notes: we don't currently know how to select and link to the correct motor using python, which is why you need to go to the McMaster Carr website to find the correct motor. If you know of an easy way to do this, please let us know!

The motor specifications are given below. [Rapid Mix motor](#)

The [motor specifications](#) indicate that the efficiency is 89.5% which is very close to the premium efficiency standard.

19.1.10 8)

How much does the motor cost? Create a variable showing the cost of the motor in USD. We have added USD to pint, so you will not find it in the original registry linked in Problem 3. The abbreviation for US Dollars is `u.USD`.

```
COST_MOTOR = 714.64 * u.USD
print('The cost of the motor is', COST_MOTOR)
```

The cost of the motor is 714.6 dollar

19.1.11 9)

Now, we want to determine the cost of electricity to operate the rapid mix unit you sized for your water treatment plant in NY.

19.1.12 9a)

Find the commercial rate for electricity in NY state at the [U.S. Energy Administration Website](#). Download the data file for NY and then import the data file using pandas (`read_csv`). There are two versions of the csv data file on this page. The one you should use in this problem is titled `Average_retail_price_of_electricity_monthly.csv`.

For this step, simply **display the resulting data table showing the most recent 12 months of electricity costs in NY.**

Note that this data file has meta information about the data in the first rows. You can **delete those rows by setting header = 4** in the function call to the `pd.read_csv`. You can read exactly one year of data by setting `nrows = 12`. This will make it easy to calculate the average cost for the past year.

```
ElectricityCostdata = pd.read_csv('Average_retail_price_of_electricity_monthly.csv',
                                header=4,nrows=12)
ElectricityCostdata
```

19.1.13 9b)

Calculate the average commercial electricity cost for the most recent 12 months. Include the appropriate units. Dont forget to correct for the fact that the prices are given in cents and not dollars.

```
ElectricityRate = np.average(np.array(ElectricityCostdata['New York : commercial_cents per kilowatthour']))/100*u.USD/(u.kWh)
print('The New York State commercial electricity cost for the past year was ',
      ElectricityRate)
```

The New York State commercial electricity cost for the past year was 0.14622 USD/kWh

19.1.14 9c)

I think that we should buy electricity in giga Joules. KiloWatt-hr is what you get when someone who loves English units decides they dont like seconds as the base unit of time that is the standard in the metric system.

How much would a giga Joule cost for the most recent 12 months? All you have to do is change the displayed units!

Look through the pint unit registry. To add orders of magnitude (like kilo, micro, giga, yotta), simply place the appropriate prefix before the usual unit abbreviation. A yotta-meter is `u.Ym`, for example, as Y represents yotta- in pint.

```
print ('The price of electricity is', ElectricityRate.to(u.USD/u.GJ))
```

The price of electricity is 40.62 USD/GJ

19.1.15 9d)

Lets look back to our design of a rapid mix unit for a water treatment plant in New York. Estimate the electricity demand * **in kilowatts** * for the rapid mix by taking the shaft power required and adjust for the efficiency of the motor that you have selected. Note that when a motor is running it does not necessarily operate at full load and hence at full power. Electric motors maintain their efficiency over a wide range of loads. To calculate the electricity consumption, use the actual shaft power required by the rapid mix and the motor efficiency. Calculate the electrical power required by the motor when the water is at its coldest.

```
PowerElectricity = (rapid_mix_shaft_power(FlowPlant,G,t,TempDesign) / MotorEfficiency) .  
                   ↪to(u.kW)  
print ('The power required by the motor is', PowerElectricity)
```

The power required by the motor is 3.30 kW

```
rapid_mix_shaft_power(FlowPlant,G,t,TempDesign).to(u.hp)
```

3.967125458280934 horsepower

19.1.16 9e)

Calculate the annual cost of electricity in dollars required to operate the rapid mix unit.

```
RMElectricityCost = (PowerElectricity*ElectricityRate).to(u.USD/u.year)  
print ('The cost of electricity to operate the rapid mix unit is', RMElectricityCost)  
  
print(PowerElectricity)  
print(ElectricityRate)
```

The cost of electricity to operate the rapid mix unit is 4232 USD/year 3.302 kilowatt 0.1462 dollar / kilowatt_hour

19.1.17 9f)

What is the total cumulative energy costs for the rapid mix unit over a period of 25 years? Note that we are not including the projected increase in electricity costs over the next 25 years. That would be a nice addition to this analysis that would make the need for energy efficiency all the more apparent, but we are ignoring that complexity for this problem.

```
YR25Electricity = RMElectricityCost*(25*u.year)  
print ('The cumulative energy costs for a period of 25 years is', YR25Electricity)
```

The cumulative energy costs for a period of 25 years is 105794.0 USD

19.1.18 10)

Write a paragraph describing what you learned from this design challenge. Include reflections on the temptation to use a standard design, the low capital cost of energy wasting designs, and the long term implications of engineering that isn't guided by a goal of sustainability.

CHAPTER
TWENTY

RAPID MIX DESIGN SOLUTION

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.core.constants as con
import numpy as np
import matplotlib.pyplot as plt
```

20.1 1)

Create a function to return maximum energy dissipation rate along the centerline of a free jet given inputs of distance downstream of the jet (along the centerline), jet velocity, and initial jet diameter. You will find the Rapid Mix powerpoint useful in finding a relationship between these variables.

Use your function to create a graph of the centerline (maximum) energy dissipation rate as a function of distance from jet origin. * Use an initial diameter of DiamJet = 10 cm and a velocity of VelJet = 1 m/s. * Use a range of 7 to 20 jet diameters when creating the graph. * Use units of meters for the x-axis and W/kg for the y-axis. Make sure to label your axes correctly.

```
DiamJet = 10*u.cm
VelJet = 1*(u.m/u.s)

def energy_dissipation_rate(x, Diam, Velocity):
    dissipation = (50 * Diam**3 * Velocity**3 / ((x - (2 * Diam))**4))
    return dissipation.to(u.m**2/(u.s**3))

DistanceDownstream = (np.array(np.linspace(7,20,100))*DiamJet).to(u.m)

print(energy_dissipation_rate(DistanceDownstream, DiamJet, VelJet)[1])
fig, ax = plt.subplots()
ax.plot(DistanceDownstream, energy_dissipation_rate(DistanceDownstream, DiamJet, VelJet), 'r-')
ax.set(xlabel='Distance from jet origin (m)')
ax.set(ylabel='Energy dissipation rate (W/kg)')
fig.savefig('Rapid_Mix/Images/Energy_dissipation_rate_vs_Distance_from_jet_origin')
plt.show()
```

0.7212 meter ** 2 / second ** 3

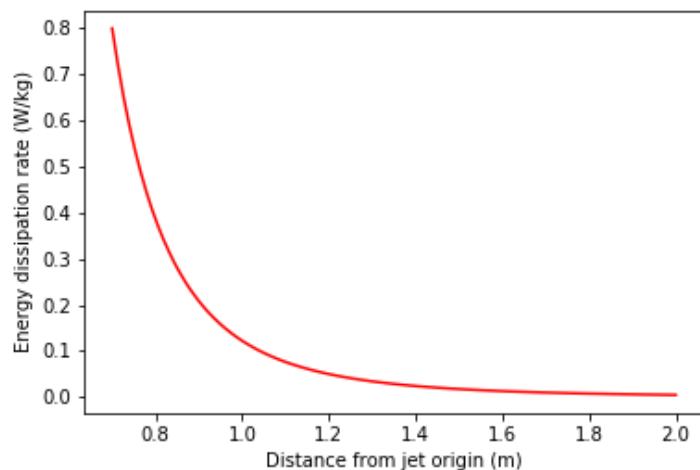


Fig. 20.1: Energy dissipation rate vs Distance from jet origin

CHAPTER
TWENTYONE

FLOCCULATION INTRODUCTION

Flocculation transforms inorganic (clays such as kaolinite, smectite, etc. and metallic oxy-hydroxides such as goethite and gibbsite) and organic (viruses, bacteria and protozoa) primary particles into flocs (particle aggregates). Flocculation doesn't remove any particles from suspension. Instead it causes particle aggregation and then floc blankets, lamellar sedimentation, and sand filtration will be used to separate those flocs from the water. Sedimentation can remove flocs more easily than it can remove primary particles because flocs have a higher terminal sedimentation velocity. Floc blankets and sand filtration rely primarily on capture based on interception and interception is much more efficient when the particles are larger. Thus the purpose of flocculation is to join **all** of the primary particles together into flocs (See movie in Fig. 21.1).

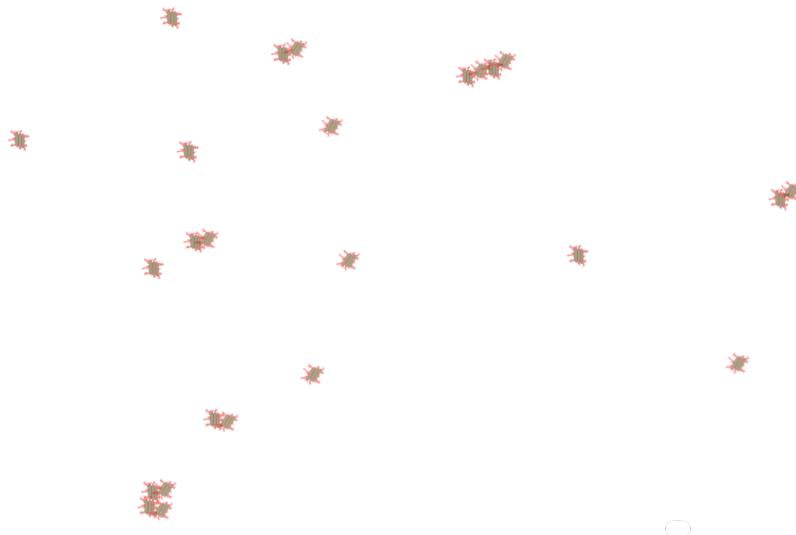


Fig. 21.1: Clay particles with attached coagulant nanoparticles collide due to fluid deformation. They grow in size quickly. The challenge is to catch the last few primary particles that failed to participate in the aggregation process.

Given that hydraulic flocculators approach plug flow conditions it is reasonable to assume that at any given location in the flocculator there is a predominance of one size of flocs. Thus collisions between similar sized flocs are most likely because that is what is present. There is also a hydrodynamic reason why similar sized flocs are favored which we will discuss later. For simplicity of modeling let's assume that flocs repeatedly double in size as suggested by the movie in Fig. 21.1. In that case, the number of primary particles in a floc is given by

$$n_{primary} = 2^{n_{collisions}} \quad (21.1)$$

If we assume (and we will show this assumption to be wrong in the next step) that the floc volume is directly proportional to the total volume of the primary particles in the floc, then we can rearrange (21.1) to solve for the number of

sequential collisions required to increase the number of primary particles by a factor of 1000,000,000.

$$n_{collisions} = \frac{\log(n_{primary})}{\log(2)} \quad (21.2)$$

```
import numpy as np

n_primary = 1000000000
n_collisions = np.log10(n_primary)/np.log10(2)
print(n_collisions)
```

30 sequential collisions would be required to produce a floc that contains 1 billion primary particles.

21.1 Flocs are Fractals

As flocs combine they don't coalesce like mist turning into rain drops. Instead they form loose aggregates that contain a higher and higher fraction of water in the voids between the solid primary particles.

Although the obvious flocculation advantage is that it produces larger aggregates that are easier to remove, it is also **possible** (this is a hypothesis that needs testing) that a difference in a physical property between primary particles and flocs plays a role in enhanced removal of flocs in floc blankets and filters. For example, the many relatively weak connection points between the primary particles in the flocs enables the flocs to deform. It is possible that deformation plays an important role right at the moment of collision. Presumably the bond strength required to lock the colliding particles together is less if the particles can deform as they are colliding.

The size change produced by flocculation is dramatic. Clay particles and pathogens have sizes that are order μm and they combine to form flocs that are order mm . A thousand fold increase in diameter suggests a billion fold increase in volume.

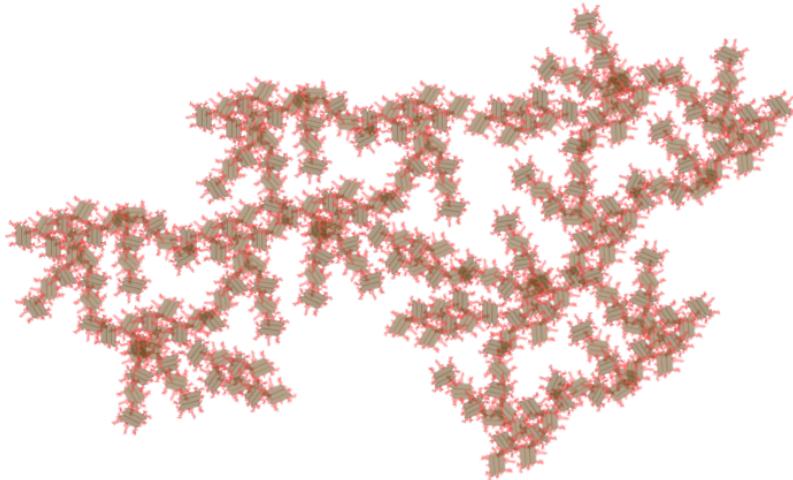


Fig. 21.2: The amount of water contained within a volume defined by the floc increases as the flocs grows.

21.2 Flocculation is (or used to be) a Slow Process

One of the mysteries of flocculation has been why it is such a slow process, requiring 30 minutes according to conventional design, and yet it appears to be a very rapid process. Plant operators observe that with high raw water

turbidities that they can see flocculation progressing after about 0.5 minutes of flocculation. We can estimate the collision potential, $G\theta$ that corresponds to making visible flocs.

$$\bar{G} = \sqrt{\frac{gh_e}{\theta\nu}} \quad (21.3)$$

```
import aquaclaracore.physchem as pc
from aquaclaracore.units import unit_registry as u
import numpy as np

HL_floc = 43*u.cm
HRT = 8 * u.min
Temperature = 20 * u.degC
G_floc = ((pc.gravity*HL_floc/(HRT*pc.viscosity_kinematic(Temperature)))**0.5).to_
    .to_base_units()
print(G_floc)
Gt_floc = G_floc*HRT
HRT_floc_visible = 0.5*u.min
Gt_floc_visible = (G_floc*HRT_floc_visible).to_base_units()
print(Gt_floc_visible)
```

Here initial flocculation is visible at a $G\theta$ of less than 3000. Given that flocculation is visible at this low collision potential, it is unclear why recommended $G\theta$ are as high as 100,000. This is one of the great mysteries that motivated the search for a flocculation model that was based on hypotheses that were consistent with laboratory and field observations.

21.3 Surface Charge Hypothesis

The mechanism of particle-particle aggregation was thought to be controlled by an average surface charge. Apparently no one was able to develop a model of how that mechanism would influence particle attachment efficiency and the result was that no predictive models for flocculation were developed. There were several observations that were at odds with conventional explanations of flocculation.

1. Efficient flocculation at coagulant dosages that led to positive surface charge. This led to a second flocculation mechanism that was called sweep floc and that was used to describe any observations that didn't fit the charge neutralization flocculation hypotheses.
2. Flocculation time for highly turbid suspensions was expected to proceed very rapidly and produce very low turbidity settled water. This expectation was not observed and led to the hypothesis that flocs were continually breaking up and producing primary particles or at least very small flocs.
3. The floc break up hypotheses led to the expectation that high turbidity suspensions would have significantly higher settled water turbidity than low turbidity suspensions. This expectation was also not observed.

Evidence that the charge neutralization hypothesis doesn't explain flocculation of surface waters has been accumulating for decades. *Sweep* flocculation has been proposed as an alternative mechanism that described common observations that didn't fit the charge neutralization hypothesis. However, similar to the charge neutralization hypothesis, the *sweep* hypothesis didn't result in the development of predictive equations to describe the process.

For example, in 1992 Ching, Tanaka, and Elimelech published their research on [Dynamics of coagulation of kaolin particles with ferric chloride](#). They found that the electrophoretic mobility which is a measure of the clay particle surface charge was never neutralized at pH 7.8 and was neutralized at $10\mu M$ at pH 6.0. The results were interpreted by the authors to mean that some combination of sweep floc and charge patchiness was responsible for the observed results.

See Fig. 21.3 showing that at pH 7.8 the ferric chloride was still negatively charged and yet succeeded in flocculating the water to almost the same extent as the ferric chloride at pH 6.0 that was positively charged (see Fig. 21.4).

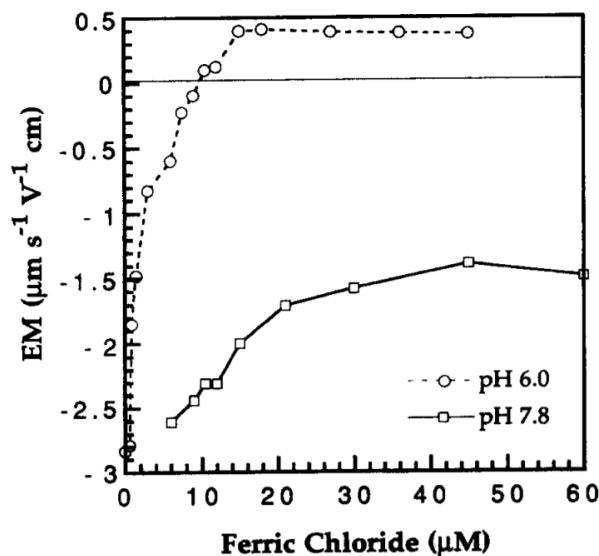


Fig. 21.3: Electrophoretic Mobility for final pH (after coagulant addition) of 6.0 and 7.8 as a function of $FeCl_3$ dose

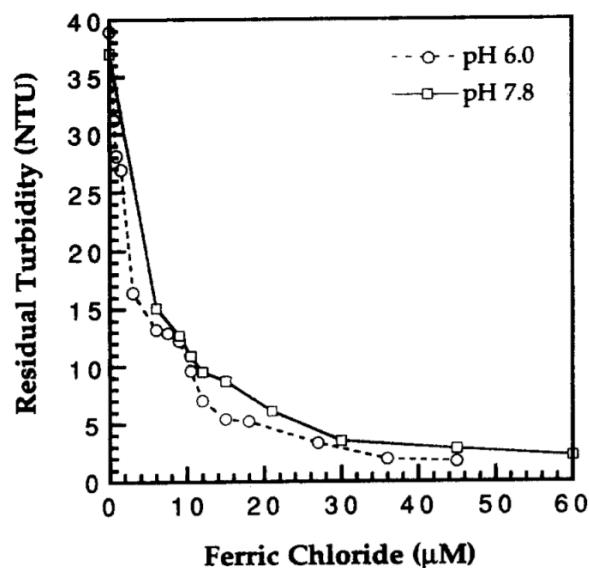


Fig. 21.4: The settled water turbidity was almost independent of pH even though the electrophoretic mobility was quite different for the two pH values tested.

At pH 6.0 the ferric hydroxide precipitates are positively charged and at pH 7.8 they are close to neutral. Thus it is apparent that neutralization of the clay surface charge can not explain these results.

Electrostatic charge neutralization hypothesis

The coagulant precipitate self aggregates – this is inconsistent with the positive charge that the electrostatic hypothesis asserts will prevent aggregation * Electrostatic repulsion extends only a few nm from the surface of a particle – and the coagulant adhesive nanoparticles are many times larger than the reach of the repulsive electrostatic force. The hypothesis that London van der Waals forces result in attachment neglects to account for the presence of water in the system. Water molecules will also be attracted to surfaces by London van der Waals forces and thus there will be competition between the coagulant and water. Thus eliminating repulsion is NOT sufficient to produce a bond between the particles. (see [hydration repulsion, page 21](#)) ‘ The theory of DLP was a great step forward in that it appeared to circumvent the whole intractable problem of many body forces through its use of measured bulk dielectric response functions. However, it must be stressed again that it is a perturbation theory. That is, it depends on the assumption that an intervening liquid between interacting surfaces has bulk liquid properties up to a molecular distance from the surfaces. This is thermodynamically inconsistent, being equivalent to the statement that surface energies (or alternatively, the positions of the Gibbs dividing surfaces) are changed infinitesimally with distance of separation. This limits the theory to large distances (Young–Laplace vs. Poisson again) where large is undefined. <[https://doi.org/10.1016/S0001-8686\(99\)00008-1](https://doi.org/10.1016/S0001-8686(99)00008-1)>’ __

21.4 AquaClara Flocculation History

2005 - We used conventional guidelines based on velocity gradient to design the first low flow vertical flocculator

2010 - We designed using energy dissipation rate and accounted for the nonuniformity of the energy dissipation rate ($\theta = 15$ minutes)

2015 – We added obstacles to decrease the distance between expansions to make all of our flocculators have maximum collision efficiency ($\theta = 8$ minutes)

2016 – Learned that particle/floc collisions are dominated by viscous shear (not by turbulent eddies). Began designing flocculators based on a target head loss of 40 cm. Used a $G\theta$ of 37,000.

2017 - Designed a pipe flocculator for the 1 L/s plant with a $G\theta$ of 20,000 and a residence time of about 100 s.

#references [Coagulation and Flocculation in Water and Wastewater Treatment](#), iwapublishing

CHAPTER
TWENTYTWO

FLOCCULATION MODEL

Particle aggregation is the fundamental mechanism that facilitates ultra low energy and low cost removal of particles and pathogens from water. Aggregation requires successful collisions. Success is defined by particles attaching when they collide.

22.1 Model assumptions

22.1.1 Key understanding: coagulant nanoparticles are sticky

Prior to the AguaClara flocculation model it was widely assumed that attachment was made possible by reducing the net surface charge of the particles. The AguaClara flocculation model is based on the understanding that coagulant nanoparticles are sticky and are much larger than the length scale of the repulsive forces due to surface charges. Thus surface charge is largely irrelevant and this explains why particle aggregation begins even with very low dosages of coagulant.

22.1.2 Key understanding: Particles follow the fluid

The collisions are caused by particles having relative motion due to fluid deformation. Particle trajectories can be different from the fluid trajectory if the density of the fluid and the particle are significantly different and if the viscous effects are small compared with inertial effects (the Stokes number). The motion of primary particles and small flocs in surface water treatment have low Stokes numbers and follow the fluid trajectory.

22.1.3 Key understanding: Long range transport is the slow part of the collision process

We need to calculate the rate of primary particle collisions. In turbulent flow flocculators the fluid deformation is caused by turbulent eddies that lose their energy to viscosity. The relative motion of particles would appear somewhat random as the small eddies have ever changing orientation and intensity. The result is that primary particles take a long meandering path before they finally approach each other and connect in a final collision. The path of relative motion prior to the collision can be thought of as having two distinct components.

- The first component is long range transport when the particles are far apart with a separation distance that is proportional to the average distance between particles.
- The second component is the short range transport at length scales less than the average particle separation distance to the final collision

The AguaClara flocculation model assumes a relatively high velocity and long distance random walk clearing a volume of fluid equal to the volume occupied by a single particle. This is followed by a slow, short, straight walk toward a

collision. The insight that the long range transport is the rate limiting step will be used to estimate the time required for particle collisions.

22.1.4 Key understanding: Primary particles can't attach to large flocs during Flocculation

In our early modeling work we assumed that collisions between primary particles and large flocs were favorable. This assumption led to the prediction that the highest quality water should be obtained when the raw water has the highest turbidity. That prediction is inconsistent with observations and led to the insight that during flocculation, primary particles are only able to collide successfully with other primary particles (or potentially with other very small flocs).

The only transport mechanism that could cause a clay particle to collide with a large floc is the fluid deformation caused by the linear velocity gradient. In our flocculators that linear velocity gradient is caused by turbulent eddies at much larger scales of the flow. We hypothesize that primary particles can not attach to large flocs because primary particles can not collide with large flocs! To understand why this collision is impossible, we need a simple insight.

The insight is that the large flocs drag fluid around as they rotate (due to the linear velocity gradient). The viscous layer around the large flocs creates a flow field in which there is no location far from the flocs that will eventually approach the surface of the flocs or even approach within the clay particle radius. If this is correct, then clay particles never collide with large flocs in a linear velocity gradient flow field.

22.1.5 Key understanding: Relative velocities between particles are dominated by viscous shear

Relative velocities between particles are dominated by viscous shear because the separation distances are smaller than the inner viscous length scale. The average particle separation distance is given by

$$\bar{\Lambda} = \frac{1}{n_P^{\frac{1}{3}}} = V_{\text{Surround}}^{\frac{1}{3}} \quad (22.1)$$

Where:

$\bar{\Lambda}$ is the average separation distance

n_P is the number of particles per volume of suspension

V_{Surround} is the suspension volume occupied by one particle

The number concentration of particles is given by

$$n_P = \frac{C_P}{V_P \rho_P} = \frac{6}{\pi \bar{d}_P^3} \frac{C_P}{\rho_P} \quad (22.2)$$

Where:

C_P is the particle concentration

V_P is the volume of a single particle

ρ_P is the particle density

\bar{d}_P is the average particle diameter

Equations (22.1) and (22.2) can be combined to obtain the relationship between separation distance and particle diameter.

$$\bar{\Lambda} = \frac{1}{n_P^{\frac{1}{3}}} = \bar{d}_P \left(\frac{\pi \rho_P}{6 C_P} \right)^{\frac{1}{3}} \quad (22.3)$$

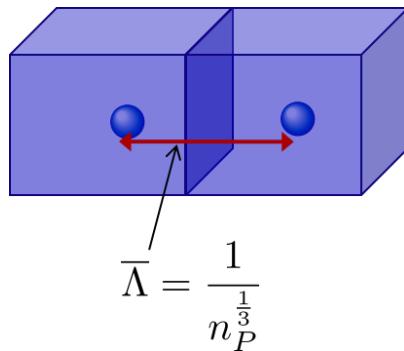


Fig. 22.1: The average particle separation distance is defined as the distance between centers of cubes that each contain the volume of the suspension occupied by a single particle.

Particle separation distance matters because it determines which transport mechanisms are at play when two particles approach for a collision. The particle separation distance is a function of the particle concentration. Surface water treatment plants commonly treat water with turbidity between 1 and 1000 NTU. We will first find the number of clay particles per liter in typical raw water suspensions.

```
import aquaclara
import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.core.constants as con
import aquaclara.research.environmental_processes_analysis as epa
import aquaclara.research.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt

C_Clay = np.arange(1,1000,1)*u.NTU
n_Clay = fm.num_clay(C_Clay,fm.Clay)
fig, ax = plt.subplots()
ax.loglog(C_Clay.to(u.NTU),n_Clay.to(1/u.L))
ax.set(xlabel='Clay concentration ($NTU$)', ylabel='Number of clay per liter')
fig.savefig('Flocculation/Images/NClay_vs_CClay')
plt.show()
```

The next step is to calculate the separation distance between the clay particles over this range of clay concentrations using equation (22.3).

```
import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.core.constants as con
import aquaclara.research.environmental_processes_analysis as epa
import aquaclara.research.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt
lamda_Clay = fm.sep_dist_clay(C_Clay,fm.Clay)
fig, ax = plt.subplots()
ax.semilogx(C_Clay.to(u.NTU),lamda_Clay.to(u.mm))
ax.set(xlabel='Clay concentration ($NTU$)', ylabel=r'Clay separation distance ($mm$)')
fig.savefig('Flocculation/Images/LambdaClay_vs_CClay')
```

(continues on next page)

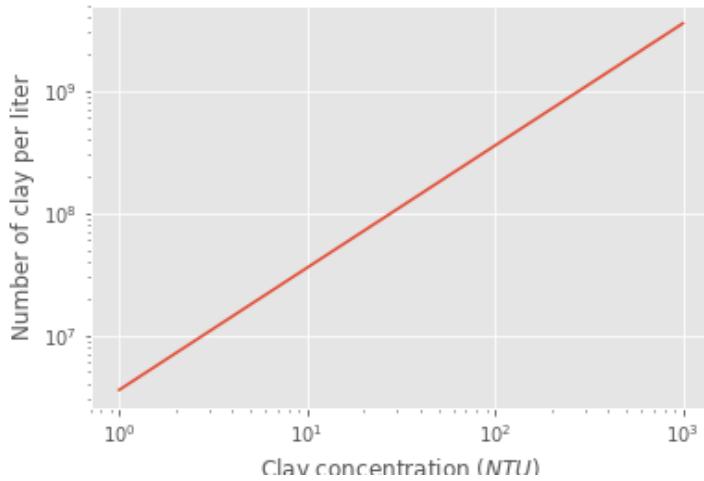


Fig. 22.2: Diagram of number of clay particles per liter as a function of the clay concentration. Note that even 1 NTU water has millions of primary particles per liter.

(continued from previous page)

```
plt.show()
```

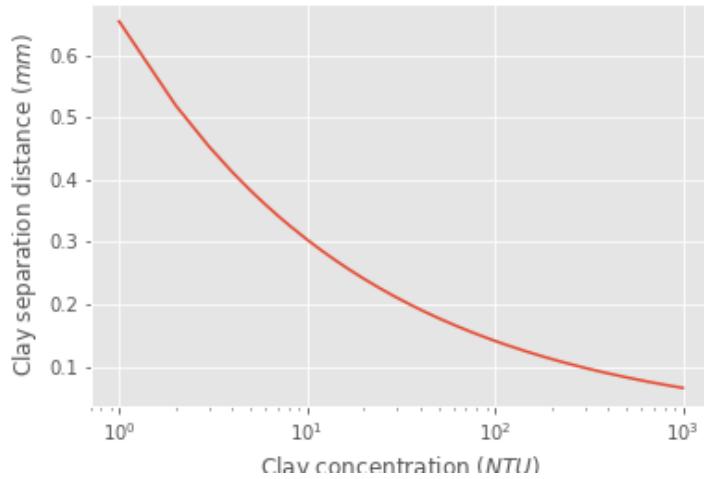


Fig. 22.3: The clay separation distance varies with the cube root of the concentration and thus varies over a relatively narrow range (0.07 mm to 0.7 mm) while the turbidity varies from 1 to 1000 NTU.

Given this range of particle separation distances the next question is whether transport of these particles relative to each other is driven by inertial or viscous dominated processes. Turbulent eddies devolve into smaller and smaller eddies until viscosity finally kills them. Viscosity damps out the effects of inertia at the inner viscous length scale. Higher intensity turbulence can generate more energetic small eddies and can resist the effects of viscosity longer. Thus the inner viscous length scale decreases as the turbulent energy dissipation rate increases.

The Camp-Stein velocity gradient used for flocculators varies from about 20 to 300 Hz. We will convert the Camp-

Stein velocity gradient to an energy dissipation rate using

$$G_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad (22.4)$$

Solving for the average energy dissipation rate, $\bar{\varepsilon}$, we obtain

$$\bar{\varepsilon} = \nu G_{CS}^2 \quad (22.5)$$

We will use the inner viscous length scale, equation (16.48) to determine whether viscous or inertial transport dominates particle collisions in surface water treatment given the range of particle separation distances (see Fig. 22.3).

```
import aquaclaracore.physchem as pc
from aquaclaracore.units import unit_registry as u
import aquaclaracore.constants as con
import aquaclararesearch.environmental_processes_analysis as epa
import aquaclararesearch.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt
Temperature = 20 * u.degC
G=np.arange(1,1000,1)*u.Hz
EDR = G**2 * pc.viscosity_kinematic(Temperature)
Inner_viscous = fm.lambda_vel(EDR, Temperature)
fig, ax = plt.subplots()
ax.semilogx(G.to(u.Hz), Inner_viscous.to(u.mm))
ax.set(xlabel='Velocity gradient (Hz)', ylabel='Inner viscous length scale (mm)')
ax.text(10, 30, 'Eddies cause mixing', fontsize=12, rotation=-30)
ax.text(3, 14, 'Viscous shear', fontsize=12, rotation=-30)
fig.savefig('Flocculation/Images/innerviscous_vs_G')
plt.show()
```

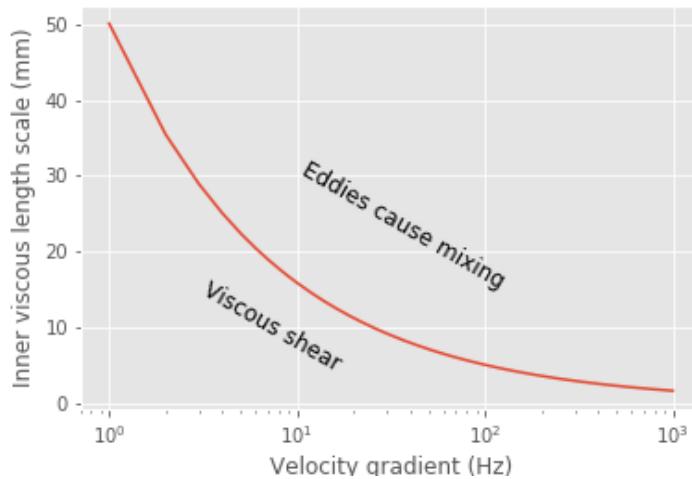


Fig. 22.4: The inner viscous length scale is approximately 3 to 10 mm for velocity gradients that are typically used in flocculators. Clay separation distances are smaller than the inner viscous length scale and thus viscous shear dominates particle collisions in flocculation.

By comparing Fig. 22.3 and Fig. 22.4 it is apparent that the particle separation distances commonly found in surface water treatment plants are much smaller than the inner viscous length scale for all practical flocculation velocity gradients. Thus viscosity will dominate the flocculation process. This key insight reveals why turbulent flow flocculators

have been designed using the dimensionless grouping $G\theta$ which is fundamentally $\sqrt{\frac{\epsilon}{\nu}}\theta$. Given that flocculation is viscous dominated implies that the flocculation process will slow down as the temperature increases and the viscosity increases.

22.1.6 Collision time estimate

Now that we know that the collisions are controlled by viscosity we can begin formulating a model that describes the long distance random walk. The long range transport is assumed to be the rate limiting step. We model a system of two particles where one particle is held fixed and we observe the second particles random motion. It may be helpful to visualize this by playing the video inside your mind in reverse starting from the moment of the collision. That way you know which two particles to follow! The random walk is illustrated in the video in Fig. 22.5.

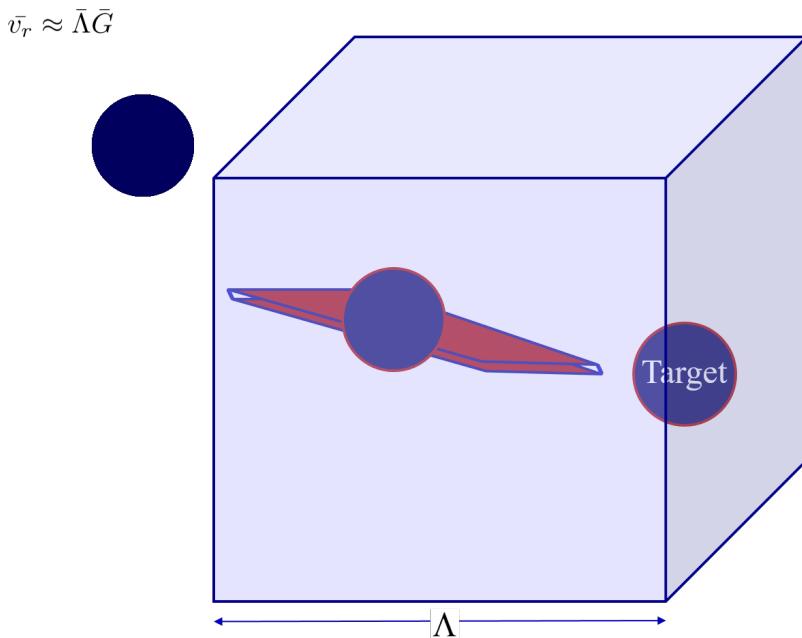


Fig. 22.5: The red volume represents the potential end zone of the random walk that will slide into a collision with a short straight slow walk. The wandering particle sweeps through a volume of water equal to the volume occupied by a single particle.

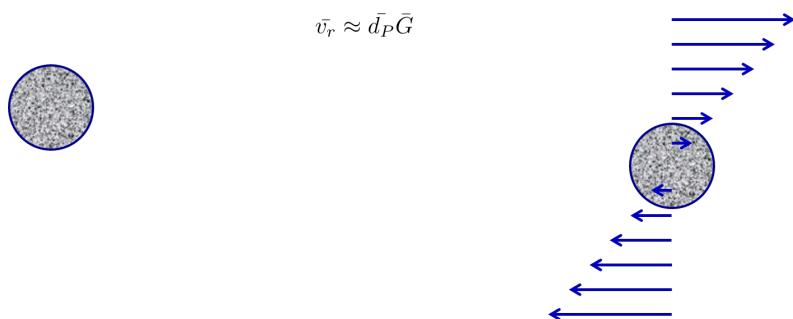


Fig. 22.6: The final approach is the slow, straight path to the collision.

The volume cleared by the wandering particle is proportional to the area defined by a circle with diameter = sum of the particle diameters. This is because the wandering particle will strike the stationary particle if the wandering particles

center is anywhere within a diameter of the center of the stationary particle.

$$V_{\text{Cleared}} \propto \pi \bar{d}_P^2 \quad (22.6)$$

The volume cleared is proportional to time

$$V_{\text{Cleared}} \propto t \quad (22.7)$$

The volume cleared is proportional to the relative velocity between the two particles.

$$V_{\text{Cleared}} \propto \bar{v}_r \quad (22.8)$$

We use dimensional analysis to get a relative velocity for the long range transport controlled by shear. The relative velocity between the two particles that will eventually collide is assumed to be proportional to the average distance between the two particles.

The assumption that the relative velocity scales with the average distance between clay particles leads to the following steps. The first step is just a proposed functional relationship. We could also have jumped to the assumption that the relative velocity is a function of the length scale and the velocity gradient.

$$\bar{v}_r = f(\bar{\varepsilon}, \nu, \bar{\Lambda}) \quad (22.9)$$

In a uniform shear environment the velocity gradient is linear. Thus the relative velocity must be proportional to the length scale.

$$\bar{v}_r = \bar{\Lambda} f(\bar{\varepsilon}, \nu) \quad (22.10)$$

The only way to for $\bar{\varepsilon}$ and ν to produce dimensions of time is to combine to create $1/\bar{G}_{CS}$.

$$\bar{v}_r \approx \bar{\Lambda} \bar{G}_{CS} \quad (22.11)$$

The volume cleared, V_{Cleared} must equal the volume occupied by one particle, V_{Surround} for a collision to occur. Combining the three equations for V_{Cleared} and the equation for v_r we obtain the volume cleared as a function of time.

$$\bar{\Lambda}^3 = V_{\text{Surround}} = V_{\text{Cleared}} \approx \pi \bar{d}_P^2 \bar{\Lambda} \bar{G}_{CS} \bar{t}_c \quad (22.12)$$

Where:

\bar{t}_c is the average time required for a collision between two particles

Solving for the collision time we obtain

$$\bar{t}_c \approx \frac{\bar{\Lambda}^2}{\pi \bar{d}_P^2 \bar{G}_{CS}} \quad (22.13)$$

In summary, a relationship for the mean time between collisions \bar{t}_c was found by proposing an average condition for a collision, successful or unsuccessful, to occur. To define this condition, it was assumed that each primary particle on average occupies a fraction of the reactor volume, $\bar{V}_{\text{Surround}}$, inversely proportional to the number concentration of particles. Furthermore, prior to a collision, a particle on average sweeps a volume, \bar{V}_{Cleared} , proportional to \bar{t}_c and to the mean relative velocity between approaching particles, \bar{v}_r . As an average condition, it was posited that for each collision, \bar{V}_{Cleared} must equal $\bar{V}_{\text{Surround}}$. From this, a relationship for a characteristic collision time, \bar{t}_c , was obtained:

22.1.7 Collision Rates

The change in the number of successful collisions (from a single particles perspective) with respect to time is equal to the mean probability that a collision will result in an attachment, $\bar{\alpha}$, divided by time for one collision, \bar{t}_c .

$$\frac{dN_c}{dt} = \frac{\bar{\alpha}}{\bar{t}_c}, \quad (22.14)$$

Where

$\frac{dN_c}{dt}$ is the rate of successful collisions between primary particles,
 $\bar{\alpha}$ is the mean probability that a collision will result in an attachment,
 \bar{t}_c is the mean time between collisions of primary particles.

The probability that two primary particles attach is expected to be equal to the probability that at least one of the colliding particles has a precipitated coagulant nanoparticle at the initial contact point. It is simpler to derive the probability of attachment from the probability that neither particle has a coagulant nanoparticle at the point where the two particles collide, since the probability of a successful collision includes the probabilities of one particle and of both particles having a coagulant precipitate. The probability of one particle colliding at a point without a coagulant precipitate is $(1 - \bar{\Gamma})$, so the probability of neither particle having a coagulant precipitate at the point of collision is $(1 - \bar{\Gamma})^2$. As this is the probability of a failed collision, the probability of a successful collision is $1 - (1 - \bar{\Gamma})^2$.

Since the model assumes an initially monodisperse population of primary particles and that collisions between differently-sized particles are unfavorable, differential sedimentation is considered negligible. Brownian motion is only significant for particles smaller than $1 \mu\text{m}$ [BL13], and so this model makes the assumption that primary particles are larger than $1 \mu\text{m}$.

The collision rate [PCWSL16] can be obtained by substituting equation (22.13) into equation (22.14).

$$\frac{dN_c}{dt} = \pi \bar{\alpha} \frac{\bar{d}_P^2}{\bar{\Lambda}^2} \bar{G}_{CS} \quad (22.15)$$

where \bar{G}_{CS} is the Camp Stein velocity gradient.

Because the flocculation performance equation will ultimately track particle concentration, the concentration of primary particles, C_P , was substituted for $\bar{\Lambda}$ using

$$\bar{\Lambda}^3 = \frac{\pi}{6} \frac{\rho_P}{C_P} \bar{d}_P^3, \quad (22.16)$$

where ρ_P is the characteristic density of primary particles. Equation (22.16) can be substituted into Equation (22.15) to result in:

$$dN_c = \pi \bar{\alpha} \left(\frac{6 C_P}{\pi \rho_P} \right)^{2/3} \bar{G}_{CS} dt. \quad (22.17)$$

Equation (22.17) reveals that $\frac{dN_c}{dt}$ increases with C_P and $\bar{\Gamma}$. During flocculation C_P will decrease and thus $\frac{dN_c}{dt}$ will also decrease.

22.1.8 Coagulant nanoparticle and primary particle Model

Continuing from [PCWSL16], the above Lagrangian differential relationships are further developed to become integrated performance prediction equations. Equation (22.17) cannot be integrated as written because the concentration of primary particles is expected to change with each collision, and thus that relationship must be specified. During the average time required for one collision it is expected that approximately e^{-1} of the particles will undergo at least one collision. The time required for a collision will change as flocculation proceeds as the average distance between

primary particles increases. The rate of loss of primary particles due to successful collisions will be first order with respect to the number of successful collisions.

$$\frac{dC_P}{dN_c} = -kC_P, \quad (22.18)$$

where k is an experimentally-derived constant that physically represents the portion of the primary particles that become settleable particles on average after each collision time, \bar{t}_c , and will depend, in part, upon the design capture velocity, v_c , used for sedimentation. Since \bar{t}_c increases over time as $\bar{\Lambda}$ increases, the above formulation is not proportional to $\frac{dC_P}{dt}$. Physically, Equation (22.18) states that, with each progressive primary particle collision, C_P decreases by some proportion. Further, Equation (22.18) states that this decrease is directly proportional to C_P . With each successive successful collision, the absolute reduction in C_P is less than the prior one. The value of k is expected to be less than 1, because not all primary particles will have a collision and grow to a size with a sedimentation velocity greater than v_c in the average time required for a collision.

Having Equation (22.18), the next step is to substitute it into Equation (22.17) and integrate. Solving Equation (22.18) for dN_c , substituting it into Equation (22.17) and rewriting the equations in terms of primary particles results in Equation (22.19),

$$\frac{dC_P}{-kC_P} = \pi\bar{\alpha} \left(\frac{6}{\pi} \frac{C_P}{\rho_P} \right)^{2/3} \bar{G}_{CS} dt, \quad (22.19)$$

It is interesting to note that rearranging Equation (22.19) in terms of $\frac{dC_P}{dt}$ gives a C_P exponent of $\frac{5}{3}$. Previous flocculation rate equations were second-order, but the observed flocculation rate was less than second-order [BL13]. The slight deviation from an exponent of two comes from the assumption of [PCWSL16] that relative velocity between colliding particles scales with Λ rather than d_P . This is to say that, in dilute suspensions characteristic of raw water, where particles are separated by $\bar{\Lambda} \gg \bar{d}_P$, the majority of \bar{t}_c is spent with the distance between particles characterized by $\bar{\Lambda}$ instead of \bar{d}_P . The time required for the final approach for a collision is hypothesized to be insignificant compared the time for $\bar{V}_{Cleared}$ to equal $\bar{V}_{Surround}$.

From Equation (22.19) it is possible to integrate and obtain equations for flocculation performance. After separation of variables, one side of the equation is integrated with respect to time from the initial time ($t = 0$) to the time of interest, generally taken to be the mean hydraulic residence time ($t = \theta$). The other side of the equation is integrated with respect to the concentration of primary particles from the value at the initial time (C_{P_0}), equivalent to the initial concentration of primary particles, to the concentration of primary particles at the time of interest (C_P). The integral becomes:

$$\frac{1}{\pi} \left(\rho_P \frac{\pi}{6} \right)^{2/3} \int_{C_{P_0}}^{C_P} C_P^{-5/3} dC_P = -k\bar{\alpha}\bar{G}_{CS} \int_0^\theta dt. \quad (22.20)$$

The integral on the left hand side assumes that ρ_P does not change as C_P changes. One assumption on the right side is that $\bar{\Gamma}$, of which $\bar{\alpha}$ is a function, does not vary with t . This requires that adsorption of coagulant to colloidal particles in rapid mix be fast enough to be approximated as completed by the beginning of flocculation. This assumption may not be valid for high rate flocculators especially under conditions of low C_{P_0} . Further work on the rate and efficacy of coagulant nanoparticle attachment to primary particle surfaces is needed.

The other assumption on the right hand side is that the mean velocity gradient, \bar{G}_{CS} , does not change over the course of the flocculation process. In mechanically-mixed flocculators, the use of a simple spatial average is not reasonable, as the velocity gradient changes dramatically from the bulk flow to the tip of the impeller blade and individual particles follow different paths that expose them to different velocity gradient zones in different sequences and durations [BB98]. The distribution of residence times in a mechanical flocculator would also need to be taken into account for the integration. For baffled hydraulic flocculators, on the other hand, the use of the spatial average, \bar{G}_{CS} , and considering it constant with t is generally a reasonable approximation, as mixing energy in a well-designed hydraulic flocculator is rather uniformly distributed spatially, the zones of higher energy dissipation rate after the baffles do not vary appreciably with time when operating at a constant flow rate, and all particles have similar residence times in the flocculator.

Integration of Equation (22.20) gives:

$$\frac{3}{2\pi} \left(\rho_P \frac{\pi}{6} \right)^{2/3} \left(C_P^{-2/3} - C_{P_0}^{-2/3} \right) = k\bar{\alpha} \bar{G}_{CS} \theta. \quad (22.21)$$

This can be put in terms of $\bar{\Lambda}$ for simplicity by using Equation (22.16) and rearranging in terms of the familiar Camp-Stein parameter, $\bar{G}_{CS} \theta$, to be

$$\bar{G}_{CS} \theta = \frac{3}{2} \frac{(\bar{\Lambda}^2 - \bar{\Lambda}_0^2)}{k\pi\bar{\alpha}\bar{d}_P^2}. \quad (22.22)$$

Equation (22.22) gives guidance for flocculator design in that higher values of $\bar{G}_{CS} \theta$ are needed for flocculators to achieve greater changes in $\bar{\Lambda}$ (or C_P) or to overcome low $\bar{\Lambda}$. It should be noted that the $\bar{\Lambda}_0$ term in Equation (22.22) will generally be very small compared to the $\bar{\Lambda}$ term for most flocculation scenarios. In this case the initial particle separation distance, $\bar{\Lambda}_0$ can be considered negligible. While simplifying the equation, this also gives the result that **flocculators must be designed** not so much for the particle concentrations they will receive but **for the particle concentrations they are intended to produce**.

Modifying Equation (22.22) to be in terms of C_P produces:

$$\bar{G}_{CS} \theta = \frac{3}{2k\pi\bar{\alpha}} \left(\frac{\pi \rho_P}{6 C_P} \right)^{2/3}. \quad (22.23)$$

A desirable way to represent flocculation performance is with the negative log of the fraction of particles remaining (also often referred to as log removal), pC^* , given in [SWSL14] as:

$$pC^* = -\log_{10} \left(\frac{C_P}{C_{P_0}} \right) \quad (22.24)$$

Likewise, a way to simplify Equation (22.21) is to put it in terms of the particle volume fraction, ϕ , defined as:

$$\phi = \frac{C_P}{\rho_P} = \frac{\pi}{6} \left(\frac{\bar{d}_P}{\bar{\Lambda}} \right)^3. \quad (22.25)$$

Putting Equation (22.21) in terms of pC^* and ϕ results in:

$$pC^* = \frac{3}{2} \log_{10} \left[\frac{2}{3} \left(\frac{6}{\pi} \right)^{2/3} k\pi\bar{\alpha} \bar{G}_{CS} \theta \phi_0^{2/3} + 1 \right]. \quad (22.26)$$

Equation (22.26) is a predictive performance model for flocculation in flows with long range particle transport toward collisions dominated by viscous forces. It is proposed as applicable to both laminar and turbulent hydraulic flocculators. Given the properties of the flocculator (\bar{G}_{CS} and θ) and its influent (ϕ_0 and $\bar{\alpha}$), flocculation performance can be predicted in terms of pC^* . The development of Equation (22.26) was the result of a team effort of Cornell University's AquaClara program and hence it will be subsequently referred to as the AquaClara flocculation model.

22.1.9 Experimental Protocols

Equation (22.26) was tested under turbulent conditions. The design scheme chosen to meet these requirements was a tube flocculator, illustrated in Fig. 22.7 and described in [PCWSL16]. This tube flocculator operated in the turbulent flow regime, which for pipe flow means that $Re > 4,000$ [Gra95]. The change in mean energy dissipation rate due to any modification to the system was approximated by

$$\bar{\varepsilon} = \frac{gh_\ell}{\theta}, \quad (22.27)$$

where g is the acceleration due to gravitational force and h_ℓ is the head loss across the flocculator. As mentioned previously, the use of $\bar{\varepsilon}$ assumes that the energy dissipation rate throughout the flocculator is completely uniform so

that it can be represented with a simple spatial average rather than a weighted average accounting for the proportion of the flow passing through different zones of energy dissipation rate. This approximation requires that the majority of energy dissipation (represented by head loss) is due to fluid shear (minor loss) in the bulk flow. If the head loss across a flocculator were primarily as a result of shear on the reactor walls (major loss), only a small fraction of the flow would experience this energy dissipation rate in the near-wall zone, and estimating the mean energy dissipation rate by this method would be invalid.

It is hypothesized, however, that the constrictions in the tube flocculator created submerged free jets downstream, generating fluid shear across the cross section of the flow [PCWSL16]. This hypothesis is supported by a calculation of the head loss due to wall shear using the Darcy-Weisbach equation [Gra95]. The turbulent tube flocculator would be expected to have a total head loss of around 7 cm if only wall shear were present, but an average head loss of 90 cm was measured across the flocculator by means of a differential pressure sensor, indicating that significant fluid shear is present.

Referring to Equation (22.27), changing the head loss by changing the constriction of the tubes or changing the water elevation difference across the flocculator would change the energy dissipation rate. Likewise, either of the above two modifications would change the mean hydraulic residence time in the flocculator. This could also be accomplished by changing the length of the flocculator.

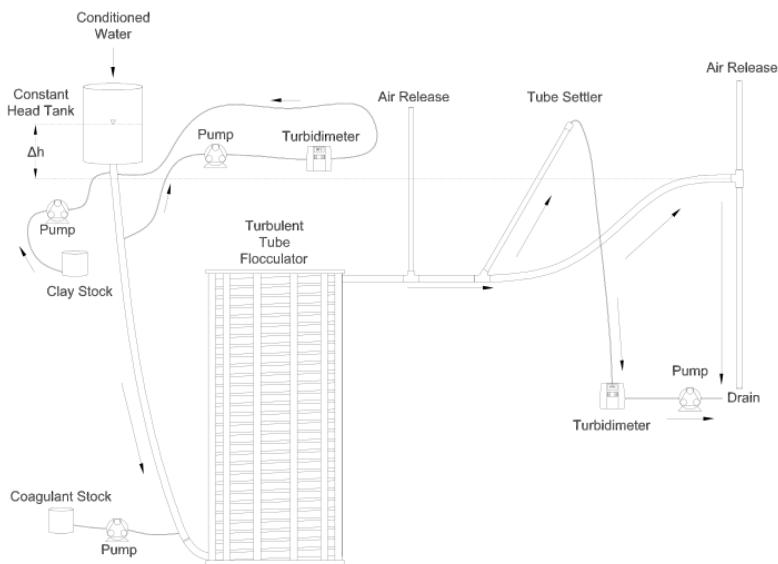


Fig. 22.7: Diagram of Turbulent Tube Flocculator adapted from [PCWSL16] with modifications made to the outlet weir system and the addition of strong base solution.

Fig. 22.7 illustrates the process sequence used in this study. At the beginning of the process, tap water from the Cornell University Water Filtration Plant came into the system with, on average, a pH of 7.67, a turbidity of 0.056 nephelometric turbidity units (NTU), a total hardness of 150 mg/L, a total alkalinity of 140 mg/L, and a dissolved organic carbon (DOC) concentration of 1.80 mg/L [BPMWSCIWSCUWS16]. This water was temperature-controlled by means of a PID (proportional-integral-derivative) controller, which regulated the relative fractions of hot water and cold water used to maintain the level in the constant head tank. The temperature-controlled water was passed through a granular activated carbon (GAC) filter to reduce the effect of dissolved organic matter (DOM) on experimental results. The water was then sent to the constant head tank, where it was bubbled with air to strip out supersaturated dissolved gases that might come out of solution during the experiment, resulting in formation of bubbles.

From the constant head tank, this conditioned water was delivered to the turbulent tube flocculator. Before entry to the flocculator, the water was set at a constant primary particle concentration by means of a computer-controlled peristaltic pump that introduced a concentrated kaolinite clay suspension (R.T. Vanderbilt Co., Inc., Norwalk, Connecticut) of about 250 g/L. A fraction of the mixed flow was sampled by a peristaltic pump and analyzed for turbidity with an

HF Scientific MicroTOL turbidimeter at a distance of greater than ten diameters downstream from the clay input and then reintroduced at the point where clay suspension was added. This turbidity reading was input into a PID control system which determined the speed of the clay pump according to the discrepancy between the influent turbidity and the experimental target value.

Along with the clay, strong base (NaOH) manufactured by Sigma-Aldrich (St. Louis, MO) was added upstream of the flocculator with a peristaltic pump to keep the pH of the water at 7.5 ± 0.5 , which was the criterion set for the pH in these experiments. In the winter, the pH of the tap water dropped close to 7, and so sufficient NaOH was added to account for seasonal variations in the natural base-neutralizing capacity (BNC) of the water and to raise the pH above 7 to around 7.5. This base addition was also sufficient to neutralize the acidity of the polyaluminum chloride (PACl) coagulant used for this study, which had been found to impact the solubility of PACl at high doses. Base doses were calculated to account for the normality of the PACl solution, based on a titration which found that the PACl solution was approximately 0.025 equivalents of strong acid per gram as Al.

Just prior to entering the flocculator, PACl coagulant (PCH-180) manufactured by the Holland Company, Inc. (Adams, Massachusetts) was added to the flow by a computer-controlled peristaltic pump which varied the coagulant dose between experiments. After entering the system, the coagulant then entered a small orifice used to accomplish rapid mix by forming a jet downstream. From there, the suspension traveled up through the flocculator made of 3.18 cm (1.25 in) inner diameter tubing. Within the flocculator, the fluid passed through constrictions in the tubing that caused the flow to contract, resulting in flow expansions afterward and achieving increased mixing and energy dissipation.

After leaving the flocculator, the flow passed a vertical tube with a free surface that served as an air release. This removed bubbles in the system so that they would not interfere with settling or analysis of the flocs. A portion of the flow was then diverted for sedimentation by means of a peristaltic pump up a clear one-inch PVC pipe angled at 60° . The flow rate through the pump was selected based on the dimensions of the tube and its angle to achieve a desired capture velocity, v_c . The supernatant from this tube settler was passed through an HF Scientific MicroTOL nephelometric turbidimeter to record the effluent turbidity for the duration of the experiment. Recording the settled effluent turbidity made it possible to calculate the pC^* term in Equations (22.26) (in terms of primary particles) and also made possible comparison with data from [SWSL14].

After data from the settled flocs had been collected, the flow from the effluent turbidimeter was sent to the drain along with the bulk flow. The bulk flow traveled past a second air release before exiting the drain. The air release gave the flow exiting the drain a free surface as it flowed over the exit weir so that the exiting water developed into a supercritical flow. Thus, the flow over the weir was not influenced by the flow downstream of the free surface, and the flow rate could be controlled by adjusting the elevation of the free surface before the drain. The outlet weir was a 1-1/4 PVC pipe within an upright 3 clear pipe, which were joined by a flexible coupling adapter. The effluent water accumulated in the clear outer pipe until it reached the elevation of the top of the inner pipe and flowed down through it. The flow rate could be adjusted by loosening the flexible coupling so that the elevation of the top of the inner pipe could be adjusted. As the bulk flow exited down out of the inner pipe to the drain, it passed over a glass electrode sensor to measure pH.

22.1.10 Results

The above process was used to conduct the experiments to test the applicability of Equation (22.26) in turbulent flocculation. The influent turbidity was set at a constant of 900 NTU. The mean energy dissipation rate was about 21.5 mW/kg, which resulted from choosing a flow rate of about 110 mL/s so that the Reynolds number was just above 4,000. These values were chosen to ensure viscous-dominated turbulent initial conditions. For these experiments, coagulant doses ranged from 0.05 to 98 mg/L as Al. A v_c of 0.12 mm/s was used for all experiments. Data from these nominally viscous experiments are shown in Fig. 22.8 as a function of coagulant dose.

The data shown in Fig. 22.8 were compared with the viscous model, as shown in Fig. 22.9. In this graph, the data are plotted in terms of Equation (22.26) and its corresponding composite parameter taken from Equation (22.15),

$$N_c \propto \bar{\alpha} \theta \bar{G}_{CS} \phi_0^{2/3}. \quad (22.28)$$

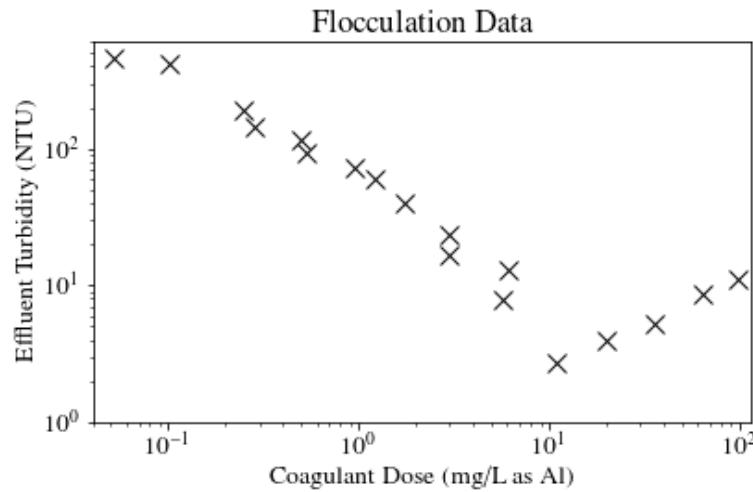


Fig. 22.8: Effluent turbidity as a function of coagulant dose for experiments performed with influent turbidity of 900 NTU, velocity gradient of 147/s, and hydraulic residence time of about 413s.

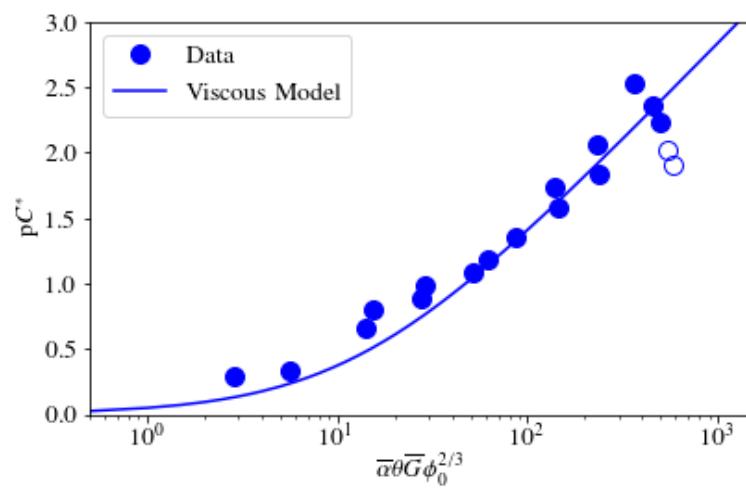


Fig. 22.9: Fit of Equation (22.26) to data from $Re \approx 4,000$ experiments. Hollow points indicate data not used in fitting the model.

At the highest values, however, a marked decrease begins. For these graphs, the model fits were done for all points where increasing performance was seen, because the model does not currently include a mechanism for the decreasing performance. The values for k were determined by the Levenberg-Marquardt algorithm, and the value for the model was 0.030. The R^2 value for the fit is 0.958 and the sum of squared errors is 0.228 (mean pC* error of 0.128).

From the values given previously, the ratio $\frac{\bar{\Lambda}_0}{\bar{\eta}}$ can be calculated for the experimental conditions. Equation (22.16) can be used to compute $(\bar{\Lambda}_0)$. For these experiments, \bar{d}_P is taken to be the average diameter of kaolinite clay particles, found by [WZL+15] and [SWSL15] to be $7 \mu\text{m}$. The concentration can be converted from NTU to the necessary mass/volume (mg/L) unit by using as a proportion the measurement reported by [WZL+15] of 68 NTU for 100 mg/L of kaolinite clay. Last, the density was assumed to be 2.65 g/cm^3 for kaolinite.

For flocculation in laminar flows, data were used from the work of [SWSL14]. Fig. 22.10 shows Equation (22.26) fit to results for a capture velocity of 0.12 mm/s at two hydraulic residence times, five influent turbidity values and a range of coagulant doses. [SWSL14] showed that the projected x-axis intercept of the linear region of the data (with a log-log slope of 1 according to her plotting of the data) was proportional to the capture velocity used for sedimentation. Correspondingly, k is expected to be a function of capture velocity.

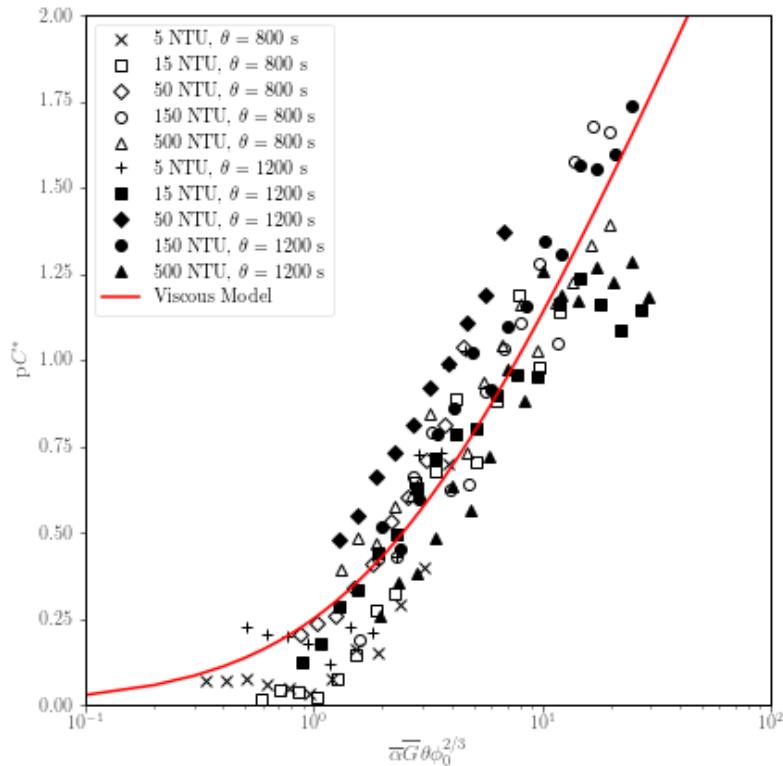


Fig. 22.10: Fit of Equation (22.26) to laminar flocculation data from [SWSL14].

Referring to Fig. 22.10, Equation (22.26) fits the data from [SWSL14] well with a k value of 0.027. The resulting R^2 for this fit is 0.844. The sum-squared error is 5.03, giving an average pC* error of 0.034 for the fit.

22.1.11 Discussion

The goodness of fit seen in Fig. 22.9 and Fig. 22.10 indicate that the model captures the important mechanisms governing flocculation performance for a wide range of coagulant doses in both laminar and turbulent hydraulic flocculation. One of the challenges in fitting the data pertained to the assumption made for the characteristic diameter of PACl

precipitate clusters, \bar{d}_C . This value has significant influence on the value of $\bar{\Gamma}$, which in turn influences the values of the composite parameter (Equation (22.28)).

It is known that PACl contains aluminum monomers and oligomers as well as Al₁₃ and Al₃₀ nanoclusters, with the larger Al₃₀ nanoclusters having a diameter of 1 nm and a length of 2 nm [MCM+12]. It has been found, however, that the components of PACl self-aggregate and go on to form larger clusters [SWSL13]. For these experiments, the value of \bar{d}_C was chosen based on sizing experiments performed by Garland (2015) with a Malvern Zetasizer Nano-ZS to analyze a 138.5 mg/L (as Aluminum) solution of PACl.

A limitation of the model can be seen in the data in Fig. 22.9 at higher values of the composite parameters. After increasing steadily for all of the preceding range of coagulant doses, the performance began to decline after the dose of 10.9 mg/L as Aluminum. A simple hypothesis for the decline in performance (which corresponds with an effluent turbidity increase over the five data points from 2.7 NTU to 11.1 NTU) is that an increase in free PACl nanoparticles made a significant contribution to the effluent turbidity. As the PACl concentration increased, the coverage of reactor and clay platelet surfaces by coagulant became more complete and the free coagulant concentration also increased. With very high coagulant doses like the ones used in the upper end of the experimental range, it is possible that the formation of PACl self-aggregates was favorable, increasing the turbidity of the suspension. Indeed, calculation of the volume fraction for the 10.9 mg/L experimental PACl dose gives a volume fraction value (for clay and coagulant combined) of 6.1×10^{-4} , while for the highest dose of 98 mg/L as Al, the value was 8.3×10^{-4} , a 37% increase due solely to the increased contribution of PACl precipitates.

Another possibility is that as $\bar{\Gamma}$ increases above 0.5, the resulting flocs are increasingly formed by PACl-PACl bonds instead of by PACl-kaolinite bonds. If the PACl-PACl bonds are weaker than PACl-kaolinite bonds, it is possible that attachment efficiency decreases for high $\bar{\Gamma}$. The weakness of PACl-PACl bonds compared with PACl-kaolinite bonds is suggested by the relative charges of PACl and kaolinite. While PACl precipitate surfaces are positively charged, the surfaces of kaolinite are mostly negatively charged [WZL+15]. Therefore, it follows that PACl precipitates will likely have more affinity for kaolinite surfaces than for other PACl precipitates. The $\bar{\Gamma}$ calculated for the peak performance was 0.52, and so it is possible that performance decreased past this point because the strength of bonds for experiments at higher doses were weaker.

Applying the AquaClara flocculation model to the design of a hydraulic flocculator indeed gives reasonable results. Assuming that a flocculator is expected to receive sufficiently high turbidities that the influent concentration can be neglected, Equation (22.23) can be used. In order for it to treat to a settled effluent of 3 NTU (pre-filtration) with sufficient PACl to achieve a surface area coverage fraction of 0.5, it would need to have a $\bar{G}_{CS}\theta$ of 99,600. [DC08] give the range of $\bar{G}_{CS}\theta$ values pertinent to flocculation of high turbidities as between 36,000 and 96,000, so this result is reasonable. This analysis does not account for removal of particles in a floc blanket that would enable use of a lower value of $\bar{G}_{CS}\theta$.

Regarding flocculator design, recommended values of \bar{G}_{CS} in flocculation range from $10\frac{1}{s}$ to $100\frac{1}{s}$, which correspond to $\bar{\varepsilon}$ values of about 0.1 to 10 mW/kg [ML00]. However, there is evidence that higher velocity gradients are advantageous, as found by [GWSL16] as well as the work done in this study, which made use of energy dissipation rates of about 22 mW/kg. For hydraulic flocculators, at least, designers should consider using higher energy dissipation rates than conventionally used, since they have a much lower ratio of maximum to average energy dissipation rate, leading to less floc breakup at high energy dissipation rates compared to mechanically mixed flocculators.

The assumption that nonsettleable particle removal is proportional to primary particle removal appears to be supported by the goodness of fit supplied by the AquaClara flocculation model to the data (see Fig. 22.9). This assumption is likely included in the values of k fit by the model. A mechanistic understanding of k will require that the proportionality between nonsettleable and primary particles be understood explicitly. It is possible that k is a function of rapid mix effectiveness, and since k predicts pC^* , it will also be dependent on v_c . Future experiments at varying v_c are planned. Currently, $\bar{\alpha}$ is calculated assuming that coagulant nanoparticle attachment to the primary particles was accomplished very early on in the flocculator, but if colloid coating by coagulant nanoparticles is dependent upon diffusion rather than exclusively on hydraulic shear, it will be a function of time in addition to $\bar{G}_{CS}\theta$, making flocculation less effective at high flow rates. Additionally, the use of $\bar{\varepsilon}$ (or \bar{G}_{CS}) assumes a uniform energy dissipation rate in the flocculator. Any spatial deviation in the laboratory flocculator from a uniform energy dissipation rate would have had an impact on the values of k relative to their theoretical values, which are dictated by the rate of conversion of primary particles to flocs.

22.1.12 Summaries

We developed a model that predicts hydraulic flocculator performance. Regardless of whether the flow is laminar or turbulent, viscous forces control the relative velocities between particles on a collision path, and the performance equation is $pC^* = \frac{3}{2} \log_{10} \left[\frac{2}{3} \left(\frac{6}{\pi} \right)^{2/3} \pi k \alpha \bar{G}_{CS} \theta \phi_0^{2/3} + 1 \right]$.

Model predictions were compared with data from [SWSL14]. To validate the first equation and the second equation in turbulent flow, experiments were conducted in turbulent flow for initial conditions of $\frac{\bar{A}}{\bar{\eta}} < 1$. It was found that the viscous equation was slightly more suitable in these conditions. Until further work is done on delineating the relative predominance of viscous and inertial forces over the range of turbulent flocculation conditions, the authors recommend using the AquaClara flocculation model. For design purposes, this model indicates that flocculator design is more sensitive to the desired effluent concentration of particles than the range of influent concentrations that might be encountered. This study also supports the use of higher energy dissipation rates (or velocity gradients) than conventionally recommended for hydraulic flocculators. Further work is needed to characterize the functional dependence of k on capture velocity and energy dissipation rate, as well as the relationship between the final concentrations of primary and primary particles.

22.2 Geometric Explanation of the Effects of Humic Acid on Flocculation

Dissolved organic matter (DOM) is found in all surface and ground waters and has a significant effect on drinking water treatment, since the presence of DOM can create a need for increased coagulant doses in addition to being a precursor of disinfection byproducts (DBPs). This work evaluated use of polyaluminum chloride (PACl) as a coagulant for a synthetic surface water to determine the effect of DOM on the settled effluent turbidity. Mechanistically-based, scalable algorithms for operation of hydraulic flocculators were developed in this research based on observations of residual turbidity. Data were obtained using a laminar-flow tube flocculator and a lamellar tube settler. The research employed a flocculation model previously published by some of the authors and considered modifications to the model algorithm to incorporate the effects of humic acid. Two adjustable model parameters were used to fit data, one related to incorporated the capture velocity used for sedimentation, and one that estimated the average size of dissolved humic acid molecules. The modified model that accounted for the presence of humic acid was able to independently predict the experimental results from 60 experiments at a different influent turbidity. The predictive model is expected to provide insights into the interactions between dissolved species and the coagulant nanoparticles.

22.2.1 Introduction

The main objective of this research was to observe and model the effects of dissolved organic matter (DOM) on flocculation to enhance the performance of a hydraulic flocculator in the context of a process train with subsequent unit processes (i.e., sedimentation). Prior research has shown that multiple variables influence the performance of hydraulic flocculators in drinking water treatment, including the concentration and type of suspended particles in the raw water, the concentration of DOM, coagulant type and dose, and hydraulic residence time and energy dissipation rate in the flocculator (Kawamura, 1991).

The design and operation of hydraulic flocculators would be assisted by a predictive model that can characterize performance of flocculator designs. A general scalable model which uses dimensionally correct relationships that are based upon relevant flocculation mechanisms was created by Pennock et al. (2018) and successfully applied to quantify the effect of varying flocculator design and operational parameters on the post-sedimentation residual turbidity that corresponded to a selected sedimentation capture velocity. However, this model did not account for the presence of varying levels of DOM.

Previous researchers have hypothesized that DOM interacts with coagulants through various mechanisms. Jarvis and Jefferson (2007) state that the aggregation mechanisms through which DOM is removed include a combination of charge neutralization, entrapment, adsorption, and complexation with coagulant metal ions into insoluble particulate

aggregates. Optimal conditions for turbidity or pathogen removal are not always the same as those for DOM removal (Hua and Reckhow, 2008). Because of the variable composition of DOM, the mechanisms of removal could be different for different types of DOM in water (Sharp and Jarvis, 2006). The hydrophobic fraction of DOM, which includes humic acids, is generally removed in coagulation more effectively than the hydrophilic fraction (Marhaba et al., 2003; Matilainen and Vepsäläinen, 2010).

Prehydrolyzed polymer coagulants, such as polyaluminum chloride (PACl), have been reported to have advantages over conventional coagulants such as alum, including less temperature or pH dependence, as well as smaller alkalinity consumption, but the characteristics of the water to be treated (e.g., alkalinity, pH, and DOM content) play a major role in the choice of a proper coagulant. Consequently, prehydrolyzed coagulants have not been consistently observed to enhance the removal efficiency of DOM (Hu, 2006).

The research described in this paper builds on the AquaClara hydraulic flocculation model developed by Pennock et al. (2018) and adds detail to the attachment efficiency coefficient describing geometric and probabilistic interactions between clay, coagulant, DOM, and reactor walls. The synthetic raw water used in experiments added one type of DOM, humic acid, to a previously studied synthetic system (Swetland et al., 2014) with the expectation that the resulting system would be sufficiently well-characterized to develop a predictive model.

The AquaClara flocculation model is based on the observation that coagulant precipitates form nanoparticles that attach to the surfaces of suspended particles (clay) and reactor walls. Swetland et al. (2014) found particle attachment efficiency in a hydraulic flocculator to be proportional to the fractional surface coverage of suspended clay by precipitated coagulant (alum and PACl) nanoparticles. The success of the surface coverage model in explaining the interactions between clay, coagulant nanoparticles, and reactor walls led to the hypothesis that hydrophobic DOM macromolecules may attach to the coagulant nanoparticles and reduce the amount of PACl surface area that is available for attachment.

22.2.2 Experimental Protocols

Experiments were conducted using the laboratory apparatus illustrated in Fig. 22.11. Cornell University tap water was pumped from an aerated and temperature-controlled reservoir and mixed with a concentrated stock suspension of kaolinite clay (R.T. Vanderbilt Co., Inc.) to form a feed-back regulated constant turbidity raw water source (Weber-Shirk, 2016).

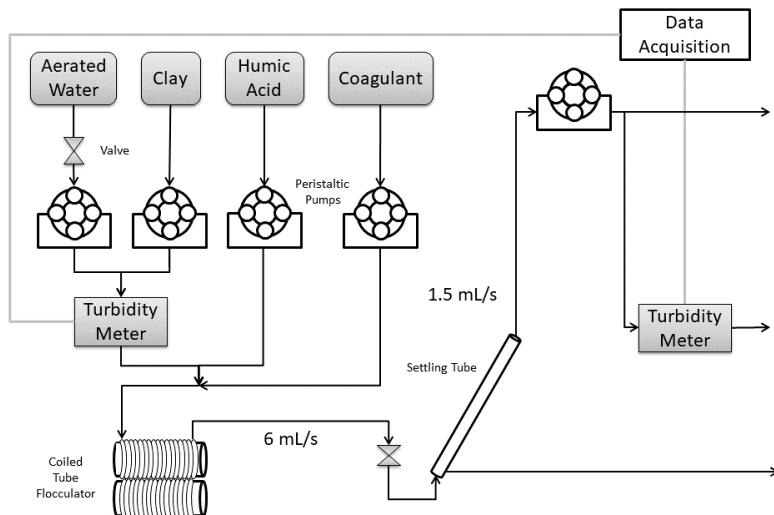


Fig. 22.11: Experimental System Schematic

Reported Cornell University tap water characteristics are listed in Table 22.1. A concentrated suspension of hu-

mic acids was mixed with the raw water source to produce humic acid concentrations ranging from 0 to 15 mg/L. The humic substances used in experiments were obtained in the form of sodium salt from Sigma-Aldrich (H16752). Polyaluminum chloride (PACl) coagulant doses (Holland Company, Adams, MA) ranging from 0.53 to 2.65 mg/L as Al were used to treat the synthetic raw water.

Table 22.1: Average properties of tap water provided by Cornell University Water Filtration Plant as reported by BP-MWS, et al., 2016.

Property	Average Value
Turbidity	0.056 NTU
Total Hardness	150 mg/L
Total Alkalinity (as CaCO_3)	140 mg/L
Dissolved Organic Carbon (DOC)	1.80 mg/L

The coagulant dosage and humic acid concentrations were regulated by adjusting the rotation speed of separate peristaltic pumps. The pH of the treated effluent was monitored in each experiment and was 7.5 ± 0.3 . Influent turbidities of 50 NTU and 100 NTU were tested. Flocculation was accomplished by laminar flow through a coiled 9.52 mm inner diameter tube. The average velocity gradient in the coiled flocculator, \bar{G}_{CS} was calculated according the equation derived by Tse et al. (2011) as

$$\bar{G}_{CS} = \bar{G}_{CS_{Straight}} \sqrt{1 + 0.033 \left[\log \left(\frac{4Q_{Plant}}{\pi D \nu} \sqrt{\frac{D}{R_c}} \right) \right]^4} \quad (22.29)$$

where $\bar{G}_{CS_{Straight}}$ is fluid velocity gradient in a straight tube, Q_{Plant} is the experimental flow rate, D is the inner diameter of the flocculator tube, R_c is the diameter of curvature of the flocculator coils, and ν is the kinematic viscosity of water, which is about $1 \times 10^{-6} \frac{\text{m}^2}{\text{s}}$ at 20°C (Kundu and Cohen, 2008). The overall experimental flow rate was 6 mL/s and the radius of curvature of the coiled tubing (R_c) was 15 cm.

The value of $\bar{G}_{CS_{Straight}}$ was calculated by first estimating the head loss in a straight tube of the equivalent diameter and material using the Hagen-Poiseuille equation for laminar flow:

$$h_L = \frac{32\nu \bar{u} L}{D^2 g} \quad (22.30)$$

where L is the length of the tube (25.45 m in these experiments), \bar{u} is the mean velocity (84 mm/s) of the flow, and g is the acceleration due to gravity (Granger, 1995). From this head loss, an average rate of the loss of kinetic energy, $\bar{\varepsilon}$ can be estimated using

$$\bar{\varepsilon} = \frac{gh_L}{\theta} \quad (22.31)$$

where θ is the mean hydraulic residence time (Pennock et al., 2018). The hydraulic residence time was 302 s as calculated by

$$\theta = \frac{L}{\bar{u}} \quad (22.32)$$

The energy dissipation rate, which was calculated to be 2.24 mW/kg, can be converted to velocity gradient, \bar{G}_{CS} , by

$$\bar{G}_{CS} = \sqrt{\frac{\bar{\varepsilon}}{\nu}} \quad (22.33)$$

which gave a velocity gradient of 50.1 s^{-1} . Using this value for $\bar{G}_{CS_{Straight}}$ in Equation 1 resulted in a value of 71.1 s^{-1} for \bar{G}_{CS} .

A coiled tube flocculator was used in this research because it is a high-Peclet-number reactor much like a baffled hydraulic flocculator and also because the average velocity gradient in laminar tube flow is well defined (Weber-Shirk

and Lion, 2010). After flowing through the flocculator, a fraction of the flow was passed through a tube settler and the settled water turbidity was recorded continuously for each experiment. The 1.37 m (4.5 ft) tube settler, with an inner diameter of 2.66 cm, had an entry port diameter of 0.95 cm (3/8 in) near the bottom and an exit port diameter of 0.635 cm (1/4 in) near the top. The capture velocity was controlled at 0.102 mm/s using a peristaltic pump with flow set by

$$Q_S = \frac{\pi}{4} D_S^2 V_c \left(\frac{L_S}{D_S} \cos \alpha_S + \sin \alpha_S \right) \quad (22.34)$$

where V_c is the capture velocity, L_S is the length of the tube settler, D_S is the diameter of the tube settler, and α_S is the angle of inclination of the tube settler, which was set at 60 degrees (Schulz and Okun, 1984).

22.2.3 Model Formation

A flocculation model considering the effects of humic acid should predict the effective collisions between colloids for a given set of conditions. The dimensionless product of the fluid velocity gradient and mean hydraulic residence time, $\bar{G}_{CS}\theta$, has been used as a measure of the collision potential provided by a flocculator that experiences laminar flow (Camp, 1955; Cleasby, 1984). It is well known that not all collisions between suspended particles result in aggregation, and average attachment efficiency, $\bar{\alpha}$, has been used to denote the fraction of successful collisions (AWWA, 1999).

The initial primary particle volume fraction, ϕ_0 , also influences coagulation (Ives, 1968; OMelia, 1972) and gives the fraction of the volume of the suspension occupied by the influent primary particles,

$$\phi_0 = \frac{C_0}{\rho_P} \quad (22.35)$$

where C_0 is the influent particle concentration (kaolinite clay in these experiments) and ρ_P is the density of influent particles (Swetland et al., 2014).

In laminar-flow flocculators, the velocity of one floc relative to another scales with the average separation distance between flocs (Swetland et al., 2014). The time between floc collisions is inversely proportional to both ϕ and the relative velocity between flocs. Because the relative velocity between flocs is proportional to separation distance, the time between collisions is proportional to $\phi^{\frac{1}{3}}$, since the average separation distance, $\bar{\Lambda}$, is given by

$$\bar{\Lambda} = d_P \left(\frac{\pi}{6\phi} \right)^{\frac{1}{3}} \quad (22.36)$$

The result is that, for laminar flow, the average time for primary particle collisions scales with $\phi^{-\frac{2}{3}}$ (Weber-Shirk and Lion 2010).

A laminar-flow hydraulic flocculator model was developed and validated based on the above analysis in Pennock et al. (2018) with the form

$$pC^* = \frac{3}{2} \log_{10} \left[\frac{2}{3} \left(\frac{6}{\pi} \right)^{\frac{2}{3}} \pi k \bar{\alpha} \bar{G}_{CS} \theta \phi_0^{\frac{2}{3}} + 1 \right] \quad (22.37)$$

where k is a fitting parameter dependent on the value of V_c used for sedimentation, $\bar{\alpha}$ is the mean fraction of collisions that are successful (i.e., result in aggregation), and pC^* is defined as

$$pC^* = -\log \left(\frac{\text{Effluent turbidity}}{\text{Influent Turbidity}} \right) \quad (22.38)$$

Equation (22.37), referred to as the AquaClara flocculation model in Pennock et al. (2018), is a Lagrangian hydrodynamic model that assumes that the aggregation of primary particles is rate-limiting. It further assumes that these particles, on average, will collide when the volume of fluid swept out as one particle approaches the other is equal to the average volume occupied by a single particle in the suspension. The time for these collisions to occur increases as flocculation proceeds, since the concentration of primary particles decreases in a way that is assumed to be first order

with respect to collisions. Thus, with each successive collision, the average volume occupied by primary particles increases, and it takes longer for the next collision to occur. In Equation (22.37), performance is linearly proportional to the logarithm of the effective collision potential, $\log(\bar{\alpha}\bar{G}_{CS}\theta\phi_0^{2/3})$.

This group of parameters is the same as the group first described by Swetland et al. (2014), with the exception that they used the estimated fractional coverage of the colloid surface by coagulant, $\bar{\Gamma}_{PACl-Clay}$, as a measure of attachment efficiency instead of $\bar{\alpha}$. Pennock et al. (2018) recognized that surface coverage of both particles participating in a collision matters, and introduced $\bar{\alpha}$ to convert the geometric information contained in $\bar{\Gamma}_{PACl-Clay}$ to a probability of a successful collision. Using data gathered by Swetland et al. (2014), Pennock et al. (2018) were able to predict the results of independent laminar flocculation experiments with no adjustable parameters in the absence of added DOM.

Experimental results obtained with added humic acid present are shown in Fig. 22.13 along with predictions based on the AguaClara flocculation model [Equation (22.37)]. It was evident that the attachment efficiency was adversely affected by the addition of humic acid. Referencing adsorption measurements by Davis (1982), a minority (his study found 20%) of added DOM would be adsorbed by kaolinite at the experimental pH of 7.5. Thus, most humic acid macromolecules were available to attach to the added coagulant nanoparticles. The following simplifying assumptions were made to account for the presence of humic acids: 1) humic acid macromolecules attach to coagulant nanoparticles to form nanoaggregates, 2) nanoaggregates attach to clay and to the reactor walls, and 3) the surfaces of precipitated coagulant nanoparticles promote adhesion, while the surfaces of bound humic acids prevent adhesion.

In this study, humic acid macromolecules and $PACl$ nanoparticles were modeled as spheres. Based on the size of coagulant nanoparticles and humic acid macromolecules, their number concentrations, N_{HA} and N_{PACl} respectively, can be estimated by

$$N_{HA} = \frac{C_{HA}}{\rho_{HA} \frac{\pi}{6} d_{HA}^3} \quad (22.39)$$

and

$$N_{PACl} = \frac{C_{PACl}}{\rho_{PACl} \frac{\pi}{6} d_{PACl}^3} \quad (22.40)$$

where C_{PACl} is the dose of coagulant in mg/L as Al; C_{HA} is the concentration of humic acid in mg/L; ρ_{PACl} is the density of the coagulant (Swetland et al. (2013) found $1,138 \frac{kg}{m^3}$); ρ_{HA} is the density of humic acid, $1,520 \frac{kg}{m^3}$ (Sigma-Aldrich, 2014); d_{HA} is the diameter of humic acid macromolecules (an adjustable model parameter); and d_{PACl} is the diameter of precipitated $PACl$ coagulant nanoparticles, taken to be 90 nm as found by Dr. Casey Garland (personal communication).

A key model assumption was that humic acid macromolecules cannot adhere to a coagulant surface that is occupied by a humic acid macromolecule, since humic acid macromolecules are assumed to not appreciably self-aggregate. The outcome of this assumption is that humic acid macromolecules attach to an uncovered surface of coagulant and do not stack on top of one another. The available surface area of the $PACl$ nanoparticle was modeled as the surface area of an equivalent sphere. The amount of that area that is occupied by an attached humic acid macromolecule was estimated as the projected area of a sphere with volume equivalent to a humic acid macromolecule. A new variable describing the coverage of coagulant nanoparticle surface area by humic acid macromolecules,

$$\bar{\Gamma}_{HA-PACl} = \frac{\frac{\pi}{4} d_{HA}^2}{\pi d_{PACl}^2} \frac{N_{HA}}{N_{PACl}} \quad (22.41)$$

was created to be incorporated into the model (within $\bar{\alpha}$) to represent the fraction of the $PACl$ nanoparticle surface area that is covered by humic acid macromolecules.

The first two steps in particle aggregation, where humic acid macromolecules attach to coagulant nanoparticles and then the resulting nanoaggregates attach to clay surfaces, were assumed to be rapid because diffusion is an effective transport process for nanoparticles (Benjamin and Lawler, 2013). Subsequent to rapid mix, the clay particles with attached nanoaggregates undergo collisions during the flocculation process and the aggregation process is governed by fluid shear (Pennock et al., 2018). The success of a collision between clay particles is hypothesized to be dependent on the properties of the contact surfaces at the initial point of contact.

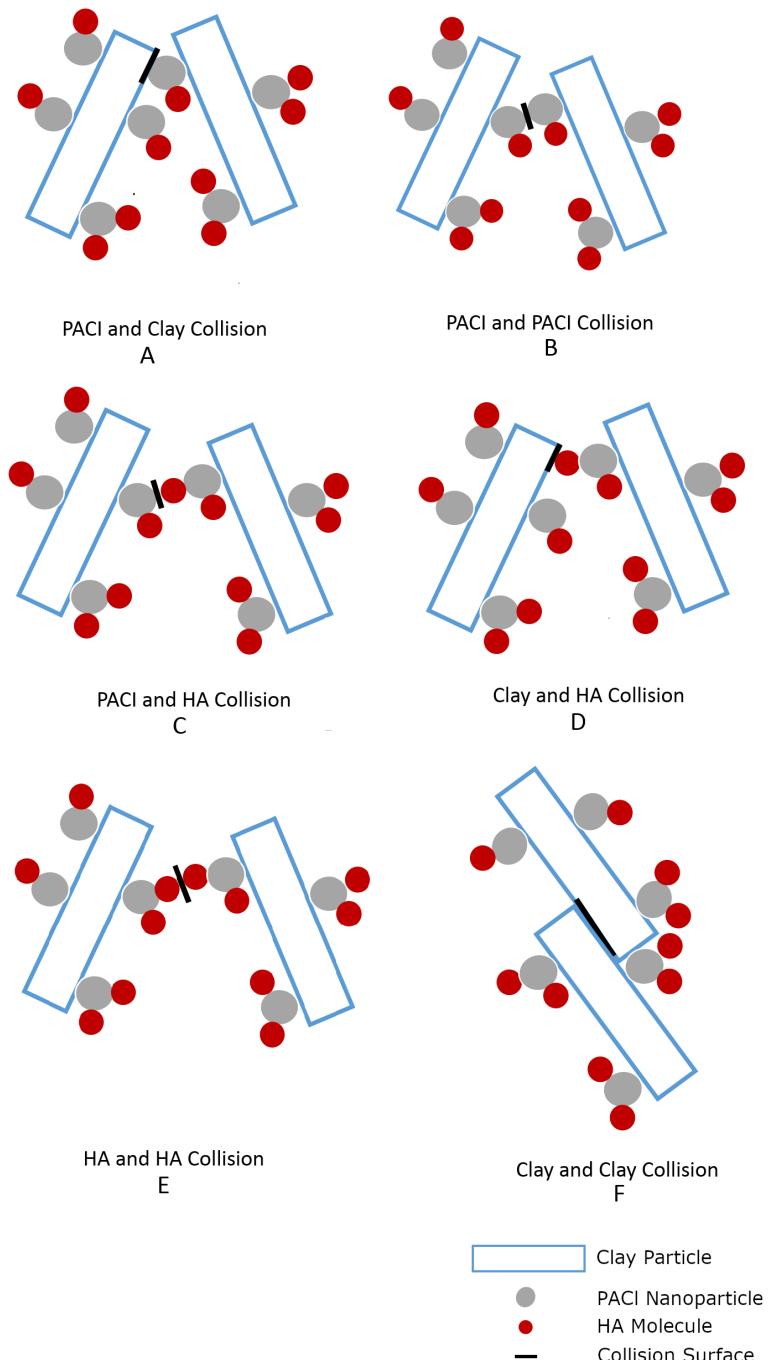


Fig. 22.12: Modes of collision between particles during flocculation.

The three types of surfaces (PACl, humic acid, clay) have 6 (3!) potential interactions as illustrated in Fig. 22.12.

Of these interactions considered in the model, the collisions that will result in attachment are assumed to involve at least one PACl nanoparticle surface (Fig. 22.12 A, B, C). The attachment efficiency is hypothesized to be the sum of probability of these three types of collisions, formally expressed as

$$\bar{\alpha} = \bar{\alpha}_{PACl-Clay} + \bar{\alpha}_{PACl-PACl} + \bar{\alpha}_{HA-PACl} \quad (22.42)$$

where the subscripts define the two surfaces that are interacting. The overbars indicate that all of these represent mean probabilities for an entire suspension rather than the probabilities for specific particles.

The probability of a clay surface colliding with a PACl surface (Fig. 22.12 A) is equal to twice the probability that the first surface is clay ($1 - \bar{\Gamma}_{PACl-Clay}$) and the second surface is the PACl surface of a PACl-HA nanoaggregate ($(1 - \bar{\Gamma}_{HA-PACl}) \bar{\Gamma}_{PACl-Clay}$), since either of two colliding particles could provide the clay surface or the PACl surface,

$$\bar{\alpha}_{PACl-Clay} = 2(1 - \bar{\Gamma}_{PACl-Clay}) [(1 - \bar{\Gamma}_{HA-PACl}) \bar{\Gamma}_{PACl-Clay}] \quad (22.43)$$

The probability of a collision between the PACl surfaces of two PACl-HA nanoaggregates ($(1 - \bar{\Gamma}_{HA-PACl}) \bar{\Gamma}_{PACl-Clay}$) (Fig. 22.12 B) is given by

$$\bar{\alpha}_{PACl-PACl} = [(1 - \bar{\Gamma}_{HA-PACl}) \bar{\Gamma}_{PACl-Clay}]^2 \quad (22.44)$$

The probability of a collision between a PACl surface of a PACl-HA nanoaggregate ($(1 - \bar{\Gamma}_{HA-PACl}) \bar{\Gamma}_{PACl-Clay}$) and an HA surface of a PACl-HA nanoaggregate ($\bar{\Gamma}_{HA-PACl} \bar{\Gamma}_{PACl-Clay}$) (Fig. 22.12 C), or vice versa, is given by

$$\bar{\alpha}_{HA-PACl} = 2 [\bar{\Gamma}_{PACl-Clay} (1 - \bar{\Gamma}_{HA-PACl})] [\bar{\Gamma}_{HA-PACl} \bar{\Gamma}_{PACl-Clay}] \quad (22.45)$$

where the factor of 2 accounts for the possibility that either colliding particle could contribute either surface type.

The model accounting for the presence of humic acids is modified from the Pennock et al. (2018) model by redefining the attachment efficiency, $\bar{\alpha}$, using Eq. 14 to account for the presence of humic acid.

The physical properties of humic acid vary with composition. The diameter of humic acid macromolecules is estimated to range from 4 nm to 110 nm (Osterberg, 1993). Because of the variation in the size of humic acid macromolecules, the characteristic diameter of the humic acid macromolecules was used as a fitting parameter. Thus, there are two adjustable model parameters, k (Equation : eq : 'eq_AquaClaraFlocculationModel'), which accounts for the sedimentation capture velocity, and d_{HA} , which accounts for coagulant precipitate surface coverage by humic acid. These parameters were fit to results from observations taken with an influent turbidity of 50 NTU; the model was then validated by independently predicting results from experiments with an influent turbidity of 100 NTU.

22.2.4 Results

The results from 60 experiments, transformed by (22.38), are shown in Fig. 22.13 for an inflow turbidity of 50 NTU with PACl doses ranging from 0.53 to 2.65 mg/L as Al and humic acid concentration ranging from 0 to 15 mg/L. A capture velocity of 0.120 mm/s was used in the experiments, which is a conservatively designed lamellar settler capture velocity (Willis, 1978). Experiments were replicated for each combination of humic acid and PACl dose.

The data show that increased coagulant dose is positively correlated with turbidity removal. The effluent turbidity was greatly increased by the presence of humic acid. Also shown is a model fit using the AquaClara flocculation model given by Pennock et al. (2018). As shown, the model can fit the performance of the 0 mg/L HA data and even the 3 mg/L HA data reasonably well, but increasing doses of humic acid decrease performance appreciably, necessitating a modification to the original model.

To apply the modified model to the raw data, the data points with 0 mg/L HA were fit by k , since their performance was not influenced by d_{HA} , resulting in $k = 0.16$. Then, the remaining data were fit using d_{HA} (with the previously

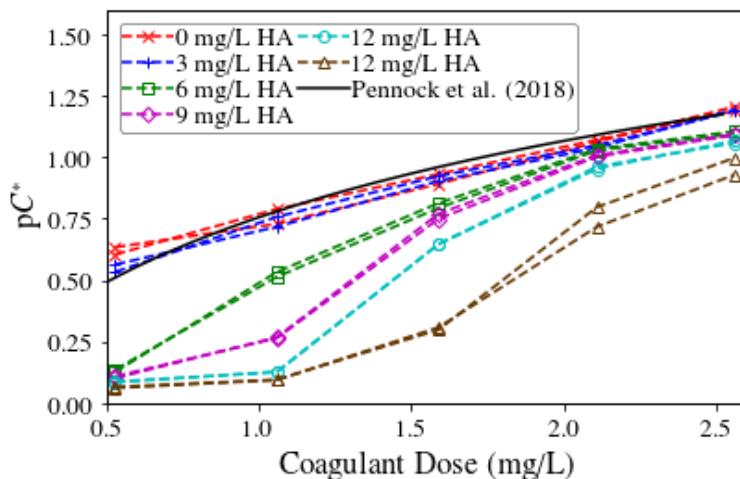


Fig. 22.13: pC^* as a function of coagulant dose for 50 NTU influent turbidity.

determined k value) to minimize the sum squared error, resulting in $d_{HA} = 77\text{nm}$ with a pC^* (dimensionless) root-mean-square error, RMSE, of 0.08. Because their performance was essentially indistinguishable from the 0 mg/L HA data. Additionally, to avoid biasing the fit by data for which the coagulant dose was insufficient to overcome the effect of humic acid, data for which performance was lower than $pC^* = 0.25$ were neglected for the fitting. Fig. 22.14 shows the fit of the model to the observations for the 50 NTU experiments.

With the given fitted value of $d_{HA} = 75\text{nm}$ for the 50 NTU influent turbidity data set, the coverage of coagulant nanoparticle surfaces by humic acid ($\bar{\Gamma}_{HA-PACl}$) changed as shown in Fig. 22.15. According to Equation (22.41), the maximum number of humic acid macromolecules with a diameter of 75 nm that will fit on a PACl precipitate particle of 90 nm diameter when fully coating it is about 6. The model predicted complete coverage of the PACl nanoparticles by humic acid for low PACl concentrations, which correlated with very low observed turbidity removal efficiency.

The relationships between the three terms included in attachment efficiency are shown in Fig. 22.16. The term corresponding to collisions between a clean coagulant nanoparticle surface and clay ($\bar{\alpha}_{PACl-Clay}$) was always dominant for the experimental conditions in this dataset, and the other terms became relatively more important but still insignificant small with respect to $\bar{\alpha}_{PACl-Clay}$ with increasing coagulant dose.

The model was validated by using it to predict turbidity removal efficiency for different experimental conditions. The predicted pC^* and the measured pC^* are compared in Fig. 22.17 for an additional 60 experiments with inflow turbidity of 100 NTU, PACl doses ranging from 0.53 to 2.65 mg/L, and humic acid concentration ranging from 0 to 15 mg/L. The resulting fit is almost as good as for the 50 NTU data, with pC^* RMSE of 0.11.

When the coagulant dose in Fig. 22.14 and Fig. 22.17 was replaced with the dimensionless group $\bar{\alpha}\bar{G}_{CS}\theta\phi^{\frac{2}{3}}$, the data collapsed to a much narrower band, implying that the composite parameter, $\bar{\alpha}\bar{G}_{CS}\theta\phi^{\frac{2}{3}}$, captures a large fraction of the trends present in the data, as seen in Fig. 22.18.

In summary, the laminar flow hydraulic flocculation model of Pennock et al. (2018) was modified to incorporate the effects of humic acid with the addition of a single fitting parameter: a characteristic dimension of the humic acid macromolecules. The required coagulant dose can be predicted based on the flocculator parameters, humic acid characteristic size and concentration, and influent turbidity. The addition of humic acid to the flocculation model increases the model applicability since natural organic matter is found in all surface and ground waters and influences the coagulant dose needed for effective turbidity removal.

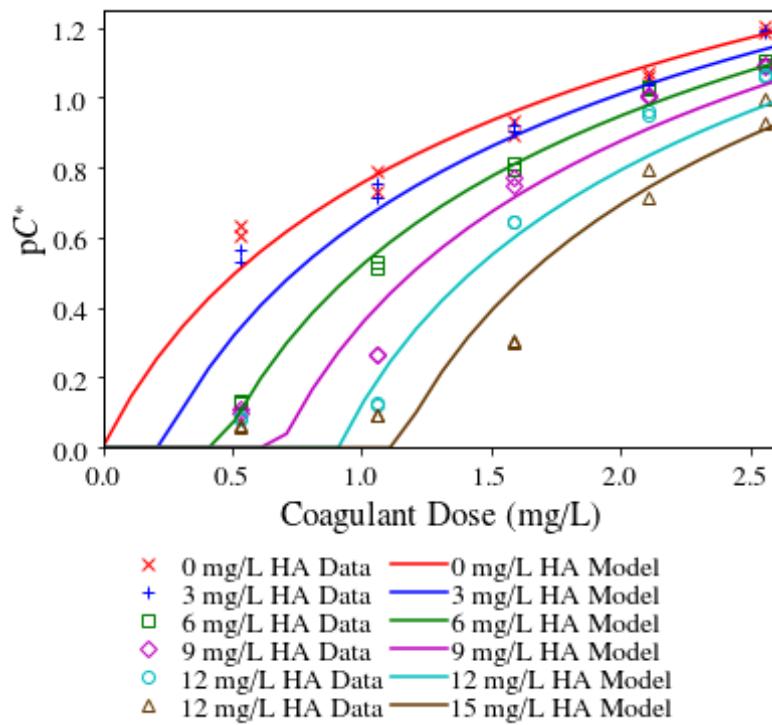
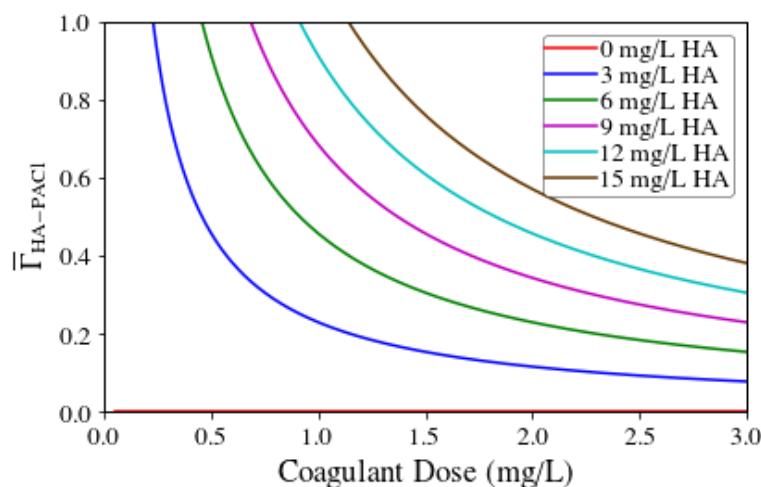
Fig. 22.14: Model fit for pC^* as function of coagulant dose for 50 NTU raw water turbidity.

Fig. 22.15: Coverage of coagulant surface by humic acid as a function of coagulant dose.

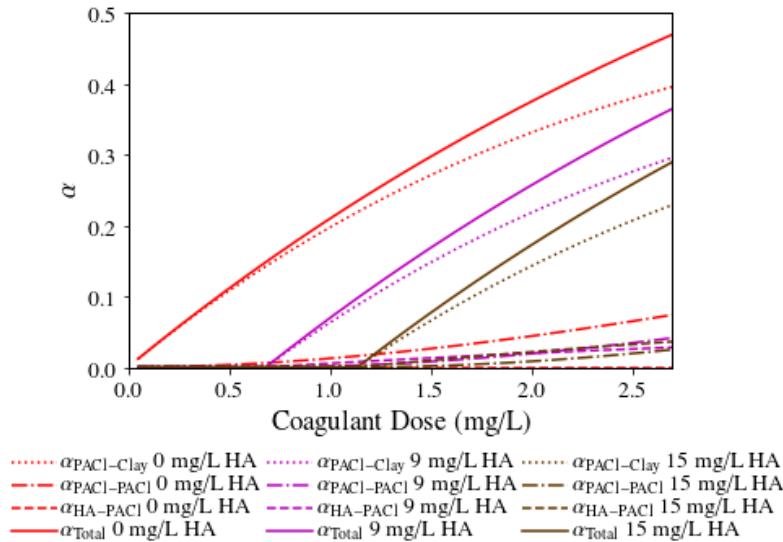


Fig. 22.16: Attachment efficiency as a function of coagulant dose.

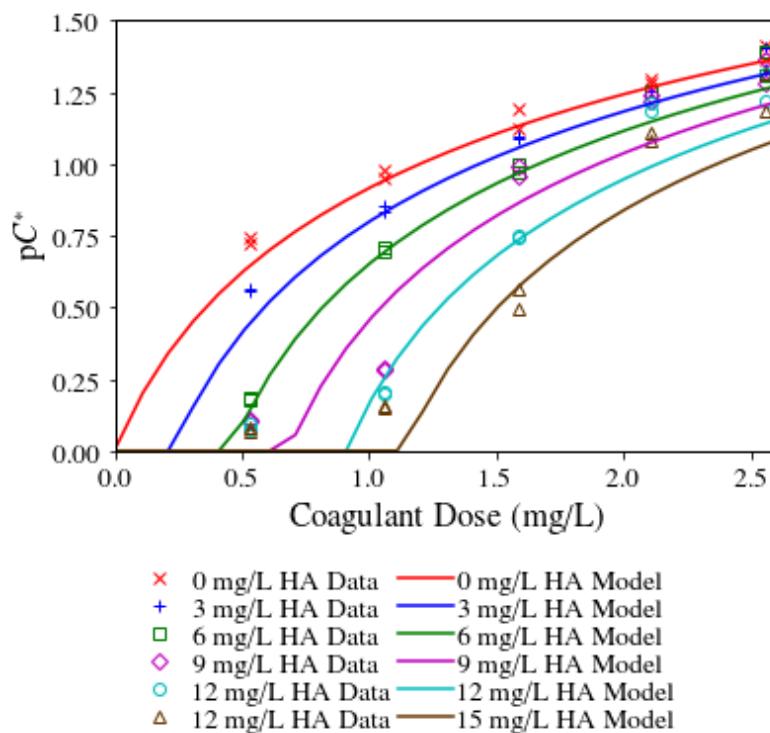


Fig. 22.17: Comparison graph between predicted data and observed data for 100 NTU influent turbidity.

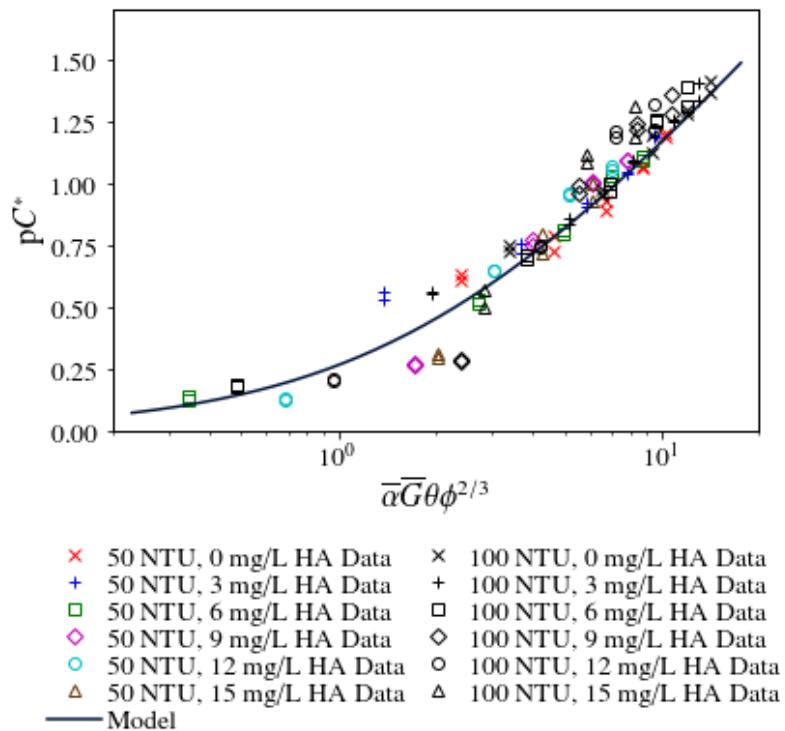


Fig. 22.18: Model fit of 50 and 100 NTU data for pC^* as function of effective collision potential. The data plotted include two replicates for each experiment.

22.2.5 Discussion

For the range of experimental conditions considered in the research, the observed influence of humic acid on flocculation performance could be explained by the fractional coverage of the coagulant nanoparticle surfaces by humic acid macromolecules. It is noteworthy that under the experimental conditions, the predictive success of the model was achieved without incorporating the charges of colloids, coagulant, and humic acids. The reader is cautioned that the observations and predictions were obtained with one test particle, one coagulant, and one form of DOM in the mixed electrolyte represented by Cornell tap water, kept within the narrow pH range where coagulant precipitation is very favorable. While the experimental pH favored PACl precipitation, pH-dependent PACl solubility is accounted for in the model with

$$N_{perClay} = \frac{[C_{PACl} - C_{PACl(aq)}] V_P \rho_P}{\frac{\pi}{6} d_{PACl}^3 \rho_{PACl} C_0} \quad (22.46)$$

where $N_{perClay}$ is the number of precipitated coagulant aggregates per clay particle, $C_{PACl(aq)}$ is the fraction of the coagulant dose that has remained in solution after precipitation using the PACl solubility observed by Van Benschoten and Edzwald (1990), and V_P is the volume of a single clay platelet (Swetland et al., 2014). Within the model, $N_{perClay}$ is used to calculate \bar{a} , since it is a component of the calculation for $\bar{\Gamma}_{PACl-Clay}$:

$$\bar{\Gamma}_{PACl-Clay} = 1 - e^{-\frac{d_{PACl}^2}{SA_{Clay}} N_{perClay} R_{Clay}} \quad (22.47)$$

where SA_{Clay} is the surface area of the surface area of the suspended clay particles and R_{Clay} is the fraction of the available surface area in the reactor (including the surface area of reactor walls) that belongs to suspended clay particles (Swetland et al., 2014).

The solubility of humic acid also is highly pH-dependent, and additional experimental results are needed to test the applicability of the model approach as a function of varying pH. The experimental conditions were designed to keep the pH relatively constant, and the pH change in the experiments was small (7.5 ± 0.3).

The model considered flocculation in the presence of humic acid as a two-step process. Firstly, humic acid macromolecules attached to precipitated coagulant nanoparticles. Then, the partially-coated coagulant nanoaggregates could bind to clay and reactor wall surfaces. Humic acid and coagulant nanoparticles were treated as spheres when estimating the attachment efficiency based on surface coverage and probability. The diameter of precipitated PACl nanoparticles was experimentally measured to be 90 nm (Garland, 2016), and a humic acid macromolecule diameter of 77 nm best fit the observations. Wall loss of coagulant precipitates with humic acid nanoaggregates was considered while direct wall loss of humic acid macromolecules was not considered.

The characteristic humic acid dimension, d_{HA} , has a physical meaning, with the fitted value, 77 nm, falling within the range (4-110 nm) reported by Osterberg (1993), and the model fits are well correlated to the observations. The predictive capability of the model was verified by predicting results under different experimental conditions with no additional adjustable parameters.

The flocculation model without the effects of humic acid shows that pC^* is directly proportional to the log of the effective collision potential, $\log(\bar{\alpha}\bar{G}_{CS}\theta\phi^{\frac{2}{3}})$, and this relationship is still present in the model with a modified attachment efficiency, $\bar{\alpha}$, based on clay surface coverage by coagulant nanoparticles as adjusted for the presence of humic acids.

Under experimental conditions, the modified flocculation model provides the fundamental basis for the relationship between coagulant dose, synthetic raw water clay, and humic acid concentrations. Extension to natural waters will undoubtedly require additional research.

The form of the flocculation model equation sets the interactions between raw water properties (ϕ_0), influent particle surface area (which contributes to $\bar{\Gamma}_{PACl-Clay}$), coagulant precipitate size and dose (which contributes to $\bar{\Gamma}_{PACl-Clay}$ and $\bar{\Gamma}_{HA-PACl}$), humic acid molecule size and concentration (which contribute to $\bar{\Gamma}_{HA-PACl}$), flocculator design ($\bar{G}_{CS}\theta$), and sedimentation tank design (k). In a gravity-powered water treatment plant operating at constant flow rate, the flocculator and sedimentation tank parameters are constant. An increase in concentration of humic acid causes an increase in $\bar{\Gamma}_{HA-PACl}$, which decreases pC^* but can be compensated for by increasing coagulant dose.

22.2.6 Summary

The development of a predictive model for laminar flow hydraulic flocculation of water containing clay and humic acid is described. The study results increase the flexibility and generality of the AquaClara hydraulic flocculation model, and the modified model provides insight into the mechanism by which humic acid causes a decrease in performance of coupled flocculation-sedimentation processes.

The model was able to predict independent experimental results for a different raw water turbidity with no additional adjustable parameters. Further tests should be done to fully validate the laminar-flow model including consideration of different experimental surrogates for DOM, different colloidal surfaces, alternative coagulants and varying solution compositions, including pH.

22.3 References

Amin, M., Safari, M., Maleki, A., Ghasemian, M., Rezaee, R., & Hashemi, H. (2012). Feasibility of humic substances removal by enhanced coagulation process in surface water. International Journal of Environmental Health Engineering. <http://www.ijehe.org/text.asp?2012/1/1/29/99323>

Benjamin, M. M., & Lawler, D. F. (2013). Water quality engineering: physical / chemical treatment processes. Hoboken, N.J.: Wiley.

BP-MWS, CIWS, & CUWS. (2016). Drinking Water Quality Report 2016. Ithaca, NY: Bolton Point Municipal Water System, City of Ithaca Water System, Cornell University Water System. Retrieved from <https://fcs.cornell.edu/content/water-system-updates-and-water-quality-reports>

Camp, T. R. (1953). Flocculation and Flocculation Basins. American Society of Civil Engineers.

- Chow, C. W. K., Fabris, R., Leeuwen, J. van, Wang, D., & Drikas, M. (2008). Assessing Natural Organic Matter Treatability Using High Performance Size Exclusion Chromatography. *Environmental Science & Technology*, 42(17), 6683–6689. <https://doi.org/10.1021/es800794r>
- Cleasby, J. (1984). Is Velocity Gradient a Valid Turbulent Flocculation Parameter? *Journal of Environmental Engineering*, 110(5), 875–897. [https://doi-org.proxy.library.cornell.edu/10.1061/\(ASCE\)0733-9372\(1984\)110:5\(875\)](https://doi-org.proxy.library.cornell.edu/10.1061/(ASCE)0733-9372(1984)110:5(875))
- Davis, J. A. (1982). Adsorption of natural dissolved organic matter at the oxide/water interface. *Geochimica et Cosmochimica Acta*, 46(11), 2381–2393. [https://doi.org/10.1016/0016-7037\(82\)90209-5](https://doi.org/10.1016/0016-7037(82)90209-5)
- Fosso-Kankeu, E., Webster, A., Ntwampe, I. O., & Waanders, F. B. (2017). Coagulation/Flocculation Potential of Polyaluminium Chloride and Bentonite Clay Tested in the Removal of Methyl Red and Crystal Violet. *Arabian Journal for Science and Engineering*, 42(4), 1389–1397. <https://doi.org/10.1007/s13369-016-2244-x>
- Granger, R. A. (1995). Fluid Mechanics. New York: Dover Publications.
- Hu, C., Hu, X., Wang, L., Qu, J., & Wang, A. (2006). Visible-Light-Induced Photocatalytic Degradation of Azodyes in Aqueous AgI/TiO₂ Dispersion. *Environmental Science & Technology*, 40(24), 7903–7907. <https://doi.org/10.1021/es061599r>
- Hua, G., & Reckhow, D. A. (2008). Relationship between Brominated THMs, HAAs, and Total Organic Bromine during Drinking Water Chlorination. In T. Karanfil, S. W. Krasner, P. Westerhoff, & Y. Xie (Eds.), Disinfection By-Products in Drinking Water (Vol. 995, pp. 109–123). Washington, DC: American Chemical Society. <https://doi.org/10.1021/bk-2008-0995.ch008>
- Integrated design of water treatment facilities: Susumu Kawamura. John Wiley & Sons, Inc.: New York, NY 1991. (pp. 658, ISBN 0-471-61591-9) \$69.95 hardcover. (1992). *Waste Management*, 12(1), 101. [https://doi.org/10.1016/0956-053X\(92\)90024-D](https://doi.org/10.1016/0956-053X(92)90024-D)
- Ives, K. J. (1968). Theory of operation of sludge blanket clarifiers. *Proceedings of the Institution of Civil Engineers*, 39(2), 243–260. <https://doi.org/10.1680/iicep.1968.8090>
- Jarvis, P., Jefferson, B., Gregory, J., & Parsons, S. A. (2005). A review of floc strength and breakage. *Water Research*, 39(14), 3121–3137. <https://doi.org/10.1016/j.watres.2005.05.022>
- Kundu, P. K., & Cohen, I. M. (2008). Fluid mechanics. Amsterdam; Boston: Academic Press. Letterman, R. D. (1999). Water quality and treatment: a handbook of community water supplies (5th ed.). New York: McGraw-Hill.
- Matilainen, A., Vepsäläinen, M., & Sillanpää, M. (2010). Natural organic matter removal by coagulation during drinking water treatment: A review. *Advances in Colloid and Interface Science*, 159(2), 189–197. <https://doi.org/10.1016/j.cis.2010.06.007>
- Marhaba, T. F., Pu, Y., & Bengraine, K. (2003). Modified dissolved organic matter fractionation technique for natural water. *Journal of Hazardous Materials*, 101(1), 43–53. [https://doi.org/10.1016/S0304-3894\(03\)00133-X](https://doi.org/10.1016/S0304-3894(03)00133-X)
- OMelia, C. R. (1972). Coagulation and flocculation. In W. J. Weber (Ed.), Physicochemical processes for water quality control. New York: Wiley-Interscience.
- Österberg, R., Lindqvist, I., & Mortensen, K. (1993). Particle Size of Humic Acid. *Soil Science Society of America Journal*, 57(1), 283–285. <https://doi.org/10.2136/sssaj1993.03615995005700010048x>
- Pennock, William H., Weber-Shirk, Monroe, & Lion, Leonard W. (2018). A Hydrodynamic and Surface Coverage Model Capable of Predicting Settled Effluent Turbidity Subsequent to Hydraulic Flocculation. *Environmental Engineering Science*, 35(12). <https://doi.org/10.1089/ees.2017.0332>
- Schulz, C. R., & Okun, D. A. (1984). Surface water treatment for communities in developing countries. New York: Wiley.
- Sharp, E. L., Jarvis, P., Parsons, S. A., & Jefferson, B. (2006). Impact of fractional character on the coagulation of NOM. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 286(1–3), 104–111. <https://doi.org/10.1016/j.colsurfa.2006.03.009>

Sigma-Aldrich. (2014). Humic acid sodium salt (H16752) (Safety Data Sheet) (p. 7). St. Louis, MO. Retrieved from <https://www.sigmaaldrich.com/MSDS/MSDS/DisplayMSDSPage.do?country=US&language=en&productNumber=H16752&brand=ALDRICH&PageToGoToURL=https%3A%2F%2Fwww.sigmaaldrich.com%2Fcatalog%2Fproduct%2Faldrich%2Fh16752%3Flang%3Den>

Soh, Y. C., Roddick, F., & Leeuwen, J. van. (2008). The impact of alum coagulation on the character, biodegradability and disinfection by-product formation potential of reservoir natural organic matter (NOM) fractions. *Water Science and Technology*; London, 58(6), 1173–1179. <http://dx.doi.org/10.2166/wst.2008.475>

Swetland, K. A., Weber-Shirk, M. L., & Lion, L. W. (2013). Influence of Polymeric Aluminum Oxyhydroxide Precipitate-Aggregation on Flocculation Performance. *Environmental Engineering Science*, 30(9), 536–545. <https://doi.org/10.1089/ees.2012.0199>

Swetland, K. A., Weber-Shirk, M. L., & Lion, L. W. (2014). Flocculation-Sedimentation Performance Model for Laminar-Flow Hydraulic Flocculation with Polyaluminum Chloride and Aluminum Sulfate Coagulants. *Journal of Environmental Engineering*, 140(3), 04014002. [https://doi.org/10.1061/\(ASCE\)EE.1943-7870.0000814](https://doi.org/10.1061/(ASCE)EE.1943-7870.0000814)

Tse, I. C., Swetland, K., Weber-Shirk, M. L., & Lion, L. W. (2011). Method for quantitative analysis of flocculation performance. *Water Research*, 45(10), 3075–3084. <https://doi.org/10.1016/j.watres.2011.03.021>

Van Benschoten, J. E., & Edzwald, J. K. (1990). Chemical aspects of coagulation using aluminum saltsI. Hydrolytic reactions of alum and polyaluminum chloride. *Water Research*, 24(12), 1519–1526. [https://doi.org/10.1016/0043-1354\(90\)90086-L](https://doi.org/10.1016/0043-1354(90)90086-L)

Weber-Shirk, M. L. (2016). ProCoDA: An Automated Method for Testing Process Parameters. Retrieved October 30, 2015, from <https://confluence.cornell.edu/display/AGUACLARA/ProCoDA>

Weber-Shirk, M. L., & Lion, L. W. (2010). Flocculation model and collision potential for reactors with flows characterized by high Peclet numbers. *Water Research*, 44(18), 5180–5187. <https://doi.org/10.1016/j.watres.2010.06.026>

Willis, R. M. (1978). Tubular SettlersA Technical Review. *Journal (American Water Works Association)*, 70(6), 331–335.

CHAPTER
TWENTYTHREE

FLOCCULATION DESIGN

This chapter will supplement the lectures by going into further detail.

Equations, universal constants, and other helpful goodies can be found in the [aguaclara code base](#). Most equations and constants you find in these summary sheets will already have been coded and will be shown here in the following format:

Variable: `pc.gravity`

Function: `pc.area_circle(DiamCircle)`.

The letters before the `.`, in this case `pc`, indicate the file within aguaclara where the variable or function can be found. In the examples above, `pc.gravity` and `pc.area_circle(DiamCircle)` show that the variable `gravity` and function `area_circle(DiamCircle)` are located inside the `physchem.py` (`pc`) file. You are strongly recommended to look up any aguaclara equations before using them, even if they are given here in this summary sheet. This is because each equation has comments in its original file describing what the specific conditions are to using it.

Important Note: This chapter introduces uncertainty and empirical design. Some of the parameters used to design AguaClara flocculators are based on what has been shown to work in the field, as opposed to having been derived scientifically. To make sure that the reader is aware of these concepts and parameters that dont yet have a thorough basis in research, they will be highlighted in red when they appear.

23.1 Hydraulic Flocculators, the AguaClara Approach

23.1.1 Important Terms

1. Collision potential
2. Energy dissipation rate
3. Baffle
4. Baffle module
5. Baffle space
6. Obstacle

23.1.2 Important Equations

1. Minor Loss equation

23.2 Introduction to Hydraulic Flocculation

The reason that flocculation is widely used in water treatment is because of sedimentation. Sedimentation is the process that actually removes particles like clay, dirt, organic matter, and bacteria from water. As you learned in the *introduction on treatment trains*, sedimentation is the process of particles falling because they have a higher density than the water, and its governing equation is:

$$\bar{v}_t = \frac{D_{particle}^2 g}{18\nu} \frac{\rho_p - \rho_w}{\rho_w} \quad (23.1)$$

Such that:

\bar{v}_t = terminal velocity of a particle, its downwards speed if it were in quiescent (still) water

$D_{particle}$ = particle diameter

ρ = density. The p subscript stands for particle, while w stands for water

To increase \bar{v}_t and make sedimentation more efficient, flocculation aims to increase the diameter d of the particles. This is done by applying a coagulant to the dirty water and helping the coagulant to stick evenly to all particles during Rapid Mix (**DOUBLE CHECK THAT THIS IS IN RAPID MIX ONCE RAPID MIX IS WRITTEN**). Being covered in coagulant allows the particles to collide, merge, and grow bigger during flocculation. Our goal in designing a flocculator is to facilitate particle collisions. How can we do this?

23.2.1 Collision Potential, $G_{CS}\theta$, and Energy Dissipation Rate, ε

Collision potential ($G_{CS}\theta$) is a term with a very straightforward name. It represents the magnitude of potential particle collisions in a fluid. It is a *dimensionless* parameter which is often used as a performance metric for flocculators; big $G_{CS}\theta$ values indicate lots of collisions (good) while small values indicate fewer collisions (not so good). AquaClara flocculators usually aim for a collision potential of $(G_{CS}\theta) = 37,000$, which has worked well in AquaClara plants historically. However, this value may change as research continues. The value for collision potential is obtained by multiplying G_{CS} , a parameter for average fluid shear with units of $\frac{1}{[T]}$, and θ , the residence time of water in the flocculator, with units of $[T]$. θ is intuitive to measure, calculate, and understand. G_{CS} is a bit more difficult. First, an intuitive explanation. See Fig. 23.1, which shows the velocity profile of flowing water.

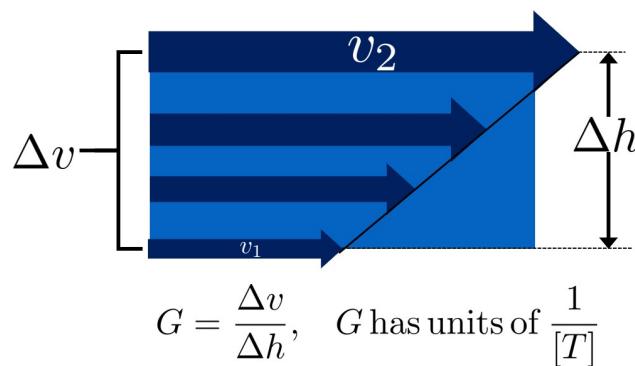


Fig. 23.1: The velocity profile of flowing fluid with uniform shear

G measures the magnitude of shear by using the velocity gradient of a fluid in space, $\frac{\Delta \bar{v}}{\Delta h}$. This is essentially the same as the $\frac{\delta u}{\delta y}$ term in fluid mechanics, which is found in the ubiquitous [fluid-shear problem](#) as sourced from [here](#).

\bar{G} represents the average $\frac{\Delta \bar{v}}{\Delta h}$ for the entire water volume under consideration, and is the parameter we will be using from now on. Unfortunately, it is unrealistic to measure $\frac{\Delta \bar{v}}{\Delta h}$ for every parcel of the water in our flocculator and take an average. We need to approximate \bar{G} using measurable parameters.

The parameter that serves as the basis for obtaining G_{CS} is ε , which represents the **energy dissipation** rate of a fluid *normalized by its mass*. The units of ε are Watts per kilogram:

$$\varepsilon = \left[\frac{W}{Kg} \right] = \left[\frac{J}{s \cdot Kg} \right] = \left[\frac{N \cdot m}{s \cdot Kg} \right] = \left[\frac{kg \cdot m \cdot m}{s^2 \cdot s \cdot Kg} \right] = \left[\frac{m^2}{s^3} \right] = \left[\frac{[L]^2}{[T]^3} \right] \quad (23.2)$$

There are at least two ways to think about ε . One is through G . Imagine that a fluid has *no viscosity*; there is no internal friction caused by fluid flow. No matter how high G becomes, no energy is dissipated. Now image a honey, which has a very high viscosity. Making honey flow fast requires a lot of energy over a short period of time, which means a high energy dissipation rate. This explanation allows us to understand the equation for ε in terms of G and ν . See this [textbook](#) for the derivation of the following equation:

$$\varepsilon = \nu G^2 \quad (23.3)$$

Which means we can solve for G :

$$G = \sqrt{\frac{\varepsilon}{\nu}} \quad (23.4)$$

Energy dissipation rate is, fortunately, easier to determine than collision potential. This is due to the second way to think about ε , which is using head loss. In any reactor, a flocculator in this case, the total energy dissipated is simply the head loss, h_L . The amount of time required to dissipate that energy is the residence time of the water in the reactor, θ . Accounting for the fact that head energy is due to gravity g , we have all the parameters needed to determine another equation for energy dissipation rate:

$$\bar{\varepsilon} = \frac{gh_L}{\theta} \quad (23.5)$$

Note that the equation above is for $\bar{\varepsilon}$, not ε . Since the head loss term we are using, h_L , occurs over the entire reactor, it can only be used to find an average energy dissipation rate for the entire reactor. Combining the equations above, $G = \sqrt{\frac{\varepsilon}{\nu}}$ and $\bar{\varepsilon} = \frac{gh_L}{\theta}$, we can get an equation for G_{CS} in terms of easily measureable parameters:

$$G_{CS} = \sqrt{\frac{gh_L}{\nu\theta}} \quad (23.6)$$

We can use this to obtain a final equation for collision potential of a reactor:

$$G_{CS}\theta = \sqrt{\frac{gh_L\theta}{\nu}} \quad (23.7)$$

Note: When we say $G\theta$ we are almost always referring to $G_{CS}\theta$.

23.2.2 Generating Head Loss with Baffles

What are Baffles?

Now that we know how to measure collision potential with head loss, we need a way to actually generate head loss. While both major or minor losses can be the design basis, it generally makes more sense to use major losses only for very low-flow flocculation (lab-scale) and minor losses for higher flows, as flocculation with minor losses tends to be more space-efficient. Since this book focuses on town and village-scale water treatment (5 L/S to 120 L/S), we will use minor losses as our design basis.

To generate minor losses, we need to create flow expansions. AquaClara does this with **baffles**, which are obstructions in the channel of a flocculator to force the flow to switch directions by 180°. Baffles in AquaClara plants are plastic sheets, and all of the baffles in one flocculator channel are connected to form a **baffle module**. Fig. 23.2 shows an AquaClara flocculator and Fig. 23.3 shows the assembly of a baffle module.

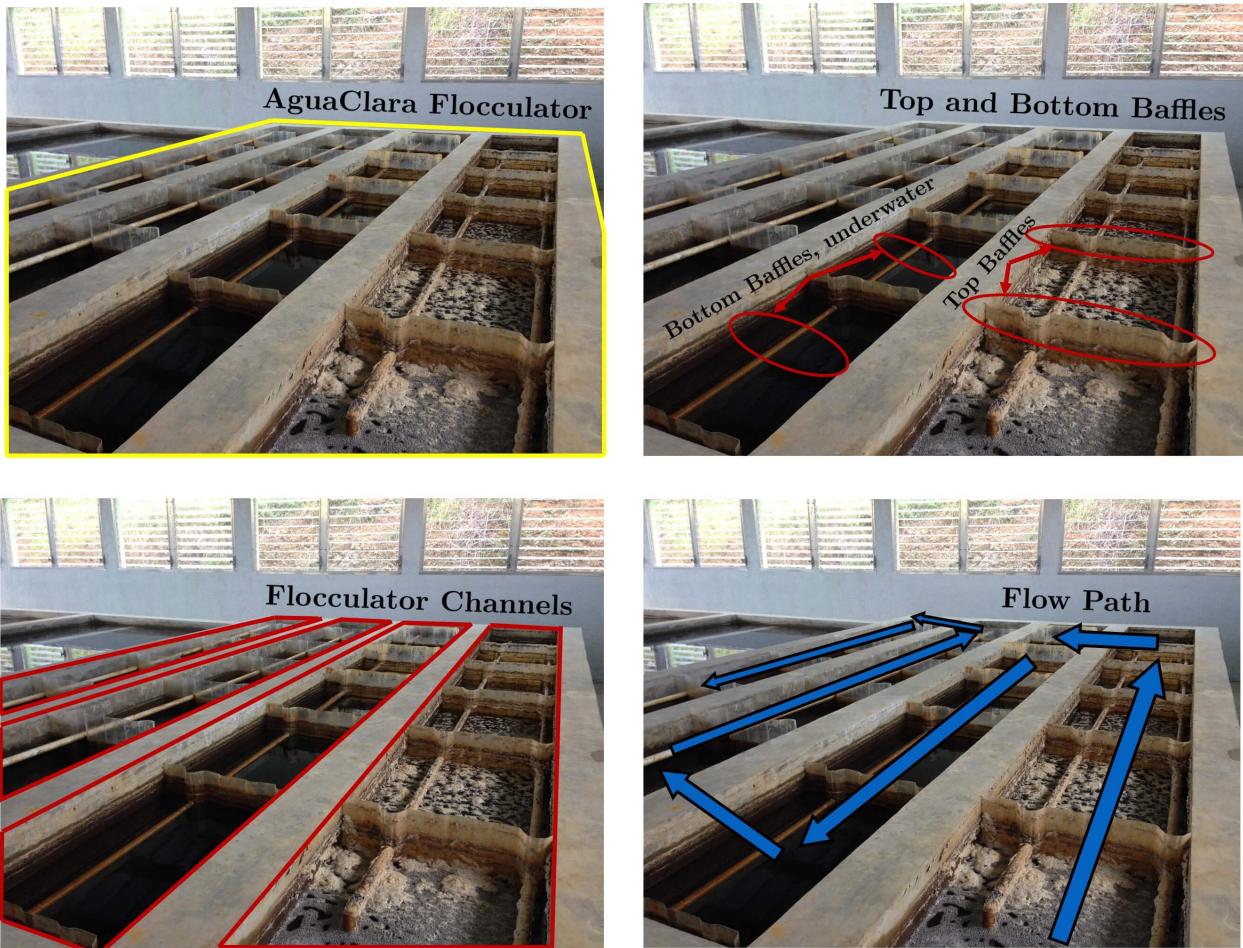


Fig. 23.2: Clockwise from the top left the images show: the outline of the entire flocculator, some top and bottom baffles in the channels, the 4 flocculator channels in this flocculator, and the flow path of water through the flocculator



Fig. 23.3: Before being inserted into the flocculator channel, the baffle module is constructed as a unit as shown here.

AquaClara flocculators, like the one pictured above, are called **vertical hydraulic flocculators** because the baffles force the flow vertically up and down. If the baffles were instead arranged to force the flow side-to-side, the flocculator would be called a **horizontal hydraulic flocculator**. AquaClara uses vertical flocculators because they are more efficient when considering plant area. They are deeper than horizontal flocculators, which allows them to have a smaller [plan-view area](#) and thus to be cheaper.

Finding the Minor Loss of a Baffle

Before beginning this section, it is important to understand how water flows through a baffled flocculator. This flow path is shown in Fig. 23.4. Take note of the thin red arrows; they indicate the compression of the flow around a baffle.

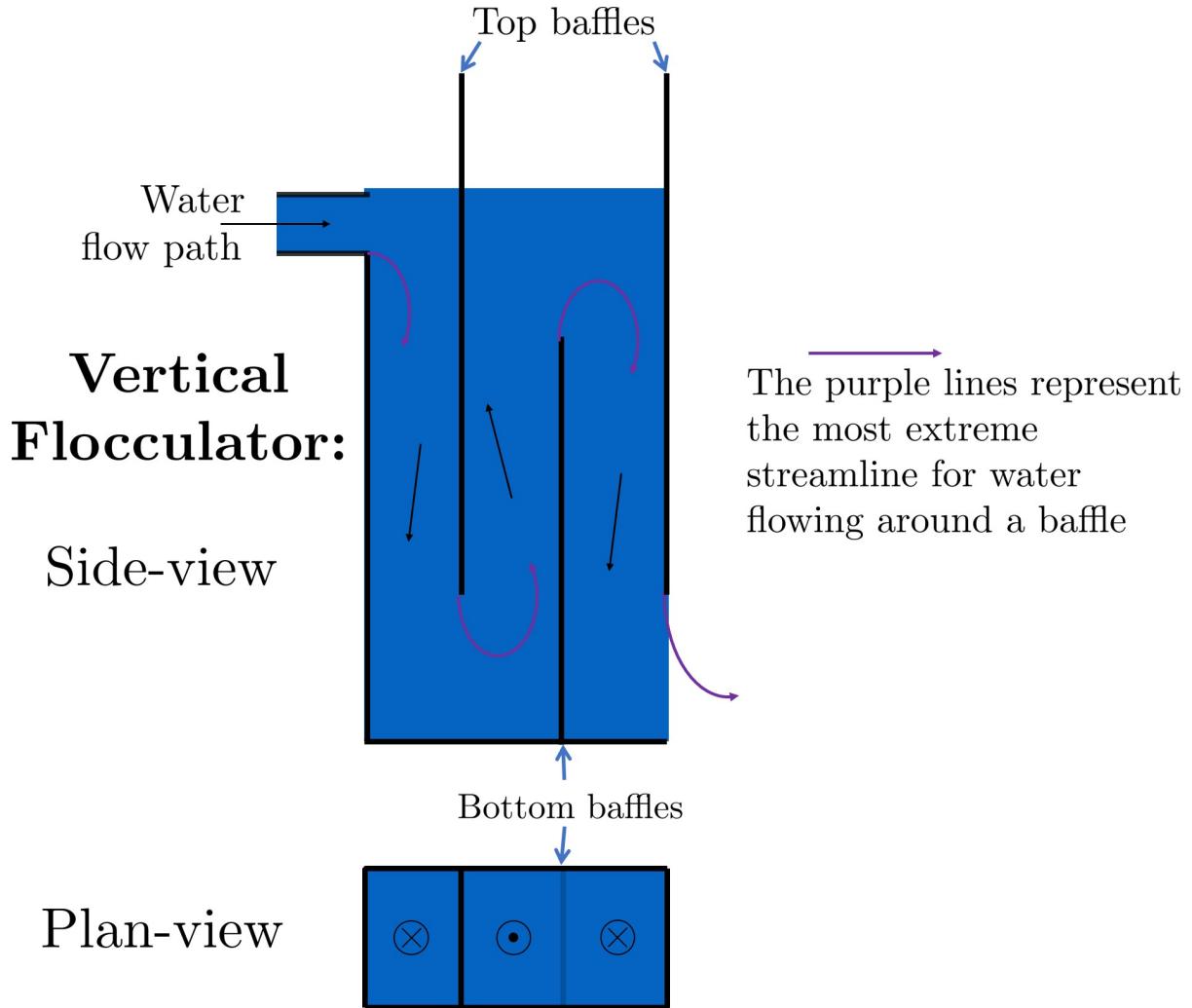


Fig. 23.4: Flow path through a vertical flow hydraulic flocculator

Since baffles are the source of head loss via minor losses, we need to find the minor loss coefficient of one baffle if we want to be able to quantify its head loss. To do this, we apply fluid mechanics intuition and check it against a computational fluid dynamics (CFD) simulation. Flow around a 90° bend has a vena contracta value of around $\Pi_{vc} = 0.62$. Flow around a 180° bend therefore has a value of $\Pi_{vc, baffle} = \Pi_{vc}^2 = 0.384$. This number is roughly confirmed with CFD, as shown in the image below.

We can therefore state with reasonable accuracy that, when most contracted, the flow around a baffle goes through 38.4% of the area it does when expanded, or $A_{contracted} = \Pi_{vc, baffle} A_{expanded}$. Through the :ref:`third form`

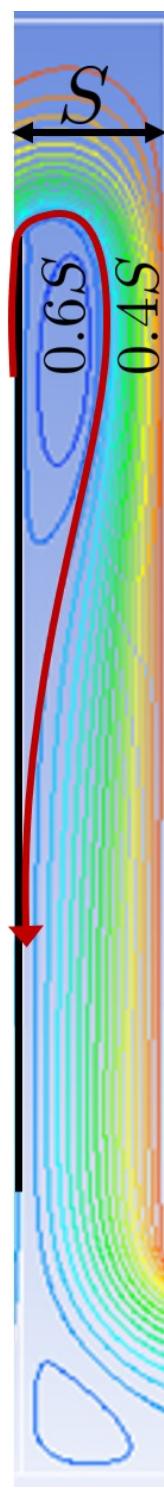


Fig. 23.5: The 180° bend at the end of a baffle results in a dramatic flow contraction with all of the flow passing through less than 40% of the space between the baffles.

of the minor loss equation <heading_minor_losses>, $h_e = K \frac{\bar{v}_{out}^2}{2g}$ and its definition of the minor loss coefficient, $K = \left(\frac{A_{out}}{A_{in}} - 1 \right)^2$, we can determine a k for flow around a single baffle:

$$\begin{aligned} K_{baffle} &= \left(\frac{A_{expanded}}{A_{contracted}} - 1 \right)^2 \\ K_{baffle} &= \left(\frac{A_{expanded}}{\Pi_{vc, baffle} A_{expanded}} - 1 \right)^2 \\ K_{baffle} &= \left(\frac{1}{0.384} - 1 \right)^2 \\ K_{baffle} &= 2.56 \end{aligned} \quad (23.8)$$

This K_{baffle} has been used to design many flocculators in Aguacalara plants. However, its value has not yet been rigorously tested for Aguacalara plants in the field. Therefore it might actually deviate from 2.56. Research and testing the K of a baffle in an Aguacalara plant is ongoing, but for now the designs made under the assumption that $K_{baffle} = 2.56$ are functioning very well in Aguacalara plants. Although research has been done by many academics on the minor loss coefficient, including [this paper by Haarhoff in 1998](#) (DOI: 10.2166/aqua.1998.20), the K_{baffle} values found are context dependent and empirically based. For Aguacalara flocculator parameters, literature suggest a K_{baffle} value between 2.5 and 4.

23.2.3 Flocculator Efficiency

When designing an effective and efficient flocculator, there are two main problems that we seek to avoid:

1. Having certain sections in the flocculator with such high local G values that our big, fluffy flocs are sheared apart into smaller flocs.
2. Having dead space. Dead space means volume within the flocculator that is not being used to facilitate collisions. Dead space occurs after the flow has fully expanded from flowing around a baffle and before it reaches the next baffle.

Fortunately for us, both problems can be quantified with a single ratio:

$$\Pi_{G_{CS}}^{G_{Max}} = \frac{G_{Max}}{G_{CS}} \quad (23.9)$$

High values of $\Pi_{G_{CS}}^{G_{Max}}$ occur when one or both of the previous problems is present. If certain sections in the flocculator have very high local G values, then G_{Max} becomes large. If the flocculator has a lot of dead space, then G_{CS} becomes small. Either way, $\Pi_{G_{CS}}^{G_{Max}}$ becomes larger.

Note: Recall the relationship between G and ε : $G = \sqrt{\frac{\varepsilon}{\nu}}$. From this relationship, we can see that $G \propto \sqrt{\varepsilon}$. Thus, by defining $\Pi_{G_{CS}}^{G_{Max}}$, we can also define a ratio for Max to average energy dissipation rate:

$$\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}} = \left(\Pi_{G_{CS}}^{G_{Max}} \right)^2 \quad (23.10)$$

Therefore, by making our $\Pi_{G_{CS}}^{G_{Max}}$ as small as possible, we can be sure that our flocculator is efficient, and we no longer have to account for the previously mentioned problems. [A paper by Haarhoff and van der Walt in 2001](#) (DOI: 10.2166/aqua.2001.0014) uses CFD to show that the minimum $\Pi_{G_{CS}}^{G_{Max}}$ attainable in a hydraulic flocculator is $\Pi_{G_{CS}}^{G_{Max}} = \sqrt{2} \approx 1.4$, which means that $\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}} = \left(\Pi_{G_{CS}}^{G_{Max}} \right)^2 \approx 2$. So how do we optimize an Aguacalara flocculator to make sure $\Pi_{G_{CS}}^{G_{Max}} = \sqrt{2}$?

We define and optimize a performance metric:

$$\frac{H_e}{S} = \Pi_{H_e S} \quad (23.11)$$

Where H_e is the distance between flow expansions in the flocculator and S is the spacing between baffles. For now, H_e is approximated as the height of water in the flocculator.

Since G_{Max} is determined by the fluid mechanics of flow around a baffle, our main concern is eliminating dead space in the flocculator. We do this by placing an upper limit on $\frac{H_e}{S}$. To determine this upper limit, we need to find the distance it takes for the flow to fully expand after it has contracted around a baffle. We base this on the rule of thumb for flow expansion, a jet doubles its initial diameter/length once it travels 10 times the distance of its original diameter/length. If this is confusing, refer to the equation and image below:

$$\frac{x}{10} = D - D_0 \quad (23.12)$$

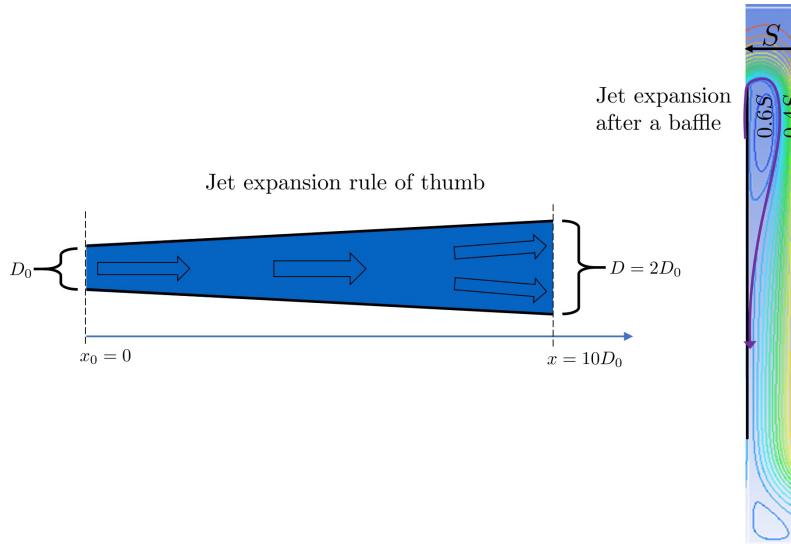


Fig. 23.6: A turbulent jet expands in width by one unit for every 10 units downstream.

Using the equation and image above, we can find the distance required for the flow to fully expand around a baffle as a function of baffle spacing S . We do this by substituting $D_0 = (0.384S)$ along with $D = S$ to approximate how much distance, $x = H_e$, the contracted flow has to cover.

$$\frac{H_e}{10} = S - (0.384S) \frac{H_e}{10} = 0.616SH_e = 6.16S \frac{H_e}{S} = 6.16\Pi_{H_e S_{Max}} = \frac{H_e}{S} = 6.16 \approx 6 \quad (23.13)$$

This is the highest allowable $\Pi_{H_e S}$ that we can design while ensuring that there is no dead space in the flocculator.

In order to have a robust design process for a baffle module, we need to have some flexibility in the $\Pi_{H_e S} = \frac{H_e}{S}$ ratio. Since we found $\Pi_{H_e S_{Max}}$ previously, we must now find the lowest functional $\frac{H_e}{S}$ ratio, $\Pi_{H_e S_{Min}}$.

AquaClara uses a fairly straightforward way of setting $\Pi_{H_e S_{Min}}$. It is based on the distance between the water level and the bottom baffle (which is the same distance between the flocculator floor and a top baffle). This distance is referred to as the slot width (Haarhoff 1998) DOI: 10.2166/aqua.1998.20) and is defined by the slot width ratio, which describes the slot width as a function of baffle spacing S . Slot width is shown in the following image:

AquaClara uses a slot width ratio of 1 for its flocculators. This number has been the topic of much hydraulic flocculation research, and values between 1 and 1.5 are generally accepted for hydraulic flocculators. See the following

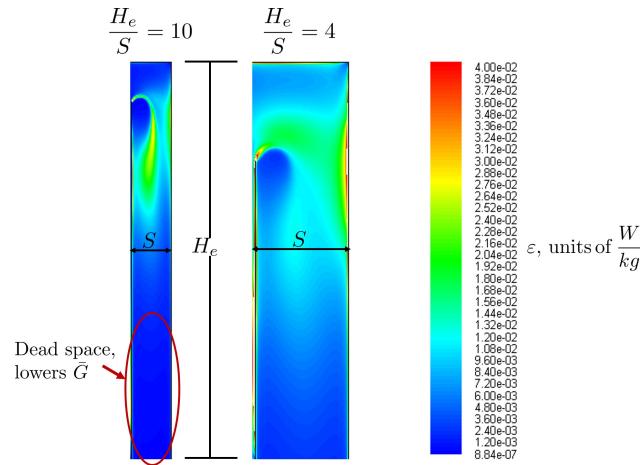


Fig. 23.7: High $\frac{H_e}{S}$ ratios result in flocculator zones with low velocity gradients that don't contribute effectively.

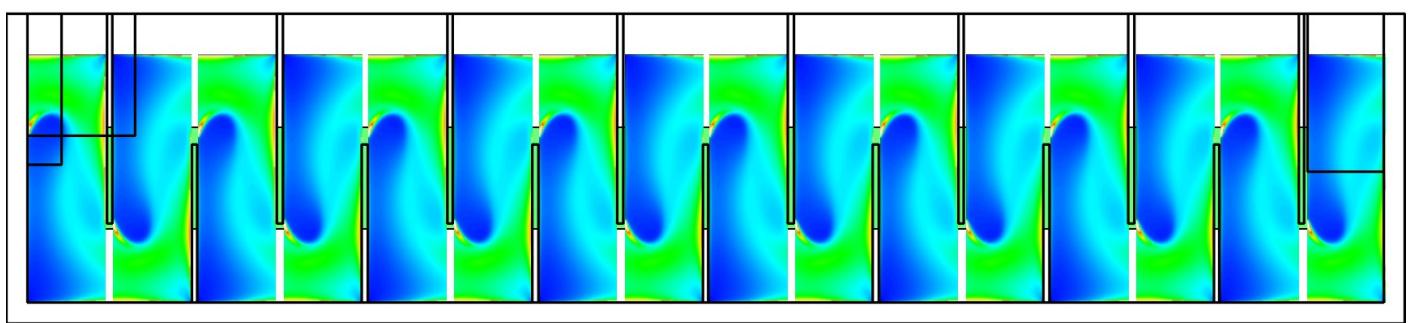


Fig. 23.8: Each bend creates a flow contraction and when the flow expands it converts kinetic energy into turbulent eddies and fluid deformation. The fluid deformation is what ultimately creates collisions between particles.

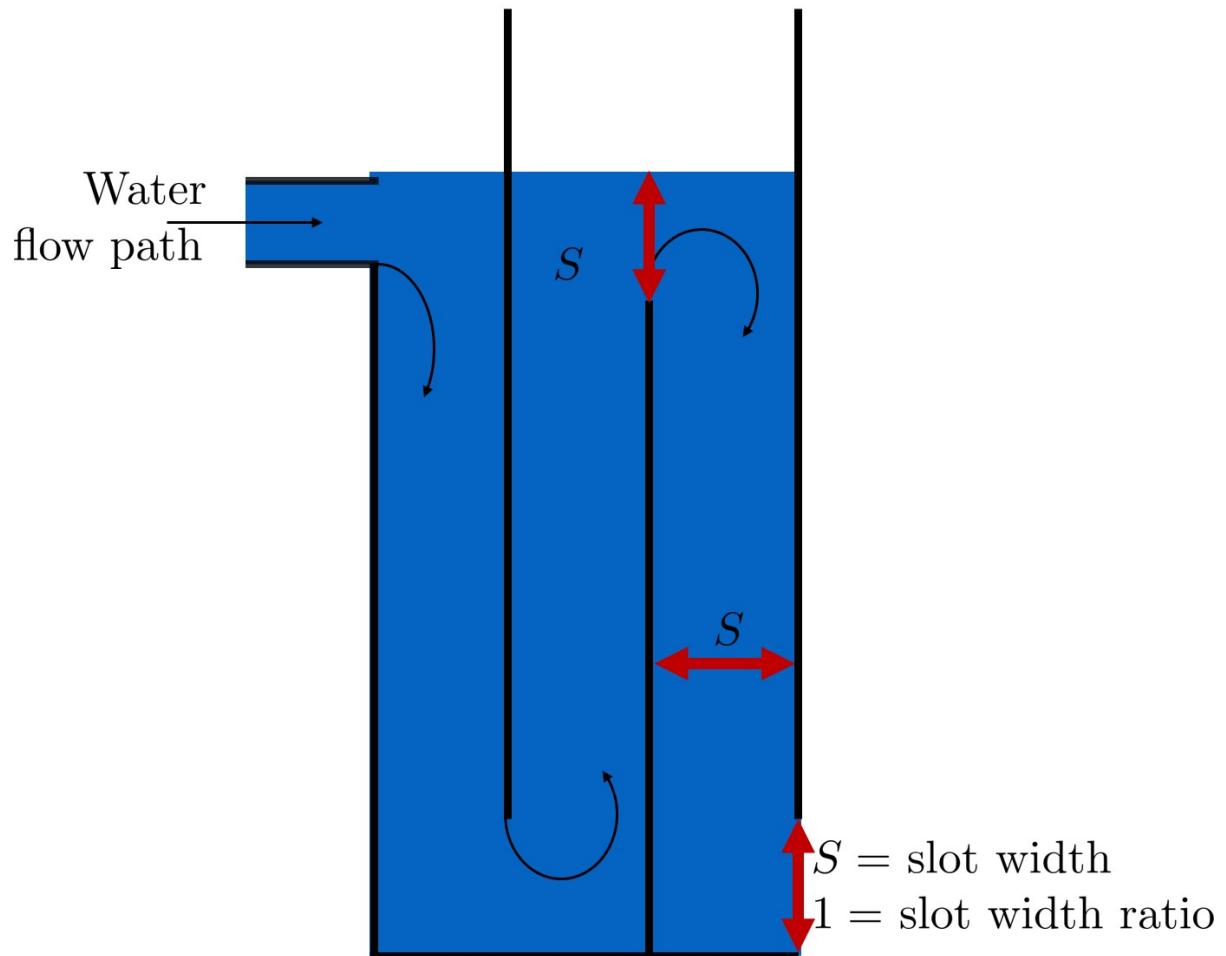
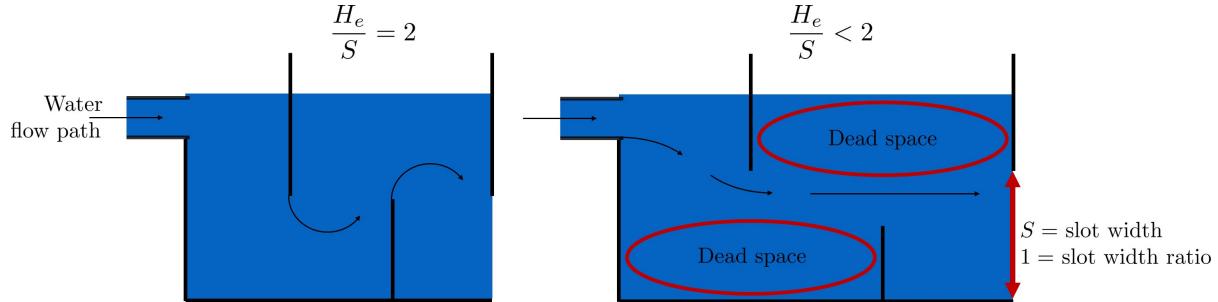


Fig. 23.9: The space between the bottom of the upper baffle and the floor of the flocculator is defined as the slot width.

paper and book respectively for more data on slot width ratios and other hydraulic flocculator parameters: [Haa98], [floc-schulz_surface_1992]. We base our slot width ratio of 1 on research done by [HvdW01] on optimizing hydraulic flocculator parameters to maximize flocculator efficiency.

The minimum $\Pi_{H_e S}$ allowable depends on the slot width ratio. If $\Pi_{H_e S}$ is less than twice the slot width ratio, the water would flow straight through the flocculator without having to bend around the baffles. This means that the flocculator would not be generating almost any head loss, and the top and bottom of the flocculator will largely be dead space. See the following image for an example:



Both images maintain AquaClara's standard slot width ratio of 1

Fig. 23.10: The minimum $\frac{H_e}{S}$ ratio is set by the need to prevent short circuiting through the flocculator.

Thus, $\Pi_{H_e S_{Min}}$ should be at least twice the slot width ratio, $\Pi_{H_e S_{Min}} = 2$. Historically, AquaClara plants have been designed using $\Pi_{H_e S_{Min}} = 3$. This adds a safety factor of sorts, ensuring that the flow does not short-circuit through the flocculator and also allowing more space for the flow to expand after each contraction.

$$\Pi_{H_e S_{Min}} = \frac{H_e}{S} = 3 \quad (23.14)$$

Finally, we describe a range of $\Pi_{H_e S}$ that we can use to design an AquaClara flocculator:

$$3 < \Pi_{H_e S} < 6 \quad (23.15)$$

Obstacles

Knowing that efficient flocculators require an $\frac{H_e}{S}$ ratio that lies between 3 and 6, we need to understand how that impacts the flocculator design. Keeping $\frac{H_e}{S}$ between two specific values limits the options for baffle spacing and quantity, due to the flocculator having certain size constraints before beginning the design of the baffles. This limitation places an upper limit on the amount of head loss that a baffled flocculator can generate, since the number of baffles is limited by space and baffles are what cause head loss. This is unfortunate, it means that baffled flocculators under certain size specifications cant be designed to generate certain values of $\bar{\varepsilon}$ and G_{CS} while remaining efficient and maintaining $3 < \Pi_{H_e S} < 6$. This problem only arises for low flow plants, usually below $Q_{Plant} = 20 \frac{L}{s}$.

To get around this problem, AquaClara included obstacles, or half-pipes to contract the flow after the flow expands around one baffle and before it reaches the next baffle. The purpose of these obstacles is to provide extra head loss in between baffles. They also generate head loss via minor losses, and one obstacle is designed to have the same K as one baffle. Introducing obstacles slightly alters how we think about H_e . In a flocculator where there are just baffles and no obstacles, then $H_e = H$, since the height of water in the flocculator is equal to the distance between expansions. When obstacles are added, however, then $H_e = \frac{H}{1+n_{obstacles}}$, where $n_{obstacles}$ is the number of obstacles between two baffles.

Baffle space is the term we use for the space between two baffles. The number of flow expansions per baffle space is $n_{expansions} = 1 + n_{obstacles}$. The 1 is because the baffle itself causes a flow expansion.

These obstacles serve as pseudo-baffles. They allow for $\frac{H}{S}$ to exceed 6, while maintaining maximum flocculator efficiency since, $\frac{H_e}{S}$ can still be between 3 and 6. Obstacles make it possible to design smaller flocculators without compromising flocculation efficiency. Fig. 23.11 and Fig. 23.12 show these obstacles and how they affect the flow in a flocculator.

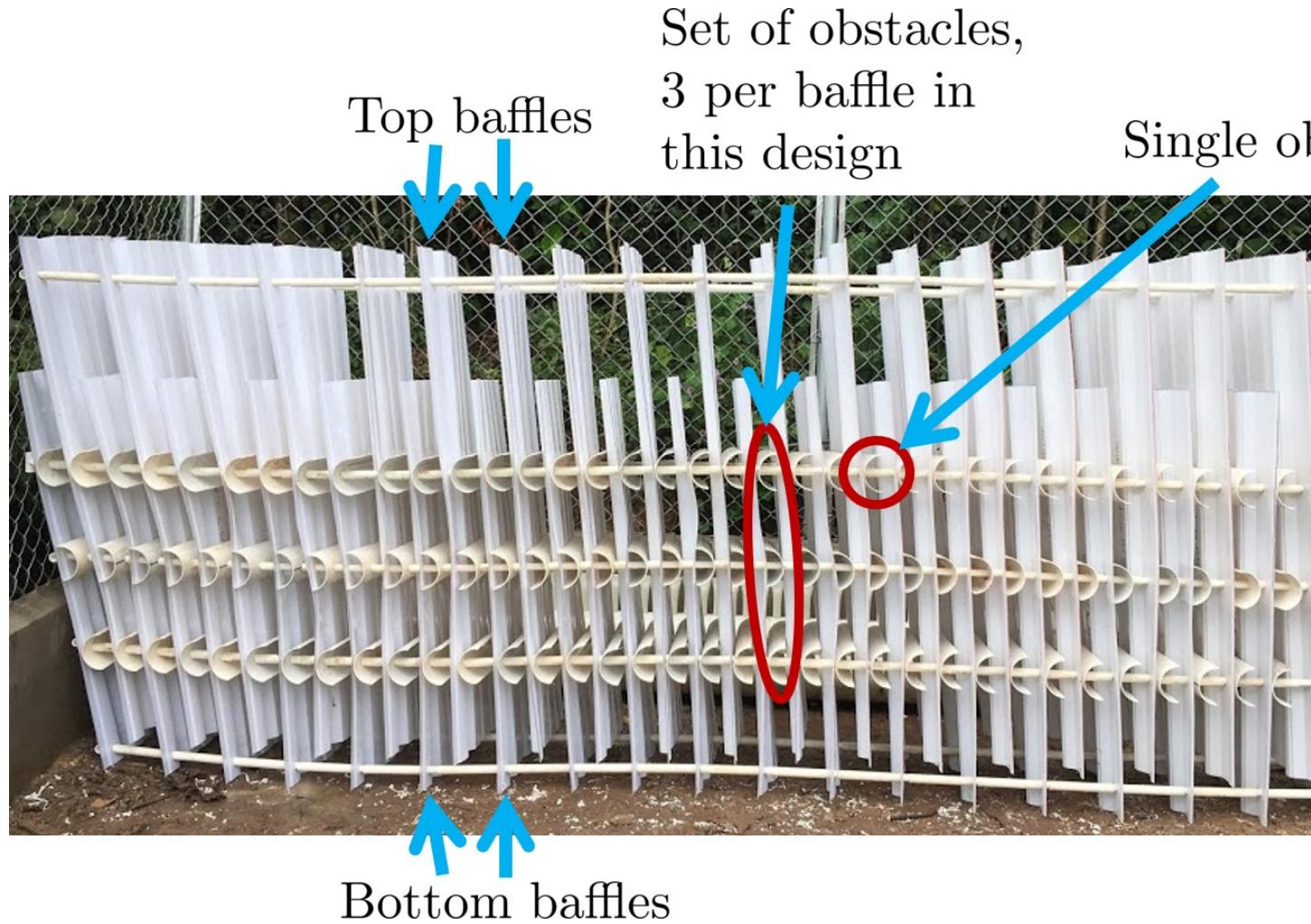


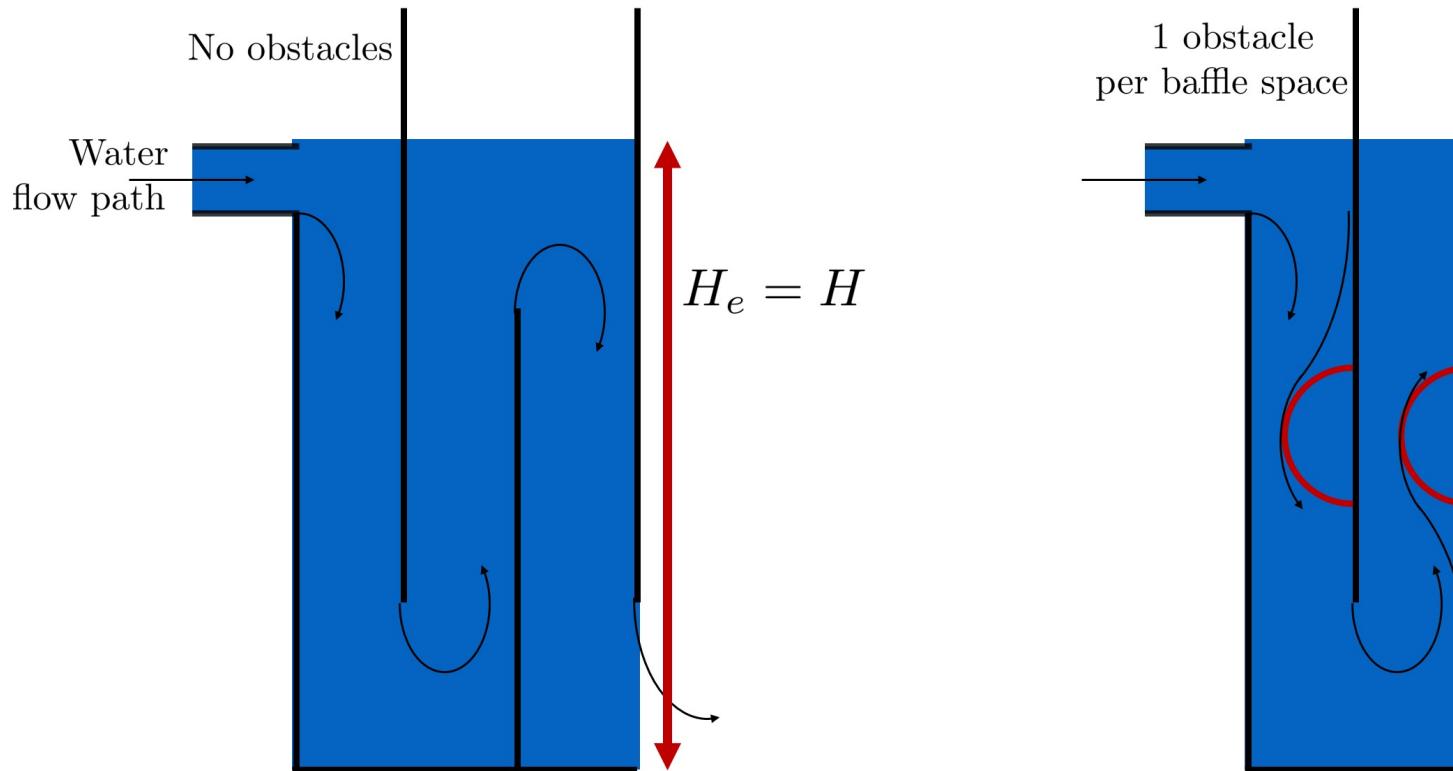
Fig. 23.11: Obstacles are added so that the flow continually contracts and expands. Additional obstacles are needed for low flow plants where the spacing between baffles is small relative to the flocculator depth.

23.3 AguClara Design of Hydraulic, Vertical Flow Flocculators

AquaClaras approach to flocculator design is the same as it is for any other unit process. First, critical design criteria, called inputs, are established. These criteria represent the priorities that the rest of the design will be based around. Once these parameters are established, then the other parameters of the design, which are dependent on the inputs, are calculated based on certain constraints.

Take the CDC as an example of this design process in *Flow Control and Measurement Design*; its inputs are $h_{L_{Max}}$, $\sum K$, Π_{Error} , and the discrete dosing tube diameters D that are available at hardware stores or pipe suppliers. Its dependent variables include the number and length of the dosing tubes and the flow through the CDC system.

The flocculator is more complex to design than the CDC, as it has more details and parameters and the equations



The volume between two baffles is called a baffle space. Mathematically, a baffle space is the volume occupied by the height of water times the width of the channel times the spacing between two baffles, $H \cdot W \cdot S$.

Fig. 23.12: Obstacles ensure that there aren't any zones with low velocity gradients.

for those details and parameters are very interdependent. Therefore, there are many ways to design an AquaClara flocculator, and many different sets of critical design criteria to begin with. Enumerated below is the current AquaClara approach.

1. Input parameters

- Specify: - $h_{L_{floc}}$, head loss - $G_{CS}\theta$, collision potential - Q , plant flow rate - H , height of water at the end of the flocculator - $L_{Max, sed}$, max length of a flocculator channel based on sedimentation tank length - $W_{Min, human}$ minimum width of a single channel based on the width of the average human hip (someones got to go down there)
- Find: - G_{CS} , average velocity gradient - θ , hydraulic retention time - V_{floc} , flocculator volume

2. Physical dimensions

- Calculate: - $L_{channel}$, actual channel length - $n_{channels}$, amount of channels - $W_{channel}$, actual channel width

3. Hydraulic parameters

- Calculate: - H_e , distance between baffle/obstacle induced flow expansions - $n_{obstacles}$, amount of obstacles per baffle space - S , baffle spacing, distance between baffles

23.3.1 Input Parameters

Specify

We start by making sure that our flocculator will be able to flocculate effectively by defining $h_{L_{floc}}$ and $G_{CS}\theta$. Fixing these two parameters initially allows us to easily find all other parameters which determine flocculator performance. Here are the current standards in AquaClara flocculators: - $h_{L_{floc}} = 40 \text{ cm}$ - $G_{CS}\theta = 37,000$

The plant flow rate Q is defined by the needs of the community that the plant is being designed for. Additionally, the height of water at the end of the flocculator, H , the maximum length of the flocculator based on the length of the sedimentation tank length, $L_{Max, sed}$, and the minimum width of a flocculator channel required for a human to fit inside, $W_{Min, human}$, are also defined initially. Ordinarily in AquaClara plants, the flocculator occupies the same length dimension as the sedimentation tanks, which is why the length constraint exists. See Fig. 23.14 for a representation of how the flocculator and sedimentation tanks are placed in a plant.

- $H = 2 \text{ m}$
- $L_{Max, sed} = 6 \text{ m}$
- $W_{Min, human} = 45 \text{ cm}$

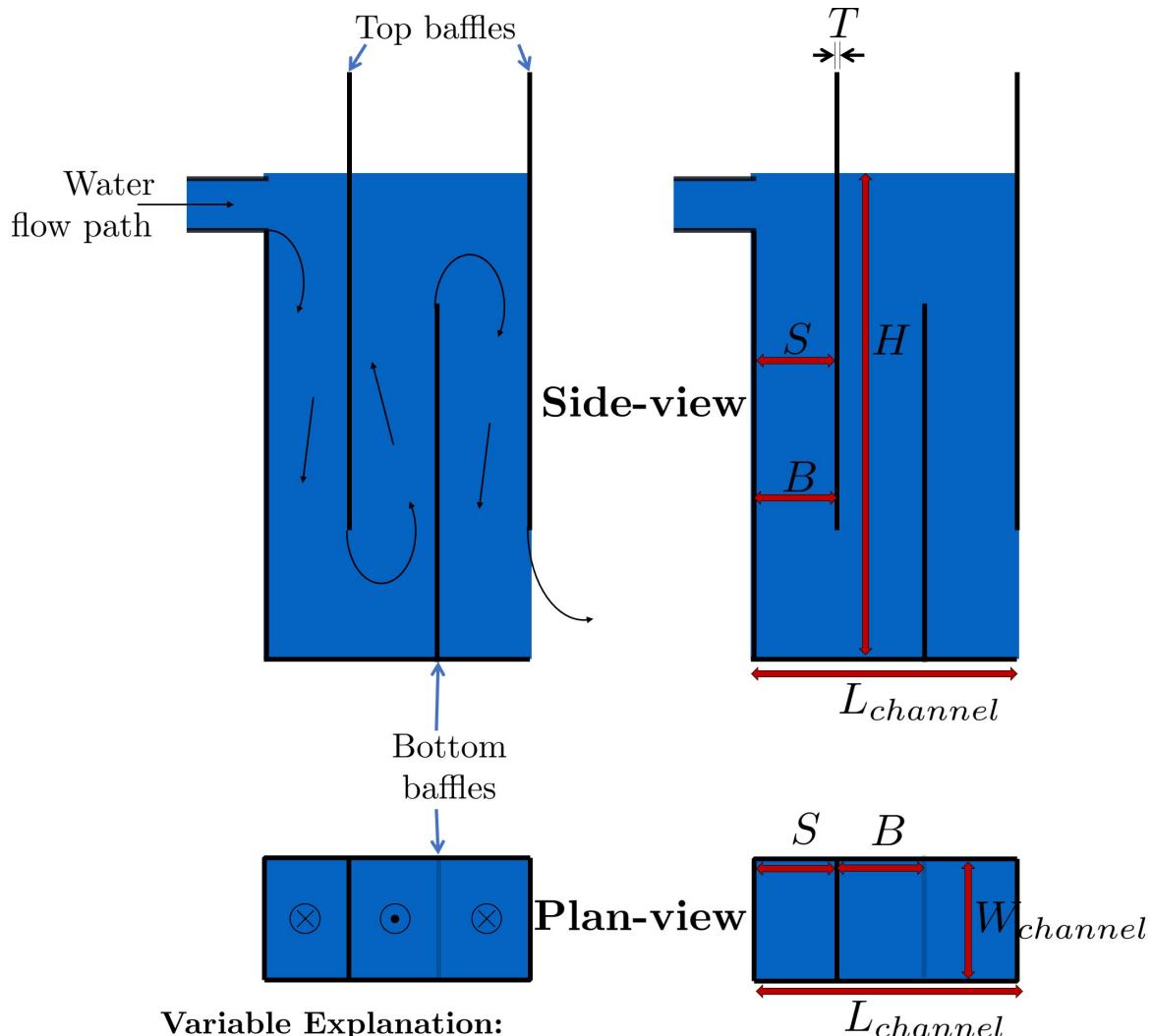
Find

We can rearrange the equation for G_{CS} from the section on collision potential, $G_{CS} = \sqrt{\frac{gh_L}{\nu\theta}}$, to solve for G_{CS} in terms of $G_{CS}\theta$:

$$G_{CS} = \frac{gh_{L_{floc}}}{\nu(G_{CS}\theta)} \quad (23.16)$$

Now that we have G_{CS} , we can very easily find θ :

$$\theta = \frac{G_{CS}\theta}{G_{CS}} \quad (23.17)$$



Variable Explanation:

$W_{channel}$ = width of flocculator channels

$L_{channel}$ = length of flocculator channels

S = distance between baffles. This is the dimension that water flows through

T = thickness of a baffle

$B = S + T$ = center-to-center distance between baffles. Accounts for thickness of each baffle

Fig. 23.13: Flocculator geometry definition including the effect of baffle thickness. Accounting for baffle thickness would be particularly important if ferrocement or wood were used for baffles.

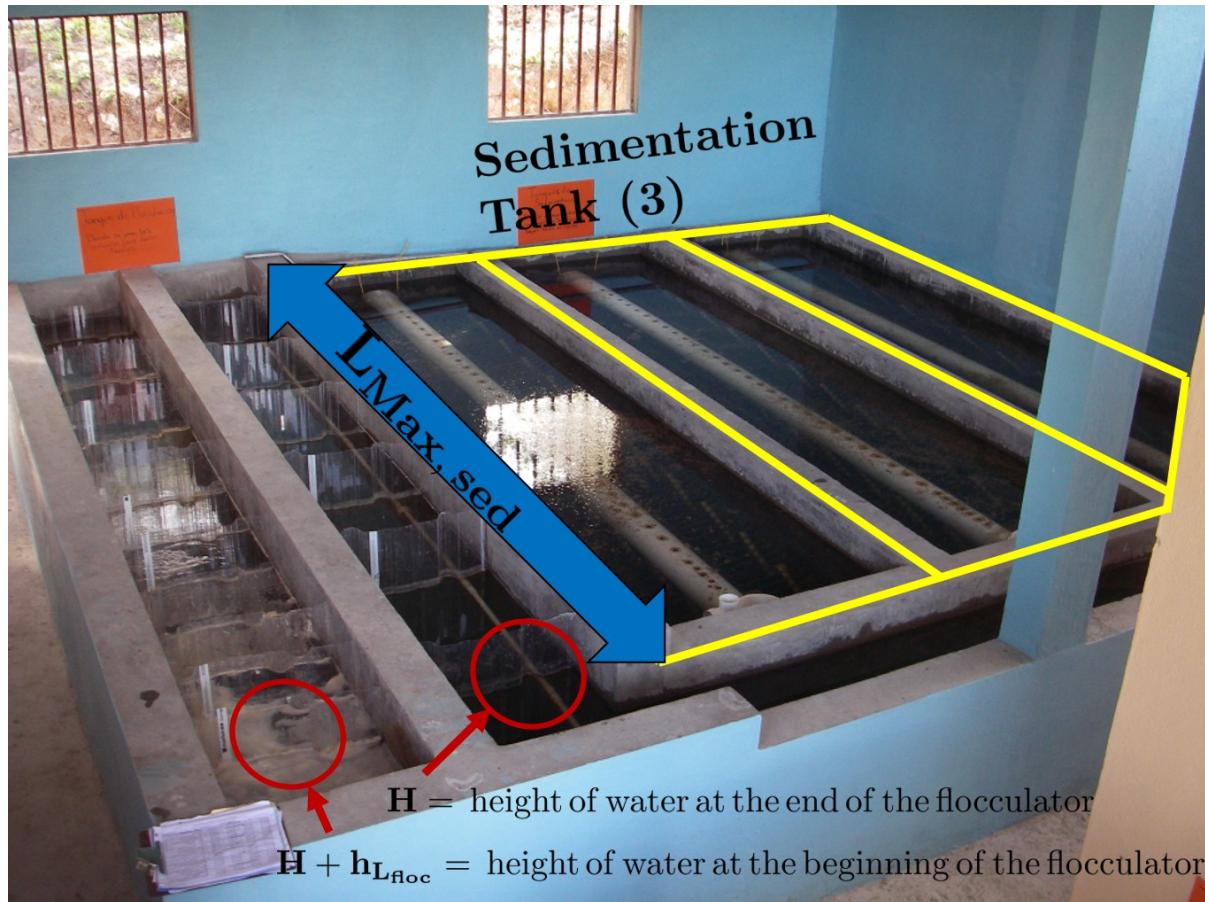


Fig. 23.14: Layout of flocculator and sedimentation tanks that was adopted starting with the 2nd AquaClara plant in Tamara, Honduras in 2008.

Finally, we take retention time θ over plant flow rate Q to get the required volume of the flocculator:

$$V_{floc} = \frac{\theta}{Q} \quad (23.18)$$

Now that we have the basic parameters defined, we can start to design the details of the flocculator, starting from the physical dimensions.

23.3.2 Physical Dimensions

Deriving the equations required to find the physical dimensions now and the hydraulic parameters (baffle/obstacle design) in the next section requires many steps. To simplify this design explanation the equation derivations are developed in *Review: Fluid Mechanics Derivations*. All complex equations which seemingly came out of nowhere will be derived in the derivation sheet.

Length

Flocculator length, $L_{channel}$ [‘] must meet two constraints: it must be less than or equal to the length of the sedimentation tanks, as the flocculator is adjacent to the sed tanks. This constraint is $L_{Max, sed}$. Next, the flocculator must be short enough to make sure the target volume of the flocculator is met, while still allowing for a human to fit inside $L_{Max, V}$. **The constraint that wins out is the one that results in the *smaller* length value.**

$$L_{Max, sed} = 6 m L_{Max, V} = \frac{V}{n_{Min, channels} W_{Min, human} H} \quad (23.19)$$

Such that:

$$n_{Min, channels} = 2$$

The reason why $W_{Min, human}$ is used is because it represents the absolute minimum of flocculator channel width. If the width ends up being larger, the length will decrease. $n_{Min, channels} = 2$ to make sure that the flow ends up on the correct side of the sedimentation tank, as the image below shows. Note that there can only be an even number of flocculator channels, as explained in the images caption.

The equation for *actual* flocculator length is therefore:

$$L_{channel} = \min(L_{Max, sed}, L_{Max, V}) \quad (23.20)$$

Width and Number of Channels

The width of a single flocculator channel must meet the following conditions: - Maintain G_{CS} at the value found in the inputs section - Allow for $3 < \frac{H_e}{S} < 6$. Recall that $\frac{H_e}{S} = \Pi_{H_e S}$ - Allow for a human to be able to fit into a flocculator channel

The first two conditions are wrapped up into the following equation, *which is derived here*

$$W_{Min, \Pi_{H_e S}} = \frac{\Pi_{H_e S} Q}{H_e} \left(\frac{K}{2H_e \nu G_{CS}^2} \right)^{\frac{1}{3}} \quad (23.21)$$

This equation represents the absolute smallest width of a flocculator channel if we consider the lowest value of $\Pi_{H_e S}$ and the highest possible value of H_e :

$H_e = H_{e_{Max}} = H = 2 \text{ m}$, this implies that there are no obstacles between baffles $\Pi_{H_e S} = \Pi_{HS_{Min}} = 3$



Fig. 23.15: There are an even amount of flocculator channels to keep the AquaClara plant layout consistent for flows greater than 12 L/s. This ensures that the entrance tank, filter box, and filters can be kept in the same places across plants.

Recall our other width constraint, $W_{Min, human} = 45$ cm, which is based on our desire to have a human be able to fit into the channels. The governing constraint is the *larger* value of W_{Min} :

$$W_{Min} = \max(W_{Min, \Pi_{H_e S}}, W_{Min, human}) \quad (23.22)$$

We can find the number of channels, $n_{channels}$ and their actual width in one last step, by finding the *total flocculator width* if there were no channels and dividing that by the minimum flocculator width, W_{Min} , found above. The equation for total flocculator width is based on our target volume:

$$W_{total} = \frac{V}{H L_{channel}} \quad (23.23)$$

Finally:

$$n_{channels} = \frac{W_{total}}{W_{Min}} \quad (23.24)$$

Such that:

$n_{channels}$ is an even number and is not 0. Usually, $n_{channels}$ is either 2 or 4.

Now that we know $n_{channels}$, we can find the actual width of a channel, $W_{channel}$.

$$W_{channel} = \frac{W_{total}}{n_{channels}} \quad (23.25)$$

23.3.3 Hydraulic Parameters

Now that the physical dimensions of the flocculator have been defined, the baffle module needs to be designed. The parameter on which most others are based is the distance between flow expansions, H_e . Recall that $H_e = H$ when there are no obstacles in between baffles.

Height Between Expansions H_e and Number of Obstacles per Baffle Space $n_{obstacles}$

We have a range of possible H_e values based on our window of $3 < \frac{H_e}{S} < 6$. However, we have a limitation and a preference which shape how we design H_e . Our limitation is that there can only be an integer number of obstacles. Our preference is to have as few obstacles as possible to make the baffle module as easy to fabricate as possible. Therefore, we want H_e to be closer to 6 than it is to 3; we are looking for $H_{e_{Max}}$.

We calculate $H_{e_{Max}}$ based on the physical flocculator dimensions. The equation for H_e is obtained by rearranging one of the equations for minimum channel width found above, $W_{Min, \Pi_{H_e S}} = \frac{\Pi_{H_e S} Q}{H_e} \left(\frac{K}{2H_e \nu G_{CS}^2} \right)^{\frac{1}{3}}$. Because we have already design the channel width, we substitute $W_{channel}$ for $W_{Min, \Pi_{H_e S}}$. Since we are looking for $H_{e_{Max}}$, we also substitute $\Pi_{HS_{Max}}$ for $\Pi_{H_e S}$. The result is:

$$H_{e_{Max}} = \left[\frac{K}{2\nu G_{CS}^2} \left(\frac{Q\Pi_{HS_{Max}}}{W_{channel}} \right)^3 \right]^{\frac{1}{4}} \quad (23.26)$$

Note that this is the *maximum* distance between flow expansions, and does not account for the limitation that there must be an integer number of obstacles per baffle space. Thus, we need to find the *actual* distance between flow expansions. To do this, we determine and round up the number of expansions per baffle space using the ceiling function:

$$n_{expansions} = \text{ceil} \left(\frac{H}{H_{e_{Max}}} \right) \quad (23.27)$$

If we had used the floor() function instead, we would find that H_e would be larger than our upper bound, $H_{e_{Max}}$. From here, we can easily get to the actual number of flow expansions per baffle spacing:

$$H_e = \frac{H}{n_{expansions}} \quad (23.28)$$

Finally, we can obtain the number of obstacles per baffle space. The -1 in the equation is because the baffles themselves provide one flow expansion per baffle space.

$$n_{obstacles} = \frac{H}{H_e} - 1 \quad (23.29)$$

Baffle Spacing

Finally, we can find the space between baffles, S . The equation for S is taken from an intermediate step in the $W_{Min, \Pi_{H_e S}}$ derivation where we obtained, $W = \frac{Q}{S} \left(\frac{K}{2H_e \nu G_{CS}^2} \right)^{\frac{1}{3}}$. Rearranging for S , we get:

$$S = \left(\frac{K}{2H_e G_{CS}^2 \nu} \right)^{\frac{1}{3}} \frac{Q}{W_{channel}} \quad (23.30)$$

Fortunately, we either know or have already designed all the parameters in this equation.

23.4 Checking the Flocculator Design

We then compare $n_{spaces, required}$ to $n_{spaces, actual}$ to make sure that they are equal.

23.4.1 Average Velocity in the Flocculator Check

As water flows through the flocculators, the flocs will get larger and larger. As a result, their terminal sedimentation velocity will increase. This is what we want. However, we need to make sure that the flocs don't settle in the flocculator; that they instead all settle in the sedimentation tank. To make sure of this, we need to make sure that the velocity of water in the flocculator is high enough to scour any flocs that fall to the bottom of the flocculator. The velocity required to scour flocs from the bottom and avoid floc accumulation is around $v_{scour} = 15 \frac{\text{cm}}{\text{s}}$. We need to check our average velocity \bar{v} against this value.

$$\bar{v} = \frac{Q}{W_{channel} S} \quad (23.31)$$

23.4.2 Residence Time of Water in the Flocculator Check

It is now time to make our final check. We need to make sure that our actual residence time is *at least* as much as we designed for. Fortunately, in our design we did not account for the change in water level throughout the flocculator due to head loss. Therefore, the actual volume of water in the flocculator is actually greater than V_{floc} . See Fig. 23.16 for clarification.

Thus, the actual average water level in the flocculator is $H + \frac{h_{L_{floc}}}{2}$. Thus, the actual residence time is:

$$\theta_{actual} = \frac{n_{channels} L_{channel} W_{channel} \left(H + \frac{h_{L_{floc}}}{2} \right)}{Q} \quad (23.32)$$

Check to see if θ_{actual} is greater than θ .

The red triangle signifies the area unaccounted for in the initial volume calculation. This makes θ_{actual} greater than the design θ . However, we do not believe that this extra triangle of space has significant G values, and therefore we do not believe that it adds any collision potential to the design.

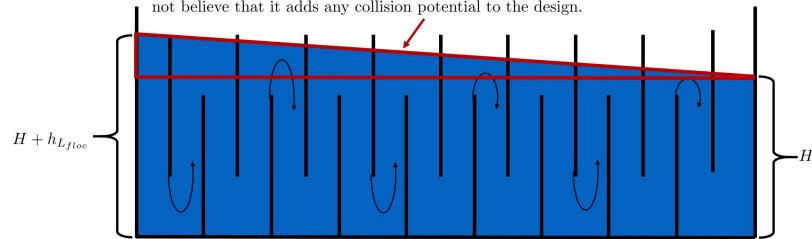


Fig. 23.16: The water level in the flocculator decreases due to head loss. Flocculators may occupy multiple channels, but this extra triangle of water exists in any case.

CHAPTER
TWENTYFOUR

FLOCCULATION DERIVATIONS

24.1 Design Equations for the Flocculator

This document contains the derivation for the minimum allowable width of a flocculator channel based on the requirements that $3 < \Pi_{H_e S} < 6$ and that we maintain the \bar{G} that serves as a basis for design. The final parameter derived is $W_{Min, \Pi_{H_e S}}$.

24.1.1 Width

Our two restrictions are: - Ensuring that we maintain the \bar{G} we get based on our input parameters - Ensuring that $3 < \frac{H_e}{S} < 6$

First, we begin by setting the two equations for energy dissipation rate, $\bar{\varepsilon} = \nu \bar{G}^2$ and $\bar{\varepsilon} = \frac{gh_{L_{floc}}}{\theta}$ equal to each other to bring \bar{G} into the equation.

$$\nu \bar{G}^2 = \frac{gh_{L_{floc}}}{\theta} \quad (24.1)$$

Very Important Note:

For the following steps, we will consider the flow through a single flow expansion H_e , not through the entire flocculator. This could be from baffle to obstacle, obstacle to baffle, obstacle to obstacle, or baffle to baffle depending on how many obstacles are in the design. This means that we are briefly redefining θ to be the time it takes for the flow to fully expand after a flow contraction. θ no longer represents the time it takes for the flow to go through the entire flocculator.

From here we make three subsequent substitutions: first $h_{L_{floc}} = K \frac{\bar{v}^2}{2g}$, then $\theta = \frac{H_e}{\bar{v}}$, and finally $\bar{v} = \frac{Q}{WS}$

$$\nu \bar{G}^2 = K \frac{\bar{v}^2}{2\theta} \quad (24.2)$$

$$\nu \bar{G}^2 = K \frac{\bar{v}^3}{2H_e} \quad (24.3)$$

$$\nu \bar{G}^2 = \frac{K}{2H_e} \left(\frac{Q}{WS} \right)^3 \quad (24.4)$$

Now we can solve this equation for channel width, W .

$$W = \frac{Q}{S} \left(\frac{K}{2H_e \nu \bar{G}^2} \right)^{\frac{1}{3}} \quad (24.5)$$

From here, we can define $\Pi_{H_e S} = \frac{H_e}{S}$ and substitute $S = \frac{H_e}{\Pi_{H_e S}}$ into the previous equation for W to get $W_{Min, \Pi_{H_e S}}$:

$$W_{Min, \Pi_{H_e S}} = \frac{\Pi_{H_e S} Q}{H_e} \left(\frac{K}{2H_e \nu \bar{G}^2} \right)^{\frac{1}{3}} \quad (24.6)$$

This equation represents the absolute smallest width of a flocculator channel if we consider the lowest value of $\Pi_{H_e S}$ and the highest possible value of H_e :

$H_e = H$, this implies that there are no obstacles between baffles

$$\Pi_{H_e S} = 3$$

CHAPTER
TWENTYFIVE

FLOCCULATION THEORY AND FUTURE WORK

FLOCCULATION EXAMPLES

26.1 Pipe flocculator

The following analysis requires complete reworking.

```
# %%
#Assumptions
Pi_VC = .62 #Vena contracta coefficient of an orifice
Ke = ((1/Pi_VC**2)-1)**2 #expansion coefficient

#Functions to calculate key parameters

def G_ave(G_theta,h_floc,Temp):
    """Calculates average G given target minimum collision potential, total headloss, ↵and design temperature
    equation from flocculation slides"""
    G_ave = (pc.gravity*h_floc/(G_theta*pc.viscosity_kinematic(Temp))).to(1/u.s)
    return G_ave

def restime(G_theta,G_ave):
    """Calculates residence time given collision potential and average G
    equation from flocculation slides"""
    theta = G_theta/G_ave
    return theta

def Dpipe(Ke,Pi_HS,Q,G_ave,Temp,SDR):
    """Calculates the actual inner diameter of the pipe
    equation from flocculation slides"""
    D_pipe = ((Ke/(2*Pi_HS*pc.viscosity_kinematic(Temp)*G_ave**2))*(4*Q.to(u.m**3/u.s)/np.pi)**3)**(1/7)
    return D_pipe

def Keactual(ID_pipe,G_ave,Temp,Pi_HS,Q):
    """estimates actual expansion coefficient given the actual inner diameter and ↵other relevant inputs
    equation from flocculation slides"""
    Ke_actual = np.pi**3*ID_pipe**7*G_ave**2*pc.viscosity_kinematic(Temp)*Pi_HS/(32*Q.to(u.m**3/u.s)**3)
    return Ke_actual

def Aorifice(ID_pipe,Ke_actual,Temp,Q):
```

(continues on next page)

(continued from previous page)

```

"""Calculates the orifice area given pipe inner diameter, expansion coefficient,
Temperature, and flow"""
A1 = (pc.area_circle(ID_pipe)).to(u.cm**2).magnitude #Pipe area
Nu = pc.viscosity_kinematic(Temp) #kinematic viscosity
Re = pc.re_pipe(Q, ID_pipe, Nu) #reynolds number

def f_orif(A2,A1,Ke_actual,Re): #root of this function is the orifice area
    return (2.72+(A2/A1)*(4000/Re))*(1-A2/A1)*((A1/A2)**2-1)-Ke_actual

A_orifice = (brentq(lambda A2: f_orif(A2,A1,Ke_actual,Re), -1, 2*A1))*u.cm**2
#numerical optimization

return A_orifice

def eave(G_ave,Temp):
    """Calculates the average energy dissipation rate"""
    e_ave = (pc.viscosity_kinematic(Temp)*G_ave**2).to(u.mW/u.kg)
    return e_ave

def Hchip(A_orifice, ID_pipe):
    """This function calculates the height of the chip based on the orifice area and
    pipe diameter
    The function uses numerical optimization to solve the transcendental equation"""
    A_flow = A_orifice.magnitude #orifice area stripped of units
    r=(ID_pipe/2).magnitude #radius stripped of units
    c = A_flow/r**2 #left hand side of equation

    def f(a,c): #roots of this function are theta
        return a-sin(a)*cos(a)-c

    theta = brentq(lambda a: f(a,c), 0, 13) #numerical optimization
    r_u = r*u.cm #radius with units
    y = r_u - r_u*np.cos(theta) #height of orifice

    H_chip = ID_pipe-y #height of chip
    return H_chip

def Cost_Length(L_pipe,ND_pipe):
    """This function calculates the total cost of the system and the total length of
    the system"""
    #Length of pipe and number of fittings needed
    OD_pipe = pipe.OD(ND_pipe)
    Total_Pipe = L_pipe + .5*u.m
    Number_T = np.ceil(Total_Pipe.magnitude)
    Number_Elbow = np.ceil(Total_Pipe.magnitude)

    if ND_pipe.magnitude == 3:
        Cost_T = 3.94*u.dollar
        Cost_Elbow = 3.53*u.dollar
        Cost_Pipe = (17.14/10*(u.dollar/u.foot)).to(u.dollar/u.m)
        Cost_Valve = 10*u.dollar
        Width_T = (3.99*u.inch).to(u.cm)
        Width_Elbow = (3.97*u.inch).to(u.cm)

    if ND_pipe.magnitude ==4:

```

(continues on next page)

(continued from previous page)

```

Cost_T = 7.16*u.dollar
Cost_Elbow = 5.40*u.dollar
Cost_Pipe = (21.5/10*(u.dollar/u.foot)).to(u.dollar/u.m)
Cost_Valve = 10*u.dollar
Width_T = (5.06*u.inch).to(u.cm)
Width_Elbow = (5.06*u.inch).to(u.cm)

if ND_pipe.magnitude ==6:
    Cost_T = 7.16*u.dollar
    Cost_Elbow = 5.40*u.dollar
    Cost_Pipe = (21.5/10*(u.dollar/u.foot)).to(u.dollar/u.m)
    Cost_Valve = 10*u.dollar
    Width_T = (5.06*u.inch).to(u.cm)
    Width_Elbow = (5.06*u.inch).to(u.cm)

    Total_Cost = Cost_Pipe*Total_Pipe + Cost_T*Number_T + Cost_Elbow*Number_Elbow +_
    ↪Cost_Valve*Number_Elbow
    Floor_Length = Number_T*(Width_T+Width_Elbow-OD_pipe).to(u.m)
    Output=[Total_Cost,Floor_Length]
return Output

```

```

#Inputs
D_Sed = 2.5*u.cm
A_Sed = pc.area_circle(D_Sed)
v_Sed = 2*u.mm/u.s
Q = (v_Sed*A_Sed).to(u.mL/u.s)
print('The flow rate is',Q)

Temp = 15*u.degC
h_floc = 50*u.cm #standard for Aguaclara plants
G_theta = 20000 #standard for Aguaclara plants
Pi_HS = 6 ##3-6 is a good range, more research needed
SDR = 41 #Standard ratio

```

```

#Calculate G average using functions listed above and given inputs
G_ave = Gave(G_theta,h_floc,Temp)
theta = restime(G_theta,G_ave)
e_ave = eave(G_ave,Temp)
print('The average G value is ',G_ave)
print('The residence time in the flocculator is ',theta)
print('The average energy dissipation rate is ', e_ave)

```

```

#Calculate the pipe diameter, both inner and nominal and determine area of pipe using_
↪inner diameter output
D_pipe = (Dpipe(Ke,Pi_HS,Q,G_ave,Temp,SDR)).to(u.cm)
#Calculate nominal diameter of pipe
ND_pipe = pipe.ND_SDR_available(D_pipe,SDR)
#Calculate nominal diameter of pipe
ID_pipe = pipe.ID_SDR(ND_pipe,SDR).to(u.cm)

ID_pipe = 5*u.mm
#Calculate inner diameter of pipe
A_pipe = (pc.area_circle(ID_pipe)).to(u.cm**2)

```

(continues on next page)

(continued from previous page)

```
print('The ideal inner diameter of the pipe would be ',D_pipe)
print('The nominal diameter of the pipe is ',ND_pipe, ', and the inner diameter is ',_
      ↪ID_pipe)
print('The area of the pipe is ', A_pipe)
```

```
#Calculate the actual Ke as a result of the calculated inner pipe diameter
Ke_actual = (Keactual(ID_pipe,G_ave,Temp,Pi_HS,Q)).to(u.dimensionless)
print('The initial expansion minor loss coefficient was ',Ke)
print('The actual expansion minor loss coefficient is ',Ke_actual)
```

```
#Calculate the orifice area
A_orifice = Aorifice(ID_pipe,Ke_actual,Temp,Q)
print('The orifice area is ',A_orifice)
```

```
# The following line of code needs to be removed once the orifice area equation is_
      ↪corrected.

H_chip = Hchip(A_orifice,ID_pipe)
print('The height of the chip is ', H_chip)
```

```
#Calculate average velocity
v_avg = (Q/pc.area_circle(ID_pipe)).to(u.m/u.s) #first calculate average velocity
print('The average velocity is ',v_avg)

#Calculate pipe length
L_pipe = (v_avg*theta).to(u.m) #then multiply velocity by residence time to get the_
      ↪required length of pipe
print('The length of the pipe is ',L_pipe)
```

CHAPTER
TWENTYSEVEN

MECHANICAL FLOCCULATOR SOLUTION

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u

import numpy as np
import matplotlib.pyplot as plt
```

In this design challenge, you will design a mechanical flocculator and power unit. You will use a flow rate of 50 L/s as your default design value. The coldest temperature that the raw water is expected to have is 10°C. You will also play with fractal flocculation model equations to get a sense of how fast flocs can aggregate. You will also explore the floc model predictions.

As you define variables, take care not to redefine parameters. One trick is to use a unique name for variables that are estimates. We often add est to any variable names that aren't the final values.

The `floc_model.py` file contains many useful functions which includes clay, coagulant and humic acid material properties, fractal model equations, and flocculation model equations. **We recommend that you open the “floc_model.py“ and have it side by side with this design challenge for easy reference.**

The course slides found on the syllabus will be very useful for obtaining and understanding relevant equations for this design challenge and ones in the future.

27.1 Conventional Design Guidelines

This table for mechanical flocculators is taken from Sincero and Sinceros 1996 textbook: *Environmental Engineering: A Design Approach*

Type	Velocity Gradient (G) (1/s)	Energy Dissipation Rate	Gt	t (min)
Low turbidity, color removal	20-70	0.4 - 4.9	50,000	-
			250,000	210
High turbidity, solids removal	70 - 180	4.9 - 32	80,000	-
			190,000	7 - 45

Dissolved organic matter impacts the color of the water. Perhaps the reason for lower G values for color removal is because flocs created from dissolved organic matter are weaker or less dense. Hence, a lower G is needed to allow them to grow large enough for removal in the sedimentation tank.

27.2 Mechanical Flocculator Design

The conventional design guidelines provide a range of values with little guidance on how to select velocity gradient or G values. One approach is to hope for the best and choose a G value in the middle of the range. We will pick the G value that is apparently appropriate for both low and high turbidity waters.

```
flow_plant = 50 * u.L/u.s
G_mech = 70 / u.s
temp_design = 10 * u.degC
```

27.2.1 1)

The **Ten State Standards** (with the hilarious acronym GLUMRB) requires the residence time in a mechanical flocculator to be 30 minutes. What is the value of Gt if you use the velocity gradient given above as G_mech?

Note: .to(u.dimensionless) ensures you dont get weird dimensions like kilometer/inch or minute/second for dimensionless parameters.

```
#answer
time_mech = 30 * u.min
Gtime_mech= (G_mech * time_mech).to(u.dimensionless)

print('The Gt is', Gtime_mech)
```

The Gt is 126000

27.2.2 2)

Calculate the equivalent average energy dissipation rate (in mW/kg) for the given velocity gradient at the design temperature.

```
#answer
ed_rate_mech_ave = (G_mech**2 * pc.viscosity_kinematic(temp_design)).to(u.mW/u.kg)
print('The equivalent average energy dissipation rate is' , ed_rate_mech_ave, '.')
```

The equivalent average energy dissipation rate is 6.4 mW/kg

27.2.3 3)

Create a function to calculate the power requirement for mechanical flocculators. Then, use your function to obtain the power requirement for the design temperature. The equation for power is the same one that we used for mechanical rapid mix units.

```
#answer
def power_floc_shaft(Q, G, t, temp):
    return (G**2 * Q * t * pc.viscosity_dynamic(temp)).to(u.kW)
power_mech_floc = power_floc_shaft(flow_plant,G_mech,time_mech,temp_design)
print('The power requirement is', power_mech_floc,'.')
```

The power requirement is 0.57 kW

Fun Fact: although the design guidelines would logically predict that the power requirements are higher for cold water, there is little evidence that mechanical flocculator plants actually increase the power input to their flocculators when the water is cold.

27.2.4 4)

We hypothesize that the maximum energy dissipation rate in a flocculator determines the size of the flocs, and those flocs may or may not be captured by the sedimentation tank. Sedimentation tank performance will deteriorate if the maximum energy dissipation rate results in flocs that are so small that their sedimentation velocity is lower than the capture velocity of the sedimentation tank. This likely occurs for very high maximum energy dissipation rates. Mechanical flocculators have traditionally been designed without insight into the variability of the energy dissipation rate and the need to keep the maximum energy dissipation rate low enough to prevent the creation of flocs that will avoid capture by the sedimentation tanks. At the same time, the traditional standards likely evolved to produce designs that were at least reasonable even in the absence of an understanding of the fluid mechanics of the mixing process.

According to the Ten State Standards, Agitators (Flocculators) shall be driven by variable speed drives with the peripheral speed of paddles ranging from 0.5 to 3.0 feet per second. Note that they do not specify the size of the propeller and yet that is a critical dimension that determines the energy dissipation rate in the wake of the spinning propeller.

The wake behind the propeller or paddle is similar in size to the small dimension of the propeller or paddle normal to the velocity. Thus we can use the propeller height to estimate the energy dissipation rate of the plate wake. The relative velocity between the wake and the surrounding fluid is approximately equal to the propeller velocity.

Calculate the maximum energy dissipation rate that occurs in the wake behind the tip of a propeller which has a height of 3 cm and is moving at 3 ft/s. In this case, assume that the plate ratio for the maximum energy dissipation rate in the wake, Π_{Plate} , is 0.04.

$$\epsilon = \Pi_{Plate} \frac{(V)^3}{W_{Plate}} \quad (27.1)$$

You may assume that the relative velocity between propeller and water is equal to 75% of the propeller velocity.

```
#answer
ratio_prop_vel = 0.75
pi_plate = 0.04
vel_prop = 3 * u.ft/u.s
height_prop = 3 * u.cm

ed_rate_prop_max = pi_plate * ((ratio_prop_vel * vel_prop)**3 / height_prop).to(u.mW/
→u.kg)

print('The maximum energy dissipation rate behind the propeller tip is', ed_rate_prop_
→max)
```

The maximum energy dissipation rate behind the propeller tip is 430 mW/kg

27.2.5 5)

Calculate the ratio of maximum to average energy dissipation rate (henceforth referred to as Max/Ave EDR) and the ratio of maximum to average velocity gradient (Max/Ave G) for the mechanical flocculator described above. What is the relationship between the two ratios?

```
#answer
ed_rate_mech_ratio = ed_rate_prop_max / ed_rate_mech_ave
print('The ratio of maximum to average energy dissipation rate is', ed_rate_mech_
→ratio,'.')

G_mech_ratio = ed_rate_mech_ratio**0.5
print('The ratio of maximum to average velocity gradient is', G_mech_ratio,'.)')
```

The ratio of maximum to average energy dissipation rate is 66.3 The ratio of maximum to average velocity gradient is 8.15

SOLUTION NOTE: this information is found on extra slide 48 in the Flocculator Design slides, heading is Results of the the CFD analysis and our model equations.

High ratios of Max/Ave for velocity gradients (and therefore also energy dissipation rates) are common in both mechanical and hydraulic flocculators. A high velocity gradient ratio is a poor design for two reasons. - This high Max/Ave G results in significant inefficiency in the use of energy for flocculation. This inefficiency requires longer residence times and/or more energy input to achieve the same extent of flocculation.

The high energy dissipation rate in the propeller wake limits the amount of energy that can be dissipated without causing excessive floc breakup. Excessive floc breakup produces flocs that are too small to be captured by the sedimentation tank.

We hypothesize that the maximum G values specified for mechanical flocculators were likely set by the constraint of not breaking flocs into small sizes. Sedimentation velocity needs to be less than capture velocity for a floc to settle - the smallest floc that will settle is referred to as the threshold floc because that is the smallest size we want to make. Because well-designed hydraulic flocculators have more uniform shear rates (as compared to mechanical flocculators), hydraulic flocculators are less likely to break up flocs. Therefore, the average G values specified for mechanical flocculators are not expected to apply to well designed hydraulic flocculators.

The high energy dissipation rates at the tip of a propeller in a flocculator may be high enough to create flocs that are too small to be captured by the sedimentation tanks. We will determine if the sedimentation tank could capture these flocs in the analysis below.

These results also suggest that the traditional emphasis on maintaining the same or lesser G value for the transport of flocculated water to the sedimentation tank may have missed the more important point of not exceeding a value of G that produces flocs that are too small to be captured by the sedimentation tank.

27.2.6 6)

A mechanical flocculator is treating high turbidity water, and therefore is using an average G value of 180 Hz.

If the design guidelines for maximum G for mechanical flocculators are correct and are based on floc breakup, then what is the largest average G that could be used for a well designed hydraulic flocculator with a Max/Ave G of $\sqrt{2}$?

Note: important distinctions to make are: - maximum G vs average G - mechanical vs hydraulic flocculators. Conceptual thought coupled with pencil and paper are recommended.

```
#answer
G_hyd_ratio = np.sqrt(2)
G_mech_ave_max = 180 * u.Hz
G_hyd_ave_max = G_mech_ave_max * (G_mech_ratio/G_hyd_ratio)
print('The maximum G for hydraulic flocculators is', G_hyd_ave_max, 'Hz')
```

The maximum G for hydraulic flocculators is 1037 Hz

These extremely high average G values are more like traditional rapid mix. It is possible that energy use constraints (too much elevation difference required to power the flocculation) will prevent use of such high G values. It is also possible that these G values would cause excessive floc break up even though they appear to meet conventional standards. It is also likely that the flocculation time required to achieve a target G θ would not be sufficient for the coagulant nanoparticles to be transported to the surfaces of suspended particles.

27.2.7 7)

How much energy in Joules per liter would be required to treat the water using this mechanical flocculator?

```
#answer
def energy_mech(Q, G, t, temp):
    return (power_floc_shaft(Q, G, t, temp) / Q).to(u.J/u.L)

print('The energy required using the mechanical flocculator is', energy_mech(flow_
    ↪plant, G_mech, time_mech, temp_design), '.')
```

The energy required using the mechanical flocculator is 11.5 J/l

27.2.8 8)

How much does the electricity cost to flocculate a million liters? It isn't necessary to actually size an electric motor for this assignment. Simply use the shaft power and assume a motor efficiency of 80%. You may assume the price of electricity is 0.15 USD/(kW-hr).

```
#answer
efficiency_motor = 0.8
electricity_rate = ((0.15 * u.USD) / (u.kW * u.hr))
electricity_cost_mech = (electricity_rate * energy_mech(flow_plant, G_mech, time_mech,
    ↪temp_design) / efficiency_motor).to(u.USD/u.ML)
print('The cost of electricity for mechanical flocculation is', electricity_cost_mech,
    ↪'.')
```

The cost of electricity for mechanical flocculation is 0.58 USD/MI

It doesn't actually cost very much to flocculate water using electricity. We will create cost savings over mechanical flocculators by designing smaller, higher performing flocculators that don't require any moving parts and thus don't require much maintenance. Our capital costs will also be lower because we use more efficient plug flow reactors to prevent short circuiting of particles through the flocculator. Thus well designed hydraulic flocculators can be smaller than mechanical flocculators.

27.2.9 9)

What is the equivalent amount of potential energy that is used to operate this mechanical flocculator (the shaft power) expressed as an elevation drop in meters? What is the required shaft power?

```
#answer
delta_height = (power_floc_shaft(flow_plant, G_mech, time_mech, temp_design) / (flow_
    ↪plant * pc.density_water(temp_design) * pc.gravity)).to(u.m)

print('The equivalent amount of potential energy to run this mechanical flocculator_
    ↪is', delta_height, '.')

print('The shaft power required for this flocculator is ', power_floc_shaft(flow_
    ↪plant, G_mech, time_mech, temp_design), '.')
```

The equivalent amount of potential energy to run this mechanical flocculator is 1.17 m 0.5730957660614814 kilowatt

27.2.10 10)

What is the required reactor volume for the mechanical flocculator?

```
#answer
vol_mech = (time_mech * flow_plant).to(u.m**3)
print('The required reactor volume for the mechanical flocculator is', vol_mech,'.')
```

The required reactor volume for the mechanical flocculator is 90.0 meter ** 3

27.2.11 11)

If this flocculator is 4 m deep, then how many square meters of plan view area are required per L/s of flow capacity? This is a measure of required size of this unit process. For comparison, an AquaClara sedimentation tank requires about $\frac{1m^2}{L/s}$ and are only 2 m deep.

```
#answer
depth_mech = 4 * u.m
area_mech = (vol_mech / (depth_mech * flow_plant))
print('The required plan view area is', area_mech)
```

The required plan view area is 0.45 meter ** 2 * second / liter

CHAPTER
TWENTYEIGHT

FLOCCULATION MODEL SOLUTION

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u

import aguaclara.research.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt
```

Many of the fractal floc equations are available in the `floc_model.py` file in the `aguaclara` repository. Look through ``floc_model.py <https://github.com/AguaClara/aguaclara/blob/master/aguaclara/research/floc_model.py>`_` within the `aguaclara` repository. The following constants are defined in that file. NTU has been defined as an approximate empirical relationship between the concentration of kaolin clay and the turbidity, such that 1 NTU is equivalent to 1.7 mg/L. The diameter of a primary clay particle is assumed to be 7 micrometers. The fractal dimension for flocs is defined as `DIM_FRACTAL` and is equal to 2.3. We are using PACl as a coagulant for this analysis, so you shall call `fm.PACl`, when a function within `floc_model.py` requires `coag` as an input.

Whenever possible, use variables defined within `floc_model.py` instead of redefining them. Relevant variables defined in `floc_model.py` include: 1. `DIM_FRACTAL` with an estimated value of 2.3.

1. PACl
2. Clay

28.1 1)

Estimate the diameter of the flocs that interact with the tip of the impeller of the mechanical flocculator analyzed above. We dont yet have a good model to predict maximum floc size as a function of velocity gradient or energy dissipation rate. We have a rough estimate, `fm.diam_floc_max(EDRmax)` based on a small amount of data.

```
ratio_prop_vel = 0.75
pi_plate = 0.04
vel_prop = 3 * u.ft/u.s
height_prop = 3 * u.cm

ed_rate_prop_max = pi_plate * ((ratio_prop_vel * vel_prop)**3 / height_prop).to(u.mW/
→u.kg)

#answer
diam_floc_mech = fm.diam_floc_max(ed_rate_prop_max).to(u.um)
print('The diameter of the flocs that interact with the impeller is', diam_floc_mech,
→'.'
```

(continues on next page)

(continued from previous page)

```
x=fm.DIM_FRACTAL  
x
```

The diameter of the flocs that interact with the impeller is 127 um.

28.2 2)

Estimate the terminal sedimentation velocity in mm/s of the flocs that interact with the tip of the impeller of the mechanical flocculator analyzed above. Use the function `fm.vel_term_floc`. You may assume that the flocs were made from a particle suspension that had 1.5 mg/L of aluminum and 100 NTU of clay.

Note: AquaClara has defined the unit NTU as u.NTU.

```
#answer  
conc_Al = 1.5 * u.mg/u.L  
conc_clay=100*u.NTU  
vel_term_floc_tip = fm.vel_term_floc(conc_Al, conc_clay, fm.PACl,  
                                      fm.Clay, fm.DIM_FRACTAL,  
                                      diam_floc_mech, temp_design).to(u.mm/u.s)  
  
print('The terminal velocity of flocs that interact with the impeller tip is   
      ↪estimated to be', vel_term_floc_tip)
```

The terminal velocity of flocs that interact with the impeller tip is estimated to be 0.738 mm/s

28.3 3)

Would these flocs be captured by a conventional design for a sedimentation tank ([10 State Standards](#)) with a capture velocity of 1.2 m/hr? The capture velocity is a property of the sedimentation tank. If the floc settles faster than the capture velocity, then theoretically the floc will be captured by the sedimentation tank.

```
#answer  
vel_capture_10_state = (1.2 * u.m/u.hr).to(u.mm/u.s)  
print('The 10 State Standards capture velocity is', vel_capture_10_state)  
print("The 10 State Standards sedimentation tank would capture the flocs that are   
      ↪able to survive the energy dissipation rate at the tip of the propeller. ")
```

The 10 State Standards capture velocity is 0.333 mm/s The 10 State Standards sedimentation tank would capture the flocs that are able to survive the energy dissipation rate at the tip of the propeller.

These flocs would be removed easily in an AquaClara sedimentation tank (capture velocity of 0.12 mm/s). However, our use of the empirical equation to predict the size of these flocs is questionable because we are extrapolating way beyond the original data. We need more experiments to characterize the size of flocs as a function of the velocity gradient.

28.4 4)

Estimate the average distance between primary clay particles at the beginning and end of flocculation given an initial turbidity of 100 NTU and a target effluent unflocculated clay concentration at the end of flocculation of less than 1 NTU. Of course, the clay concentration is actually constant in flocculation since particles are not actually being

removed. But here we are referring to the primary clay particles that have escaped aggregation and thus are still unattached.

You can do this by figuring it out empirically (brownie points!) or by looking for a function that finds average distance between particles.

A little extra to think about (not necessary to answer): The AquaClara floccuation model assumes that primary clay particles mostly attach to other primary clay particles and not to larger flocs (aggregates of clay particles). Can you think of why this is?

```
#answer
init_sep_dist_clay = fm.sep_dist_clay(100 * u.NTU, fm.Clay).to(u.mm)
final_sep_dist_clay = fm.sep_dist_clay(1 * u.NTU, fm.Clay).to(u.mm)
print('The average distance between clay particles at 100 NTU is', init_sep_dist_clay)
print('The average distance between clay particles at 1 NTU is', final_sep_dist_clay)
```

The average distance between clay particles at 100 NTU is 0.141 mm The average distance between clay particles at 1 NTU is 0.654 mm

28.5 5)

What is the inner viscous length scale in the mechanical flocculator at the maximum energy dissipation rate? Given that this is a very high energy dissipation rate for flocculation, it corresponds to a very small inner viscous length scale. This means that eddies are able to survive down to a small size before viscosity damps their motion. If the separation distance between clay particles that havent turned into flocs is less than this inner viscous scale, then it is reasonable to assume that all flocculation is dominated by viscosity. The function within `floc_model.py` that does this is confusingly named `lambda_vel()`.

```
#answer
print('The inner viscous length scale is', fm.lambda_vel(ed_rate_prop_max, temp_
    ↪design).to(u.mm))
```

The inner viscous length scale is 2.39 mm

28.6 6)

Below is a graph showing the inner viscous length scale that divides flows that are dominated by inertia (eddies) from flows where viscosity is significant. **Add the data point** representing the maximum energy dissipation rate vs the maximum clay separation distance at the end of flocculation for the mechanical flocculator you have been designing.

```
#This code is provided to help you make your graph

#Creates the array for energy dissipation rates (EDRs)
x = np.logspace(np.log10(1), 4)*u.mW/u.kg

plt.figure('Inner Viscous Scale', (6, 6))
plt.title('Inner Viscous Scale vs Energy Dissipation Rate')

ax.set(ylabel='Inner Viscous Scale (mm)')
ax.set(xlabel='Energy Dissipation Rate (mW/kg)')

plt.yscale('log')
plt.xscale('log')
```

(continues on next page)

(continued from previous page)

```

plt.grid(b=True, which='major', color='k', linestyle='-', linewidth=1)
plt.grid(b=True, which='minor', color='k', linestyle='--', linewidth=0.5)

#fm.lambda_vel, which returns the inner viscous length scale,
#is being applied to the array of EDRs for our design temperature
y = fm.lambda_vel(x,temp_design)
ax.plot(x, y.to(u.mm), 'r')

#-----
#----- YOUR CODE BELOW -----
#-----

#answer
ax.plot(ed_rate_prop_max.to(u.mW/u.kg), final_sep_dist_clay.to(u.mm), 'ko')

print(final_sep_dist_clay)
print(ed_rate_prop_max)

plt.show()

```

0.6541789493547243 millimeter 422.57902694348155 milliwatt / kilogram

28.7 7)

According to the plot and analysis above, are the collisions between clay particles at a concentration of 1 NTU dominated by inertia or by viscosity? Explain why!

The final spacing between clay particles is still smaller than the inner viscous length scale at which eddies are damped by viscosity. This suggests that all collisions in flocculation are dominated by viscosity.

28.7.1 Real-world considerations of flocculation

Now that you have an augmented understanding of flocculation theory, we can consider a few ways in which the theory applies to real-world flocculators.

In this section, there are no calculations for you to do or code for you to write - everything has been provided for you. This was done to shorten this design challenge while still detailing relevant and important information.

There are two conceptual questions for you to answer at the end of the section. Read through and focus on understanding the concepts before you try to answer the questions.

Coagulant distribution in a reactor

The flocculation model accounts for loss of coagulant nanoparticles to the reactor walls. The loss of coagulant nanoparticles is assumed to scale with the area of the flocculator walls divided by the total area of clay and flocculator walls. This loss is significant for low turbidity and small scale flocculators, such as the 1 liter per second flocculator AquaClara recently designed.

Here we will consider a flocculator built out of pipe, not one contained within a rectangular reactor. The diam_tube parameter is the flocculator diameter and is needed to estimate how much of the coagulant is lost to the walls of the flocculator. We will assume the flocculator tube is for the 1 L/s plant and has a diameter of 7.5 cm

We will evaluate the situation where the turbidity is 10 NTU and the coagulant dose is 1 mg/L of aluminum. The code below does the following:

- Estimates the fraction of coagulant nanoparticles lost to the flocculator walls.
- Estimates the fraction of the clay surface area that is coated with nanoparticles.

```
diam_tube = 7.5 * u.cm
conc_clay = 100*u.NTU
conc_Al = 0.5*u.mg/u.L

#fm.ratio_area_clay_total() returns fraction (between 0 and 1) that represents the_
#surface area of
#the clay particle over the sum of the surface area of the clay and reactor walls

wall_loss = 1 - fm.ratio_area_clay_total(conc_clay, fm.Clay, diam_tube, fm.RATIO_
#HEIGHT_DIAM)
print('The fraction of the coagulant lost to the walls is', wall_loss)

#fm.gamma_coag() returns the fraction of clay that is covered by coagulant. This is a_
#very hard parameter
#to actually measure, so this is just an estimate.

fraction_coated = fm.gamma_coag(conc_clay, conc_Al, fm.PACl,
                                 fm.Clay, diam_tube, fm.RATIO_HEIGHT_DIAM)
print('The fraction of the clay surface area that is is coated is', fraction_coated)
```

Time scale of flocculation

Now we want to estimate the average time required for an initial successful collision between two primary clay particles that are partially coated with coagulant nanoclusters. Note that for the first collision, the current floc size is the same as the clay size. We will use the average energy dissipation rate for the mechanical flocculator as found above.

```
ed_rate_mech_ave = 6.4*u.mW/u.kg

time_first_collision = fm.time_col_laminar(ed_rate_mech_ave, 10*u.degC,
                                             conc_Al, conc_clay, fm.PACl,
                                             fm.Clay, fm.Clay.Diameter, diam_tube,
                                             fm.DIM_FRACTAL, fm.RATIO_HEIGHT_DIAM) .
#to (u.s)

print('The time required for the first succesful collision is', time_first_collision)
```

This collision time is quite fast and is the origin of the question, why does flocculation require 30 minutes? as mandated in the Ten State Standards.

AquaClara flocculation model

We will now briefly consider an AquaClara flocculator design with an average energy dissipation rate of approximately 11 mW/kg and a residence time of 8.1 minutes. The design temperature is 15 degC.

Below is a calculation for the Gt value of this flocculator.

```
#answer
ed_rate_floc_aguaclara = 11*u.mW/u.kg
time_floc_aguaclara = 8.1*u.minute
temp_design_aguaclara = 15*u.degC

#This equation for G can be found in the course slides (all equations you see in_
#design challenges can be found in the slides)
G_floc_aguaclara = np.sqrt(ed_rate_floc_aguaclara/pc.viscosity_kinematic(temp_design_
#aguaclara))
Gtime_floc_aguaclara = (G_floc_aguaclara*time_floc_aguaclara).to(u.dimensionless)
print('The AquaClara Gt value is', Gtime_floc_aguaclara)
```

Coagulant coverage fraction of a particle

This section solves the integrated flocculation model for Γ . We simplify the model by recognizing that the spacing between particles at the end of the flocculation process is much greater than the initial particle spacing. This means that the raw water turbidity drops out of the equation. The value of the rate constant for collisions is $k = 0.24$. We start with the equation below:

$$\Gamma = \frac{3}{2} \cdot \frac{\Lambda^2}{k\pi d_p^2 Gt} \quad (28.1)$$

We then estimate the required coagulant coverage of clay, Γ , for the AquaClara flocculator to achieve a 2 NTU settled water turbidity when starting with a raw water that is 50 NTU.

Note that the specified flocculation model applies to both hydraulic and mechanical flocculators.

```
#Fitting constant/Sedimentation tank factor
k = 0.24

#gamma_aguaclara_design uses functions in floc_model.py to solve the equation in the_
#problem statement
conc_clay_goal = 2 * u.NTU
gamma_aguaclara_design = (3/2) * (fm.sep_dist_clay(conc_clay_goal, fm.Clay)**2
                                   / (k * np.pi * (fm.Clay.Diameter * u.m)**2
                                      * Gtime_floc_aguaclara
                                   )
                               ) .to(u.dimensionless)

print('The Gamma value is', gamma_aguaclara_design)
```

Residence time and coagulant coverage

If you doubled the residence time of the flocculator, the required coagulant coverage of clay changes according to the model. By doubling the residence time, the required coagulant coverage is reduced by a factor of 2.

Modeling flocculation in the presence of humic acid, with pC* as the performance metric

The flocculation model predicts the settled water turbidity given the composition of the raw water, the flocculator characteristics, and a fitting parameter that must be a function of the sedimentation tank characteristics. This fitting parameter is k , which is the same as the rate constant for collisions described above. The model is far from complete - it doesn't yet describe the effects of floc blankets. Below we have created a plot showing model predictions for a

range of coagulant and humic acid (dissolved organic matter) concentrations. The plot uses our approximation for pC^* described in class and shown below:

$$pC^* = \frac{3}{2} \log\left(\frac{2}{3}\pi k \frac{d_p^2}{\Lambda_0^2} Gt\alpha + 1\right) \quad (28.2)$$

```
#Define the range of coagulant. This is necessary to create plots of pC* as a
#function of coagulant dose.
coag_graph = np.linspace(0.01, 2.5, 100) * u.mg/u.L

# Graph results of a particular NTU. Note that you can change this value to see how
#the graph responds.
# A change here even changes the graph title changes!
plot_NTU = 50*u.NTU

plt.figure(str(plot_NTU), (6,6))
plt.title(str(plot_NTU) + ' Graph for Various Humic Acid Concentrations')
ax.set(ylabel='pC*')
ax.set(xlabel='coagulant dosage (mg/L)')

# Create an array of humic acid concentrations
plot_humic_acid = np.linspace(0,15,6)*u.mg/u.L

# Create a function that only has inputs for the values that we will change between
#plots.
#All other variables are taken from predefined values. This simplifies the function
#call for use in generating the plots.
def plot_pc(conc_humic_acid):
    k = 0.24
    # The energy dissipation rate for aquaclar designs
    ed_rate = 11*u.mW/u.kg
    #The inner diameter of the flocculator tube is important because a significant
    #fraction of the
    #coagulant ends up attaching to the flocculator walls
    tube_diam = 3/8 * u.inch
    time_floc = 8.1 * u.minute
    temp = 15 * u.degC
    #pc_viscous is the solution for the equation in the problem statement. It returns
    #pC*
    plot_pc = fm.pc_viscous(ed_rate, temp, time_floc, tube_diam,
                            plot_NTU, coag_graph, conc_humic_acid,
                            fm.HumicAcid, fm.PACl, fm.Clay,
                            k, fm.RATIO_HEIGHT_DIAM)
    return plot_pc

x = coag_graph.to(u.mg/u.L)

ax.plot(x, plot_pc(plot_humic_acid[0]), 'r',
        x, plot_pc(plot_humic_acid[1]), 'b',
        x, plot_pc(plot_humic_acid[2]), 'g',
        x, plot_pc(plot_humic_acid[3]), 'm',
        x, plot_pc(plot_humic_acid[4]), 'c',
        x, plot_pc(plot_humic_acid[5]), 'y')

#We can use the array of humic acid concentrations to directly create the legend!
plt.legend(plot_humic_acid, loc = 'best')
plt.show()
```

There is a lot to learn from this graph!!!!!! It appears that for any given coagulant dose, humic acid concentration

significantly affects pC*. Additionally, notice the diminishing returns of adding more coagulant. This effect appears to be independent of humic acid concentration (see the red curve).

Modeling flocculation in the presence of humic acid, with settled water turbidity as the performance metric

We will now display a similar plot which shows settled water turbidity instead of pC*. Our initial turbidity is 10 NTU, and we will four curves for separate humic acid concentrations.

```
#answer
plot_NTU = 10*u.NTU

plt.figure(str(plot_NTU), (6,6))
plt.title(str(plot_NTU) + ' Graph')
ax.set(ylabel='Settled water turbidity (NTU)')
ax.set(xlabel='coagulant dosage (mg/L)')

def plot_conc_clay(conc_clay, conc_nat_org_mat):
    k = 0.24
    ed_rate = 11*u.mW/u.kg
    #The inner diameter of the flocculator tube is important because a significant
    ↪fraction of the
    #coagulant ends up attaching to the flocculator walls
    tube_diam = 3/8 * u.inch
    conc_clay = plot_NTU
    time_floc = 8.1 * u.minute
    temp = 15 * u.degC
    #s_t calls on fm.pc_viscous() like the previous cell of code, but also uses the
    ↪fm.invp()
    #function to turn pC* back into units of settled water turbidity.
    s_t = fm.invp(fm.pc_viscous(ed_rate, temp, time_floc, tube_diam,
                                 conc_clay, coag_graph, conc_nat_org_mat, fm.HumicAcid, fm.PACl, fm.
    ↪Clay,
                                 k, fm.RATIO_HEIGHT_DIAM), conc_clay)

    return s_t

#Creates array of humic acid concentrations. This was done in the previous code cell
↪with np.linspace,
#but because we don't want evenly spaced concentrations here we input our desired
↪values manually
plot_humic_acid = np.array([0,1,5,20])*u.mg/u.L

x = coag_graph.to(u.mg/u.L)
ax.plot(x, plot_conc_clay(plot_NTU, plot_humic_acid[0]), 'r',
        x, plot_conc_clay(plot_NTU, plot_humic_acid[1]), 'b',
        x, plot_conc_clay(plot_NTU, plot_humic_acid[2]), 'g',
        x, plot_conc_clay(plot_NTU, plot_humic_acid[3]), 'y')

plt.legend(plot_humic_acid, loc = 'best')
plt.show()
```

Looking at the interactions between coagulant, clay, and humic acid from this perspective yields even more fun discoveries! For increasing humic acid concentration, more coagulant is required to even begin the process of flocculation.

28.8 8)

Why does the AquaClara flocculation model predict that adding 1 mg/L of aluminum has no effect on turbidity when the humic acid concentration is 20 mg/L?

At low concentrations of coagulant every coagulant nanoparticle surface is completely coated with humic acid and thus they aren't sticky at all.

28.9 9)

It is tempting to assume that all the coagulant dosed gets attached to clay particles. However, if a plant operator were to make this assumption, their plant would produce low-quality water.

Identify and explain two significant reasons as to why this assumption fails.

1. Coagulant is lost to the walls of the reactors
2. Coagulant is lost to humic acid (Students need to write more than this)

CHAPTER
TWENTYNINE

FLOCCULATOR DESIGN SOLUTION

```
#import statements have been cleaned up
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.core.constants as constants
import numpy as np
import matplotlib.pyplot as plt
# plt.switch_backend('TKAgg')
```

In this design challenge you will design a vertical flow hydraulic flocculator. You will use the flow rate of `flow_plant = 20 L/s` as your default design value.

Although we dont require that you use the nomenclature from the AguacLara variable naming guide, it provides a reasonable basis for organizing your variable names. We use the convention that the first part of the variable name describes the type of variable and always has a unique dimension. The subsequent parts of the variable name start from the big picture and slowly add more and more detail to arrive at the precise component that you are describing. Thus `space_floc_baffle` is the spacing between baffles in the flocculator and `n_floc_channel_baffle` should be the number of baffles in a flocculator channel. `width_floc_channel_port` should be the width of the port connecting two flocculator channels.

In this design challenge we will use functions to calculate each step of the design. Whenever we need to access a previously calculated result we will use the function call to access that value. Using this method of nested function calls will make the entire design be only a function of a few input parameters. This makes it possible to have flexible design code and ease the creation of new plant designs.

The minimum input parameters to define a flocculator are (`flow_plant`, `headloss_floc_BOD`, `Gt_BOD`, `T_BOD`). We could have included a longer list of input parameters (`height_floc_end`, `width_PC_sheet`, `K_e`, etc.) to make our functions even more general. But to keep our code more concise we will focus on only 4 of the input parameters. This will make it possible to easily change the input parameters to obtain new designs. We will use this capability to plot the results of varying the design flow rate.

29.1 Hydraulic Vertical Flow Flocculator Design

This challenge is design a hydraulic flocculator using the core concepts of the AguacLara design methodology.

Below are the inputs for the design. BOD stands for Basis Of Design.

```
# head loss through the flocculator
headloss_floc_BOD = 40 * u.cm

# collision potential based on recent designs for AguacLara Plants
Gt_BOD = 37000
```

(continues on next page)

(continued from previous page)

```

# water depth at the end of flocculator where it flows into the inlet channel of the
# sedimentation tank
height_floc_end = 2 * u.m

flow_plant = 20 * u.L/u.s

# This is the estimate for larger plants where the flocculator is as long as the
# sedimentation tanks.
# For lower flow plants we will need to reduce this length because of the constraint
# that the channels must
# be wide enough to construct.
length_channel_max = 6 * u.m

# minimum and maximum ratios of distance between expansions to baffle spacing
Pi_HS_min = 3
Pi_HS_max = 6

Pi_vc = constants.VC_ORIFICE_RATIO
Pi_vc_baffle = Pi_vc**2

# width of the polycarbonate sheets used to make baffles
width_PC_sheet = 1.067*u.m

# this is a reasonable constraint to keep the channel constructable by humans
width_floc_min_BOD = width_PC_sheet / 2
width_floc_max_BOD = width_PC_sheet

# expansion minor loss coefficient for 180 degree bend
K_e = (1 / Pi_vc_baffle - 1)**2

# this is the minimum temperature of the raw water
T_BOD = 15* u.degC

```

29.1.1 Design Algorithm Steps

1. Calculate the total volume of flocculator given head loss and collision potential
2. Calculate the number of channels by taking the total width and dividing by the maximum channel width
3. Calculate the channel width (total width over number of channels)
4. Calculate the minimum number of obstacles and spacing between obstacles by assuming a maximum H/S ratio
5. Use the actual H/S ratio based on obstacle spacing to calculate the spacing between baffles
6. Calculate the obstacle width to obtain the same jet expansion conditions as produced by the 180 degree bend.

This algorithm does not yet handle the integer number of baffles in a channel or the thickness of the baffles

For this assignment **create functions for each question** and then use those function calls whenever that calculation result is required in a subsequent step. Include flow_plant, headloss_floc_BOD, Gt_BOD and T_BOD as the inputs that can be varied for each function.

Whenever you reference a previously calculated value in a function, use the function call for that value so that dependency is not broken.

That way you will be able to easily vary any of the three main input parameters (flow, Gt, head loss) to see their effect on the design.

1)

Estimate the average velocity gradient of a flocculator given head loss and collision potential. I'm going to solve this problem for you so you see how to use function calls.

```
def G_avg(hl, Gt, T):
    G = (pc.gravity * hl) / (Gt * pc.viscosity_kinematic(T_BOD))
    return G.to(1/u.s)

print ('The average velocity gradient of flocculator is', G_avg(headloss_floc_BOD, Gt_BOD, T_BOD))
```

The average velocity gradient of flocculator is 93.24 / second

2)

Estimate the residence time of flocculator given the target head loss and collision potential. (Note that this ignores the decrease in water depth caused by head loss. We hope to improve this design process further in the near future.)

```
def t_floc_est(hl, Gt, T):
    theta = Gt / G_avg(hl, Gt, T)
    return theta.to(u.s)

print ('The residence time of flocculator is', t_floc_est(headloss_floc_BOD, Gt_BOD, T_BOD))
```

The residence time of flocculator is 396.8 second

3)

Plot $G\theta$ as a function of the operating temperature given the head loss and residence time for this design. In this step you are assuming that you have built this hydraulic flocculator and you want to see how the collision potential, $G\theta$, varies with temperature of operation. Vary the temperature from 0°C to 30°C. The following equation makes it clear that the velocity gradient originates from head loss that creates fluid deformation that is limited by viscosity.

$$\bar{G}\theta = \sqrt{\frac{gh_e\theta}{\nu}} \quad (29.1)$$

```
def Gt_operation(hl,Gt,T_BOD,T):
    Gt_operation = np.sqrt(pc.gravity * hl * t_floc_est(hl, Gt, T_BOD)/pc.viscosity_kinematic(T))
    return Gt_operation.to(u.dimensionless)
Temp_Operation=np.linspace(0, 30)*u.degC
ypoints=(Gt_operation(headloss_floc_BOD, Gt_BOD, T_BOD,Temp_Operation))
fig, ax = plt.subplots()
ax.plot(Temp_Operation,ypoints,'-')
#Note the use of latex. The r tells matplotlib to interpret the following strings as raw text.
#Without the "r" the \theta would be parsed as a tab (\t) followed by heta!
ax.set(xlabel=r'Temperature ($^\circ$C)')
ax.set(ylabel=r'G$\theta$')
fig.savefig('Flocculation/Images/Gtheta_vs_temperature')
plt.show()
```

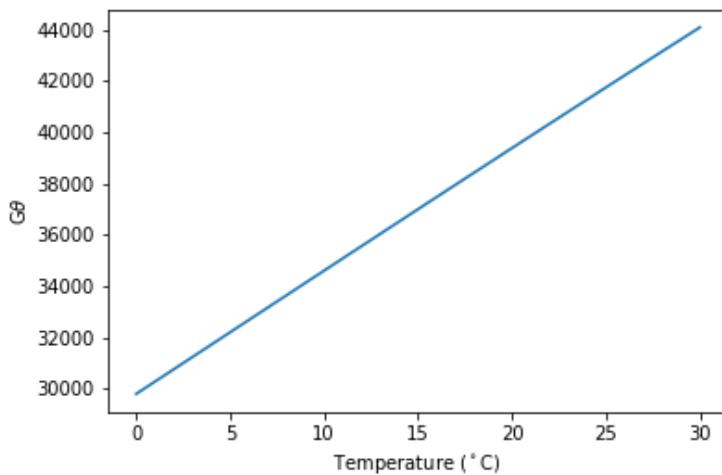


Fig. 29.1: The $\bar{G}\theta$ delivered by a flocculator increases with temperature. Note that the apparent linearity of this graph is because the inverse of the square root of kinematic viscosity varies linearly with temperature over this range!

4)

The Equation (22.26) creates the link between $\bar{G}\theta$ and flocculator performance. What does the floc model and Fig. 29.1 tell you about flocculator performance and flocculator design? Explain why performance varies with temperature. Explain how temperature influences collisions between particles. What temperature should be used to design flocculators?

Flocculator performance improves with warmer temperatures because the fluid deforms more given the same input energy. The fluid deforms more because it is less viscous at higher temperatures. Flocculators should be designed to perform well based on the coldest operating temperature that they will encounter.

5)

Calculate the volume of flocculator. Note that this volume does not take into account the extra volume that flocculator will have due to the changing water level caused by the head loss. Simply estimate the volume based on the residence time and the flow rate.

```
def vol_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    vol = t_floc_est(headloss_floc_BOD, Gt_BOD, T_BOD )*flow_plant
    return vol.to(u.m**3)

print('The volume of flocculator is', vol_floc(flow_plant, headloss_floc_BOD, Gt_BOD,
    ↴T_BOD))
```

The volume of flocculator is 7.936 meter ** 3

6)

Calculate the actual length of the flocculator channels. This must meet two constraints. First, it must be less than or equal to the maximum channel length. Second, the channel length is limited by the flocculator volume, height, minimum number of channels, and minimum width of the channels. This second constraint is important for low flow

rates so that the flocculator has the correct target volume. Make sure to use this floc channel length in subsequent calculations.

```
def length_channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    #The factor of two below is because the minimum number of channels in the
    #flocculator is two.
    #This is a result of the orientation of the sedimentation tanks and the plumbing
    #in the plant.
    #Unless the design is for very low flows (<10 L/s), there will always be an even
    #number of flocculator channels.
    length = vol_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)/width_floc_min_
    #BOD/2/height_floc_end
    return min(length,length_channel_max)

print('The maximum channel length is',length_channel(flow_plant, headloss_floc_BOD,
    #Gt_BOD, T_BOD))
```

The maximum channel length is 3.719 meter

7)

Calculate the combined total width of the flocculator channels (not including walls) based on the given length and depth.

```
def width_floc_total(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    width_floc = vol_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)/(length_
    #channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)*height_floc_end)
    return width_floc.to(u.m)

print ('The total width of the flocculator channels is is', width_floc_total(flow_
    #plant, headloss_floc_BOD, Gt_BOD, T_BOD))
```

The total width of the flocculator channels is 1.067 meter

8)

Calculate the minimum channel width required to achieve H/S>3. The channel can be wider than this, but this is the absolute minimum width for a channel. The minimum width occurs when there is only one expansion per baffle and thus the distance between expansions is the same as the depth of water at the end of the flocculator.

$$W_{Min} = \frac{\Pi_{HS}Q}{H_e} \left(\frac{K_e}{2H_e \nu G^2} \right)^{\frac{1}{3}}$$

```
def width_floc_min_est(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    Gavg = G_avg(headloss_floc_BOD, Gt_BOD, T_BOD)
    nu = pc.viscosity_kinematic(T_BOD)
    width_floc_min = Pi_HS_min*((K_e/(2 * height_floc_end * (Gavg**2) * nu))** (1/
    #3))*flow_plant/height_floc_end
    return width_floc_min.to(u.cm)

print('The minimum channel width is', width_floc_min_est(flow_plant, headloss_floc_
    #BOD, Gt_BOD, T_BOD))
```

The minimum channel width is 11.64 centimeter

9)

What is the minimum channel width given the additional constraint that must be built by humans? Use the max function to find the true minimum channel width given both constraints.

```
def width_floc_min(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    return max(width_floc_min_est(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD), width_
    ↪floc_min_BOD)

print('The minimum channel width is', width_floc_min(flow_plant, headloss_floc_BOD, ↪
    ↪Gt_BOD, T_BOD))
```

The minimum channel width is 0.5335 meter

10)

Calculate the number of channels by taking the total flocculator width (see step 7) and dividing by the minimum channel width (round down). Include the requirement that the number of channels must be even (Use the numpy floor function - look it up!). To make this function robust, make sure that it can't ever return zero channels (the max function might be useful here)! You can convert the float to an integer with the int() function.

```
def num_channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    num = (width_floc_total(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD) /
           (width_floc_min(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD))).to(u.
    ↪dimensionless)
    # floor function with step size 2
    num = np.floor(num/2)*2
    return int(max(num, 2))

print('There are', num_channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD),
    ↪'channels.')
```

There are 2 channels.

11)

Calculate the actual channel width based on the number of channels and the total flocculator width.

```
def width_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    width_total = width_floc_total(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)
    num_c = num_channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)
    return (width_total/num_c).to(u.cm)

print('The actual flocculator channel width is', width_floc(flow_plant, headloss_floc_
    ↪BOD, Gt_BOD, T_BOD))
```

The actual flocculator channel width is 53.35 centimeter

12)

Calculate the *maximum* distance between expansions. This occurs for the largest allowable H/S ratio. Note that this isn't accounting for the integer requirement for the number of baffle spaces per channel yet.

$$H_{e_{Max}} = \left[\frac{K_e}{2\nu\bar{G}^2} \left(\frac{Q\Pi_{HS_{Max}}}{W} \right)^3 \right]^{\frac{1}{4}} \quad (29.2)$$

```

def height_exp_max(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    g_avg = G_avg(headloss_floc_BOD, Gt_BOD, T_BOD)
    nu = pc.viscosity_kinematic(T_BOD)
    term1 = (K_e/(2 * (g_avg**2) * nu))**(1/4)

    term2 = (Pi_HS_max*flow_plant/width_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_
    ↵BOD))** (3/4)

    height_exp = term1*term2
    return height_exp.to(u.m)

print('The maximum distance between expansions', height_exp_max(flow_plant, headloss_
    ↵floc_BOD, Gt_BOD, T_BOD))

```

The maximum distance between expansions 1.074 meter

13)

Calculate the minimum number of expansions per baffle space.

```

def num_expansions(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    return int(np.ceil(height_floc_end/(height_exp_max(flow_plant, headloss_floc_BOD,_
    ↵Gt_BOD, T_BOD))).to(u.dimensionless))

print('The number of expansions is', num_expansions(flow_plant, headloss_floc_BOD, Gt_
    ↵BOD, T_BOD))

```

The number of expansions is 2

14)

Calculate the actual distance between expansions given the integer requirement for the number of expansions per flocculator depth.

```

def height_exp(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    return height_floc_end/num_expansions(flow_plant, headloss_floc_BOD, Gt_BOD, T_
    ↵BOD)

print('The actual distance between expansions is', height_exp(flow_plant, headloss_
    ↵floc_BOD, Gt_BOD, T_BOD))

```

The actual distance between expansions is 1 meter

15)

Calculate the spacing between baffles based on the target velocity gradient using (23.30).

```

def spacing_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    g_avg = G_avg(headloss_floc_BOD, Gt_BOD, T_BOD)
    nu = pc.viscosity_kinematic(T_BOD)

```

(continues on next page)

(continued from previous page)

```
term1 = (K_e/(2 * height_exp(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD) * (g_
↪avg**2) * nu))**(1/3)

ans = term1*flow_plant/width_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)
return ans.to(u.m)

print ('The spacing between baffles is', spacing_floc(flow_plant, headloss_floc_BOD,_
↪Gt_BOD, T_BOD))
```

The spacing between baffles is 0.1832 meter

16)

How many baffle spaces would fit in the channel(s) given the length of the flocculator and the baffle spacing? Round to the nearest integer.

```
def num_baffles(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    num = round(num_channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)*length_
↪channel(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)/spacing_floc(flow_plant,_
↪headloss_floc_BOD, Gt_BOD, T_BOD))
    return int(num)

print ('The number of baffle spaces that would fit in the channels is', num_
↪baffles(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD))
```

The number of baffle spaces that would fit in the channels is 41

17)

How many baffle spaces are needed to create the required collision potential? Note that this isn't necessarily the same number as found in Problem 16. Calculating the collision potential per baffle space is the advised first step.

```
def Gt_baffle(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    term1 = spacing_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)*width_
↪floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD)
    ans = term1*G_avg(headloss_floc_BOD, Gt_BOD, T_BOD)*height_floc_end/flow_plant
    return ans.to(u.dimensionless)

print ('The collision potential (Gt) per baffle space is', Gt_baffle(flow_plant,_
↪headloss_floc_BOD, Gt_BOD, T_BOD))

def num_baffle_min(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    ans = round(Gt_BOD/Gt_baffle(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD))
    return int(ans)

print ('The minimum number of baffles required is', num_baffle_min(flow_plant,_
↪headloss_floc_BOD, Gt_BOD, T_BOD))
```

The collision potential (Gt) per baffle space is 944 dimensionless. The minimum number of baffles required is 41

18)

Do the two estimates of the number of baffle spaces agree?

```
if num_baffles(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD) == num_baffle_min(flow_
    ↪plant, headloss_floc_BOD, Gt_BOD, T_BOD):
    print('Yes')
else:
    print('No')
```

Yes

19)

Calculate the average velocity of the water in the flocculator. This is the velocity after the flow has expanded through each baffle/obstacle.

```
def vel_floc_ave(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    ans = flow_plant/( spacing_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_
        ↪BOD)*width_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD))
    return ans.to(u.m/u.s)

print ('The average velocity of the water in the flocculator is', vel_floc_ave(flow_
    ↪plant, headloss_floc_BOD, Gt_BOD, T_BOD))
```

The average velocity of the water in the flocculator is 0.2046 meter / second

20)

Calculate the depth of water at the beginning of the flocculator based on the design head loss.

```
HEIGHT_WATER_FLOC_START = height_floc_end + headloss_floc_BOD
print ('The depth of the water at the beginning of the flocculator is', HEIGHT_WATER_
    ↪FLOC_START)
```

The depth of the water at the beginning of the flocculator is 2.4 meter

21)

Estimate the residence time in the hydraulic flocculator taking head loss into account. It is okay if your estimate doesn't capture all of the details of the flocculator. You don't need to account for the volume of the baffles. Simply account for the added water due to head loss. You can approximate the extra depth as a triangle.

```
def theta_floc(flow_plant, headloss_floc_BOD, Gt_BOD, T_BOD):

    theta_est = t_floc_est(headloss_floc_BOD, Gt_BOD, T_BOD)
    ans = theta_est*((height_floc_end + (headloss_floc_BOD/2))/height_floc_end)
    return ans.to(u.min)

print ('The residence time in the hydraulic flocculator is', theta_floc(flow_plant,
    ↪headloss_floc_BOD, Gt_BOD, T_BOD))
```

The residence time in the hydraulic flocculator is 7.275 minute

22)

Create plots showing number of channels, number of expansions per water depth, total number of baffles, and channel width for a flow range from 10-100 L/s. Note that the functions that we created in this design challenge are not able to handle arrays as inputs. Use for loops to create the numpy arrays of y data needed for these graphs. Use 100 points to define each plot. Remember to initialize the numpy arrays before

```
plot_points=100
flow_plant=np.linspace(10,100, num=plot_points, endpoint=True)*u.L/u.s

expansions=np.zeros(plot_points)
for i in range(plot_points):
    expansions[i] = num_expansions(flow_plant[i], headloss_floc_BOD, Gt_BOD, T_BOD)
fig, ax = plt.subplots()
ax.plot(flow_plant,expansions,'-')
ax.set(xlabel='Plant Flow Rate (L/s)')
ax.set(ylabel='Expansion per baffle space')
fig.savefig('Flocculation/Images/Expansions_per_baffle_space_vs_plant_flow')
plt.show()

channels=np.zeros(plot_points)
for i in range(plot_points):
    channels[i]=num_channel(flow_plant[i], headloss_floc_BOD, Gt_BOD, T_BOD)
fig, ax = plt.subplots()
ax.plot(flow_plant,channels,'-')
ax.set(xlabel='Plant Flow Rate (L/s)')
ax.set(ylabel='Number of channels')
fig.savefig('Flocculation/Images/Number_of_channels_vs_plant_flow')
plt.show()

baffles=np.zeros(plot_points)
for i in range(plot_points):
    baffles[i]=num_baffles(flow_plant[i], headloss_floc_BOD, Gt_BOD, T_BOD)
fig, ax = plt.subplots()
ax.plot(flow_plant,baffles,'-')
ax.set(xlabel='Plant Flow Rate (L/s)')
ax.set(ylabel='Number of Baffle Spaces')
fig.savefig('Flocculation/Images/Number_of_baffle_spaces_vs_plant_flow')
plt.show()

width_floc_channel=np.zeros(plot_points)*u.m
for i in range(plot_points):
    width_floc_channel[i]=width_floc(flow_plant[i], headloss_floc_BOD, Gt_BOD, T_BOD)
fig, ax = plt.subplots()
ax.plot(flow_plant,width_floc_channel,'-')
ax.set(xlabel='Plant Flow Rate (L/s)')
ax.set(ylabel='Floc Channel Width (m)')
plt.title('Floc channel width vs plant flow')
fig.savefig('Flocculation/Images/Floc_channel_width_vs_plant_flow')
plt.show()
```

23)

Read from the graphs to determine

1. At what flow rate is it no longer necessary to add extra obstacles in the flocculator?
2. At what flow rate does the flocculator switch from 2 channels to 4 channels?

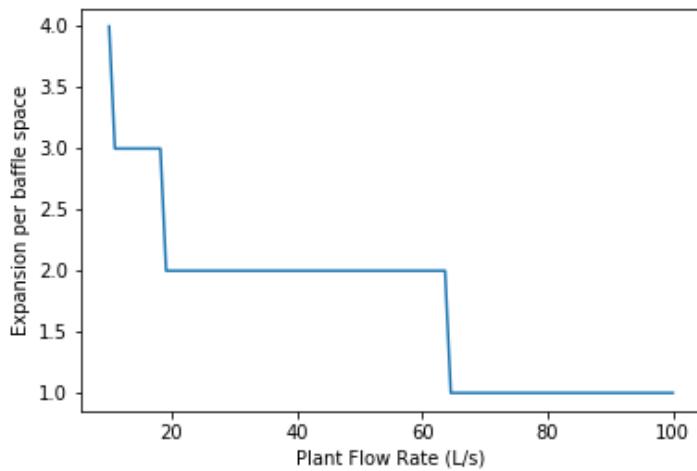


Fig. 29.2: The number of expansions per baffle space decreases rapidly with flow rate. These results are a function of the flocculator depth and of the target head loss used for the design.

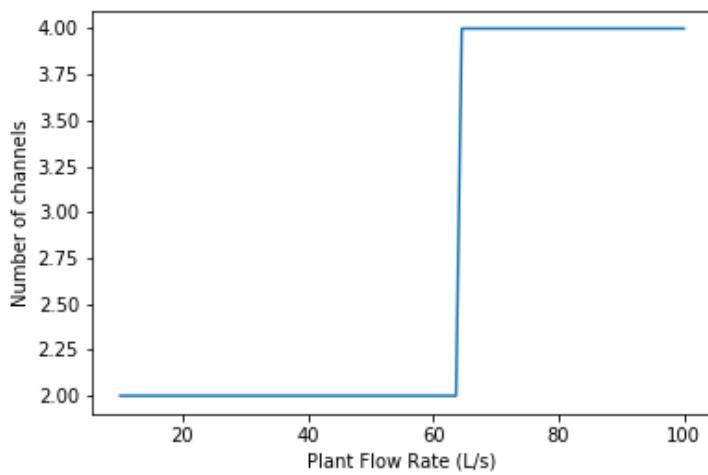


Fig. 29.3: The number of channels is less than 2 for a wide range of flows. This is because the required residence time in the flocculator is so low. We may eventually want to create a design that makes it possible to have an odd number of flocculator channels so that we can use a single channel for low flow rates.

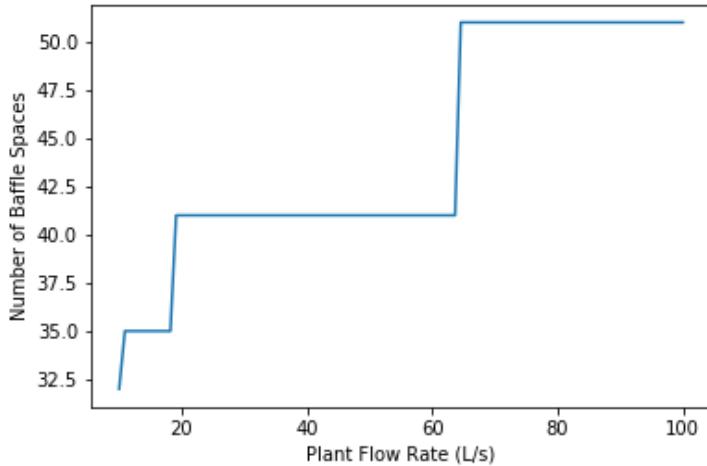


Fig. 29.4: The number of baffle spaces is lower for low flow rates because those flocculators have additional expansions. At high flow rates the number of baffles spaces increases because the residence time per baffle space decreases. The residence time per baffle space decreases when the number of channels jumps from 2 to 4.

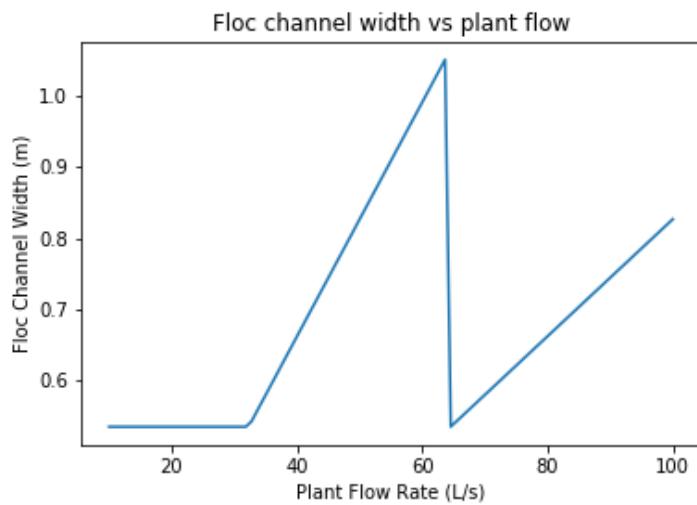


Fig. 29.5: The flocculator channel width varies linearly with flow rate for a constant number of channels. The exception is for flows below about 35 L/s. In that flow rate the requirement of 2 channels that are the length of the sedimentation tank is excessive and results in more collision potential than needed. We need a new design solution to handle this suboptimal design for flows lower than 35 L/s.

3. Why did the flocculator switch from 2 to 4 channels?
4. No obstacles are needed for flows greater than about 65 L/s.
5. The flocculator switches from 2 to 4 channels at 65 L/s.
6. The channels had reached their maximum width at that flow rate.

24)

Change Gt_BOD to 20,000 and run the code again. Identify at least 3 changes in the design.

1. The number of channels would be 2 for all of the flows explored here.
2. The value of G increased
3. The spacing between baffles decreased
4. The flow expansions became closer together (because the spacing between baffles decreased)
5. The number of baffles decreased

FLOCCULATION DESIGN CHALLENGE

30.1 1)

How far will two clay particles with a diameter of $5 \mu\text{m}$ travel relative to each if they are in a uniform velocity gradient of 100 Hz for 400 s and separated (in the direction of the velocity gradient) by their average separation distance based on a turbidity of 1 NTU? Note that in a uniform velocity gradient $\bar{G} = G_{CS}$.

30.2 2)

Use the flocculation model to estimate the settled water turbidity (assuming no floc blanket) given a surface coverage, $\Gamma = 0.3$, no DOM, a $G_{CS}\theta = 40,000$. Use the model that neglects the initial particle concentration.

30.3 3)

A hydraulic flocculator has 50 cm wide channels, an exit water depth of 2 m, a baffle spacing of 40 cm with no intermediate obstacles, and a design temperature of 15°C .

1. What is the flow rate that would result in a G_{CS} of 100 Hz?
2. If the $G_{CS}\theta$ for the flocculator is 40,000, then what is the total head loss for the flocculator?
3. How many baffles does the flocculator need?
4. What is the velocity
5. If the flow rate is decrease by 50%, then what is the new G_{CS} and $G_{CS}\theta$ for the flocculator?

30.4 4)

A 60 L/s hydraulic flocculator has 63 cm wide channels, an exit water depth of 2 m, a baffle spacing of 39.3 cm with no intermediate obstacles and a total of 63 flow expansions. Plot $G_{CS}\theta$ as a function of temperature from 0°C to 35°C .

1. Why does $G_{CS}\theta$ increase with temperature?
2. How does head loss through a flocculator vary as a function of temperature? Explain why.

SEDIMENTATION INTRODUCTION

The improved performance is due to 3 factors. First, the inlet manifold has a diffuser system that straightens the fluid jets that are exiting the manifold so that they have no horizontal velocity component. This is critical because even a small horizontal velocity causes a large scale circulation that transports flocs directly to the top of the sedimentation tank. Inlet manifolds without flow straightening diffusers are commonly used in vertical flow sedimentation tanks including designs by leading competitors.

Second, the diffusers create a line jet that spans the entire length of the sedimentation tank. The line jet enters a jet reverser and the vertical upward jet momentum is used to resuspend flocs that have settled to the bottom of the sedimentation tank. The resuspended flocs form a fluidized bed (floc blanket) with a suspended solids concentrations of approximately 1-5 g/L. The high concentration of particles leads to an increase in collisions and particle aggregation. The floc blanket reduces settled water turbidity by a factor of 10 (Garland et al., 2017) and provides two additional benefits. The floc blanket creates a uniform vertical velocity of water entering the plate settlers and the floc blanket transports excess flocs to a floc hopper for final removal by opening a small drain valve. Third, the bottom geometry is shaped so that all flocs that settle are transported to the jet reverser. Thus there is no accumulation of settled flocs in the main sedimentation basin. Sludge that is allowed to accumulate in the bottom of sedimentation tanks in tropical and temperate decomposes anaerobically and generates methane. The methane forms gas bubbles that carry suspended solids to the top of the sedimentation tank and cause a reduction in particle removal efficiency. The AguaClara sedimentation tank bottom geometry prevents sludge accumulation. The hydraulic self cleaning sedimentation tank with a high performing floc blanket, zero sludge accumulation, and with no moving parts outperforms conventional sedimentation tanks on capital cost, performance, and maintenance costs. Mechanical sludge removal systems are well known to be costly to install and a challenge to maintain.

31.1 Floc blankets

See the Pan American Health Organization, (PAHO) manual on [theory of rapid sand filtration plants rapid sand filters](#) (page 289) for reasons why floc blankets should not be used! According to PAHO, floc blankets are not recommended for small communities who lack highly trained personal to operate the plant and floc blanket should only be used where plant flow rates and water quality are constant. Each of these constraints was due to the inadequate design of previous floc blanket reactors that made operation difficult.

31.1.1 Floc blanket hypotheses

The floc blanket mechanism responsible for reduced settled water turbidity has been elusive. - not flocculation between particles delivered from the flocculator because $G\theta$ generated by the shear of the settling flocs and the hydraulic residence time of the floc blanket is insufficient to cause significant - The floc blanket consists of settling flocs that are maintained in suspension by the upwardly flowing water.

31.2 Floc Hopper

31.3 Plate Settlers

31.4 Manifold Hydraulics

CHAPTER
THIRTYTWO

SEDIMENTATION DESIGN

CHAPTER
THIRTYTHREE

SEDIMENTATION EXAMPLES

Design a tube settler for a laboratory scale sedimentation tank. The vertical section of the sedimentation tank has a net upflow velocity of 3 mm/s. This velocity is maintained in the tube settler, V_α . The target capture velocity is 0.2 mm/s. The tube settler diameter is 2.54 cm.

$$\frac{\bar{v}_\uparrow}{v_c} = \frac{L}{D} \cos \alpha \sin \alpha + \sin^2 \alpha \quad (33.1)$$

$$\bar{v}_\uparrow = \bar{v}_\alpha \sin \alpha \quad (33.2)$$

Solve for the length of the tube settler.

$$L = \frac{D}{\cos \alpha} \left(\frac{\bar{v}_\alpha}{\bar{v}_c} - \sin \alpha \right) \quad (33.3)$$

```
from aguaclara.core.units import unit_registry as u

import numpy as np

v_alpha = 3 * u.mm/u.s
v_c = 1 * u.mm/u.s
D = 2.54 * u.cm
alpha = 60 * u.deg

def L_settler(D,alpha,v_alpha,v_c):
    return D/np.cos(alpha)*(v_alpha/v_c - np.sin(alpha))

print(L_settler(D,alpha,v_alpha,1*u.mm/u.s))
print(L_settler(D,alpha,v_alpha,0.2*u.mm/u.s))
```

The tube settler above the floc hopper needs to be 72 cm long. The tube settler should provide a capture velocity of at least 1 mm/s prior to the floc hopper. Thus there should be 11 cm below the floc hopper.

CHAPTER
THIRTYFOUR

SEDIMENTATION THEORY AND FUTURE WORK

34.1 Floc recycle

We hypothesize that the flocs in floc blankets serve as collectors that primary particles attach to. We suspect that collisions between primary particles and large flocs are possible in the sedimentation tank because the rotational velocity of the flocs is small relative to the sedimentation velocity of the flocs. If the rotational velocity of the flocs is small, then a stagnation point will exist on the floc and a finite flow of fluid will come within a primary particle radius of the floc. Thus we expect primary particle removal in floc blankets to be proportional to the number of collectors that a primary particle passes while in the floc blanket.

The number of collectors that a primary particle passes is proportional to the solids concentration (a surrogate for the number concentration of flocs), the primary particle residence time in the floc blanket, and the sedimentation velocity of the flocs. The sedimentation velocity of the flocs is important because that is what causes a relative velocity between the primary particles and the flocs.

As we have explored increasing the upflow velocity in sedimentation tanks the performance has dropped markedly. This is undoubtedly due in part to the combined effective of a very dilute floc blanket at high upflow velocities AND a low residence time for the primary particles.

Would it be possible to increase the concentration of the floc blanket and thus increase the collision rate? At 3 mm/s upflow velocity there are very few flocs that can stay in the floc blanket. We need a mechanism to transport flocs to the bottom of the floc blanket and return them again after they are carried to the top of the floc blanket.

We propose to test this by installing a settled floc recycle line. The recycle line will connect to the bottom surface of the tube settler below the location of the floc weir. From there it will carry concentrated sludge to the very bottom of the sedimentation tank where it will pass through the wall of the sedimentation tank. Increasing the amount of recycle flow will both increase the solids concentration in the floc blanket and decrease the primary particle residence time in the floc blanket.

There must be an optimal amount of recycled flocs for a floc blanket. Of course, one possibility is that the optimal recycle is zero. Recycled flocs increase the floc blanket concentration and thus increase the rate of collisions between primary particles and flocs. The recycled flocs also decrease the residence time in the floc blanket and thus decrease the total number of collisions between primary particles and flocs. It may be more complicated than this because the hindered sedimentation velocity of the flocs in the floc blanket is also a function of their concentration.

Our goal is to find the optimal recycle ratio. Optimal is defined as the maximum collision potential. Collision potential for the floc blanket is proportional to the collision rate times the hydraulic residence time. The collision rate is proportional to the solids concentration and the hindered sedimentation velocity of those flocs. The collision potential is thus proportional to the total number of flocs that a primary particle passes on its way through the floc blanket.

$$CP_{fb} \propto C_{fb} \theta_{fb} v_{hindered} \quad (34.1)$$

The residence time in the floc blanket is given by

$$\theta_{fb} = \frac{H_{fb}}{v_{fb}} \quad (34.2)$$

$$v_{fb} = \frac{Q_{plant} + Q_{recycle}}{A_{fb}} \quad (34.3)$$

$$Q_{recycle} = \Pi_{recycle} Q_{plant} \quad (34.4)$$

The velocity up through the floc blanket without recycle is defined as

$$v_{up} = \frac{Q_{plant}}{A_{fb}} \quad (34.5)$$

$$v_{fb} = v_{up} (1 + \Pi_{recycle}) \quad (34.6)$$

Now we need equations for the concentration in the floc blanket. This is based on mass conservation such that the mass in the floc blanket is constant. There is a hindered sedimentation velocity of the flocs that results in a reduction of the mass flux out of the top of the control volume.

$$C_{fb} \left(\frac{Q_{plant} + Q_{recycle}}{A_{fb}} - v_{hindered} \right) A_{fb} = C_{plant} Q_{plant} + C_{recycle} Q_{recycle} \quad (34.7)$$

$$C_{fb} \left(\frac{Q_{plant} + \Pi_{recycle} Q_{plant}}{A_{fb}} - v_{hindered} \frac{Q_{plant}}{Q_{plant}} \right) A_{fb} = C_{plant} Q_{plant} + C_{recycle} \Pi_{recycle} Q_{plant} \quad (34.8)$$

$$C_{fb} \left(1 + \Pi_{recycle} - \frac{v_{hindered}}{v_{up}} \right) = C_{plant} + C_{recycle} \Pi_{recycle} \quad (34.9)$$

$$C_{fb} = \frac{C_{plant} + C_{recycle} \Pi_{recycle}}{\left(1 + \Pi_{recycle} - \frac{v_{hindered}}{v_{up}} \right)} \quad (34.10)$$

Now we can substitute to get the collision potential as a function of the flow rates.

$$CP_{fb} \propto \frac{C_{plant} + C_{recycle} \Pi_{recycle}}{\left(1 + \Pi_{recycle} - \frac{v_{hindered}}{v_{up}} \right) (1 + \Pi_{recycle})} \frac{H_{fb} v_{hindered}}{v_{up}} \quad (34.11)$$

We estimate the hindered sedimentation velocity to be 1 mm/s since that is what occurs in a 1 mm/s upflow velocity floc blanket. Ideally we would have a hindered sedimentation velocity as a function of the concentration of flocs in the floc blanket. The concentration of recycled flocs is assumed to be approximately 20 g/L based on Casey Garland measurements of the solids concentration in the floc hopper sludge.

```
import aguacela.core.physchem as pc
from aguacela.core.units import unit_registry as u

import numpy as np
import matplotlib.pyplot as plt

D_fb=2.5*u.cm
A_fb = pc.area_circle(D_fb)
H_fb = 1 * u.m
v_hindered = 1 * u.mm/u.s
C_fb_conventional = 3 * u.g/u.L
C_recycle = 20 * u.g/u.L
C_plant = 100 * u.NTU
v_up = 3 * u.mm/u.s

def CP(H_fb,v_up,v_hindered,Pi_recycle,C_plant,C_recycle):
    return (H_fb*v_hindered/v_up*(C_plant+C_recycle*Pi_recycle)/((1+Pi_recycle)*(1+Pi_recycle-v_hindered/v_up))).to_base_units()
Pi_recycle_max = 2
```

(continues on next page)

(continued from previous page)

```

Pi_recycle = np.arange(0,Pi_recycle_max,0.1)
fig, ax = plt.subplots()
x=np.array([0,Pi_recycle_max])
yscale = (C_fb_conventional*H_fb*v_hindered/ (1*u.mm/u.s)).to_base_units()
yscale
y=np.array([1,1])*yscale
ax.plot(x,y)
ax.plot(Pi_recycle,CP(H_fb,v_up,v_hindered,Pi_recycle,C_plant,C_recycle))
imagepath = 'Sedimentation/Images/'
ax.set(xlabel='recycle ratio', ylabel='Collision Potential (kg/m^2)')
ax.legend(["no recycle at 1 mm/s","with recycle at 3 mm/s"])
fig.savefig(imagepath+'fb_recycle_ratio')
plt.show()

```

Here are the results.

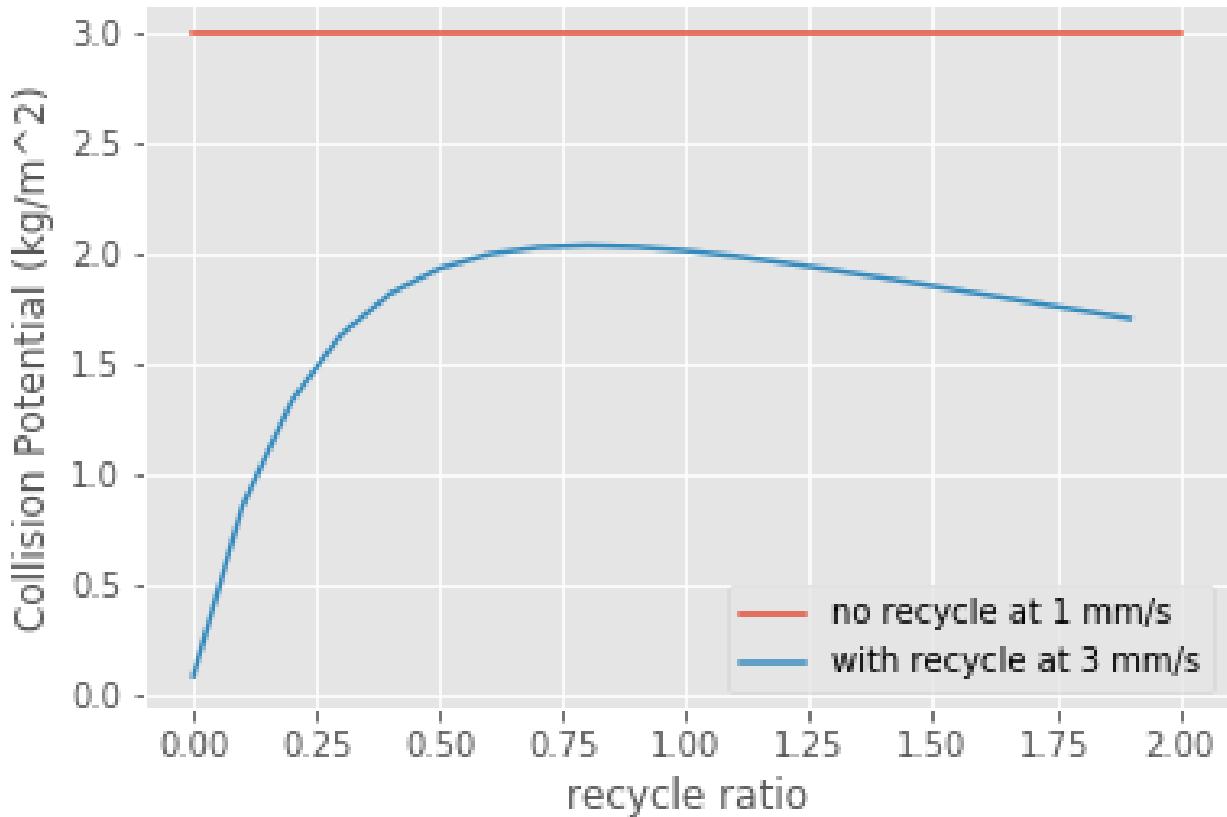


Fig. 34.1: Collision potential comparison in a 1 m deep floc blanket.

This analysis suggest that a recycle flow rate that is between 0.5 and 1.5 at a net upflow velocity of 3 mm/s could produce collision potential that is 2/3 of the collision potential with a 1 mm/s upflow velocity. Thus a 3 mm/s sed tank with 1.5 m of floc blanket and recycle might be able to perform at the same level as a 1 mm/s sed tank with a 1 m floc blanket.

The next step is to design the recycle tube. The recycle tube could be inclined to promote additional consolidation to reduce the amount of water that is recycled. The slope would need to be about 60 degrees. We could experiment with the design of the recycle line if it were made of flexible tubing.

It is expected that the consolidated sludge will flow by gravity because of its higher density. The big unknown is what diameter recycle line is needed for a lab scale test with a 2.5 cm diameter sedimentation tank.

The recycle sludge has a density given by

$$\rho_{sludge} = \left(1 - \frac{\rho_{H_2O}}{\rho_{Clay}}\right) C_{sludge} + \rho_{H_2O} \quad (34.12)$$

The piezometric head (measured in equivalent change in height of the recycle line liquid) that is causing the flow through the recycle line is equal to the difference in density between the recycled sludge and the floc blanket times the height of the floc blanket normalized by the recycle line density.

$$H_l = H_{fb} \frac{\rho_{sludge} - \rho_{fb}}{\rho_{sludge}} \quad (34.13)$$

Substitute to replace the sludge and floc blanket densities.

$$H_l = H_{fb} \frac{\left(1 - \frac{\rho_{H_2O}}{\rho_{Clay}}\right) C_{sludge} + \rho_{H_2O} - \left[\left(1 - \frac{\rho_{H_2O}}{\rho_{Clay}}\right) C_{fb} + \rho_{H_2O}\right]}{\left(1 - \frac{\rho_{H_2O}}{\rho_{Clay}}\right) C_{sludge} + \rho_{H_2O}} \quad (34.14)$$

Simplify the equation for the head loss in the recycle tube.

$$H_l = H_{fb} \frac{C_{sludge} - C_{fb}}{C_{sludge} + \frac{\rho_{H_2O} \rho_{Clay}}{\rho_{Clay} - \rho_{H_2O}}} \quad (34.15)$$

The recycle tube is assumed to be sloped at 60 degrees from the horizontal to enable further consolidation. The length of the recycle tube is

$$L_{tube} = H_{fb} / \sin(60) \quad (34.16)$$

We will assume that the dynamic viscosity of the sludge is the same as the dynamic viscosity of water. We will calculate the kinematic viscosity of the sludge by dividing the dynamic viscosity of water by the density of the recycle.

Now we can solve for the required tube diameter

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u

import numpy as np
import matplotlib.pyplot as plt

Temperature= 20*u.degC
D_fb=2.5*u.cm
A_fb = pc.area_circle(D_fb)
H_fb = 1.5 * u.m
Angle_tube = 60*u.deg
L_tube = H_fb/np.sin(Angle_tube)
density_clay=2650*u.kg/u.m**3

H_l = H_fb* (C_recycle-C_fb) / (C_recycle+((pc.density_water(Temperature)*density_clay)/
    ↪(density_clay-pc.density_water(Temperature)))))

H_l
Q_plant=v_up*A_fb
```

(continues on next page)

(continued from previous page)

```

Pi_recycle=0.5
density_recycle = (1 - pc.density_water(Temperature)/density_clay)*C_recycle + pc.
    ↵density_water(Temperature)
nu_recycle = pc.viscosity_dynamic(Temperature)/density_recycle
D_recycle = pc.diam_pipe(Q_plant*Pi_recycle,H_l,L_tube,nu_recycle,0.01*u.mm,2)
D_recycle.to(u.mm)
D_recycle.to(u.inch)

```

The head loss in the recycle tube is approximately 1.6 cm in a 1.5 m deep floc blanket.

The recycle line will be installed between the bottom of the tube settler and the inlet to the sedimentation tank. The recycle line will connect directly to the side of the sedimentation tank to minimize minor losses. We will use a 0.25 ID, 3/8OD clear flexible tube for the recycle line. We will use PVC glue to attach the flexible tube to the rigid clear PVC tubing.

It is possible that it will be necessary to prevent flow in the recycle line initially so that it doesn't flow upward. Once the tube begins filling with solids it should be possible for it to start flowing downwards.

34.2 Floc Volcanoes

Floc volcanoes are caused by differences in temperature between the water that is in a sedimentation tank and the incoming water. If the incoming water is warmer than the water that is already in the sedimentation tank, then the incoming water will be buoyant and will rise quickly to the top of the sedimentation tank and carry flocs to the effluent launder.

Temperature fluctuations can be especially pronounced with small scale water supplies where small streams and small diameter transmission lines can be exposed to the sun and can warm up dramatically during a few hours of sunshine. Given that temperature changes and density changes can not easily be engineered, the only solution that we have is to reduce the time that water spends in the sedimentation tank so that the influent water is closer to the average temperature of the water in the sedimentation tank. Solar heating causing the raw water temperature to go from a minimum at 6 am to a maximum at 1 pm. AquaClara sedimentation tanks currently have a residence time of approximately 2 m / (1 mm/s) or 2000 s. We anticipate that by increasing the upflow velocity and by introducing floc recycle that the effects of temperature induced floc volcanoes will be reduced.

SEDIMENTATION DESIGN SOLUTION

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import numpy as np
import matplotlib.pyplot as plt
```

35.1 Introduction

The Aguacalara team would like to understand how to design larger sedimentation tanks. In this design challenge you will learn how some of the sedimentation tank components work together. The design for an entire sedimentation tank is somewhat complex and you are welcome to review the sedimentation tank design file to see the latest Aguacalara design team solution. The sedimentation tank design has evolved rapidly since 2005 and our fabrication techniques have evolved as well.

In Aguacalara sedimentation tanks the influent and effluent channels are located above the space occupied by the floc hopper. If you want to see how those geometric constraints are handled you can review the Aguacalara sedimentation tank design file. Agua Para el Pueblo is building a 120 L/s plant for Gracias, the capital of Lempira, Honduras. This plant has 2 treatment trains and thus we can design each train for 60 L/s. For this design challenge you may assume the following:

For each step in the solution define a variable with a name that is easily understood so that it can be used later if needed. Consider using the Aguacalara naming convention.

Use numpy for ceil, floor, sin, and pi.

The following cell has constants defined for you to use in designing your sedimentation tank. Read through and run it so you know what variables have been defined for you. See [Table 14.5](#) for the plane jet constant.

```
# We are experimenting with using head loss at the exit of the diffusers as
# a reasonable design constraint for the design of the
# inlet manifold/diffuser system. This head loss in the exit of the diffuser
# will allow higher velocities (and pressure recovery) in the manifold pipe
# and thus will enable use of smaller diameter manifold pipes.
# -----
# We've set a maximum head loss in the sedimentation tank inlet:
headloss_sed_inlet_max = 1 * u.cm

# The manifold and diffuser pipes used in the sedimentation tank have an SDR of 26.
SDR=26

# Pi_jet_plane is used to estimate the maximum energy dissipation rate in a plane jet.
# A plane jet is uniform in one dimension. A plane jet can be formed by a long narrow_
# slot.
```

(continues on next page)

(continued from previous page)

```

# see table_EDR_G_equations
Pi_jet_plane = 0.0124

# Design temperature will be helpful to calculate viscosity
T_design = 15*u.degC

# Plant Flowrate:
flow_plant = 60 * u.L/u.s

# Upflow velocity at the top of the floc blanket:
V_sed_up = 1 * u.mm/u.s

# The corrugated plastic sheets used to make the plate settlers:
W_sed = 42 * u.inch
thickness_sed_plate = 2 * u.mm

# The plate settlers are angled 60° from the horizontal:
angle_sed_plate = 60 * u.deg

# The plate settlers are spaced 2.5cm apart (this is the perpendicular
# distance between plates, not the horizontal distance between plates):
s_sed_plate = 2.5 * u.cm

# Plate settler capture velocity:
V_sed_capture = 0.12 * u.mm/u.s

# The minimum port flow (from the first port) divided by the maximum port
# flow (from the last port) for flow division between sedimentation tanks
# and for flow distribution from the inlet manifold should be at least:
Pi_sed_manifold_flow = 0.8

g = pc.gravity

```

35.2 Diffusers

35.2.1 1:

We will start our design of the sedimentation tank by considering the diffusers.

- Calculate the maximum velocity of water leaving the diffuser based on the maximum head loss. Assume that the majority of head loss is the kinetic energy of the flow exiting the diffuser slot (this assumption will be checked later). Assume K=1.
- Calculate the minimum inner width of the diffuser. Assume that the diffuser slot is continuous over the entire length of the sedimentation tank to get an initial estimate (it isn't actually continuous because it is made from many flattened diffuser pipes).

Define your answers as variables and then print those variables.

```

# minor loss equation with K=1
V_diffuser_max = (np.sqrt((2 * g * headloss_sed_inlet_max))).to(u.m / u.s)
print('The maximum velocity of the sed tank diffusers is', V_diffuser_max)

# mass conservation

```

(continues on next page)

(continued from previous page)

```
W_diffuser_inner_min = ((V_sed_up / V_diffuser_max) * W_sed).to(u.mm)
print('The minimum width of the sed tank diffusers is',W_diffuser_inner_min)
```

The maximum velocity of the sed tank diffusers is 0.4429 meter / second The minimum width of the sed tank diffusers is 2.409 millimeter

35.2.2 2:

Diffusers are made by deforming PVC pipe. Softened PVC pipe is forced onto a mold that shapes it into the rectangular shape of the diffuser (see slides from the Sedimentation Lecture).

- What metal plate thickness should be used to make the mold for the diffusers? This value will be the minimum diffuser width. Metal plates are available in 1/16 increments of thickness. The minimum thickness of plate that is strong enough for a mold is 1/16.

Note: you can use the `ceil_nearest` function defined in `utility.py` to do this elegantly. You can create an array of available metal plates using `numpy.arange`.

```
W_diffuser = ut.ceil_nearest(W_diffuser_inner_min, np.arange(1/16,1/4,1/16)*u.inch)

print('The width of sed tank diffuser is',W_diffuser.to(u.cm))
```

The width of sed tank diffuser is 0.3175 centimeter

```
Pi_Error = 0.1
Headloss_DosingTube_Max = 20*(u.cm)
K_Minor = 2

Vel_Tube_Max = (((Pi_Error * 2 * Headloss_DosingTube_Max * g) / K_Minor)**(1/2)).to(u.
meter/u.s)
```

$$Vel_{Tube,Max} = \left(\frac{2 \Pi_{Error} Headloss_{DosingTube,Max} g}{K_{Minor}} \right)^{\frac{1}{2}} \quad (35.1)$$

35.2.3 3:

The PVC pipe that forms the diffusers changes in shape and wall thickness during the molding process. The inner width of the rectangle is created by forcing the pipe over a rectangular wedge that is the thickness you calculated above. During the molding process, PVC pipe wall cross-sectional area is conserved. The pipe wall is stretched in total length approximately 20%. Another way to think about this is that the thickness of the wall is reduced by a factor of 1/1.2 because the mass of PVC is conserved and the density is unchanged. Thus, volume and cross-sectional area are conserved.

- Start by drawing a picture of what is happening to the deformation of the pipe as it is converted from the circular pipe to the rectangular diffuser slot (assume that the slot is a rectangle with perfectly square corners). You do not need to submit your sketches for this design challenge.
 - Draw the initial circular pipe. Label the diagram with the appropriate variables for inner diameter, outer diameter, and wall thickness.
 - Draw the final rectangular diffuser slot. Label the diagram with the appropriate variables for length, width, and wall thickness.

Area is given using the following equation:

$$Area_{PVC} = 2(B_{diffuser} + W_{diffuser}) - Usetheequationfor:math: 'Area_{PVC}'tocalculate -theouterlengthoftherectangulardiffuserset, :math: 'B_{diffuser}': -theinnerlengthoftherectangulardiffuserset, :math: (35.2)$$

Answering this question will require using functions from the `pipedatabase` file, imported here as `pipe`.

```
SDR=26
```

```
# Assumed stretch of the PVC pipes as they are heated and molded:
Pi_PVC_stretch = 1.2

# Nominal diameter of the sed tank diffuser
ND_sed_diffuser = 1 * u.inch
```

```
#The cross-sectional area of the pipe wall is:
area_PVC = (np.pi/4) * ((pipe.OD(ND_sed_diffuser)**2)
                        - (pipe.ID_SDR(ND_sed_diffuser, SDR)**2)
                        )

#The thickness of the wall is reduced by the stretch factor:
thickness_sed_diffuser_wall = ((pipe.OD(ND_sed_diffuser)
                                 - pipe.ID_SDR(ND_sed_diffuser, SDR))
                                 / (2 * Pi_PVC_stretch)
                                 )

# From geometry of the rectangular diffuser opening (assuming perfectly square_
#corners) we have:
B_diffuser = ((area_PVC / (2 * thickness_sed_diffuser_wall))
              - W_diffuser
              ).to(u.cm)

print("Sed diffuser outer length:", B_diffuser)

S_diffuser = B_diffuser - (2 * thickness_sed_diffuser_wall)
print("Sed diffuser inner length:", S_diffuser)
```

Sed diffuser outer length: 5.736 centimeter Sed diffuser inner length: 5.522 centimeter

35.2.4 4:

Each diffuser serves a certain width and length of the sedimentation tank. Assume that the diffusers are installed so that they touch each other.

- Determine the flow through each diffuser.
- Determine the velocity through each diffuser.

```
flow_max_diffuser = V_sed_up * W_sed * B_diffuser

V_diffuser = (flow_max_diffuser
              / (W_diffuser * S_diffuser)).to(u.m / u.s)
print('The flow of water leaving a sed tank diffuser is', flow_max_diffuser.to(u.ml/u.
              s))
print('The velocity of water leaving the sed tank diffuser is', V_diffuser)
```

The flow of water leaving a sed tank diffuser is 61.19 milliliter / second The velocity of water leaving the sed tank diffuser is 0.349 meter / second

35.2.5 5:

What is the Reynolds number of the jet exiting the diffusers?

Note: you will likely need to force Pint to display this as a dimensionless number.

```
Re_diffuser_jet = ((W_diffuser * V_diffuser) / pc.viscosity_kinematic(T_design)).to(u.
    ↪dimensionless)
print('The Reynolds number for this jet is', Re_diffuser_jet)
```

The Reynolds number for this jet is 974.6 dimensionless

35.2.6 6:

What is the Reynolds number of the vertical flow up through the top of the floc blanket?

```
Re_sed = ((W_sed * V_sed_up) / pc.viscosity_kinematic(T_design)).to(u.dimensionless)
print('Reynolds number through floc is', Re_sed)
```

Reynolds number through floc is 938.2 dimensionless.

35.2.7 7:

Compare the two values for Reynolds numbers that you found for Problems 5 and 6. What do the Reynolds numbers for these very different flows tell you?

The Reynolds number is almost the same because mass conservation requires $V \cdot W$ to be a constant. The only difference in the Reynolds number is due to the fact that the plane jet isn't quite continuous. It is broken by twice the thickness of the pipe wall between diffusers.

35.2.8 8:

Next, we want to determine the energy dissipation rate for the flow leaving the jet reverser. For this process, you can assume that the jet remains laminar. The flow spreads to fill the gaps created by the walls of the diffuser tubes by the time it traverses the jet reverser. Jet velocity and flow rate are conserved as the jet changes direction in the jet reverser.

- Calculate the thickness of the jet after it does the 180 degree bend of the jet reverser.
- Calculate the energy dissipation rate for the flow leaving the jet reverser.

Convert your final answer to milliwatts per kilogram.

```
#Calculate the thickness of the jet when it leaves the diffuser. B_diff = S_diff
W_jet_reversed = W_sed * V_sed_up / V_diffuser
#Calculate the maximum energy dissipation rate
EDR_inlet_jet = Pi_jet_plane *((( V_diffuser)**3)
    / W_jet_reversed).to(u.mW / u.kg)
```

(continues on next page)

(continued from previous page)

```
print('The energy dissipation rate for inlet jet is', EDR_inlet_jet)
```

The energy dissipation rate for inlet jet is 172.5 milliwatt / kilogram

35.2.9 9:

In designing AguaClara plants, it is critical to account for all forms of significant head loss. In the sedimentation tank, effluent launders (the pipes that collect settled water at above the plate settlers) provide about 4 cm of head loss. That 4 cm of head loss is the majority of the head loss through an AguaClara sedimentation tank. We want to calculate the exit head loss for water leaving the inlet diffusers to determine whether it is a significant addition to the total head loss through the sedimentation tank.

Calculate this diffuser exit head loss in two ways. * First, calculate the head loss making sure to account for the upflow velocity in the sed tank. * Second, calculate the head loss but assume that the upflow velocity is negligible.

- Is it reasonable to neglect the upflow velocity in the sed tank when calculating this head loss?

$$h_e = \frac{(V_{in} - V_{out})^2}{2g} \quad (35.3)$$

You will find that the exit head loss for water leaving the diffuser is high enough that we need to account for head loss in the sed tank inlet piping for our designs.

```
hl_sed_diffuser_exit1 = (((V_diffuser - V_sed_up) ** 2) / (2 * g)).to(u.cm)
hl_sed_diffuser_exit2 = (((V_diffuser) ** 2) / (2 * g)).to(u.cm)
hl_sed_diffuser_error=(hl_sed_diffuser_exit2-hl_sed_diffuser_exit1)/hl_sed_diffuser_
exit1

print('The best estimate of the exit head loss for the diffuser is', hl_sed_diffuser_
exit1)
print('The 2nd estimate of the exit head loss for the diffuser ignoring the upflow_
velocity is', hl_sed_diffuser_exit2)
print('It is reasonable to neglect the effect of the upflow velocity. The error is',
hl_sed_diffuser_error)
```

The best estimate of the exit head loss for the diffuser is 0.6176 centimeter The 2nd estimate of the exit head loss for the diffuser ignoring the upflow velocity is 0.6211 centimeter It is reasonable to neglect the effect of the upflow velocity. The error is 0.005755 dimensionless

35.3 Manifold and Launders

Flow distribution between and within sedimentation tanks is an important design component to ensure good sedimentation performance. We need to distribute flow uniformly between sedimentation tanks and also between diffusers on the inlet manifolds.

The following variable definitions and equations will be useful in answering later questions. * $hl_{ParallelPath}$ is the head loss (flow resistance) in the parallel paths leaving the manifold. The head loss in the parallel path is the total head loss from where the flow leaves the manifold to the point where the parallel flows reunite. * $\Delta H_{Manifold}$ is the variability in piezometric head in the manifold that is driving the flow through the parallel paths.

- The ratio of minimum (first diffuser port) to maximum (last diffuser port) flow is given by:

$$\Pi_{DiffuserFlow} = \sqrt{\frac{hl_{ParallelPath} - \frac{\Delta H_{Manifold}}{2}}{hl_{ParallelPath} + \frac{\Delta H_{Manifold}}{2}}} \quad (35.4)$$

-The change in piezometric head is given by :

$$\Delta H_{Manifold} = \frac{Velocity_{Manifold}^2}{2g} \quad (35.5)$$

-The maximum allowable velocity in the manifold is given by :

$$\Pi_{DiffuserFlow}^2 * \left(hl_{ParallelPath} + \frac{\Delta H_{Manifold}}{2} \right) = hl_{ParallelPath} - \frac{\Delta H_{Manifold}}{2} \quad (35.6)$$

$$(\Pi_{DiffuserFlow}^2 - 1) hl_{ParallelPath} + (\Pi_{DiffuserFlow}^2 + 1) \frac{\Delta H_{Manifold}}{2} = 0 \quad (35.7)$$

$$\left(\frac{1 - \Pi_{DiffuserFlow}^2}{\Pi_{DiffuserFlow}^2 + 1} \right) hl_{ParallelPath} = \frac{\Delta H_{Manifold}}{2} \quad (35.8)$$

$$\left(\frac{1 - \Pi_{DiffuserFlow}^2}{\Pi_{DiffuserFlow}^2 + 1} \right) hl_{ParallelPath} = \frac{Velocity_{Manifold}^2}{4g} \quad (35.9)$$

35.3.1 10:

Now, we want to find the maximum velocity for an inlet manifold which is dependent on the given flow distribution constraint, $\Pi_{DiffuserFlow}$, and the head loss in the parallel paths, $hl_{ParallelPath}$.

- Determine the relationship between diffuser exit velocity and the head loss in the parallel paths.
- Determine an equation for maximum velocity for an inlet manifold in terms of diffuser exit velocity and the flow distribution constraint.
- Write a **function** for maximum velocity for an inlet manifold using the equations you just found.

Exit losses from the diffusers dominate the head loss because the velocity in the diffuser slots is much higher than the velocity at the entrance to the diffuser pipes. Using the insight from the previous problem, it is reasonable to neglect the effect of the upflow velocity when calculating the exit head loss for the manifold diffusers.

```
#hl_jet = V_jet^2/(2*g)

def Vel_sed_manifold_max(Pi_diffuser_flow, V_diffuser):
    return (V_diffuser * np.sqrt(2 * ((1-(Pi_diffuser_flow**2))
                                      / ((Pi_diffuser_flow**2)+1)
                                      )
                                ))
```

35.3.2 11:

Head loss in the sedimentation tank is impacted by multiple forms of head loss, including head loss through the effluent launder and diffusers. Head loss through the effluent launder is about 4 cm. You found head loss through the diffusers in Problem 9.

- Which form of head loss (effluent launder or diffuser) is in the parallel path, $hl_{ParallelPath}$?

Use the function that you wrote for Problem 10 to calculate the maximum velocity in the inlet manifold of the sedimentation tank. Use the value for `Pi_sed_manifold_flow` given above. Use the diffuser exit velocity you found in Problem 4.

```
print("Only the diffuser head loss is in the parallel paths.")

V_sed_manifold_max = Vel_sed_manifold_max(Pi_sed_manifold_flow, V_diffuser)

print('The maximum velocity in the sedimentation tank manifold is',V_sed_manifold_max)
```

Only the diffuser head loss is in the parallel paths. The maximum velocity in the sedimentation tank manifold is 0.2313 meter / second

35.3.3 12:

The ratio of manifold pipe cross-sectional area to total diffuser cross-sectional area determines the flow distribution between diffusers.

- Calculate the ratio of manifold pipe cross-sectional area to total diffuser cross-sectional area. You can use the velocities of the manifold and the diffusers to calculate the areas.
- What is the significance of the flow area ratio that you found? What does it tell you about the relative areas?

Note: the flow distribution will be more uniform if the diffuser velocity is higher than the manifold velocity.

```
print('The flow area ratio of manifold pipe to diffusers is',(V_diffuser / V_sed_
↪manifold_max).to(u.dimensionless))
print("This means that the manifold flow area is larger than the total diffuser area.
↪")
```

The flow area ratio of manifold pipe to diffusers is 1.509 dimensionless This means that the manifold flow area is larger than the total diffuser area.

35.3.4 13:

The maximum sed tank flow rate is currently set by the constraint of using a single length of pipe for the manifold and launder. The maximum length of the upflow region of the sedimentation tank is 5.8 m, as given below. What is the corresponding sedimentation tank flow rate?

```
L_sed_upflow_max = 5.8 * u.m

flow_sed_max = (L_sed_upflow_max * V_sed_up * W_sed).to(u.L / u.s)
print("The maximum flow rate in one sedimentation tank is",flow_sed_max)
```

The maximum flow rate in one sedimentation tank is 6.187 liter / second

35.3.5 14:

The maximum sed tank flow rate dictates the required pipe diameter for the manifold and launder.

- What is the minimum inner diameter of the sedimentation tank manifold?
- What is the required nominal pipe diameter given this flow rate? Use the `pipe.ND_SDR_available` function.

SDR is the same as given in Problem 3 (SDR = 26).

```
D_sed_manifold_min= pc.diam_circle(flow_sed_max / V_sed_manifold_max)

ND_sed_manifold = pipe.ND_SDR_available(D_sed_manifold_min, SDR)

print('The minimum inner diameter of the sedimentation tank manifold is',D_sed_
    ↴manifold_min.to(u.inch))
print('The nominal diameter of the sedimentation tank manifold is',ND_sed_manifold)
```

The minimum inner diameter of the sedimentation tank manifold is 7.266 inch The nominal diameter of the sedimentation tank manifold is 8 inch

35.4 Sedimentation Tank Bays and Number of Diffusers

35.4.1 15:

What is the total required plan area for the sedimentation tanks? Calculate this using the design flow rate and the upflow velocity between the floc blanket and plate settlers.

Give your final answer in square meters.

```
A_sed_flocblanket_total = (flow_plant / V_sed_up).to(u.m**2)

print('The plant view area of the floc blanket is',A_sed_flocblanket_total)
```

The plant view area of the floc blanket is 60 meter ** 2

35.4.2 16:

What is the total length of the floc blanket zone for all tanks? Calculate this using the total required plan area for the sedimentation tank and the sedimentation tank width.

This total length will enable you to calculate how many sed tanks are required.

```
L_sed_flocblanket_total = (A_sed_flocblanket_total / W_sed).to(u.m)

print(L_sed_flocblanket_total)
```

56.24 meter

35.4.3 17:

How many sedimentation tanks are required to treat the total plant flow? Calculate this using the the total plant flow rate and the maximum sed tank flow rate. The plant flow rate is the basis of design and the maximum sed tank flow rate is based on the manifold diameter.

Your answer should be an integer value.

```
N_sed_tanks = int(np.ceil(flow_plant / flow_sed_max))

print('The required number of sedimentation tanks is',N_sed_tanks)
```

The required number of sedimentation tanks is 10

35.4.4 18:

How much water (in L/s) can all of the sedimentation tanks for the plant treat? Assume that all tanks have been built to maximum length.

```
flow_sed_tanks_total = flow_sed_max * N_sed_tanks  
print(flow_sed_tanks_total)
```

61.87 liter / second If we use this design approach the plant design capacity will slightly exceed the requested 120 L/s. We are still working on setting the optimal design for a single sedimentation tank.

35.4.5 19:

How many diffusers are required in each tank? Assume the maximum length of the upflow region of the sedimentation tank is used. Use the np.floor function to round down to an integer value.

```
N_sed_tank_diffusers = int(np.floor(((L_sed_flocblanket_total/N_sed_tanks) / B_  
diffuser).to(u.dimensionless)))  
  
print('The number of diffuser pipes per sed tank is',N_sed_tank_diffusers)
```

The number of diffuser pipes per sed tank is 98

35.5 Plate Settlers

You may assume that the active area of the sedimentation tank is equal to the top area of the floc blanket zone. This isn't quite right because of the geometric constraints from the floc hopper, inlet channel, settled water channel, and angled plates. However, it is a good approximation for these long tanks. We will use this approximation to determine the plate settler details.

35.5.1 20:

What is the required length of the plate settlers? Do not neglect the thickness of the plate settlers.

```
L_sed_plate = ((s_sed_plate * ((V_sed_up/V_sed_capture)-1)  
+ thickness_sed_plate * (V_sed_up/V_sed_capture))  
/ (np.sin(angle_sed_plate) * np.cos(angle_sed_plate)))  
.to(u.m)  
  
print('The minimum length of the plate settlers is',L_sed_plate)
```

The minimum length of the plate settlers is 0.4619 meter

35.5.2 21:

What is the horizontal spacing (center to center) of the plate settlers?

```
B_sed_plate_horizontal = ((thickness_sed_plate + s_sed_plate)
                           / np.sin(angle_sed_plate)).to(u.cm)

print('The horizontal center to center spacing of the plate settlers is',B_sed_plate_
      ↪horizontal)
```

The horizontal center to center spacing of the plate settlers is 3.118 centimeter

35.5.3 22:

Approximately how many plate settlers spaces are needed in each sedimentation tank? Assume the maximum length of the upflow region of the sedimentation tank is used. Neglect the lost space at the end of the sedimentation tank due to the angle of the plate settlers.

Round your answer to the closest integer value.

```
N_sed_plates_pertank = int(round(((L_sed_flocblanket_total/N_sed_tanks) / B_sed_
      ↪plate_horizontal).to(u.dimensionless)))
print('The number of plate settlers per sedimentation tank is',N_sed_plates_pertank)
```

The number of plate settlers per sedimentation tank is 180

Congratulations on making it this far! Although we havent designed every component in the sedimentation tank, you have a good idea of the analysis that is required for systematic parametric design. We have not covered the topics of the inlet channel, the launder that removes clean water from the top of the sedimentation tank, the floc weir or floc hopper, or the system of equations used to calculate the final depth of the sedimentation tank. The detailed design required to create a high-performing sedimentation tank is sophisticated, complicated, and if you are successful the resulting sedimentation tank is high-performing and easy to maintain!

CHAPTER
THIRTYSIX

SEDIMENTATION DESIGN CHALLENGE

1.

CHAPTER
THIRTYSEVEN

FILTRATION INTRODUCTION

If the water is dirty, then filter it. That is a common misunderstanding of what we should do if the water is dirty. And that misunderstanding has led to many, many failures of water treatment technologies. There are slow sand filters and multistage filters abandoned in many corners of Honduras. The first Aguacala plant at Marcala, Honduras was built in an abandoned filtration plant. In many cases the failed filters were installed in locations where the raw water carried far too much sediment to be treated by filtration. This common misunderstanding of the capabilities of sand filters has a long history.



Fig. 37.1: Abandoned multiple stage filtration plant in the valley north of Tegucigalpa, Honduras.

37.1 History of Filtration

Starting in 1829 surface water treatment plants in England used slow sand filters. Decades later in 1885 they discovered that slow sand filters actually remove bacteria. That was quite a surprise given that the pores in slow sand filters are perhaps 50 times larger than the bacteria that they remove! Clearly filters were doing something more interesting than simply straining out particles that were bigger than the pores.

Slow sand filtration ran into big trouble when they were installed in cities along the Ohio and Mississippi Rivers in the United States. The high sediment load carried by those rivers could not be removed using slow sand filters. The need to treat water with high sediment loads (high turbidities) led to the innovation of coupling flocculation, sedimentation, and a new kind of filter - rapid sand filters. The early emphasis on filtration as the primary process persisted and continues even today. For example, the surface water treatment plant that provides drinking water to the Cornell University campus is called Cornell Water Filtration Plant.

The process that removes most of the particles in the Cornell Water Filtration Plant is the sedimentation tank with big assistance from the flocculators. The rapid sand filters are a polishing step that removes approximately 1 NTU or about 1 mg/L of solids. This is not to diminish the importance of filtration. After all, this chapter is all about filtration. The goal is to put filtration in context and remind all of us that both slow and rapid sand filters are limited to polishing water that is already very clean.

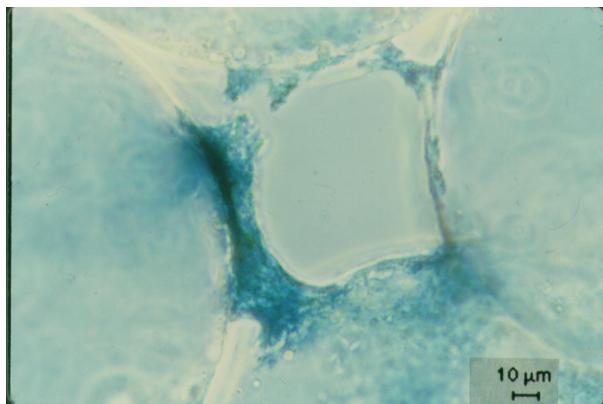


Fig. 37.2: Sample taken from the top of a slow sand filter made with glass beads. Note the constriction that is building in the narrowest part of the void between 4 glass beads.

37.1.1 Slow Sand Filters

Slow sand filters don't use coagulants or pretreatment. They are suited for surface waters coming from pristine watersheds. They are a passive technology in that there is no way for an operator to improve performance other than to clean the filter.

Slow sand filters are sometimes called bio filters, a misunderstanding that has persisted from the early observations that biofilms grow (or are imported) in the sand bed. Although biofilms do form in slow sand filters it is unlikely that they play a role in particle removal. Weber-Shirk demonstrated that removal of *E. coli* by slow sand filters was insignificant even when the filters had 1 m of head loss due to biofilm growth. Weber-Shirk also demonstrated that slow sand filters perform well even when poisoned with azide. Weber-Shirk also demonstrated that predation by a small protozoan, a chrysophyte, can play a role in the removal of bacteria. The chrysophyte is too small to ingest protozoan pathogens such as cryptosporidium or Giardia lamblia and thus those organisms are likely only removed by physical-chemical means. Similarly it is unlikely that the chrysophyte wastes energy on ingesting virus particles. The dominant particle removal mechanism in slow sand filters is physical-chemical and is the same mechanisms at work in rapid sand filters. Naturally occurring aluminum creates sticky surfaces in the sand bed and makes some of the influent particles sticky. It is likely that aluminum content in the raw water plays a significant role in slow sand filter performance. The aluminum hydroxide nanoparticle glue is likely able to effect the removal of a wide range of particle sizes.

Slow sand filters remove the majority of the particles in the top cm of the sand bed. A schmutzdecke (filter cake or layer of dirt) generally forms on top of the sand and this layer of dirt causes the majority of the head loss thru the filter bed. It is this thin layer that is removed when the head loss is too high to achieve the desired flow rate through the filter. Most slow sand filters are cleaned by hand using shovels to remove a thin layer of the schmutzdecke. The cleaning process is labor intensive and requires taking the filter offline for a day or longer.

37.1.2 Multistage filters

Multistage filters were invented to extend the range of applicability of slow sand filters. Slow sand filters were viewed as an appropriate technology for rural communities, but their very limited ability to treat water with suspended solids made them inappropriate for communities that have surface water sources with high turbidity during runoff events. Roughing filters made with various sizes of gravel were added to remove sediment. The roughing filters were essentially inefficient sedimentation tanks with 60% of their volume occupied by rocks! Roughing filters were also difficult to clean and without using coagulants their particle removal efficiency was poor. Nonetheless, multistage filters have been built in many rural communities in Latin America.



Fig. 37.3: Slow sand filters at Jesus de Otoro, Honduras. Slow sand filters require extensive areas of flat terrain or extensive earthwork to create a level spot. In the tropics the filters are open to the sky and that results in nuisance growth of algae. In climates where freezing is a problem the filters are covered.

37.1.3 Rapid Sand Filters

It is always interesting to try to understand why different technologies were invented and specifically what problem they were trying to solve. One possibility is that goal was to reduce the labor required for cleaning slow sand filters by devising a system to clean the sand in place. Given the idea of fluidizing the bed of sand with a high upflow velocity it then became tempting evolve to higher flow rates during filtration because having to clean the filter frequently was no longer a problem. Another possibility is that rapid sand filters were invented to handle higher turbidity water, but that isn't supported by the capabilities of rapid sand filters. Slow and rapid sand filters have very similar maximum influent water turbidity limits. Thus the real advantage to rapid sand filters is that they can be cleaned quickly.

Many rapid sand filters use 2 or 3 different densities and sizes of filter media to achieve a system with large pores at the surface of the filter and smaller pores deeper within the filter. If sand with a wide range of sizes is fluidized the small sand grains quickly rise to the top and the large sand grains fall to the bottom. Thus a poorly sieved sand will have small pores at the top and large pores at the bottom after backwash. The trick to creating a mix of filter media that results in big pores on the top is to use low density, large diameter media with a lower terminal velocity than a higher density, small diameter media. During backwash the less dense media ends up on top and thus this reverse stratification is achieved.

Given that many installed rapid sand filters are dual media it is worth reflecting on what problem the dual media filters are designed to solve. One possible problem is surface clogging of the sand filter when large, strong flocs bridge across sand pores and then support a layer of flocs above the sand bed. This surface layer would be similar to the schmutzdecke in slow sand filters and could be the location of significant head loss. It appears that rapid sand filters do suffer from surface clogging in some instances. Of course, it is possible that this failure is related to the use of flocculent aids that result in excessively strong flocs and/or sedimentation tanks that don't effectively capture large flocs.

Additional potential benefits of dual media filtration include the possibility that they can hold a larger mass of particles before requiring backwash or that they may be able to achieve higher particle removal efficiency. [Zouboulis et al.](#) compared single and dual media filtration and demonstrated that the main advantage to dual media filtration was that it had a lower rate of head loss accumulation. Given that the rate of increase of head loss for the dual media filters was constant over time suggests that the pores that were doing the filtration were exclusively in the larger of the two media. This suggests that larger diameter media (rather than dual media) might be a better choice for rapid sand filters.

Dual media filters face more challenges during cleaning. The use of relatively low density media for the top layer results in a lower density fluidized bed. This lower density fluidized bed makes it possible for flocs to form mudballs that are sufficiently dense to fall through the top layer of media during backwash (page 221 of Water and Wastewater Technology, 6th edition, by Hammer and Hammer, 2008). The formation of these mudballs is one of the reasons that dual media filters use air scour before or during backwash to breakup the mudballs.

The addition of flocculation and sedimentation prior to filtration changes the properties of the incoming particles. The settled water contains small flocs and primary particles with some coagulant nanoparticles attached. These particles are sticky and the stickiness undoubtedly influences the physics of the particle attachment processes inside the sand bed. We will talk more about the mysteries of life inside a filter bed later in this chapter.

Rapid sand filters rely on a backwash system. It takes a relatively high velocity of water to fluidize a bed of sand and that water has to come from somewhere. In conventional water treatment plants that water comes from a clearwell that is filled with filtered water. During backwash water from the clearwell is pumped through the filter and then either wasted or returned to the head of the plant for another pass through the treatment processes. The need for pumps is one of the big disadvantages of conventional rapid sand filters. Conventional rapid sand filters fail if they lose power or if there isn't enough filtered water in the clearwell when the filter needs to be backwashed. This raises an interesting question about how rapid sand filters are initially commissioned at water treatment plants given that the clearwell is initially empty and given that filters MUST BE FILLED WITH WATER FROM THE BOTTOM. If you dump water on top of a dry filter bed the water will create a few paths down through the sand and the majority of the sand pores will remain filled with air even when the level of water floods the top of the filter bed. Thus the first operation in rapid sand filter startup must be backwash!

One solution for eliminating the need for backwash pumps is to design a set of filters that work together to backwash one of the filters. This eliminates the need for a clearwell for backwash water. This system of a set of filters is used by CEPIS in their design of [rapid sand filters](#). This system is an elegant solution that works well for small cities, but the requirement of 6 filters is onerous for towns where 2 filters should have been sufficient.



Fig. 37.4: 6 filters form a complete set such that 5 filters can work together to backwash one filter without using pumps.

37.1.4 Stacked Rapid Sand Filters

Stacked Rapid Sand, StaRS, filters were invented in 2010 by the AquaClara Cornell program in response to the need for a new technology that would both eliminate the need for backwash pumps AND not require the construction of 6 filters for small towns. StaRS filters use 6 20 cm deep layers of sand (no dual media here) with the layers stacked vertically. The six layers give a total sand depth of 1.2 m. The filter operates with the same design flow rate for both backwash and filtration modes and uses settled water for backwash. This eliminates the startup problem for rapid sand filters that don't have an initial source of backwash water. The shallow filter layers take advantage of new insights into the filtration mechanisms (more on that soon).

StaRS filters come in two configurations. Open StaRS (OStaRS) are used for flow rates greater than about 8 L/s. An 8 L/s filter has a plan view area of 85 cm x 85 cm. This is considered the minimum size that can be constructed with a human working inside the filter. For lower flow rates Enclosed StaRS (EStaRS) filters can be assembled using PVC pipe as the body of the filter. The inner plumbing can be accessed through openings in the top and bottom of the main filter body.

```
from aquaclaracore.units import unit_registry as u
import numpy as np

Q_Filter = 8 * u.L/u.s
V_filter_backwash = 11 * u.mm/u.s
A_Filter = Q_Filter/V_filter_backwash
W_Filter = np.sqrt(A_Filter).to(u.m)
print('The width of a filter designed to treat ',Q_Filter,' is ', W_Filter,'.')
```

Table 37.1: Relative velocities and areas for different unit processes.

Filter Type	Velocity (mm/s)	Cleaning (mm/s)	Max (NTU)	pC*	Area (m^2) for 1 L/s
Dynamic	0.4			0	2.5
Roughing	0.17	5.5 mm/s down-flow		0.5	5.9
Slow	0.04	Scrape surface	10	0.8	25
Multistage	0.03		100	1.3	33.4
Rapid	0.7 - 2.8	11 mm/s back-wash	5	1	0.55
AC Entrance	8				0.125
AC Flocculation	4				0.25
AC Sedimentation	1			2.5	1
AC Floc Hopper	5				0.2
AC Stacked Rapid	1.8*6	11 mm/s back-wash	3	1	0.093
AC total	0.6		1000	3.5	1.67

The slowest (and hence largest) unit process in the AquaClara surface water treatment train is sedimentation. Thus sedimentation represents the most likely opportunity for significant cost reduction. The entire AquaClara treatment train is a fraction of the size of the dynamic filters that are used in multiple stage filtration plants.

Filters remove the least amount of contaminants in a surface water treatment plant.

Filters have a huge range in velocities that correspond to a huge range in size. This is illustrated in the city of Jesus de Otoro, Honduras. They have both a 20 L/s multiple stage filtration plant and a 20 L/s AquaClara plant serving different sections of the city. The plants can both be seen in [Google Maps](#).

37.2 Filtration Theory

Filters are used to remove particles and thus we'd like to be able to predict particle removal efficiency in a filter. Unfortunately we don't yet have equations that describe particle removal by sand filtration. This is a very unpleasant surprise. It is as if we were designing a suspension bridge and didn't have any equations describing the relationship between the tension in the cables and the load they are supporting. We only have an equation describing what the cables do when there isn't any additional load. In the case of filtration we only have clean bed filtration models that attempt to describe what happens before the filter begins to remove particles.

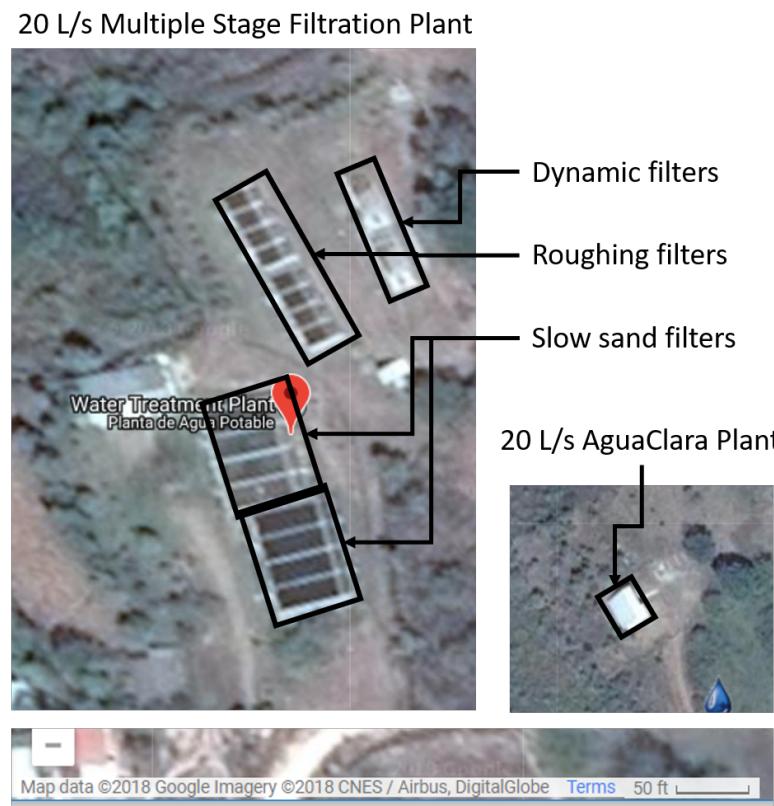


Fig. 37.5: Two water treatment plants using different technologies and serving the same city illustrate the high construction cost of low technologies simply based on the required size of the low tech facilities.

Reflection: How did we get to 2018 without a model for filter performance? There may be several reasons for the lack of a filtration model.

- The lack of data acquisition systems in university laboratories means that very few rapid sand filters were operated and evaluated for full filter runs in laboratory settings
- The Environmental Engineering fixation on jar tests as the way to model water treatment plants provided no method to test filtration and thus most university laboratories only experimented with batch operation and not continuous flow.
- Filtration models were borrowed from air filtration (Yao et al, 1971) and thus did not take into account that the coagulant nanoparticles made particle attachment to surfaces very favorable.
- Filtration models only modeled the clean bed phase(the first few minutes of a filter run) before particles were deposited and began altering the geometry of the pores.

Clean bed filtration models include an equation first presented by Iwasaki in 1937 that suggested that particle removal occurred as a first order process with respect to depth. This simplifies to

$$pC^* \propto \frac{L}{D_{sand}} \quad (37.1)$$

where L is the depth of the sand in the column. This suggests that increasing the depth of sand in a filter would dramatically improve performance.

It is quite amazing that we have no useful models for sand filter performance after more than a century of using sand filters as a required process in converting surface waters into safe drinking water. Fortunately we have plenty of clues suggesting what is happening inside filters and at the level of the particles traveling through the pores.

The movie in Fig. 37.6 illustrates that the classic performance of a rapid sand filter as a function of time could be explained by the presence of an active filtration zone that slowly progresses down through the filter as the pores become fully loaded. But that begs the question of what determines fully loaded.

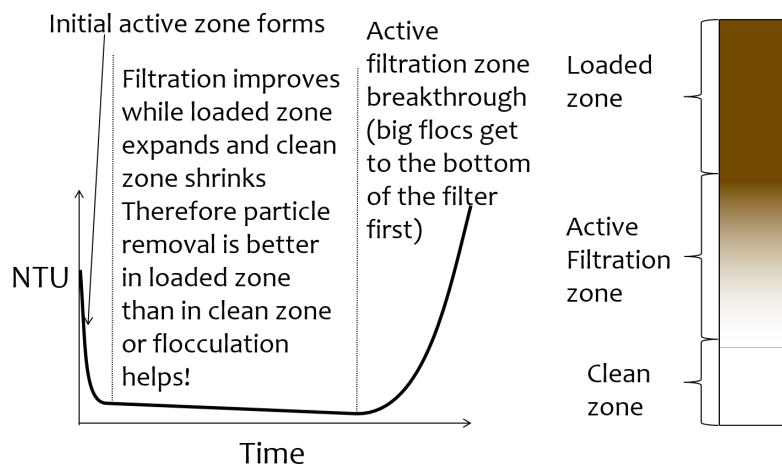


Fig. 37.6: Movie illustrating how effluent turbidity connects to deposition of particles within a sand bed.

The linear increase in head loss with time (see Fig. 37.7) is a remarkable and surprising property of depth filtration. The deposition of particles in a pore would change the flow geometry, increase the flow velocity, and increase the head loss across the pore. As each particle is deposited in the pore it would be expected to have an increasing impact on head loss as the flow area is decreased and the flow velocity increases. Head loss due to the flow expansion downstream from the flow constriction is proportional to the velocity squared. Thus we would not expect head loss across a pore to increase linearly with time.

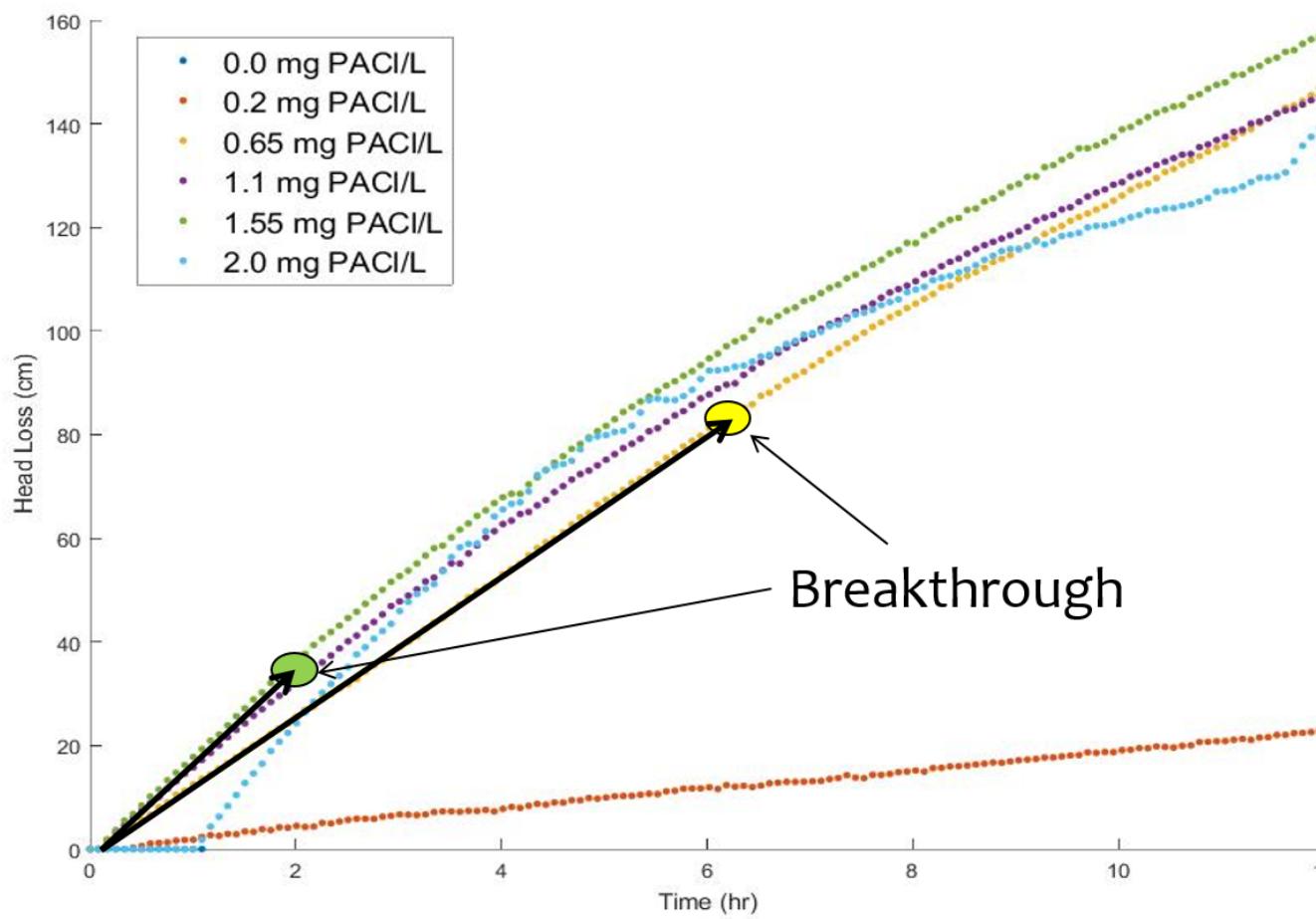


Fig. 37.7: Head loss through a filter increases linearly with time under conditions of constant influent turbidity.

The data is trying to tell us something. The head loss through a filter does increase linearly with time and with the amount of solids that have been captured by the filter. Our hypothesis is that the number of fully loaded pores is increasing linearly with time. To understand how this might be working we need to delve into the filter and imagine what must be happening in the pores as particles are flowing by.

[Fig. 37.8](#) illustrates that particles that are carried by the fluid can come into contact with the walls of the pores where the streamlines converge. Clean bed filtration models predict that this process of interception is significant even for clean beds. As a pore begins to fill due to particle deposition, the flow constriction becomes even more severe and thus the capture efficiency increases. This would suggest that a pore would rapidly fill with flocs until it became so clogged that the flow of water through the pore effectively stopped.

No removal by interception in straight tube

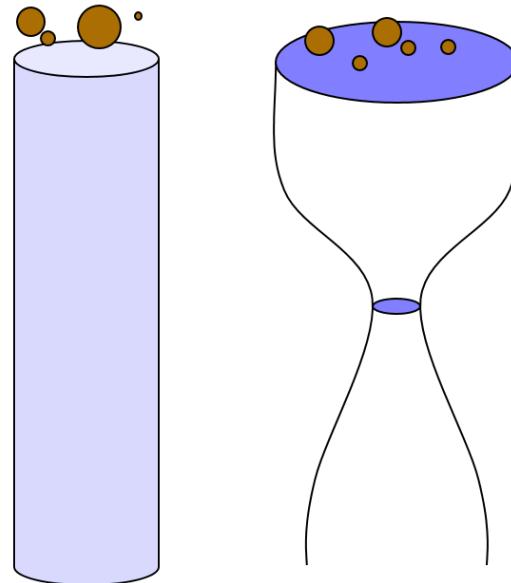


Fig. 37.8: Movie illustrating that particles are transported closer to sand surfaces where streamlines converge.

If pores clogged completely, then sand filters would clog as soon as one layer of pores was filled. There must be another process that is preventing pores from clogging. We hypothesize that flocs cant attach to the filter surfaces when the fluid drag on the floc exceeds the strength of the bonds between the coagulant nanoparticles and the particle and pore surfaces. Thus as pores grow smaller due to deposition it becomes more difficult for incoming flocs to attach. Instead, the flocs are shoved right through the pore even if it means the floc has to undergo significant deformation to squeeze through.

The story is beginning to emerge. Converging streamlines result in flocs touching previously deposited particles in the flow constriction and attaching. As the flow constriction becomes even smaller the fluid drag on particles that attempt to attach to the pore becomes too great and the particles are forced through the constriction. At that point the pore is fully loaded. However, fully loaded does not mean that the void is full of flocs. It just means that the constriction is as small as it can get given the strength of the coagulant nanoparticle bonds and the fluid drag on the flocs.

Pores at the upstream end of the filter are fully loaded first and then remain relatively unchanged for the duration of the filter run. Thus the active filtration zone is pushed deeper into the filter bed. At any one time there are likely a series of pores that are partially loaded and thus actively filling. The number of pores in series that are actively filling is not yet known, but it is clear that the particle removal efficiency of a filter must be related to the number of actively



Fig. 37.9: Movie showing flocs flowing through a pore created by sand grains. (Credit Filter_Constrictions team Spring 2018)

filling pores in series.

The active filtration zone is progressing through the filter at a constant velocity. The number of fully loaded pores is increasing linearly with time! That is why head loss increases linearly with time.

The poor performance at the beginning of a filter run is because the clean zone isn't a good filter. This suggests that dual media filters are pointless because the smaller media at the downstream end of the filter remains clean and doesn't actually do anything. If this is correct, then multimedia filters can be replaced with a single larger media size.

As coagulant dose increases

- flocs are larger and thus they fill the pores faster and thus the active zone moves faster through the filter
- large flocs are removed more efficiently and thus the active zone is thinner (fewer active pores in series)
- thinner active zone results in faster failure (higher slope of turbidity vs time during failure)
- larger flocs results in faster failure (higher slope of turbidity vs time during failure)

As sand size increases

- There are fewer pores and thus fewer constrictions and thus less head loss at the end of the filter run
- Each constriction requires more particles to build and thus there is less head loss per mass of particles removed.

37.2.1 What about particle removal efficiency?

This is the multi-decade old question that challenges us to continue our research. What determines how many particles sneak through a water treatment plant? We've learned that flocculation runs out of steam because the primary particles only want to collide with other primary particles and thus they start taking forever to collide as they become scarce. The floc blanket likely acts like a series of collectors (can't say it is like a filter because it doesn't have stationary constrictions). This would suggest that more floc blanket is always better. For some reason some primary particles make it through the floc blanket. What determines how many of those primary particles make it through the filter? It must depend on the geometry of every constriction. That is so complicated. Is there another way to think about this? Large flocs are easy to capture in a sand filter. Primary particles are much more difficult to capture. Large flocs tend to fill up the first unfilled pore they come to. Thus large flocs tend to take active pores out of service. This suggests that the influent floc size distribution might influence filter performance. See [Shear big flocs to improve filter performance](#) for an analysis of the feasibility of breaking up flocs at the point of injection into the sand bed.

Zouboulis et al. found that dual media filter produced water of slightly higher turbidity and it produced significantly less head loss than a single media filter. This comparison was

CHAPTER
THIRTYEIGHT

FILTRATION DESIGN

This section deals with search for a self-backwashing filter which is functional over a wide range of flows. While in basic concept, running water through a sand bed, filters are the simplest of the unit processes, they are probably the most complex in design within an AguaClara plant because they are not inherently self-cleaning. Additionally. The search to overcome this problem has led to the development of a Stacked Rapid Sand Filter (StaRS Filter) a novel filter design which provides a hydraulic backwash system and works over a large range of flows with some adaptation for small flows.

38.1 Important Terms and Equations

Terms:

1. Porosity
2. StaRS Filter
3. Backwash
4. Pressure Recovery

Equations:

- 1.

38.2 AguaClara Flow Control and Measurement Technologies

38.2.1 Filter Types

The principal difference in various filters is the difference in velocity of water through them. This in turn determines the plan view area of the filter for a particular flow.

For a multistage filter system, the filter areas for the Dynamic, Roughing, and Slow Sand filters are included in a total area. For a given flow, and the velocity for each filter type the total area is:

$$A_{Total} = \frac{Q}{v_{Dynamic}} + \frac{Q}{v_{Rough}} + \frac{Q}{v_{Slow}} \quad (38.1)$$

Using this formula it becomes easy to see the relative sizes of different filter systems, we as how complex filter system quickly add up in terms of size. TABLE:*_table_net_velocities* shows the relative area of various filtration technologies.

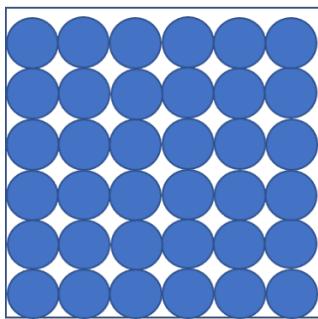
Understanding the amount of area requires for this component makes it easy to see why certain systems would be preferable to others, but also that overall filtration is only a polishing step and cannot treat as well as other unit processes, predominantly the flocculation-sedimentation combination.

38.3 Porosity

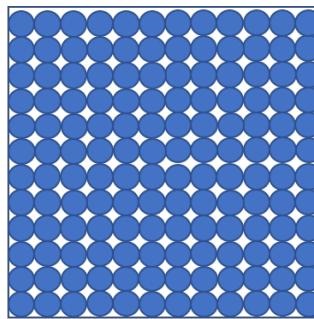
In understanding how sand filtration works, porosity is one of the most important concepts to be familiar with. Porosity refers to the ratio of the void volume to the total volume of a control volume.

$$\phi_{FiSand} = \frac{V_{voids}}{V_{total}} \quad (38.2)$$

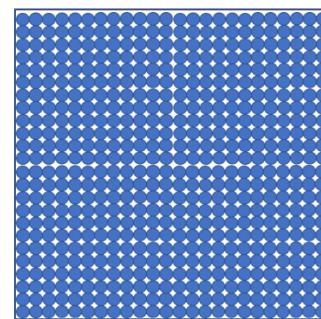
Porosity is determined by the geometry of the material in the control volume, but also by the size of the particles involved. If you have three different sized spheres (such as .0um clay, .2mm sand, and 1 cm gravel) in three different buckets, each bucket will have the same porosity as seen in Fig. 38.1. To minimize the porosity, the three materials could be mixed because the smaller materials would be filling the pore space of the larger material.



LargeDiameterSpheres



MediumDiameterSpheres



SmallDiameterSpheres

Fig. 38.1: Within each box, the spheres are different sizes. However the total porosity is the same. To minimize the pore space, the smaller particles could be used to fill the pore space between the larger particles, though in a filter this is not necessarily ideal.

One way that the relative size of particles is characterized is by describing the size of the smallest 10% of grains, and the smallest 60% of grains. That is:

D_{10} = the sieve size that passes 10% by mass of sand through

D_{60} = the sieve size that passes 60% by mass of sand through

D_{10} is used for particle removal models, and D_{60} is used for hydraulic modeling.

The ratio of the two is the uniformity coefficient:

$$UC = \frac{D_{60}}{D_{10}} \quad (38.3)$$

The uniformity coefficient describes the uniformity of the sand. A $UC = 1$ indicates that every grain of sand is the same size, which is the ideal case. A large UC is indicative of a wide range of grain sizes which can cause trouble during filter operation and backwash, as stratification occurs and the porosity changes with respect to depth in the filter, which affects the requirements for backwash.

During backwash, the sand is fluidized and the sand bed expands. This expansion causes a change in porosity of the sand bed (as the volume of water occupied by the sand is increased). The porosity and height of the sand bed are directly related through the following equation:

$$\phi_{FiSandBw} = \frac{\phi_{FiSand} H_{FiSand} A_{Fi} + (H_{FiSandBw} - H_{FiSand}) A_{Fi}}{H_{FiSandBw} A_{Fi}} \quad (38.4)$$

Such that:

$\phi_{FiSandBw}$ = sand porosity during backwash

ϕ_{FiSand} = settled sand porosity

H_{FiSand} = height of sand in the filter

$H_{FiSandBw}$ = height of sand during backwash

A_{Fi} = filter area

From this it becomes possible to directly relate porosity (as above) to the filter expansion ratio, which is simply the ratio of the heights of the expanded sand bed and the settled sand bed:

$$\text{label : filter expansion ratio} \quad (38.5)$$

$$\Pi_{FiBw} = \frac{H_{FiSandBw}}{H_{FiSand}}$$

Such that:

Pi_{FiBw} = the expansion ratio value

H_{FiSand} = height of sand in the filter

$H_{FiSandBw}$ = height of sand during backwash

38.4 Headloss Requirements

One of the key parameters in design of a filter is the headloss through the system because it determines the required fluid velocity for backwash. The Carman Kozeny Equation, an adaptation of the Hagen-Poiseille equation (ref from elsewhere, not linked yet) describes the headloss through a clean bed during filtration.

$$\frac{h_l}{H_{FiSand}} = 36k \frac{(1 - \phi_{FiSand})^2 \nu V_{Fi}}{\phi_{FiSand}^3 g D_{60}^2} \quad (38.6)$$

Such that:

h_l = headloss in sand bed

H_{FiSand} = the sand bed depth/length of flow paths

ϕ_{FiSand} = porosity of sand

ν = kinematic viscosity

V_{Fi} = the water velocity in the filter

D_{60} = the size of the sand

g = gravity

k = Kozeny constant (5 for most filtration cases)

This equation is valid for Reynolds numbers less than 6. Where: $Re = \frac{D_{60} V_{Fi}}{\nu}$

The headloss during backwash is taken as the design parameter, so other values are constructed around it.

The following equation describes the headloss through the fluidized bed:

$$\frac{h_{l_{FiBw}}}{H_{FiSand}} = (1 - \phi_{FiSand}) \left(\frac{\rho_{Sand}}{\rho_{Water}} - 1 \right) \quad (38.7)$$

Such that:

$h_{l_{FiBw}}$ = the headloss in the fluidized bed

H_{FiSand} = the depth of the settled sand bed

ϕ_{FiSand} = the settled sand porosity

ρ_{Sand} = the sand density

ρ_{Water} = the water density

Using these two equations the minimum velocity for sand fluidization can be found!

$$V_{MinFluidization} = \frac{\phi_{FiSand}^3 g D_{60}^2}{36k\nu (1 - \phi_{FiSand})} \left(\frac{\rho_{Sand}}{\rho_{Water}} - 1 \right) \quad (38.8)$$

From this equation it can easily be seen that if the diameter of the sand at the top is half the diameter of the sand at the bottom, it will fluidize at one quarter the velocity. This result indicates that fluidization occurring at the top of the filter is **not** indicative of fluidization at the bottom.

This parameter is the most important parameter to consider as it is a property of the sand not of the water!

38.5 Backwash

When considering backwash design, there are two main factors that constitute a dilemma. The first, backwash velocity must be greater than filtration velocity (to expand the sand bed), and second, the backwash water must be clean water (cleaning with dirty water introduces more particles into the filter). This limits the paths water can take during the backwash process. The conventional options include pumping it back from the storage tank, using a set of parallel filters to backwash one filter at a time, or storing the filtered water at an adequate elevation. Due to energy limitations and space constraints, the conventional solutions are simply not feasible for this system. Examples that illustrate why they cannot work can be found in the derivations sections(?) (or the examples?)

brief example here?

To avoid electricity, pumps can be immediately ruled out.

Parallel filters would require too much area and wouldn't work well under low flow conditions:

Given:

$$Q_{Plant} = 6 \frac{L}{s} \quad V_{Fi} = 1.8 \frac{mm}{s} \quad V_{Bw} = 9 \frac{mm}{s} \quad (38.9)$$

As the ratio of the backwash velocity to the filter velocity is 5, 5 filters will be needed to provide enough flow to backwash one: Therefore the number of parallel filters is 6:

$$N_{Fi} = \frac{V_{Bw}}{V_{Fi}} + 1 = 6$$

In this system, the water exiting five of the filters would be diverted to backwash one of the other filters. In addition to requiring the plan view area of 6 filters, each filter would need to be backwashed independently, meaning it would take 6x longer and use 6x the water as compared to just having one filter. Another detriment to this system is that in low flows (such as drought conditions) not enough water would be passing through the system to backwash at points since all the water is diverted to backwash.

The third option, elevating the filtered water to provide enough head to cause backwash, is also unfeasible.

add the third one at some later point if its useful

How can we find a solution?

If the velocities could be more similar the filter could work!

This could be accomplished in several ways: such as decreasing the media density thus lowering velocity to fluidize it, decrease the media diameter thus lowering the fluidization velocity, or make a more compact filter which filters in parallel and backwashes in series.

As changing the material characteristics of the sand is challenging, a more compact filter is the chosen design. As it happens this innovation results in a more conceptually difficult filter. In the design, six layers of sand are stacked, there are four inlets, and three outlets which are all in use during filtration. During backwash only one inlet is used

and the backwash water is discarded through a separate manifold. Throughout this section, figures and images will be the best methods to understand the design flow through the system, and will be supplemented by the text.

This overall design can be seen in Figure XXXXX.

(figure of the full system)

Tasks for Clare for Thursday + Friday morning: insert images! none of them are in yet. Save as pngs. streamline the way you want this to work as well. like overall structure

In is most basic schema, the filter is a series of pipes leading into a deep box with 1.2 meters of sand (for most filters)

As a parcel of water traveling in the filter the first part of the filter is the inlet box. The inlet box is a shallow box with four holes in the bottom. The holes lead into four pipes which lead into different levels of the sand filter. At the outlet of each of these pipes into the sand filter is a structure designed to spread the flow over the entire footprint area of the filter. These structures have slots which allow water out of the inlet pipes into the sand bed. Across a layer of sand from the inlet is an outlet pipe in the same shapes as the slotted pipe inlets. Water passes into the pipes and up to the filter outlet box where it only needs to be chlorinated before being distributed.

Steps of designing a filter. 1. Calculate backwash velocity from sand diameter - complicating factors: – wall shear – lift as a unit and fall together stratification will always occur if a range of sizes exists 2.

38.6 Layer Thickness as a function of diameter

Two heights are defined here. Both are defined as functions:

$H_{FiLayerF}$ the height of a standard filter layer. It is defined as the maximum value between the defined layer height, the outer radius (OR) of the trunk (as a function of nominal diameter) plus the minimum filter trunk spacing , the outer diameter of the fitting of the trunk (also as function of ND), and the fernco outer diameter (again a function of ND). Each of these is rounded to the nearest 1cm.

and

$H_{FiBottomLayerF}$ the height of The bottom filter layer. This is defined as the maximum value between 1. The defined layer height, the OR of the trunk + OR of the backwash trunk (as functions of NDs of the trunk and BW trunk) + minimum filter trunk spacing, the sum of fitting ORs for the trunk and BW trunk (function of NDs), and the average of the ferco ODs as a function of NDs. Each rounded to the nearest 1cm.

38.7 Siphon

Backwash is initiated through the activation of a siphons system. In the siphoning process two things must happen for a success:

1. Backwash can be initiated at any point in time
2. Backwash can be prevented with the siphon as well

In under the siphon, the ideal gas law, **Add a ref**, is the most important equation to understand how siphoning works..

38.8 Design from MathCad adapted for me

Overall Goals: - Uniformity in the sand bed through appropriate headloss

Note: Pressure recovery occurs when fluid velocity slows down. This happens in the filter at the end of a manifold pipe, because the fluid must stop at the end. This stoppage causes the pressure to increase locally, which then, when the fluid passes into the outlet, increases the local velocity leading to non-uniform flow.

Constraints:

Filtration Constraints - inlet manifolds need to have small piezometric head relative to clean sand bed - trunk and branches are short manifolds (fL/d) (**friction factor, L, diam?**), therefore: piezometric head variability dominated by pressure recovery as given by $\frac{V^2}{2g}$ with the velocity as the initial velocity of the manifold – this is the limiting constraint for the velocity in the trunk and branches, which can be relaxed with added headloss in the slots - limit the imbalance between inlets that carry water for two layers and inlets that carry water for one layer. – *perhaps top/bottom inlet should have half the number of slots* – current assumption: backwash branches will have half the slot areas - if slots are used to generate head loss (to improve flow distribution) sand blocking, $(1 - \epsilon)$ during filtration must be accounted for - the slots are not blocked during backwash (except maybe at initiation)

Backwash Constraints: - flow rate from each slot/hole must be close to average (within 20%) - as there is no head loss in fluidized bed, the manifold must be designed with the appropriate head loss for independent uniform flow distribution - headloss through slots/holes will be 36x greater during backwash because of 6x increase in velocity - to make flow more uniform: – decrease head loss + pressure recovery in manifold – increase head loss in slots - slot head loss must not impede the filtration mode capacity - this sets maximum headloss for each mode. – during filtration <10% clean bed head loss – during backwash <36x the filtration limits

Additional concerns: - there are two manifold systems, the trunks into the branches and the branches into the slots. Thus pressure recovery must be small in both trunk and branches so that the slots can reconcile the changes in flow, because flow distribution is fairly uniform in short manifold systems when total port area is equal to or less than manifold area.

38.9 Sand Layer Thickness as Function of trunk diameters

38.10 Flow Distribution constraint: ratio of pressure recovery to clean bed head loss

38.11 Number of filters

38.12 Clean bed head loss

38.13 Auxilliary box widths and plumbing

38.14 Number of manifold branches

38.15 Filter box dimensions and manifold inlet pipes

38.16 Manifold slot/orifice design

38.17 Plumbing head losses

38.18 Total Sand depth and backwash head loss

38.19 Path head loss calculations and flow distribution between layers

38.20 Siphon design

38.21 Inlet channel and elevations

38.22 Elevations

38.23 Inlet Weir Design

38.24 Entrance and overflow box lengths (X-direction)

38.25 Siphon Valve sizes

38.26 Sand Removal Pipe

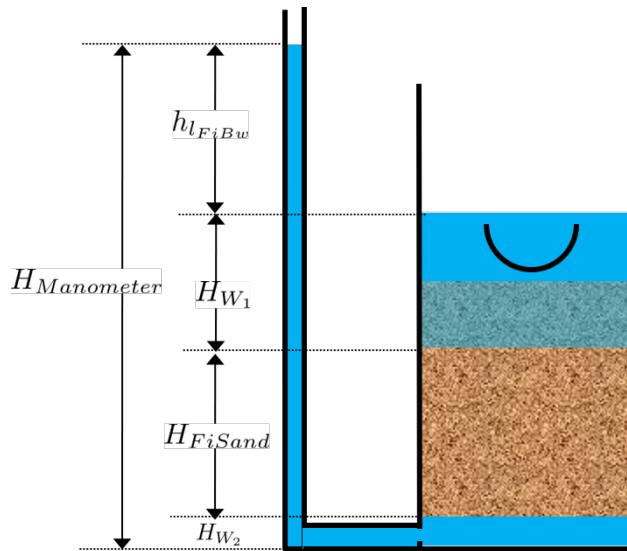
38.27 Trunk Line purge valves

38.28 Main plant drain channel

38.29 Sand Volume

FILTRATION DERIVATIONS

To determine the head loss during backwash a force balance should be performed between the water and the sand per unit of filter area (thus pressure values will be yielded). A schematic for this system is shown below:



The pressure from the water:

$$P_{Manometer} = \rho_{Water} g (H_{W_1} + H_{W_2} + \phi_{FiSand} H_{FiSand}) + \rho_{Sand} g (1 - \phi_{FiSand}) H_{FiSand} \quad (39.1)$$

Such that:

$P_{Manometer}$ = total height from the bottom of the filter to the inlet box

ρ_{Water} = density of water

H_{W_1} = the distance from the top of the settled sand bed to the water surface in the filter

H_{W_2} = the height of the water below the sand bed but within the filter

ϕ_{FiSand} = porosity of sand

H_{FiSand} = height of the filter bed

ρ_{Sand} = density of sand

The pressure from the sand:

$$P_{Manometer} = \rho_{Water} g (H_{W_1} + H_{W_2} + H_{FiSand} + h_{FBw}) \quad (39.2)$$

Such that:

$h_{l_{FiBw}}$ = the difference in height of the inlet and water surface height during backwash; the backwash head loss

Setting them equal for a force balance:

$$\rho_{Water}g(H_{W_1} + H_{W_2} + \phi_{FiSand}H_{FiSand}) + \rho_{Sand}g(1 - \phi_{FiSand})H_{FiSand} = \rho_{Water}g(H_{W_1} + H_{W_2} + H_{FiSand} + h_{l_{FiBw}}) \quad (39.3)$$

Which simplifies to:

$$h_{l_{FiBw}} = \frac{\rho_{Sand} - \rho_{Water}}{\rho_{Water}} (1 - \phi_{FiSand}) H_{FiSand} \quad \text{or} \quad (39.4)$$

$$h_{l_{FiBw}} = H_{FiSand} (1 - \phi_{FiSand}) \left(\frac{\rho_{Sand}}{\rho_{Water}} - 1 \right)$$

This result gives a ratio of the head loss during backwash to the height difference during forward operation. With $\phi_{FiSand} = 0.4$ and $\rho_{Sand} = 2650\text{kg/m}^3$ the value of this ratio is:

$$(1 - \Phi_{FiSand}) \left(\frac{\rho_{FiSand}}{\rho_{Water}} - 1 \right) = 0.99 \quad (39.5)$$

Thus:

$$h_{l_{FiBw}} = H_{FiSand} * 0.99 \quad (39.6)$$

39.1 Siphon Derivation

The siphon is defined by its airtrap!

More things here!

Overall Goals: - Uniformity in the sand bed through appropriate headloss

Note: Pressure recovery occurs when fluid velocity slows down. This happens in the filter at the end of a manifold pipe, because the fluid must stop at the end. This stoppage causes the pressure to increase locally, which then, when the fluid passes into the outlet, increases the local velocity leading to non-uniform flow.

Constraints:

Filtration Constraints - inlet manifolds need to have small piezometric head relative to clean sand bed - trunk and branches are short manifolds (fL/d) (**friction factor, L, diam?**), therefore: piezometric head variability dominated by pressure recovery as given by $\frac{V^2}{2g}$ with the velocity as the initial velocity of the manifold 1. this is the limiting constraint for the velocity in the trunk and branches, which can be relaxed with added headloss in the slots - limit the imbalance between inlets that carry water for two layers and inlets that carry water for one layer. -1 *perhaps top/bottom inlet should have half the number of slots* -2 current assumption: backwash branches will have half the slot areas - if slots are used to generate head loss (to improve flow distribution) sand blocking, $(1 - \epsilon)$ during filtration must be accounted for - the slots are not blocked during backwash (except maybe at initiation)

Backwash Constraints - flow rate from each slot/hole must be close to average (within 20%) - as there is no head loss in fluidized bed, the manifold must be designed with the appropriate head loss for independent uniform flow distribution - headloss through slots/holes will be 36x greater during backwash because of 6x increase in velocity - to make flow more uniform: 1. decrease head loss + pressure recovery in manifold 2. increase head loss in slots - slot head loss must not impede the filtration mode capacity - this sets maximum headloss for each mode: 1. during filtration <10% clean bed head loss 2. during backwash <36x the filtration limits

Additional concerns: - there are two manifold systems, the trunks into the branches and the branches into the slots. Thus pressure recovery must be small in both trunk and branches so that the slots can reconcile the changes in flow, because flow distribution is fairly uniform in short manifold systems when total port area is equal to or less than manifold area.

39.2 Flow Distribution constraint: ratio of pressure recovery to clean bed head loss

There are three flow distribution problems in the filter design: 1. Between slots along manifold branches 2. Between branches along manifold trunks 3. Between filter layers

The relative distribution of the flow through a particular path is defined as:

$$\Pi_Q = \frac{Q_{long}}{Q_{short}} = \sqrt{\frac{C_{pshort}}{C_{plong}}} \quad (39.7)$$

Such that:

Π_Q = the ratio of flow

Q_{long} = the flow through the longest filter path (lowest layer, at the furthest slot on the furthest branch)

Q_{Long} = the flow through the shortest filter path (top layer, closest slot on the first branch)

C_{pshort} = pressure coefficient at the end of the shortest path

C_{plong} = pressure coefficient at the end of the longest path

C_p is defined in Fluids review (**Make this actually be defined here**)

$$\Pi_Q = \frac{Q_{long}}{Q_{short}} = \sqrt{\frac{H_{LSand} - PR}{H_{LSand}}} \quad (39.8)$$

Such that:

H_{LSand} = the head loss in the sand bed

PR = pressure recovery (as defined by: $\frac{V^2}{2g}$)

$$PR = H_{LSand}(1 - \Pi_Q^2)$$

These relationships define the head loss constraints of the filter.

The ratio, $\Pi_{ManifoldQ}$ has been arbitrarily given a value of 0.85, meaning the flow exiting the longest path is 85% of the flow exiting from the shortest path.

Thus from above:

$$1 - \Pi_{ManifoldQ}^2 = .278 = \Pi_{ManifoldHeadLoss}$$

Where the ratio of the pressure recovery in the branches to the head loss through the clean bed (or through just the slots/holes in backwash) is:

$$\Pi_{ManifoldHeadLoss} = \frac{PR}{H_{LSand}}$$

Though the piezometric head profiles for the inlet and outlet manifolds for the middle layers may be parallel, meaning the pressure recovery is less constrained for a good flow distribution, we still need a tight constraint for the outer manifolds where the velocity is 1/2 and the PR is 1/4 that of the inner layer, while smaller still in the bottom-most manifold where the velocity head is tiny as the diameter is larger.

39.3 Number of filters

Parameters: $PR_{ManBranchEst} = 0.8\text{cm}$ $ND_{BwTrunkMax} = 8\text{in}$ $HL_{BWSlotsEst} = 10\text{cm}$ $PR_{BwManifoldMax} = HL_{BWSlotsEst} * \Pi_{ManifoldHeadLoss}$

Explain where these came from

First Constraint: Pressure Recovery in trunks during forward Filtration;

Use the Kozeny equation to find the headloss through the clean bed. Assume the depth of the sand bed as calculated above in the _heading_sand_layer_from_trunk_diam section

Second Constraint: Pressure recovery in lowest trunk during backwash

$$V_{BwManTrunkMaxPR} = \sqrt{2g * (HL_{BWSlotsEst} * \Pi_{ManifoldHeadLoss} - PR_{ManBranchEst})} \quad (39.9)$$

This velocity is used to find the flow possible in the pipe, using the inner diameter of the pipe, and is rounded to the nearest 1 L/s.

$$Q_{PRBwTrunk} = \pi * ID_{BwTrunkMax}^2 * V_{BwManTrunkMaxPR} \quad (39.10)$$

The flow set by the maximum pressure recovery is then the lesser of the flow calculated from forward filtration or backwsh. This value is the maximum flow through on filter.

Knowing the maximum flow through one filter, finding the number of filters is simple.

If the plant flow is less than 16 L/s, EStaRS should be used, as having two filters is ideal, but the minimum filter flow is 8 L/s, which is not possible below 16 L/s. In all other cases at least 2 filters should be used to allow for backwash during low flows.

Thus the number of filters for plants is:

$$\max\left(\frac{Q_{Plant}}{Q_{MaxPR}}, 2\right) \quad (39.11)$$

The flow through each filter given the number of filters:

$$Q_{Fi} = \frac{Q_{Plant}}{N_{Fi}} \quad (39.12)$$

Within a filter, the flow through each layer:

$$Q_{Layer} = \frac{Q_{Fi}}{N_{Layer}} \quad (39.13)$$

This is the flow that sets the pipe size for each trunk within each layer of the filter. The Nominal Diameter (ND) of the trunk pipes is then determined using the available pipe sizes. This design assumes SDR 26 to be conservative and avoid looping.

First, find the diameter based on the flow and velocity. A doubled flow is used because the two middle trunks must carry flow for two layers (**check this for correctness**)

$$\frac{2Q_{Layer}}{V_{ManTrunkMaxPR}} = A_{TrunkCalc} = \frac{\pi * ID_{TrunkCalc}^2}{4} \quad (39.14)$$

Within the set of available inner diameters for SDR 26 pipes, this $D_{TrunkCalc}$ value is rounded up to the nearest real size. This size is found in the specified pipe database, and kept as a ND. This functioanlly obtained ND value is compared to the maximum filter trunk ND ($ND_{TrunkMax}$). The lesser of the two values is chosen. The lesser value is chosen because selecting the maximum pressure recovery in a previous step can result in a filter flow rate slightly larger than the maximum for the max trunk diameter. The max trunk diameter is still used in this case, though it just barely violates the pressure recovery constraint

That process is repeated to find the size of the backwash trunk, the only difference is the flow rate and velocity used are those for the backwash trunk.

$$\frac{Q_{Fi}}{V_{BwManTrunkMaxPR}} = A_{BwTrunkCalc} = \frac{\pi * ID_{BwTrunkCalc}^2}{4} \quad (39.15)$$

Finding the pipe sizes lets the pressure recovery be determined for the trunk in forward filtration:

$$PR = \frac{V^2}{2g} \rightarrow \frac{\frac{2Q_{Layer}}{\pi * D_{Trunk}^2}}{2g} = PR_{ForwardTrunk} \quad (39.16)$$

and in backwash:

$$\frac{\frac{2Q_{Fi}}{\pi * D_{TrunkBw}^2}}{2g} = PR_{BwTrunk} \quad (39.17)$$

These values allow the necessary height of sand in each layer to be determined, as in the following section.

39.4 Sand Layer Thickness as Function of trunk diameters

To make construction easier, all sand depths get rounded up to the nearest centimeter.

The following are the functions which determine the heights of the all the sand layers.

$$H_{layer} = f(ND_{trunk}) = \quad (39.18)$$

This relates to the distance between the manifold branches and how the water will distribute among the layers.

<Insert picture of side view of filter>

39.5 Clean bed head loss

The headloss through a bed of sand is determined with the Kozeny Equation (**ref for this eventually**)

The head loss between the lowest layer is different than the other five layers because, it is slightly thicker as calculated just above.

39.6 Auxilliary box widths and plumbing

not today

39.7 Number of manifold branches

Constraints for number of manifold branches: 1. Even number, because branches on both sides of the trunks 2. max allowable flow through backwash branches 3. allowable **average** flow for pressure recovery term

First, the maximum pressure recovery in backwash branch:

$$PR_{BwManBranchMax} = HL_{BWSlotsEst} * \Pi_{ManifoldHeadLoss} - PR_{BwTrunk} \quad (39.19)$$

the resulting velocity from this pressure recovery:

$$V_{BwManBranchMax} = \sqrt{2g * PR_{BwManBranchMax}} \quad (39.20)$$

The maximum allowable flow through this branch is then:

$$Q_{BwBranchMaxPR} = V_{BwManBranchMax} * \pi * IR_{BranchManifold}^2 \quad (39.21)$$

The allowable average flow is also necessary, which is derived thusly:

$$\Pi_Q = \frac{Q_{min}}{Q_{max}} \quad (39.22)$$

Assume linear flow distribution between branches:

$$\begin{aligned} Q_{ave} &= \frac{Q_{max} + Q_{min}}{2} \\ Q_{ave} &= \frac{Q_{max} + \Pi_Q Q_{max}}{2} \\ Q_{ave} &= Q_{max} \left(\frac{1 + \Pi_Q}{2} \right) \\ \Pi_Q &= \sqrt{1 - \Pi_{HL}} \\ Q_{ave} &= Q_{max} \left(\frac{1 + \sqrt{1 - \Pi_{HL}}}{2} \right) \end{aligned} \quad (39.23)$$

Thus, the average allowable flow:

$$Q_{BwBranchAveMaxPR} = Q_{BwBranchMaxPR} \left(\frac{1 + \sqrt{1 - \frac{PR_{BwTrunk}}{HL_{BW SlotsEst}}}}{2} \right) \quad (39.24)$$

Then the number of manifold branches (rounded up to the nearest even number) because of backwash flow distribution:

$$N_{BwManBranchMin} = \frac{Q_{Fi}}{Q_{BwBranchAveMaxPR}} \quad (39.25)$$

To determine the minimum for the forward filtration flow distribution is a similar process with flows and PR adjusted for forward flow.

$$PR_{ManBranchMax} = HL_{CleanLayerMin} * \Pi_{ManifoldHeadLoss} - PR_{ForwardTrunk} \quad (39.26)$$

The resulting velocity from this pressure recovery:

$$V_{ManBranchMax} = \sqrt{2g * PR_{ManBranchMax}} \quad (39.27)$$

The maximum allowable flow through this branch is then:

$$Q_{ForwardBranchMaxPR} = V_{ManBranchMax} * \pi * IR_{BranchManifold}^2 \quad (39.28)$$

The average allowable flow:

$$Q_{ForwardBranchAveMaxPR} = Q_{ForwardBranchMaxPR} \left(\frac{1 + \sqrt{1 - \frac{PR_{ForwardTrunk}}{HL_{CleanLayer}}}}{2} \right) \quad (39.29)$$

Then the number of manifold branches (rounded up to the nearest even number) because of backwash flow distribution:

$$N_{ManBranchMin} = \frac{2Q_{Layer}}{Q_{ForwardBranchAveMaxPR}} \quad (39.30)$$

Then the design minimum number of branches is the maximum of the two minimums:

$$N_{ManBranchMin} \text{ and } N_{BwManBranchMin} : \quad (39.31)$$

In addition to the minimums based on the flow constraints, a maximum number of branches exists for geometry reasons, so that the filter box width matches the auxiliary box width (when there is more than one filter)

This maximum is determined by the following rounded down to a whole number:

$$N_{ManSideBranchMaxMult} = \frac{\frac{A_{Active}}{W_{ActiveMin}} - OD_{BwManBranch} + L_{ManFerncoCoupling} + S_{BranchToWall}}{B_{ManifoldBranch}} \quad (39.32)$$

Such that:

$N_{ManSideBranchMaxMult}$ = the maximum number of branches per side

A_{Active} = the active area of the filter (perpendicular to flow direction)

$W_{ActiveMin}$ = the width of the filter area (**length of trunk?**)

$OD_{BwManBranch}$ = the outer diameter of the branch piping

$L_{ManFerncoCoupling}$ = the length of the fernco coupling (**in length dir of branches?**)

$S_{BranchToWall}$ = the spacing from the end of the branch to the inner edge of the filter

$B_{ManifoldBranch}$ = the **length?** of the branches

(should have a sketch for dimensions!)

As there are branches on both sides of the trunk, this number is multiplied by 2 to get the total maximum number of branches.

$$N_{ManBranchMaxMult} = 2 * N_{ManSideBranchMaxMult} \quad (39.33)$$

Include only one filter????????? that design is in here????? the red text is here with a note from skyler or the error if the geo branches doesn't meet flow distribution constraint????

39.8 Filter box dimensions and manifold pipe lengths

Note: In this section and beyond, B is used to describe the center-to-center distance between features (such as slots or orifices) and W is the width of a feature (such as the width of a slot)

From the number of manifold branches the lengths of the manifold pipes as well as the various filter box dimensions can be determined.

The total length of the filter (which is in the direction of the trunk) is determined by adding the spaces the branches take up.

In this section several expert inputs become relevant. Expert input are parameters that are the same across all plants due to constraints during construction or set sizes of equipment. The other use of expert inputs is for values determined to be functional, but without the same research pressure to make them perfectly correct. An example of this is the spacing between manifold branches.

The total length of the filter is the sum of several things:

$$L_{Fi} = \left(\frac{N_{ManBranch}}{2} - 1 \right) B_{ManifoldBranch} + OD_{BwManBranch} + L_{ManFerncoCoupling} + S_{BranchToWall} \quad (39.34)$$

The active width of the filter is the active area divided by the length:

$$W_{Active} = \frac{A_{Active}}{L_{Fi}} \quad (39.35)$$

This width is the active filter width, which does not include the projected area of the trunk lines, though it is assumed that the space between the branch holders contributes to the active area.

The total width is:

$$W_{Fi} = W_{Active} + OD_{BWTrunk} \quad (39.36)$$

The width of the filter entrance is the sum of half the width of the filter, the outer radius of the cap of the backwash trunk, the minimum space between fittings in a tank or fittings and the wall of the tank, the thickness of the filter walls, and the thickness of the filter box walls. That is:

$$W_{Entrance} = \frac{W_{Fi}}{2} + OR_{Fitting} + S_{Fitting} + T_{FiWall} + T_{FiBoxWall} \quad (39.37)$$

The width of the filter overflow depends on the number of filters. If there is one filter, then $W_{FiOverflow} = W_{FiOverflowMin}$. In most cases when there are 2 or more filters, the width of the overflow box is:

$$W_{FiOverflow} = W_{Fi} + T_{FiWall} - W_{FiEntrance} - 2T_{FiBoxWall} \quad (39.38)$$

The length of the filter with regards to the manifold branches is the next calculable attribute:

$$L_{ManBranch} = \frac{W_{Fi}}{2} - OR_{Fitting} - OR_{BranchHolder} + 2L_{ManBranchExt} - S_{ManAssembly} \quad (39.39)$$

This length is the length of the filter per manifold branch. Thus the total length of the filter where the manifold branches are is this length multiplied by the number of branches:

$$L_{ManBranchTot} = L_{ManBranch}N_{ManBranch} \quad (39.40)$$

The length of the backwash manifold is calculated similarly.

$$\begin{aligned} L_{BwManBranch} &= \frac{W_{Fi}}{2} - OR_{BwTrunk} - OR_{FittingBWBranchHolder} - OR_{BWBranchHolder} + 2L_{ManBranchExt} - S_{ManAssembly} \\ L_{BwManBranchTot} &= L_{BwManBranch}N_{ManBranch} \end{aligned} \quad (39.41)$$

Next, the length of the trunks:

$$\begin{aligned} L_{Trunk} &= L_{Fi} - \frac{L_{ManFerncoCoupling}}{2} - T_{TrunkCap} - S_{ManAssembly} \\ L_{BWTrunk} &= L_{Fi} - \frac{L_{ManFerncoCoupling}}{2} - T_{BwTrunkCap} - S_{ManAssembly} \end{aligned} \quad (39.42)$$

And the lengths of the branch holders:

$$\begin{aligned} L_{BranchHolder} &= L_{Fi} - 2T_{BranchHolder} - 2S_{ManAssembly} \\ L_{BwBranchHolder} &= L_{Fi} - 2T_{BwBranchHolder} - 2S_{ManAssembly} \end{aligned} \quad (39.43)$$

39.9 Manifold slot/orifice design

Note Previously all manifold branches were slotted, and the middle two inlets and the three outlets had identical slotted pipes. The top and bottom inlets were different because they only received half the flow but we wanted the same slot head loss through all manifolds.

Now the design has changed to having inlet branches with orifices instead of slots to avoid clogging. Sand is kept out with downward-facing U-channels around the pipes that create gravity exclusion zones. The outlet manifold are still slotted because we don't yet have a design that would keep sand out of the orifices during normal filtration.

The basis of the design of the orifices is the head loss through the bottom manifold orifices during backwash in order to get good flow distribution along the manifold.

$$\begin{aligned}
 HL_{BwOrificeEst} &= \frac{PR_{BwManTotal}}{\Pi_{ManifoldHeadLoss}} \\
 A_{BwManOrificeEst} &= \frac{Q_{Fi}}{\Pi_{VC} \sqrt{2g(HL_{BwOrificeEst})}} \\
 A_{BwManOrificeEst} &= 2A_{BwManOrificeEst} \\
 A_{TopManOrificeEst} &= A_{BwManOrificeEst} \\
 A_{OutletManSlots} &= A_{InletManOrificesEst}
 \end{aligned} \tag{39.44}$$

Outlet Design:

Parameters:

$$\begin{aligned}
 B_{ManSlot} &= \frac{1}{8}in = 3.175mm \\
 N_{SlotRows} &= 2
 \end{aligned} \tag{39.45}$$

Lengths:

$$\begin{aligned}
 L_{OutletManSlotBranchTotal} &= \frac{A_{OutletManSlots}}{W_{ManSlots} N_{ManBranch}} \\
 L_{OutletBranchSlotted} &= L_{ManBranch} - 2L_{ManBranchExt} - B_{ManSlot}
 \end{aligned} \tag{39.46}$$

Number of outlet slots per branch, then rounded down to the nearest integer:

$$N_{OutletManSlotsPerRow} = \frac{L_{OutletBranchSlotted}}{B_{ManSlot}} \tag{39.47}$$

Then the number of the slots per branch:

$$N_{OutletManSlotsPerBranch} = N_{OutletManSlotsPerRow} N_{SlotRows} \tag{39.48}$$

Thus the length of the slot is:

$$L_{OutletManSlotCurvedInner} = \frac{L_{OutletManSlotBranchTotal}}{N_{OutletManSlotsPerBranch}} \tag{39.49}$$

This amount should be checked to ensure that it does not exceed what is possible given the 1/2 inner circumference of the branch.

Inlet Design:

Parameter: $B_{ManOrificeEst} = 1cm$.. is this a diameter of the orifice?

For the backwash branches the maximum number of orifices is (rounded down an integer):

$$N_{BwManBranchOrificeMax} = \frac{L_{BwManBranch} - 2L_{ManBranchExt} - B_{ManOrificeEst}}{B_{ManOrificeEst}} \tag{39.50}$$

The maximum number of orifices on the backwash manifold is:

$$N_{BwManOrificeMax} = N_{ManBranch} N_{BwManBranchOrificeMax} \tag{39.51}$$

On the other inlets an identical calculation is done :

$$N_{ManBranchOrificeMax} = \frac{L_{ManBranch} - 2L_{ManBranchExt} - B_{ManOrificeEst}}{B_{ManOrificeEst}} \quad (39.52)$$

$$N_{ManOrificeMax} = N_{ManBranch} N_{ManBranchOrificesMax}$$

Determining the diameter of the orifices is next. In this design only drill bit sizes of 1/4 and 1/8 are considered. The total area of the orifices is divided by the maximum number of orifices. From this a radius is calculated and double to get a diameter. This diameter is compared to the available drill bit options and rounded up to the closest size. This is done for the backwash manifold, inner inlet manifolds, and the top inlet manifold as shown below: .. why are these the only two drill bit sizes considered?

$$D_{BwManOrifice} = 2\sqrt{\frac{A_{BwManOrificeEst}}{\pi * N_{BwManBranchOrificesMax}}} \quad (39.53)$$

$$D_{InletManOrifice} = 2\sqrt{\frac{A_{InletManOrificeEst}}{\pi * N_{ManOrificesMax}}}$$

$$D_{TopManOrifice} = 2\sqrt{\frac{A_{TopManOrificeEst}}{\pi * N_{ManOrificesMax}}}$$

Thus the hole size is determined!

The actual number of inlet orifices can then be calculated. A calculation is done for each type of inlet, if that number is larger than the maximum allowed number of orifices then the maximum is used. Otherwise the calculated number is used.

The calculations for number of orifices based on orifice size is:

$$N_{BwManOrificesCalc} = \frac{A_{InletManOrificesEst}}{\frac{\pi}{4} * D_{BwManOrifice}^2} \quad (39.54)$$

$$N_{InletManOrificesCalc} = \frac{A_{InletManOrificesEst}}{\frac{\pi}{4} * D_{InletManOrifice}^2}$$

$$N_{TopManOrificesCalc} = \frac{A_{FiTopManOrificesEst}}{\frac{\pi}{4} * D_{TopManOrifice}^2}$$

Comparing these calculations to the output provides values for $N_{BwManOrifices}$, $N_{InletManOrifices}$, and $N_{TopManOrifices}$ which are then used to find the actual areas of the inlet orifices

$$A_{BwManOrifices} = \frac{\pi}{4} D_{BwManOrifice}^2 * N_{BwManOrifices} \quad (39.55)$$

$$A_{InletManOrifices} = \frac{\pi}{4} D_{InletManOrifice}^2 * N_{InletManOrifices}$$

$$A_{TopManOrifices} = \frac{\pi}{4} D_{TopManOrifice}^2 * N_{TopManOrifices}$$

Recalling the outlet area, a conservative headloss calculation can be made accounting for sand blockage. Currently, this is not included in the total path headloss; it is unclear if it applies.

$$HL_{OutletSlotForward} = \frac{\left(\frac{2Q_{FiLayer}}{A_{OutletManSlots} * \epsilon_{Sand} * \Pi_{VCOrifice}} \right)^2}{2g} \quad (39.56)$$

Additionally the headloss from each of the inlets can be found.

$$\begin{aligned}
 HL_{BwManOrifice} &= \frac{\left(\frac{Q_{Fi}}{A_{BwManOrifices} * \epsilon_{Sand} * \Pi_{VCOrifice}}\right)^2}{2g} \\
 HL_{BwManOrificeForward} &= \frac{\left(\frac{Q_{FiLayer}}{A_{BwManOrifices} * \epsilon_{Sand} * \Pi_{VCOrifice}}\right)^2}{2g} \\
 HL_{InletManOrificeForward} &= \frac{\left(\frac{2Q_{FiLayer}}{A_{InletManOrifices} * \epsilon_{Sand} * \Pi_{VCOrifice}}\right)^2}{2g} \\
 HL_{TopManOrificeForward} &= \frac{\left(\frac{Q_{FiLayer}}{A_{TopManOrifices} * \epsilon_{Sand} * \Pi_{VCOrifice}}\right)^2}{2g}
 \end{aligned} \tag{39.57}$$

39.10 Plumbing head losses

39.11 Total Sand depth and backwash head loss

Here, H is used as height of sand, though as the filter extends downward it may be more intuitive to think of this as a depth of sand.

The overall depth of sand is:

$$H_{Sand} = (N_{Layer} - 1)H_{Layer} + H_{BottomLayer} + OR_{BwManBranchFitting} + OR_{ManBranch} \tag{39.58}$$

During filtration not all of the sand is actively involved in the process, the amount of sand that is active is all the sand contained in the sand layers.

$$H_{SandActive} = H_{Sand} - OR_{BwManBranch} - OR_{ManBranch} \tag{39.59}$$

The height of the fluidized sand bed is the height of the settled sand bed times the expansion ratio:

$$H_{FluidizedBed} = \Pi_{Fluidized} H_{Sand} \tag{39.60}$$

Knowing the depth of the fluidized bed allows the steady state headloss to be calculated for backwash.

$$HL_{BwSandSS} = \frac{H_{Sand}(\rho_{Sand} - \rho_{H2O})(1 - \epsilon_{Sand})}{\rho_{H2O}} \tag{39.61}$$

The value of this HL is very close to the total depth of sand in the filter, though it should be noted that this value does neglect losses in the plumbing.

$$HL_{Terminal} = HL_{CleanLayer} + HL_{Dirty} \tag{39.62}$$

$$HL_{BwInitiation} = HL_{BwSF} + HL_{BwSandSS} \tag{39.63}$$

Note: What are these terms? have them explained! also explain how the terms below are gotten

How much does the filter bed over-fluidize when the entrance box level lowers after backwash initiation!

$$H_{FluidizedBedVariation} = \frac{(HL_{BwInitiation} - HL_{BwSandSS})(W_{FiEntrance}L_{FiEntranceEst} - 3\pi OR_{Trunk}^2 - \pi OR_{BwTrunk}^2)}{A_{FiActive}} \tag{39.64}$$

39.12 Path head loss calculations and flow distribution between layers

Now that the sand layer depth is set the Kozeny Head Loss can be determined for the clean bed and the headloss through various flow paths can be determined.

First use the Kozeny Equation ([ref kozeny](#)) to find the HL in the central layers and also the bottom layer with H_{Layer} and $H_{BottomLayer}$

As the filter has 6 layers there are six possible paths for the water to take.

Overview of the algorithm of this section:

In each path the path headloss is the sum of : inlet plumbing, sand layer, and outlet plumbing (for the relevant flow)

Find the max headloss through the respective paths (Q1 - Q6) Find the min headloss through the respective paths (Q1 - Q6) Find the average headloss of the paths (sum(HL)/6) ^a theoretical term because the flow distribution will sort itself out because that's how headloss works Find Π_{layer} . The ratio of the flow distribution. Goal is close to 1.

Because the flow distribution will change to make the headlosses even, it can be taken as true that $HL_{Path1} = HL_{Path2}$ and so on for each path. This assumes the clean bed headlosses and also accounts for the varying flows in each path.

It is also known that the flow must add up to the flow in the filter. Knowing that a system of equations can be set up to use the headloss and total flow requirement to solve for the flow in each layer. Those flows are then taken as the flow through each layer.

39.13 Siphon design

39.14 Inlet channel and elevations

39.15 Elevations

39.16 Inlet Weir Design

39.17 Entrance and overflow box lengths (X-direction)

39.18 Siphon Valve sizes

39.19 Sand Removal Pipe

39.20 Trunk Line purge valves

39.21 Main plant drain channel

39.22 Sand Volume

39.23 Backwash Lagoon

FILTRATION THEORY AND FUTURE WORK

40.1 Filtration model

The filtration model is based on the insight that rapid sand filters have an active filtration zone that slowly progresses down through the filter as it fills to maximum capacity at the upstream end. A clean bed of sand is ineffective at capturing small particles as evidenced by the poor initial performance after backwash. Thus it is apparently that previously deposited particles play a key role in subsequent capture of particles.

Interception! Flow constrictions - converging streamlines move particles closer to the pore wall. Previously deposited particles form constrictions. The constrictions are the most likely location for particles to collide with deposited particles. Thus particles do not fill the pores and clog the filter. Instead particle form constrictions where streamlines already converged.

Maximum shear sets a minimum constriction opening size! As particles gradually deposit in an ever shrinking flow constriction the velocity through the constriction increases and the velocity gradient at the wall increases. Eventually the bond strength of the coagulant nanoparticles is not great enough to capture suspended particles that collide with the deposited particles. The flow constriction reaches a minimum diameter and subsequent suspended flocs flow right through the constriction.

The gradual creation of more flow constrictions results in an almost linear increase in head loss as a function of the volume of deposited flocs.

Table 40.1: Hypotheses, evidence, and why it matters

#	Hypotheses	Evidence	Why it matters
1	Particle removal is primarily due to converging streamlines that move particles close to the sand surface or to previously deposited particles	Filtration theory and estimates of void volume occupied by particles	Basis for our filtration model
2	The particle size distribution entering the filter is set by the plate settlers	Estimate of velocity gradient in inlet waterfall and in filter bed	High velocity injection would be required to reduce the floc size
3	The jets that the constrictions create dissipate almost all of their energy in the downstream void BEFORE entering the next constriction	Laminar flow jets dissipate energy very quickly	Allows a simple relationship between pore head loss and constriction velocity
4	Flocs are captured with VERY high efficiency and thus primary particle removal limits filter performance	Clean bed filtration models	Need to optimize filters for primary particle removal
5	Constrictions form a continuous barrier across the filter at each sand grain layer	Any gaps would receive higher flow rate and thus higher flux of particles	Filter automatically prevents short circuiting
6	The thickness of the deposit scales with the width of the ring or perhaps simply with the pore size or sand size	Thickness cant scale with diameter of the flocs because that would predict more total mass retained with increased coagulant dose	Required to predict head loss vs mass deposited.

40.2 Filter Head loss model

The maximum velocity in a constriction is set by the drag on a primary particle that is large enough to prevent the primary particle from attaching. Given that the constriction is likely sharp edged on the upstream face the flow velocity is likely close to uniform at the entrance to the constriction. There is likely a vena contracta effect and that effect would prevent deposition of particles downstream of the leading edge of the constriction because particles are diverted away from the constriction walls.

The maximum constriction velocity is thus likely independent of the filtration velocity and pore size. The maximum velocity at the inlet to the constrictions is likely proportional to the fractional surface coverage of the primary particles by coagulant nanoparticles. Thus as the coagulant dose increases the head loss per pore and the head loss at the time of particle breakthrough increases.

$$h_{l_{constriction}} = \frac{\bar{v}_{constriction}^2}{2g} \quad (40.1)$$

The maximum drag that a primary particle can

- need to connect velocity to drag
- Use experimental data from filtration to find critical velocity
- expect velocity to be independent of pore size (sand size)
- Create model for total head loss in a filter as a function of sand size
- Need a connection between pore size and volume of particles it can hold.

The average distance between sand grains in a filter bed is obtained by taking the cube root of the total volume occupied by a sand grain including the pore space. We also assume that there is a one-to-one correspondence between pores and

sand grains and thus the distance between pores is the same as the distance between sand grains.

$$\Lambda_{sand} = \Lambda_{pore} = \left(\frac{\pi}{6(1-\phi)} \right)^{\frac{1}{3}} D_{sand} \quad (40.2)$$

The flow rate per pore is the approach velocity multiplied by Λ^2 .

$$Q_{pore} = v_a \Lambda_{pore}^2 \quad (40.3)$$

The velocity through a constriction in a pore is

$$Q_{pore} = v_a \Lambda_{pore}^2 = \frac{\pi}{4} D_{constriction}^2 v_{constriction} \quad (40.4)$$

The constriction diameter is thus given by

$$D_{constriction} = \Lambda_{pore} \sqrt{\frac{4v_a}{\pi v_{constriction}}} \quad (40.5)$$

The Reynolds number of the jet issuing from the constriction is obtained by using equation (40.5) to eliminate the dependence on diameter.

$$Re_{jet} = \frac{\Lambda_{pore}}{\nu} \sqrt{\frac{4v_a v_{constriction}}{\pi}} \quad (40.6)$$

The jet issuing from the constrictions is laminar for Reynolds numbers below 500 (see Ungate et al., 1975 page 13). The constriction is likely a relatively thin (washer-like) deposit and thus the flow through the constriction is likely close to uniform. In order to estimate the head loss in the flow expansion that occurs after the constriction we need to understand how much the flow expands. The flow expansion may be limited by the geometry of the pores or it might be limited by the rate at which laminar flow jets expand. WE dont have a way to know which constraint will set the expansion and thus we need to calculate the expansion rate for a laminar jet to see if that expansion could occur given the pore geometry.

The rate of jet expansion can be obtained from an analytical solution of the Navier Stokes equation as shown by Pai, S.I., Fluid dynamics of jets. (D. Van Nostrand Company, Inc., Princeton, NJ, 1954 (pages 78-79)

The momentum in the direction of flow is conserved and is given by

$$M_0 = 2\pi\rho \int_0^\infty u^2 r dr \quad (40.7)$$

For the case of uniform velocity through the constriction we can replace ∞ with $r_{constriction}$

$$M_0 = 2\pi\rho v_{constriction}^2 \int_0^{r_{constriction}} r dr \quad (40.8)$$

$$M_0 = \pi\rho v_{constriction}^2 r_{constriction}^2 = \rho Q_{pore} v_{constriction} \quad (40.9)$$

The velocity in the direction of the jet is given by (equation 4.39 in Pai, 1954)

$$u = \frac{3M_0}{8\pi\mu} \frac{1}{x} \frac{1}{\left[1 + \left(\frac{\zeta^2}{4}\right)\right]^2} \quad (40.10)$$

$$\zeta = \frac{1}{4\nu} \left(\frac{3M_0}{\pi\rho} \right)^{\frac{1}{2}} \frac{r}{x} \quad (40.11)$$

and thus ζ is zero at the centerline. We will use the centerline velocity at distance Λ_{pore} from the constriction to estimate the head loss caused by one constriction.

$$v_{exp} = \frac{3\rho Q_{pore} v_{constriction}}{8\pi\mu\Lambda_{pore}} \quad (40.12)$$

We can substitute for Q_{pore} to obtain

$$v_{exp} = \frac{3v_a\Lambda_{pore}}{8\pi\nu} v_{constriction} \quad (40.13)$$

We can use mass conservation and the equation for Q_{pore} to replace v_a with $v_{constriction}$

$$v_{exp} = \frac{3}{32} \frac{D_{constriction}}{\Lambda_{pore}} Re_{jet} v_{constriction} \quad (40.14)$$

It isn't yet clear if this always means that v_{exp} will be very small compared with $v_{constriction}$, but that seems to be the logical conclusion. Thus when calculating the head loss for a pore it is reasonable to assume that the kinetic energy of the jet is all lost before entering the next constriction.

```

import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.research.floc_model as fm

import numpy as np

D_sand = 0.5 * u.mm
Porosity = 0.4
Temperature = 20 * u.degC
v_a = 1.85 * u.mm/u.s
he_filter = 40 * u.cm
H_filter = 20 * u.cm
Lambda_pore = (np.pi/(6*(1-porosity)))**(1/3)*D_sand
Lambda_pore

# Use measured fully constricted filter head loss to estimate the constricted velocity
v_constriction = (np.sqrt(2*pc.gravity*Lambda_pore*he_filter/H_filter)).to(u.mm/u.s)
print('The velocity in the constriction is estimated (for these conditions) to be',v_
      ↴constriction,'.')

Re_constriction =(Lambda_pore/pc.viscosity_kinematic(Temperature) * np.sqrt(4*v_a*v_
      ↴constriction/np.pi)).to(u.dimensionless)
print('The jet is laminar because the Reynolds number is ',Re_constriction.magnitude,
      ↴' which is well within the laminar range because it is less than 500.')
D_constriction = Lambda_pore * np.sqrt(4*v_a/v_constriction/np.pi)

#velocity of the jet after it travels one pore length
v_exp = (3*v_a*Lambda_pore*v_constriction/(8*np.pi*pc.viscosity_
      ↴kinematic(Temperature))).to(u.mm/u.s)
print('The velocity of the jet after it travels one pore length is ',v_exp,' which_
      ↴contains an insignificant amount of kinetic energy compared with the original jet.')

#Now find the Camp Stein velocity gradient in the pore
G_CS = (v_constriction * np.sqrt(v_a/(2*pc.viscosity_kinematic(Temperature)*porosity_
      ↴* Lambda_pore))).to(u.Hz)
print('The constricted jet induced Camp Stein velocity gradient in the pores is ',G_
      ↴CS,'.')

#Calculate the Camp Stein velocity gradient at the StaRS injection site
W_inject = 2.5 *u.cm
S_inject = 10 *u.cm
n_layers_per_inject = 2
v_a_inject = v_a * S_inject*n_layers_per_inject/W_inject
print('The StaRS injection velocity is ',v_a_inject,'.')
#The clean bed camp velocity gradient given this injection approach velocity
def Re_Erdon(v_a, D_sand, Temperature, Porosity):
    return (v_a*D_sand/(pc.viscosity_kinematic(Temperature)*(1-Porosity))).to(u.
      ↴dimensionless)

```

(continues on next page)

(continued from previous page)

```

def f_Erdon(v_a, D_sand, Temperature, Porosity):
    return 300/Re_Erdon(v_a, D_sand, Temperature, Porosity) + 3.5

def hf_Erdon(v_a, D_sand, Temperature, Porosity, L):
    return (f_Erdon(v_a, D_sand, Temperature, Porosity)*L/D_sand*v_a**2/(2*pc.
    ↪gravity)*(1-Porosity)/Porosity**3).to(u.m)

def G_CS_Ergun(v_a, D_sand, Temperature, Porosity):
    return np.sqrt(f_Erdon(v_a, D_sand, Temperature, Porosity) * v_a**3 * (1-Porosity)/
    ↪(2 * pc.viscosity_kinematic(Temperature) * D_sand * Porosity**4)).to(u.Hz)

G_CS_inject = G_CS_Ergun(v_a_inject, D_sand, Temperature, Porosity)
print('The velocity gradient at the StaRS injection sites (in the two middle inlets) is ', G_CS_inject)

```

40.3 Sedimentation plates set the size of the flocs in the filter

The characteristic floc size could be set by the plate settlers that preferentially remove large flocs or it could be set by any of the high shear events on the path to or through the filter. We will estimate floc size from each of the potential shear and terminal velocity events. The calculations are done in *python below*.

40.3.1 Sedimentation tank floc diameter

The floc diameter that is not completely captured by plate settlers with a capture velocity of 0.12 mm/s (the capture velocity used by AquaClara) is $25 \mu\text{m}$.

40.3.2 Constricted pore velocity gradient

Almost all of the kinetic energy of the jet issuing from the constriction is dissipated in the downstream pore. We will assume that the majority of the head loss is due to the jet (rather than wall shear). The volume of a pore is given by

$$V_{pore} = \phi \Lambda_{pore}^3 \quad (40.15)$$

The hydraulic residence time in a pore is obtained from the volume and flow rate through a pore.

$$\theta_{pore} = \frac{V_{pore}}{Q_{pore}} = \frac{\phi \Lambda_{pore}^3}{v_a \Lambda_{pore}^2} = \frac{\phi \Lambda_{pore}}{v_a} \quad (40.16)$$

The Camp Stein velocity gradient in a pore is obtained by assuming that all of the input kinetic energy is dissipated through viscosity during the time that the water spends in the pore.

$$G_{CS} = \sqrt{\frac{gh_L}{\nu \theta}} \quad (40.17)$$

The head loss in the pore is from the jet kinetic energy. Substitute the equations for pore volume, residence time, and head loss to obtain the fundamental equation for the velocity gradient in a pore.

$$G_{CS} = \bar{v}_{constriction} \sqrt{\frac{v_a}{2\nu \phi \Lambda_{pore}}} \quad (40.18)$$

The estimated value of this velocity gradient is 300 Hz .

40.3.3 Influent sand injection velocity gradient

We need to compare the constricted jet velocity gradient with the clean bed injection velocity gradient that occurs due to the fact that the approach velocity at the point of flow injection in StaRS filters is much higher than the average approach velocity. The flow injection area is approximately 2.5 cm wide and serves an effective filter area (up and down!) of 20 cm (spacing between injection points is 10 cm.). Thus the approach velocity in current StaRS filters is 14.6 mm/s and the corresponding Camp Stein velocity gradient is 1440 Hz.

40.3.4 Filter inlet box water fall velocity Gradient

EstaRS filters have water falls from the filter inlet channel to inlet box. In recent designs the water flows through a narrow slot and then falls into the inlet box. The slot is about 3.4 cm wide and the water falls about 40 cm. The vertical velocity after dropping 40 cm is 2.8 m/s. The geometry of this water fall is complicated and We will assume the resulting jet is circular at impact with the water surface. The diameter of the jet is thus obtained from continuity and is equal to 9.5 cm. Now we use the energy dissipation rate of a round jet to obtain 4300 Hz. Thus the floc size could be set by the water fall in the inlet box since this is the highest velocity gradient on the path into and through the filter. However, our rough estimate based on wild extrapolation suggests that this would yield a 36 μm diameter floc. This is larger than the flocs produced by the sedimentation tank plate settlers and thus it is possible that even with this waterfall, that the floc size distribution is set by the sedimentation tank plate settlers.

Note that the water fall height diminishes during a filter run as the water level in the inlet box increases due to head loss through the filter. Thus it is likely that the waterfall doesn't significantly break flocs.

40.4 Maximum velocity in constrictions

The maximum velocity in a pore is hypothesized to be set by the bond strength of the coagulant nanoparticles and the fluid drag on the primary particle that is attaching. It is assumed that the last particles that are able to deposit in a pore are primary particles because they can fill in the last available volume before the pore velocity is too high for any other particles to attach. It is possible that the attachment strength of the primary particles is a function of the fraction of their surface area that is covered by coagulant nanoparticles, Γ . The total force acting downward on a primary particle that attaches to a constriction is the sum of the drag and the particle buoyant weight. These forces are counteracted by the force of the coagulant bonds.

$$F_{coag_{bonds}} = F_{drag} + F_{weight} - F_{buoyancy} \quad (40.19)$$

The drag force on a clay particle that has attached to the

$$F_{drag} = C_D \frac{\pi}{4} D_{clay}^2 \rho_{H_2O} \frac{v_{constriction}^2}{2} \quad (40.20)$$

At Reynolds numbers (based on primary particle diameter) less than about 10 the drag coefficient is given by

$$C_D = \frac{24}{Re} = \frac{24\nu}{v_{constriction} D_{clay}} \quad (40.21)$$

Thus the drag on a clay particle is given by

$$F_{drag} = 3\pi\nu v_{constriction} D_{clay} \rho_{H_2O} \quad (40.22)$$

$$F_{coag_{bonds}} = 3\pi\nu v_{constriction} D_{clay} \rho_{H_2O} + (\rho_{clay} - \rho_{H_2O}) g \frac{\pi}{6} D_{clay}^3 \quad (40.23)$$

The drag force is assumed to be set by the average pore water velocity because the deposition occurs near the entrance to the constriction before the boundary layer on the wall can develop.

The force of the coagulant bonds is presumed to be proportional to the fractional coverage of the clay with coagulant and the intrinsic shear strength of the coagulant bonds to the clay surface

$$F_{coagbonds} = \Gamma \tau_{bonds} \frac{\pi}{4} D_{clay}^2 \quad (40.24)$$

where τ_{bonds} is the intrinsic shear strength of the coagulant bonds to the clay surface. Solving for the maximum constriction velocity we obtain

$$v_{constriction_{max}} = \frac{\Gamma \tau_{bonds} \frac{\pi}{4} D_{clay}^2 - (\rho_{clay} - \rho_{H_2O}) g \frac{\pi}{6} D_{clay}^3}{3\pi\nu D_{clay} \rho_{H_2O}} \quad (40.25)$$

It is likely that the weight of the clay particle is a small contribution to the force balance. In that case the equation simplifies to

$$v_{constriction_{max}} = \frac{\Gamma \tau_{bonds} D_{clay}}{12\nu \rho_{H_2O}} \quad (40.26)$$

Unfortunately, we do not have a measure of the intrinsic bond strength of the coagulant nanoparticles to clay surfaces, τ_{bonds} . This equation does provide a possible means to back calculate this property.

The minimum diameter of a particle deposition constriction is set by the maximum constriction velocity, $v_{constriction_{max}}$.

$$D_{constriction_{min}} = \Lambda_{pore} \sqrt{\frac{4v_a}{\pi v_{constriction_{max}}}} \quad (40.27)$$

The head loss through a flow constriction can be estimated from the head loss through a flow expansion. We will use the form of the expansion equation (7.20) that is based on the contraction velocity. The jet is assumed to expand sufficiently so that the residual kinetic energy is insignificant.

$$h_{e_{constriction}} = \frac{\bar{v}_{constriction_{max}}^2}{2g} \quad (40.28)$$

The number of deposited constrictions per unit depth in a filter is

$$N_{constrictions_{series}} = \frac{H_{filter}}{\Lambda_{pore}} \quad (40.29)$$

The total head loss in a filter if taken to the point where the active filtration zone exited the filter and all pores were constricted would be

$$h_{e_{filtermax}} = \frac{H_{filter}}{\Lambda_{pore}} \frac{\bar{v}_{constriction_{max}}^2}{2g} \quad (40.30)$$

The effect of increasing the pore size on terminal head loss is to decrease the *final* head loss when the active zone reaches the bottom of the filter because of the effect of Λ_{pore} ‘in the first term of equation : eq : ‘eq_{hefilter}. Note that this does not yet address the rate of head loss accumulation which is expected to be a function of sand grain diameter.

We can solve equation (40.30) for maximum constriction velocity based on experimental measurements of the head loss at filter failure that is due to constrictions. Note that this head loss does NOT include the clean bed head loss.

$$v_{constriction_{max}} = \sqrt{\frac{2g\Lambda_{pore}}{H_{filter}} h_{e_{filtermax}}} \quad (40.31)$$

From Fig. 37.7 we have an estimate of 35 to 80 cm of head loss through a 20 cm bed of 0.5 mm diameter sand. This gives an estimate of 137 mm/s for the constriction velocity. This could be used to obtain an estimate of the bond strength of the coagulant nanoparticles.

40.5 Shear big flocs to improve filter performance

Here we explore the possibility of breaking flocs as they enter the filter bed to eliminate large flocs that may be reducing filter performance.

Primary particles have the lowest probability of hitting the wall in a constriction. Thus primary particles can travel the greatest distance through the active zone and still have a very small chance of being deposited near the end of the active zone. Thus it is possible that primary particles set the maximum length of the active zone and flocs tend to fill in the active zone at the upstream end. The larger the floc the more likely it will fill in an upstream constriction and thus shorten the active zone.

This suggests that one way to improve filter performance is to have a zone of very high shear that rips flocs apart so that they don't fill in the upstream pores in the active zone so quickly. This is because smaller flocs will not be removed as efficiently by each constriction and thus they will penetrate deeper into the active zone. One possible method to create a high shear zone is to size the flow injection area to achieve high shear through the first sand grains. The idea is to shred incoming flocs so they have a lower probability of being removed per pore and thus more of these small flocs penetrate deeper into the active filtration zone before being captured. Smaller flocs are also more dense and thus don't fill up the available volume in the constrictions as fast as the large flocs that they came from.

We need an estimate of the shear through the first pores as the water enters the sand. The Kozeny equation is valid up to a particle Reynolds number of 1 ((16.104)). The Reynolds number at this proposed flow injection site will be much larger than 1 and thus the Erdun equation ((16.105)) that is valid for laminar and turbulent flow in porous media will be used.

We will use the Camp Stein velocity gradient to estimate injection velocity required to create very small flocs. The important parameter for floc break up is a force that can be obtained from the velocity gradient multiplied by the dynamic viscosity.

Solving (16.109) for the approach velocity, v_a , we obtain

$$v_a = \left(G_{CS}^2 \frac{2\nu D_{sand}}{f_\phi} \frac{\phi^4}{(1-\phi)} \right)^{\frac{1}{3}} \quad (40.32)$$

to estimate the injection area that should be used to break up flocs entering the sand bed.

40.6 Floc size and velocity gradient calculations

```
import aquaclara.core.physchem as pc
from aquaclara.core.units import unit_registry as u
import aquaclara.research.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt

N_layers = 6
v_filter_backwash = 11 * u.mm/u.s
v_a = v_filter_backwash/N_layers
Porosity = 0.4
D_sand = 0.5 * u.mm
# the following is just a guess at pore size
D_pore = 0.2 * D_sand
Temperature = 20 * u.degC
L = 20 * u.cm
Q_waterfall = 20 * u.L/u.s
v_waterfall=(np.sqrt(pc.gravity*2*40*u.cm)).to(u.m/u.s)
```

(continues on next page)

(continued from previous page)

```

v_waterfall
#use continuity to get diameter of waterfall
D_waterfall = (np.sqrt(Q_waterfall/v_waterfall *4/np.pi)).to(u.cm)
D_waterfall
Pi_JetRound = 0.08
G_Max_waterfall = (v_waterfall*np.sqrt(Pi_JetRound*v_waterfall/(pc.viscosity_
↪kinematic(Temperature)*D_waterfall))).to(u.Hz)
print('The maximum velocity gradient in the plunging jet in the filter inlet box is ',_
↪G_Max_waterfall,'.')
# now estimate the size of this floc
EDR_waterfall = (G_Max_waterfall**2*pc.viscosity_kinematic(Temperature)).to(u.mW/u.kg)
EDR_waterfall
D_floc_waterfall = (fm.diam_floc_max(G_Max_waterfall**2*pc.viscosity_
↪kinematic(Temperature))).to(u.um)
print('The diameter of flocs after the waterfall is estimated to be',D_floc_waterfall,
↪'.')

# Calculate maximum diameter of flocs leaving the sedimentation tank
D_floc= fm.diam_floc_vel_term(0*u.mg/u.L,10*u.mg/u.L,fm.PACl,fm.Clay,fm.DIM_FRACTAL,_
↪0.12*u.mm/u.s,20*u.degC)
print('The maximum diameter of flocs leaving the sedimentation tank is',D_floc.to(u.
↪um),'.')
v_graph = np.linspace(0.1, 100, 500) * u.mm/u.s

fig, ax = plt.subplots()
ax.plot(v_graph, hf_Erdon(v_graph, D_sand, Temperature, Porosity, L)/L).to(u.
↪dimensionless),'-')
ax.plot(v_graph, (pc.headloss_kozeny(L,D_sand, v_graph, Porosity, pc.viscosity_
↪kinematic(Temperature) )/L).to(u.dimensionless),'-')
ax.set(xlabel='approach velocity (mm/s)')
ax.set(ylabel='head loss per m')
ax.legend(['Ergun', 'Kozeny'])
fig.savefig('Filtration/Images/Head_loss_Ergun_and_Kozeny')
plt.show()

```

```

# Need to use a root finding method here because f_phi is a function of velocity
# Will use a graphical solution for now

fig, ax = plt.subplots()
ax.plot(v_graph,G_CS_Ergun(v_graph, D_sand, Temperature, Porosity),'-')
ax.set(xlabel='approach velocity (mm/s)')
ax.set(ylabel='velocity gradient (Hz)')
fig.savefig('Filtration/Images/G_vs_approach_velocity')
plt.show()

```

```

#We guess at a velocity gradient by extrapolating wildly to a 20 um floc.
G_CS =np.sqrt((fm.ener_dis_diam_floc(40*u.um))/pc.viscosity_kinematic(Temperature)) .
↪to(u.Hz)
print('A wild guess at the velocity gradient required to break up flocs is ',G_CS,'.')
#from the graph above we'd need an approach velocity of about 80 mm/s to achieve a G_
↪of 10,000 Hz.
v_inject = 80*u.mm/u.s
(v_inject/v_a).to(u.dimensionless)
injection_port_spacing = 10 * u.cm

```

(continues on next page)

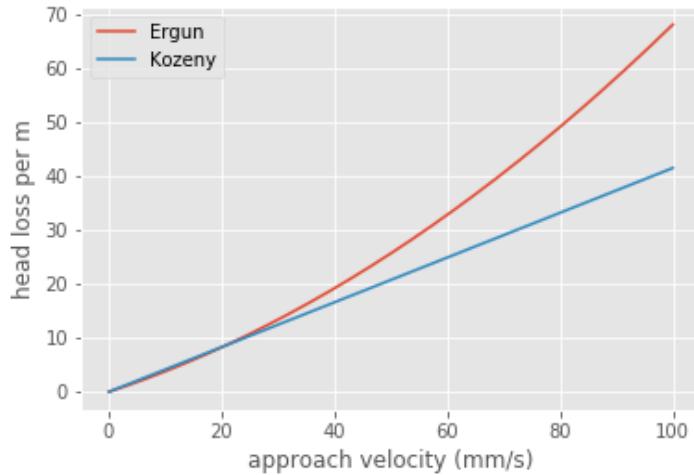


Fig. 40.1: The Ergun and Kozeny equations are very similar even at approach velocities that are much larger than are used in rapid sand filtration. At very high velocities the turbulent term in the Ergun equation begins to be significant.

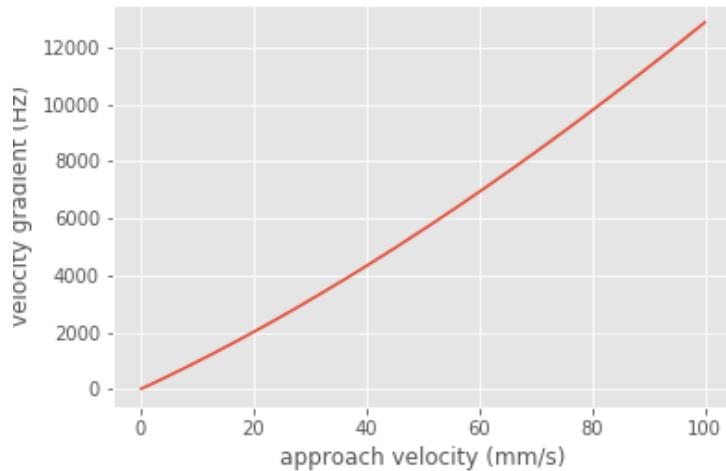


Fig. 40.2: The Camp Stein velocity gradient increases rapidly with approach velocity.

(continued from previous page)

```
injection_port_width = (injection_port_spacing/(v_inject/v_a)).to(u.mm)
print('The injection port width would be ',injection_port_width,'.')
print('The injection velocity would be ',v_inject.to(u.mm/u.s),'.')
```

The analysis above suggests that the approach velocity required to break floes down to a dimension of $20\mu\text{m}$ is approximately 80 mm/s. This is based on a VERY bad guesstimate of the relationship between floc size and shear.

We need to know how much energy would be expended to force the water through this high velocity injection zone. Once the water enters the sand it will spread radially in all directions. As the water spreads it will slow down and the head loss per distance traveled will decrease. We need to integrate this head loss over the first few centimeters to get an estimate of the injection head loss.

The velocity at distance r from the center of the injection line can be calculated from the velocity at r_0 by continuity.

$$v_{a_r}r = v_{a_{r_0}}r_0 \quad (40.33)$$

$$v_{a_r} = v_{a_{r_0}} \frac{r_0}{r} \quad (40.34)$$

$$dh_f = \frac{dr}{2gD_{sand}} \left(300 \frac{\nu(1-\phi)^2}{D_{sand}\phi^3} v_a + 3.5 \frac{(1-\phi)}{\phi^3} v_a^2 \right) \quad (40.35)$$

Now substitute for the approach velocity

$$\frac{dh_f}{dr} = \frac{1}{2gD_{sand}} \left[300 \frac{\nu(1-\phi)^2}{D_{sand}\phi^3} \left(v_{a_{r_0}} \frac{r_0}{r} \right) + 3.5 \frac{(1-\phi)}{\phi^3} \left(v_{a_{r_0}} \frac{r_0}{r} \right)^2 \right] \quad (40.36)$$

$$dh_f = \frac{v_{a_{r_0}}r_0}{2gD_{sand}} \left[300 \frac{\nu(1-\phi)^2}{D_{sand}\phi^3} \left(\frac{1}{r} \right) + 3.5 \frac{(1-\phi)v_{a_{r_0}}r_0}{\phi^3} \left(\frac{1}{r} \right)^2 \right] dr \quad (40.37)$$

We will create terms to make the integration easier

$$a_0 = \frac{v_{a_{r_0}}r_0}{2gD_{sand}} \quad (40.38)$$

$$a_1 = 300 \frac{\nu(1-\phi)^2}{D_{sand}\phi^3} \quad (40.39)$$

$$a_2 = 3.5 \frac{(1-\phi)v_{a_{r_0}}r_0}{\phi^3} \quad (40.40)$$

Now we set up the numerical integration and integrate from the injection site to the radius where the velocity is equal to the filtration velocity.

```
#This is for an inlet that serves 2 layers (up and down)
v_filter = (11 * u.mm/u.s)/N_layers*2
S_branch = 10 * u.cm
v0 = 80 * u.mm/u.s
r0 = (2*v_filter/v0 * S_branch/(2*np.pi)).to(u.mm)
r0

#create coefficients for the constant terms in the equation
a0 = v0 * r0 / (2*pc.gravity*D_sand)
a1 = (300*pc.viscosity_kinematic(Temperature)*(1-Porosity)**2) / (D_sand*Porosity**3)
a2 = 3.5*(1-Porosity)* v0 * r0/Porosity**3
r1 = S_branch/(2*np.pi)
r1
#create an array of r values with each value centered in the ring that it represents.
#We will use simple
```

(continues on next page)

(continued from previous page)

```

n_points = 10000
dr = ((r1-r0)/(n_points)).to(u.mm)
r_array = np.linspace((r0).to(u.mm), (r1).to(u.mm), n_points)*u.mm
y_array = (a0*(np.divide(a1,r_array) + np.divide(a2,np.multiply(r_array,r_array)))).
    ↪to(u.dimensionless)
#Use the trapezoidal rule to integrate
#need to reattach units to np.trapz.
hf_inlet = np.trapz(y_array,x=r_array, dx = dr)*u.mm
print('The head loss through the sand between the injection point and where it ↪reaches the filtration velocity is ', hf_inlet)

```

The analysis above suggests that a high velocity and high velocity gradient injection into the sand bed with the goal of breaking flocs into pieces that are $20 \mu\text{m}$ in diameter would require about 12 cm of head loss. This is based on the assumption that the water would be able to flow radially from the injection point and thus rapidly slow down. Thus the head loss rapidly decreases with distance from the injection point.

This is an experiment worth trying. It will help us understand if large flocs result in poorer filter performance.

40.7 Volume of flocs per pore

The volume of solids deposited in one pore can be obtained based on the average diameter of clean pore constrictions, the diameter of the constricted pore after solids deposition, and the thickness of the deposit. We already have an estimate for the diameter of the constricted pore after solids deposition. The thickness of the deposit must be proportional to some other length scale. We initially hypothesized that the thickness of the deposit scaled with the diameter of the flocs that make it up. That led to the conclusion that increased coagulant dose would increase the total mass of solids that could be retained by the filter before breakthrough. That doesn't match experimental data and thus we now propose that the average thickness of the deposit scales with the sand grain size, or pore size, or pore separation distance. We will use the pore separation distance as our scaling parameter.

There are several options for estimating the areal extent of the constriction. We already have an estimate of the inner diameter of the constriction and thus all we need is an estimate of the outer extent of the deposited constriction. One option would be to take the area of a circle defined by 3 spheres coming close together. That is a clear underestimate because the constriction must extend into the gaps between the spheres. A second option would be to use the porosity to estimate the average plane view area of the pores. If we assume that the pore must connect vertically and thus has a height Λ_{pore} , then the plane view area is given by

$$A_{pore} = \phi \Lambda_{pore}^2 \quad (40.41)$$

The area of the deposit is obtained by subtracting the constriction opening from the previous equation.

$$A_{deposit} = \Lambda_{pore}^2 \left(\phi - \frac{v_a}{v_{constriction}} \right) \quad (40.42)$$

The volume of the deposit is thus

$$V_{deposit} = \Pi_{pore}^{deposit} \Lambda_{pore}^3 \left(\phi - \frac{v_a}{v_{constriction}} \right) \quad (40.43)$$

where $\Pi_{pore}^{deposit}$ a number much less than 1 that represents the fixed ratio between the thickness of the deposit and the pore separation distance.

The head loss per volume of particles deposited can be obtained by dividing the head loss per pore by the volume of particles per pore.

$$h_{l_{per deposit}} = \frac{\bar{v}_{constriction}^2}{2g\Pi_{pore}^{deposit} \Lambda_{pore}^3 \left(\phi - \frac{v_a}{v_{constriction}} \right)} \quad (40.44)$$

If the primary goal for filter design were to decrease head loss per volume of solids deposited, then selecting larger sand (increasing Λ_{pore}) would be the clear strategy. Increasing the sand diameter by a factor of two should decrease the head loss by a factor of 8. Increasing the approach velocity v_a , results in a small increase in the head loss per volume of deposited material.

We need a method to connect turbidity removed by a filter into volume of deposited flocs. We will make this connection by first assuming that the flocs have a characteristic size based on a high shear event on their way into the filter. We begin with the relationship between the number of clay particles in a floc and the floc diameter.

$$D_{floc} = D_{clay} n_{clay}^{\frac{1}{\Pi_{Fractal}}} \quad (40.45)$$

Where $\Pi_{Fractal}$ is the volume based fractal dimension of a floc. We estimate $\Pi_{Fractal}$ to have a value of 2.3. We can rearrange this equation and solve for the number of clay particles in a floc.

$$n_{clay} = \left(\frac{D_{floc}}{D_{clay}} \right)^{\Pi_{Fractal}} \quad (40.46)$$

Now we can create a relationship for the concentration of clay in a floc dividing the mass of clay by the volume of the floc.

$$C_{clay_{floc}} = \frac{n_{clay} V_{clay} \rho_{clay}}{V_{floc}} = \frac{n_{clay} D_{clay}^3 \rho_{clay}}{D_{floc}^3} = \frac{D_{clay}^3 \rho_{clay}}{D_{floc}^3} \left(\frac{D_{floc}}{D_{clay}} \right)^{\Pi_{Fractal}} \quad (40.47)$$

With one more simplification we obtain the desired equation for the clay concentration in a floc of given diameter.

$$C_{clay_{floc}} = \rho_{clay} \left(\frac{D_{clay}}{D_{floc}} \right)^{3 - \Pi_{Fractal}} \quad (40.48)$$

The mass of clay per pore is obtained by multiplying the deposit volume by the concentration of the flocs.

$$M_{clay_{pore}} = \rho_{clay} \Pi_{pore}^{deposit} \Lambda_{pore}^3 \left(\phi - \frac{v_a}{v_{constriction}} \right) \left(\frac{D_{clay}}{D_{floc}} \right)^{3 - \Pi_{Fractal}} \quad (40.49)$$

The mass of clay per plan view area of the filter is obtained by multiplying by the number of pores per depth of the filter and dividing by the plan view area of a pore, Λ_{pore}^2 .

$$M_{clay_{filter}} = H_{filter} \rho_{clay} \Pi_{pore}^{deposit} \left(\phi - \frac{v_a}{v_{constriction}} \right) \left(\frac{D_{clay}}{D_{floc}} \right)^{3 - \Pi_{Fractal}} \quad (40.50)$$

According to this model, the mass of clay that can be held by a filter increases linearly with filter depth. The retained mass is independent of the sand size. This is an easy hypothesis to test. Note, however, that this model does not account for the depth of the active zone. Presumably the active zone depth may be greater for larger diameter media and thus breakthrough may occur sooner for larger diameter media.

If coagulant dose increases it will have two effects. The primary particle attachment strength will increase, the constricted velocity will increase, and the mass retained will increase. The size of the flocs will also increase and that will result in a slight decrease in the retained mass.

Thus it isn't immediately clear how changing the coagulant dose will change the maximum mass of retained particles. The evidence from the AguClara filter theory team is that the mass of clay retained decreases as the coagulant dose increases.

The head loss per mass of particles deposited can be obtained by dividing the head loss per pore by the mass per pore.

$$h_{e_{permassclay}} = \frac{\bar{v}_{constriction_{max}}^2}{2g\rho_{clay} \Pi_{pore}^{deposit} \Lambda_{pore}^3 \left(\phi - \frac{v_a}{v_{constriction}} \right)} \left(\frac{D_{floc}}{D_{clay}} \right)^{3 - \Pi_{Fractal}} \quad (40.51)$$

The head loss per mass of solids removed is significantly lower for larger sand sizes. When the coagulant dose increases the head loss increases rapidly because the constricted velocity increases and the floc diameter increases. Unfortunately we do not yet have a model describing floc size as a function of both velocity gradient and coagulant nanoparticle coverage.

40.8 Particle removal efficiency

Particle removal is complicated. We hypothesize that flocs form the deposits that change the flow from being wall shear dominated with a parabolic velocity profile to being uniform velocity flow through the constrictions. This uniform velocity profile transports a very small fraction of clay particles close enough to the deposit to be captured.

During filter ripening the particles that pass through the filter would be expected to be the primary particles because removal efficiency increases very rapidly with size. During the main part of the filter run the escaping particles are primary particles that weren't captured by the actively growing deposits. Breakthrough at the end of the filter run is caused by both primary particles and flocs.

The flocs form a series of actively growing deposits. The number of actively growing deposits in series is possibly a function of the average volume of the flocs (smaller flocs result in more active deposits) and the volume fraction of the flocs normalized by the volume fraction of the primary particles. If this dimensionless volume fraction increases there may be more active deposits and hence improved removal of primary particles.

40.9 Proposed experiments and inventions

1. Compare different sizes of sand media. Expect to get poorer removal efficiency with larger sand sizes, similar mass of particles retained at breakthrough, and much lower head loss.
2. dual media. expect to find less head loss and poorer performance than single small media. And expect the smaller media to not contribute anything.
3. How small could the sand media be? We could get better filter performance if we used smaller sand. Shallow sand beds are apparently fine and if we used smaller diameter sand the filter layer depth could be reduced even more. Why not use a 0.2 mm sand and 5 cm sand layers? If we offset the inlet and outlet branches (with branches spaced on 10 cm centers and inlet and outlet branches offset by 5 cm) there would still be a significant path length.
4. Floc amendment. We could add floc hopper particles to the filter to increase the ratio of flocs to primary particles. Presumably this would reduce effluent turbidity IF there aren't many primary particles in the floc hopper. We could compare the prospects of using smaller sand grains vs adding floc amendment as strategies to get higher performing filters.
5. Now that we know that sand doesn't remove clay without the help of flocs, could we invent a filter that could capture clay and other primary particles without requiring ripening? We need a filter media that has sharp edges or sudden constrictions that create high velocity near the edge. Washers with holes the size of constrictions aren't available.

FILTRATION DESIGN SOLUTION

```
import aguaclara.core.physchem as pc
from aguaclara.core.units import unit_registry as u
import aguaclara.research.floc_model as fm

import numpy as np
import matplotlib.pyplot as plt
```

41.1 DC Stacked Rapid Sand Filtration

The stacked rapid sand filter at Tamara, Honduras is treating 12 L/s of water with six 20 cm deep layers of sand. The sand has an effective size of 0.5 mm and a uniformity coefficient of 1.6. The backwash velocity of the filter is 11 mm/s. Defined below are many of the necessary inputs for the filtration analysis.

```
flow_plant = 12 * u.L / u.s
density_sand = 2650 * u.kg / u.m**3
k_Kozeny = 5
H_filter_layer = 20 * u.cm
N_filter_layer = 6

#We will use 20 deg C as the typical operating temperature. Honduras is quite warm.
T_design = 20*u.degC

# Notes on Uniformity Coefficient and Effective size from http://www.clean-water-for-
#laymen.com/sand-sieve.html

#Effective Size (ES) - This is defined as the size of screen opening that will pass 10
#% of the sand sample (see more explanation below)
#Uniformity Coefficient (UC) - This is defined as a ratio and is calculated as the_
#size opening that will just pass 60% of the sand (d60 value) divided by the size_
#opening that will just pass 10% of the sand sample (d10 value).
#ES is basically a value describing the average size of sand grains in a sand sample.
#UC is basically a value describing the range of grain sizes present in a sample

D_filter_sand_ES = 0.5 * u.mm
UC_filter_sand = 1.6
porosity_filter_sand = 0.4
V_filter_backwash = 11 * u.mm / u.s
```

41.1.1 Remember: dont break continuity!

Ensure that you use the variables defined above in your code, do not hard code any numbers if you do not have to.

1)

Calculate the total sand depth of all 6 sand layers.

```
H_filter_sand_total = (H_filter_layer * N_filter_layer).to(u.m)
print('The total depth of the filter sand is', H_filter_sand_total,'.')
```

The total depth of the filter sand is 1.2 meter

2)

Calculate the diameter that is larger than 60% of the sand (D60 of the filter sand).

```
D60_filter_sand = D_filter_sand_ES * UC_filter_sand
print('The D60 for the sand grain size is', D60_filter_sand,'.')
```

The D60 for the sand grain size is 0.8 millimeter

3)

What is the total filter bed plan view area for both filters in Tamara?

```
A_filter_total = (flow_plant / V_filter_backwash).to(u.m**2)
print('The filter bed plan view area is ',A_filter_total,'.')
```

The filter bed plan view area is 1.091 meter ** 2

4)

What is the velocity of water through a filter during filtration? Recall that the flow through the filter is the same in filter and backwash modes.

```
V_filter = V_filter_backwash / N_filter_layer
print('The filtration velocity is', V_filter,'.')
```

The filtration velocity is 1.833 millimeter / second

5)

Create a function to calculate the head loss through the filter at the beginning of filtration with a clean filter bed. Then use that function to find the head loss through the clean bed of the Tamara filter. Assume that each flow path receives the same flow.

Recall: - If you have flow paths in parallel, the head loss is NOT the sum of the head loss in each path. - Instead, the head loss in each path is the same as the total head loss.

```
def headloss_kozeny(PorosityFilterSand, D60_filter_sand, VelocityFilter, DepthFilter):
    return (36*k_Kozeny*((1 - PorosityFilterSand)**2 / PorosityFilterSand**3) * (
        pc.viscosity_kinematic(T_design) * VelocityFilter)/(pc.gravity * D_filter_
    ↪sand_ES**2) * DepthFilter).to(u.cm)

hl_filter_clean = headloss_kozeny(porosity_filter_sand, D60_filter_sand, V_filter, H_
    ↪filter_layer)
print('The headloss through the clean filter sand is', hl_filter_clean,'.')
```

The head loss through the clean filter sand is 15.20 cm

6)

Create a function to estimate the minimum fluidization velocity for this filter bed. Then use that function to find the minimum fluidization velocity of the Tamara filter. Fluidization occurs at the beginning of backwash as all of the water flows through the bottom inlet. Note that this is not the actual velocity used for backwashing the sand.

```
def velocity_backwash_min(porosity_sand, D60_filter_sand,T):
    return ((porosity_sand**3 * pc.gravity * D60_filter_sand**2) / (
        36 * k_Kozeny * pc.viscosity_kinematic(T) * (1 - porosity_sand)) * (
            density_sand / pc.density_water(T)-1)).to(u.mm/u.s)

print('The minimum fluidization velocity for this filter bed is', velocity_backwash_
    ↪min(porosity_filter_sand, D60_filter_sand,T_design),'.')
```

The minimum fluidization velocity for this filter bed is 6.1 mm/s

7)

First, plot the minimum backwash velocity as a function of water temperature from 0°C to 30°C. Then use your plot to answer the following question: if you have a water treatment plant with a single filter and there is a drought that is reducing flow to the plant, when should you backwash the filter? Should you backwash when the water is coolest or when the water is warmest?

```
T_graph=np.linspace(0,30,50)*u.degC
ypoints = velocity_backwash_min(porosity_filter_sand, D60_filter_sand,T_graph)
fig, ax = plt.subplots()
ax.plot(T_graph,ypoints, '-')

ax.set(xlabel=r'Temperature (°C)')
ax.set(ylabel='Minimum backwash velocity')
fig.savefig('Filtration/Images/Minimum_backwash_velocity_vs_water_temperature')
plt.show()
```

8)

What is the residence time of water in the filter during backwash, when the bed is fluidized? You may assume the sand bed expansion ratio is 1.3.

```
Pi_filter_backwash_expansion = 1.3
t_res_filter_backwash = ((porosity_filter_sand * H_filter_sand_total + (
    Pi_filter_backwash_expansion - 1) * H_filter_sand_total
```

(continues on next page)

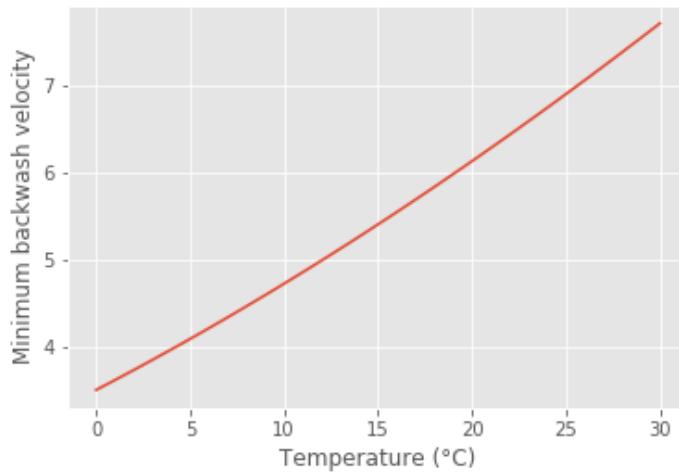


Fig. 41.1: The minimum backwash velocity increases with temperature. Thus it is best to backwash when the water is coolest.

(continued from previous page)

```
) / V_filter_backwash).to(u.s)

print('The residence time in the fluidized bed during backwash is', t_res_filter_
    ↪backwash, '.')
```

The residence time in the fluidized bed during backwash is 76.36 second

Our next overall goal is to determine the ratio of water wasted in a Stacked Rapid Sand (StaRS) Filter to water treated in a StaRS. Given that the backwash water that ends up above the filter bed never returns to the filter it isn't necessary to completely clear the water above the filter bed during a backwash cycle. Therefore we anticipate that backwash can be ended after approximately 3 expanded bed residence times. In addition it takes about 1 minute to initiate backwash by lowering the water level above the filter bed.

9)

To start, estimate the time between beginning backwash and finishing the cleaning of the bed.

```
t_filter_backwash_drain = 1 * u.min
t_filter_backwash = (t_res_filter_backwash * 3 + t_filter_backwash_drain).to(u.s)

print('The time to backwash the filter is', t_filter_backwash, '.')
```

The time to backwash the filter is 289.1 second

10)

Estimate the total **depth** of water that is wasted while backwash is occurring.

```
H_filter_backwash_water = (V_filter_backwash * t_filter_backwash).to(u.m)

print('The total depth of water that is wasted is', H_filter_backwash_water, '.')
```

The total depth of water that is wasted is 3.18 meter

11)

Estimate the total depth of water that is lost due to refilling the filter box at the end of backwash plus the slow refilling to the maximum dirty bed height. You may ignore the influence of plumbing head loss and you may assume that the dirty bed head loss is about 40 cm. The water level in the filter during backwash is lower than the water level at the end of filtration by both the head loss during backwash AND the head loss at the end of filtration. There is also an additional 20 cm of lost water that is required for the hydraulic controls.

To reiterate, the three components that contribute to the depth of water lost in refilling the filter box after backwash are as follows:

1. Head loss during clean-bed filtration.
2. Difference in head loss between clean-bed filtration and dirty-bed filtration, just before backwash.
3. Height of the pipe that initiates backwash, also called the hydraulic control. This is actually the pipes diameter, since it is laying sideways in the filter.

```
hl_filter_final = 40 * u.cm
H_hydraulic_controls = 20*u.cm
H_filter_backwash_water_refill = H_filter_sand_total + H_hydraulic_controls + hl_
↪filter_final

print('The total depth of water that is lost due to refilling the filter box is', H_
↪filter_backwash_water_refill)
```

The total depth of water that is lost due to refilling the filter box is 1.8 meter

12)

Now calculate the total length (or depth) of water that is wasted due to backwash by adding the two previous lengths. The length found in problem 10 represents water wasted while backwash is occurring, while the length in problem 11 represents the water lost in the transition to and from backwash.

```
H_filter_backwash_water_wasted = H_filter_backwash_water + H_filter_backwash_water_
↪refill
print('The depth of the water that is wasted due to backwash is', H_filter_backwash_
↪water_wasted)
```

The depth of the water that is wasted due to backwash is 4.98 meter

13)

Assume that the filter is backwashed every 12 hours. This means that the filter is producing clean water for 12 hours before it need to be backwashed. What is the total height (or length) of water that would be treated by the filter during this time? This length when multiplied by the area of the filter would give the total volume of water processed by a filter.

```
t_filter_cycle = 12 * u.hr
H_water_filtered_per_cycle = (t_filter_cycle * V_filter_backwash).to(u.m)
print('The height of water that would enter the filter in 12 hours is', H_water_
↪filtered_per_cycle)
```

The height of water that would enter the filter in 12 hours is 475.2 meter

14)

Finally, what is the ratio of water lost due to backwash and related water level changes in the filter box to water treated?

```
Pi_water_wasted = (H_filter_backwash_water_wasted / H_water_filtered_per_cycle)
print('The fraction of the total water that is lost due to backwash is', Pi_water_
      ↵wasted)
```

The fraction of the total water that is lost due to backwash is 0.01048 dimensionless

15)

Now we will evaluate the very first data set from a full scale SRSF. The performance data given below is the settled water turbidity and then the filtered water turbidity during one filter run. The time step is 5 minutes. Plot pC* for the filter as well as effluent turbidity as a function of time on two separate graphs.

```
filter_influent = np.array([7.06201, 7.14465, 7.00537, 6.33032, 5.4502, 4.98511, 4.64221, 4.
                           ↵23853, 3.93707, 3.72717, 3.67126, 3.55866, 3.55292, 3.45593, 3.48163, 3.50226, 3.45093, 3.
                           ↵50232, 3.41095, 3.55341, 3.46643, 3.50732, 3.49146, 3.51208, 3.487, 3.58893, 3.54315, 3.61469,
                           ↵3.58429, 3.55835, 3.72723, 3.7829, 3.74201, 3.82398, 3.74725, 3.83423, 3.72717, 3.72705, 3.
                           ↵87079, 3.77338, 3.70148, 3.74762, 3.76227, 3.58875, 3.63477, 3.67566, 3.52789, 3.59296, 3.
                           ↵66577, 3.51709, 3.63965, 3.57843, 3.47546, 3.66016, 3.58386, 3.53259, 3.57886, 3.51392, 3.
                           ↵63983, 3.64972, 3.64606, 3.49121, 3.51691, 3.5119, 3.61414, 3.5835, 3.46558, 3.63965, 3.60962,
                           ↵3.50147, 3.51678, 3.46039, 3.49133, 3.48566, 3.50104, 3.5943, 3.47778, 3.53766, 3.55811, 3.
                           ↵50635, 3.42535, 3.48077, 3.54242, 3.55274, 3.59369, 3.48596, 3.53296, 3.51746, 3.45056, 3.
                           ↵59387, 3.5426, 4.26868, 3.99201, 3.45569, 3.86975, 3.54407, 3.49097, 3.9823, 3.58325, 3.77789,
                           ↵3.70111, 3.59839, 4.09424, 3.56769, 3.83325, 3.76019, 3.49121, 4.52917, 3.63037, 4.21228, 3.
                           ↵60858, 4.72827, 4.00757, 3.71674, 3.87647, 3.8288, 3.44067, 3.74219, 3.64941, 3.87439, 3.
                           ↵79327, 4.10486, 4.16638, 3.84418, 4.11792, 3.46082, 3.71661, 3.5061, 3.48596, 4.42175, 3.
                           ↵57806, 4.01294, 3.63965, 3.80408, 3.60431, 3.67572, 3.61981, 3.66022, 3.67554, 3.46076, 3.
                           ↵72669, 3.56287, 3.66003, 3.60004, 3.4812, 3.55823, 6.19641, 3.74146, 3.88025, 4.18713, 4.
                           ↵27881, 3.6496, 3.45111, 3.8656, 3.90106, 3.68597, 3.66101, 3.88513, 3.74652, 3.70123, 3.79828,
                           ↵3.84369, 3.59418, 3.60968, 3.49127, 3.45081, 3.40424, 3.35852, 3.32788, 3.29211, 3.21008, 3.
                           ↵79279, ]) * u.NTU
filter_effluent = np.array([0.926376, 0.645981, 0.69725, 0.625855, 0.600449, 0.472694, 0.
                           ↵381546, 0.340439, 0.309883, 0.289848, 0.27813, 0.254097, 0.249432, 0.253769, 0.203117, 0.
                           ↵253891, 0.34063, 0.223385, 0.365952, 0.264389, 0.259193, 0.340576, 0.279671, 0.309776, 0.
                           ↵304878, 0.279404, 0.299896, 0.340294, 0.25399, 0.350731, 0.34053, 0.487877, 0.391518, 0.
                           ↵309715, 0.360901, 0.442352, 0.30967, 0.391251, 0.457253, 0.447685, 0.386322, 0.549149, 0.
                           ↵33036, 0.426437, 0.513458, 0.279022, 0.462692, 0.589768, 0.381073, 0.532669, 0.538529, 0.
                           ↵350472, 0.559036, 0.51828, 0.462578, 0.594772, 0.47242, 0.538681, 0.605087, 0.402168, 0.
                           ↵538818, 0.66658, 0.447243, 0.574585, 0.691544, 0.513382, 0.655701, 0.655884, 0.605408, 0.
                           ↵666153, 0.599976, 0.651047, 0.702515, 0.92572, 0.90535, 0.732117, 0.605026, 0.849335, 0.
                           ↵691925, 0.584976, 0.803314, 0.783569, 0.706787, 0.783066, 0.640701, 0.732269, 0.803497, 0.
                           ↵625458, 0.625687, 1.00089, 1.03265, 1.00687, 0.834213, 0.732224, 0.859665, 0.559174, 1.08862,
                           ↵0.93634, 0.722717, 0.966141, 1.03278, 0.890366, 0.874756, 0.813675, 0.864746, 0.874863, 0.
                           ↵625504, 0.874847, 0.889893, 0.931473, 0.880112, 1.30231, 0.813965, 0.961502, 0.966324, 0.
                           ↵839432, 0.889969, 0.930618, 1.20074, 0.93631, 0.951584, 0.78331, 0.961609, 0.996796, 0.
                           ↵940994, 0.966675, 1.04788, 0.951645, 0.966888, 1.01251, 0.96167, 1.0076, 0.900436, 0.996628,
                           ↵1.12949, 0.910263, 1.03256, 1.04788, 0.997604, 1.05817, 1.05801, 1.06311, 1.05805, 1.00711, 1.
                           ↵05811, 1.11896, 1.06299, 1.07318, 0.966049, 1.07343, 1.09918, 1.15497, 1.07303, 1.13937, 1.
                           ↵12402, 1.155, 1.15436, 1.2056, 1.14954, 1.17526, 1.15457, 1.20569, 1.17517, 1.0376, 0.915253,
                           ↵0.935471, 0.884979, 0.89035, 0.940781, 0.885071, 0.874817, 0.828796]) * u.NTU

t_delta = 5 * u.min
```

(continues on next page)

(continued from previous page)

```
t_end = np.shape(filter_effluent)[0]
t_data = np.arange(0,t_end)*t_delta
c_dim = np.divide(filter_effluent,filter_influent)
filter_pc = -np.log10(c_dim)
fig, ax = plt.subplots()
ax.plot(t_data.to(u.hour),filter_pc)
ax.set(xlabel='Filter run time (hrs)')
ax.set(ylabel='pC*')
fig.savefig('Filtration/Images/Filter_run_time_vs_removal_efficiency')
plt.show()

fig, ax = plt.subplots()
ax.plot(t_data.to(u.hour),filter_effluent)
ax.set(xlabel='Filter run time (hrs)')
ax.set(ylabel='Effluent turbidity (NTU)')
fig.savefig('Filtration/Images/Filter_run_time_vs_effluent_turbidity')
plt.show()
```

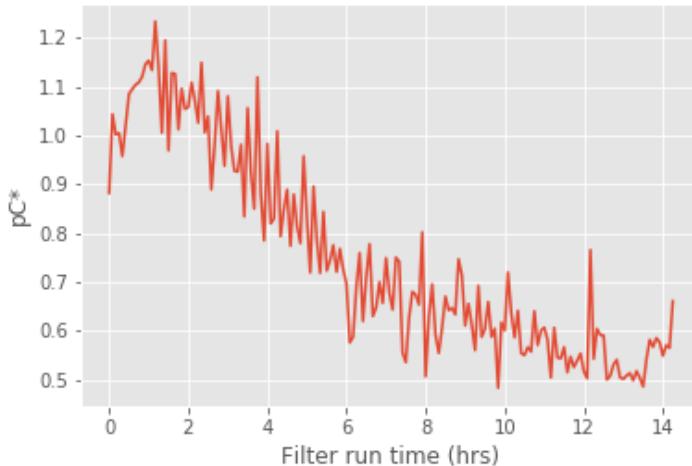


Fig. 41.2: The pC* for this filter run was not very good and suggests that either some particles were being released by the new sand or the coagulant dose was not optimal.

16)

How many kg of suspended solids per square meter of filter were removed during this filter run. Use the plan view area for the filter (dont multiply by the number of layers)

```
M_filter_solids = (np.sum((filter_influent - filter_effluent) * (
    flow_plant * t_delta) / A_filter_total)
    ).to(u.kg/u.m**2)

print('The mass of the suspended solids removed is', M_filter_solids)
```

The mass of the suspended solids removed is 2.94 kg/m²

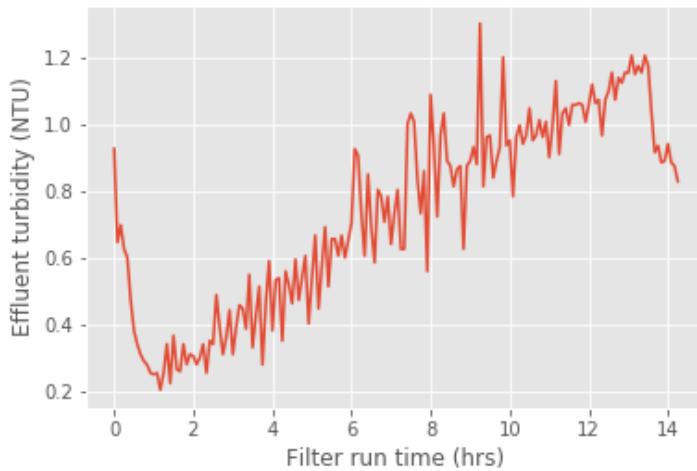


Fig. 41.3: The filter performance deteriorated over the length of the filter run. This does not match the expectations that we have based on laboratory experiments with filters. AquaClara has limited data of filter performance as a function of time. However, the [recent data from Tamara](#) (select Tamara from the drop down menu of plants) suggests that filtered water turbidity is consistently lower than in this first run of the filter that you plotted above.

17)

Another useful way to express the solids capacity of the filter is to calculate the turbidity removed the run time and then express the results with units of NTU hrs. What was the capacity of the filter in NTU hrs?

```
solids_capacity_filter = (np.sum((filter_influent-filter_effluent) * t_delta)).to(u.  
                                NTU * u.hr)  
  
print('The filter capacity is',solids_capacity_filter,)
```

The filter capacity is 43.72 NTU * hour

18)

How long was the filter run?

```
t_filter_cycle = t_data[np.size(t_data)-1]  
print('The filter was run for', t_filter_cycle.to(u.hour))
```

The filter was run for 14.25 hour

19)

What is the total volume of pores per square meter (plan view area) of StarS filter bed (includes all 6 layers) (in L/m²)?

```
volume_filter_pores = (H_filter_sand_total * porosity_filter_sand).to(u.L/u.m **2)  
print('The total volume of pores is', volume_filter_pores)
```

The total volume of pores is 480 liter / meter ** 2

20)

The next step is to estimate the volume of flocs per plan view area of the filter. Assume the density of the flocs being captured by the filter are approximated by the density of flocs that have a sedimentation velocity of 0.10 mm/s (slightly less than the capture velocity of the plate settlers). (see slides in flocculation notes for size of the floc and then density of that floc. I've provided this value below to simplify the analysis

```
density_floc = pc.density_water(T_design) + 100 * u.kg/u.m**3
density_clay = 2650 * u.kg/u.m**3
```

Given the floc density, calculate fraction of floc volume that is clay.

Given that floc mass is the sum of clay mass and water mass and given that floc volume is the sum of clay volume and water volume, derive an equation for the volume of flocs per plan view area of a stacked rapid sand filter (includes all 6 layers) given the floc, clay, and water densities and the mass of the clay. Show the equations that you derive using Latex

Mass conservation gives

$$Vol_{Floc} \cdot \rho_{Floc} = M_{Clay} + M_{Water} \quad (41.1)$$

M_{Water} is an unknown.

$$M_{Water} = Vol_{Floc} \cdot \rho_{Floc} - M_{Clay} \quad (41.2)$$

Volume conservation gives

$$Vol_{Floc} = Vol_{Clay} + Vol_{Water} \quad (41.3)$$

$$Vol_{Floc} = \frac{M_{Clay}}{\rho_{Clay}} + \frac{M_{Water}}{\rho_{Water}} \quad (41.4)$$

Substitute to eliminate M_{Water}

$$Vol_{Floc} = \frac{M_{Clay}}{\rho_{Clay}} + \frac{Vol_{Floc} \cdot \rho_{Floc}}{\rho_{Water}} - \frac{M_{Clay}}{\rho_{Water}} \quad (41.5)$$

Solve for Vol_{Floc}

$$Vol_{Floc} - \frac{Vol_{Floc} \cdot \rho_{Floc}}{\rho_{Water}} = \frac{M_{Clay}}{\rho_{Clay}} - \frac{M_{Clay}}{\rho_{Water}} \quad (41.6)$$

$$Vol_{Floc} \left(1 - \frac{\rho_{Floc}}{\rho_{Water}}\right) = M_{Clay} \left(\frac{1}{\rho_{Clay}} - \frac{1}{\rho_{Water}}\right) \quad (41.7)$$

$$Vol_{Floc} = M_{Clay} \left(\frac{\frac{1}{\rho_{Clay}} - \frac{1}{\rho_{Water}}}{1 - \frac{\rho_{Floc}}{\rho_{Water}}}\right) \quad (41.8)$$

$$Vol_{Floc} = \frac{M_{Clay} \rho_{Water}}{\rho_{Floc} - \rho_{Water}} \left(\frac{1}{\rho_{Water}} - \frac{1}{\rho_{Clay}}\right) \quad (41.9)$$

```
vol_floc = ((M_filter_solids * pc.density_water(T_design) / (density_floc - pc.density_water(T_design))) *
            ((1 / pc.density_water(T_design)) - 1 / density_clay)).to(u.l / u.m **2)
print('The volume of the flocs per plan view area is', vol_floc)
```

The volume of the flocs per plan view area is 18.34 liter / meter ** 2

21)

What percent of the filter pore volume is occupied by the flocs? This fraction of pore space occupied is quite small and suggests that much of the filter bed has a very low particle concentration at the end of a filter run.

```
Pi_flocvolume_porevolume = (vol_floc / volume_filter_pores)

print('The fraction of filter pore volume that is occupied by flocs is', Pi_
      ↴flocvolume_porevolume)
```

The fraction of filter pore volume that is occupied by flocs is 0.0382

This result is surprising and intriguing. It indicates that the pores in the filters are 96% empty when the filter run is complete! Thus filters don't fail because the pores get full. There is a different mechanism at play here.

41.2 Filter constriction hypothesis

The following analysis is completed for you and is intended to illustrate the hypothesis that flocs that are removed by the filter form a small diameter flow constriction at each place where the sand grains form a flow constriction.

Final head loss for the filter was 50cm. Assume that this is caused by minor losses due to creation of a floc orifice (constriction) in each pore. Find the minor loss contribution by subtracting the clean bed head loss to find the head loss created by the flow constrictions that were created by the flocs.

```
hl_filter_final = 50 * u.cm
hl_constriction = hl_filter_final-hl_filter_clean

print('The minor loss contribution is', hl_constriction)
```

The minor loss contribution is 34.8 centimeter

If we assume that at the end of the filter run every pore in the filter had a flow constricting orifice from the deposition of flocs in the pore, then what was the diameter of each of the flow constrictions? We will calculate this in several steps. To begin, estimate how many flow constrictions are created by the sand grains before any flocs are added with the assumption that there is one flow constriction per sand grain. How many sand grains are there per cubic meter of filter bed? Use D60_filter_sand to estimate the number of sand grains. We will assume there is a one to one correspondence between sand grains and flow constrictions.

```
vol_filter_sand_grain = D60_filter_sand**3 * np.pi/6
vol_filter_sand_grain_and_pore = vol_filter_sand_grain / (1 - porosity_filter_sand)
N_sand_grains = 1 / (vol_filter_sand_grain_and_pore)

print('There are this many sand grains in a cubic millimeter', N_sand_grains.to(1/u.
      ↴mm**3))
```

There are this many sand grains in a cubic millimeter 2.238 / millimeter ** 3

Estimate the average vertical distance between flow constriction based on the cube root of the volume occupied by a sand grain

```
L_grain_separation = (vol_filter_sand_grain_and_pore ** (1/3)).to(u.mm)

print('The distance between flow constriction is', L_grain_separation)
```

The distance between flow constriction is 0.7645 millimeter

On average, how many sand grain flow constriction does a water molecule flow through on its way through the filter?

```
N_constriction = (H_filter_layer/L_grain_separation).to(u.dimensionless)

print('A water molecule flows through', N_constriction, 'constriction through the',
      'StaRS filter')
```

A water molecule flows through 261.6 dimensionless constriction through the StaRS filter

What is the head loss per flow constriction?

```
hl_per_constriction = (hl_constriction / N_constriction).to(u.mm)
print('The head loss per constriction is', hl_per_constriction)
```

The head loss per constriction is 1.33 millimeter

If each constriction was partially clogged with flocs at the end of the filter run, estimate the velocity in the constriction using the expansion head loss equation. You can use the average pore water velocity as a good estimate of the expanded flow velocity.

$$h_e = \frac{(V_{in} - V_{out})^2}{2g} \quad (41.10)$$

```
V_pore = V_filter / porosity_filter_sand
V_constriction = ((2 * pc.gravity * hl_per_constriction)**(1/2) + V_pore).to(u.mm / u.s)

print('The velocity in the constriction is', V_constriction)
```

The velocity in the constriction is 166.1 millimeter / second

What is flow rate of water through each pore in $\mu\text{L/s}$? You can estimate this from the number of pores per square meter given the average separation distance.

```
N_pore_per_area = 1 / L_grain_separation**2
flow_per_pore = (V_filter / N_pore_per_area).to(u.microliter/u.s)

print('The flow rate through each pore is', flow_per_pore)
```

The flow rate through each pore **is 1.071 microliter / second**

What is the inner diameter of the flow constriction created by the flocs if the vena contracta is 0.62?

```
Pi_vena_contracta = 0.62
A_constriction = flow_per_pore / V_constriction / Pi_vena_contracta
D_constriction = pc.diam_circle(A_constriction)
print('The inner diameter of the flow constriction created by the flocs is', D_
      .constriction.to(u.micrometer))
```

The inner diameter of the flow constriction created by the flocs is 115.1 micrometer

This suggests that this flow constriction is stable because the high velocity results in shear levels that are too high for flocs to attach. Thus once the constriction forms and reaches the shear level that prevents deposition it remains stable.

Plot the fractional removal per constriction as a function of particle size.

```
D_clay = 7 * u.micrometer

#create an array of floc sizes from clay diameter up to the diameter of the
      'constriction'
```

(continues on next page)

(continued from previous page)

```

def D_floc(D_constriction):
    return (np.linspace(D_clay.to(u.micrometer), D_constriction.to(u.micrometer)))*u.
    ↵micrometer

#below is an estimate of the floc removal efficiency as a function of the floc size
def c_star_constriction(D_constriction):
    return ((D_constriction - D_floc(D_constriction))
        / D_constriction)**2

c_star = c_star_constriction(D_constriction)
fig, ax = plt.subplots()
ax.plot(D_floc(D_constriction),c_star)
ax.set(xlabel='Particle diameter(micrometer)')
ax.set(ylabel='Fractional remaining')
fig.savefig('Filtration/Images/Diameter_vs_fractional_remaining')
plt.show()

```

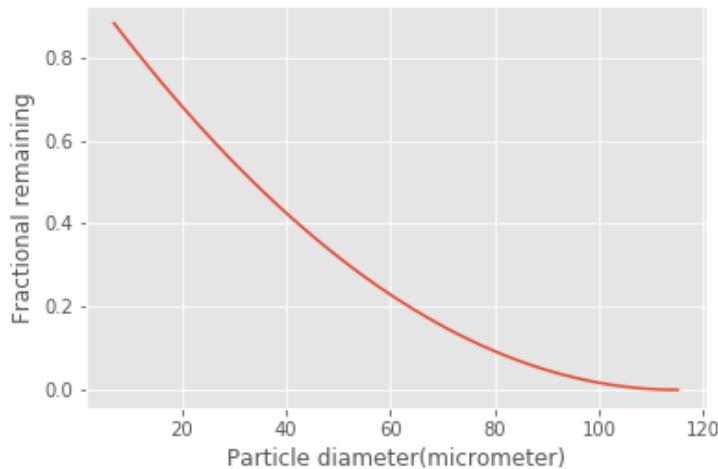


Fig. 41.4: There are many constrictions in series and the filter fraction remaining is the pore fraction remaining raised to the power of the number of pores in series.

CHAPTER
FORTYTWO

FILTRATION DESIGN CHALLENGE

42.1 1)

Calculate the turbidity of the backwash water. Ideally use POST data to do this. An alternative would be to use CUWFP data.

42.2 2)

An engineer suggests that you should install a roughing filter uphill from an AguaClara plant to improve its ability to treat turbid water. This is a real question from a conference presentation in Honduras. How would you reply to this suggestion. Consider how to answer without offending the questioner. Use the required sizes of the unit processes to guide your response.

42.3 3)

Why dont StaRS filters experience surface clogging?

42.4 4)

Explain why particles are captured at flow constrictions inside sand filters.

42.5 5)

Explain the role of coagulant nanoparticles in the capture of particles by flow constrictions.

42.6 6)

Explain why a flow constriction in a filter has a limited capacity to collect particles.

CHAPTER
FORTYTHREE

DISSOLVED GAS INTRODUCTION

Dissolved gas supersaturation and subsequent bubble formation can cause significant performance deterioration in drinking water treatment plants. Supersaturation means that the dissolved gas concentration is greater than the equilibrium concentration. The liquid phase equilibrium concentration of a gas is directly proportional to the absolute partial pressure of the gas in the gas phase. This relationship is described by Henry's law.

43.1 Henrys Law

43.2 Effects of Gas Bubbles in Water Treatment Plants

The American Water Works Association recognized the detrimental effects of gas bubble evolution in drinking water treatment plants and sponsored research at Virginia Polytechnic Institute and State University. [Paolo Scardinas Dissertation](#) provides an excellent review of the problems that are caused by gas bubbles.

Dissolved gas removal is also a goal after anaerobic wastewater treatment for enhanced removal of dissolved methane.

43.2.1 Flocculation

Minimal adverse effects. Foam produced can dry and be blown around the water treatment plant. Nuisance and aesthetically unpleasing.

43.2.2 Floc Blankets

Disrupts floc blanket, breaks flocs, carries flocs into plate settlers.

43.2.3 Sedimentation

Bubbles carry flocs rapidly to the top of the sedimentation tank where they form a surface mat of flocs. The surface mat of flocs isn't necessarily harmful to plant performance, but it is a nuisance and requires that the operators scrape the flocs to remove them from the sedimentation tank. Some of the rising flocs are swept into the exit manifold and thus the settled water turbidity can be significantly increased by these rising flocs.

43.2.4 Filtration

Filtration performance can be severely harmed by gas bubbles. The bubbles form in the pores of the filter and are unable to leave the pore due to surface tension. Filter media can fill rapidly with gas bubbles and this results in increased head loss or decreased flow through the filter. Bubble formation can significantly reduce the filter run time.

The amount of bubble formation is strongly dependent on the pressure in the filter. Filters that operate with a deep column of water on top of the filter bed can be unaffected by dissolved gasses if the high pressure within the filter raises the equilibrium dissolved gas concentration above the actual dissolved gas concentration. Enclosed Stacked Rapid Sand Filters are more vulnerable to bubble formation than Open Stacked Rapid Sand Filters because of the difference in pressure in the sand bed. This pressure analysis also reveals that gas bubble formation will tend to occur at the lowest pressure zone in the filter bed. This low pressure zone can occur in a down flow filter right below the partially clogged section of the bed because of the decrease in pressure due to head loss.

At the El PODA Nicaragua AquaClara plant there was so much dissolved air in the influent water that the filters could only operate one hour before requiring backwash.

43.3 Dissolved Gas Sources

- high pressure regions in transmission lines where the pipeline flow capacity exceeds the need AND where the flow control is restricted by upstream supply rather than by downstream head loss control (meaning by partially closing a valve!).
- Temperature increase that reduces the equilibrium concentration. This gas source is observed in the AquaClara lab facilities where hot and cold incoming tap water are mixed to produce room temperature water. The increase in water temperature results in bubble formation.

43.4 Dissolved Gas Concentration Reduction

43.4.1 Methods to provide nucleation sites

- aeration - air can be pumped through diffusers - air can be sucked into the diffusers if the reactor operated at less than atmospheric pressure
- fluidized bed containing particles that have nucleation sites

The fluidized nucleation site particles must be dense enough and large enough so that attached gas bubbles can't carry them out of the degassing reactor.

43.4.2 Methods to reduce degassing reactor pressure

The top of the degassing reactor can be located several meters above the entrance tank of the water treatment plant. The higher the tank (up to a max of 10 m) the more pronounced the pressure reduction will be and the more effective the reactor will be at degassing. The degassing reactor will produce a water and bubble effluent that must be

CHAPTER
FORTYFOUR

FLUORIDE INTRODUCTION

References

Fluoride, natural organic matter, and particles : the effect of ligand competition on the size distribution of aluminum precipitates in flocculation

CHAPTER
FORTYFIVE

WASTEWATER THEORY AND FUTURE WORK

Although AguaClara began with a focus on drinking water treatment, we have always been keenly aware that adequate wastewater treatment is absolutely essential to reduce harm to the environment and harm to downstream communities.

One of the core ideas of the AguaClara design process is that reactor geometry and hydraulic design are critical to obtain the target performance. Environmental engineers have tended to focus on the microbiology and chemistry of unit processes and have sometimes neglected the interactions between fluids, particles, and reactor geometry. We hypothesize that it will be possible to significantly improve on the conventional UASB design by inventing an anaerobic digester that accounts for the interactions between fluids, particles, and reactor geometry. Similarly, we hypothesize that it will be possible to dramatically improve the design of ultra low energy atmospheric oxygen transfer into aerobic reactors.

45.1 Anaerobic Pulsed Bed

Upflow anaerobic settled bed (UASB) are conventionally known as upflow anaerobic sludge blanket reactors. The word blanket is frequently used in the field of water and wastewater treatment to refer to a fluidized bed of suspended particles (see floc blanket). Unfortunately that definition is not clearly communicated by the term blanket and this has led to confusion of the fundamental mechanisms at play in UASB reactors.

Fluidized bed reactors require inlet and bottom geometry configurations that prevent settled particles from accumulating anywhere on the bottom of the reactor. Many UASB reactors have flat bottoms and the inlets are not designed to ensure continuous resuspension of settled particles. Thus the use of the term blanket for conventional UASB reactors is a misnomer.

UASB reactors typically require hydraulic residence times of x to y hours and have a height of z meters. The result is a maximum upflow velocity of y/z mm/s. This upflow velocity is orders of magnitude lower than the terminal velocity of the particles and thus it is clear that UASB reactors are primarily settled beds of sludge.

The flow distribution through settled sludge is very unlikely to be uniform. The flow is likely to erode a mostly vertical path the shortest distance between the inlet and the top of the settled sludge. There doesn't appear to be any mechanism that would lead to the idealized uniform flow distribution. Thus conventional UASB reactors are evidently plagued by short circuiting with actual hydraulic residence times a fraction of the design value. (Cite literature in support of this hypothesis.) This leads to short-circuiting and formation of preference flow patterns in sludge bed which in turn leads to dead zones in the sludge as well as improper treatment (Pena, 2006)

The upflow velocity required to maintain a fully fluidized bed of the anaerobic granules is approximately (cite AguaClara UASB research by Cho, et al. who measured the sedimentation velocity of anaerobic granules) x mm/s. At this velocity the height of the reactor would need to be x m in order to achieve the target hydraulic residence time of y hrs. This is not a practical design for community scale reactors and thus it would be advantageous to invent an alternate system for providing more uniform flow through the solids that contain the microorganisms in a UASB reactor.

Our proposed solution to this mismatch between required upflow velocity for a fluidized bed and target hydraulic residence time

- a small fraction of the depth of the UASB (perhaps less than 10% to ensure that no fresh wastewater can jet through the entire UASB in the time that the sludge settled again)
- large enough to provide a flow passage underneath the lifted bed without requiring flow velocities that are so high that the bed is scoured near the inlet jet. This translates to larger than a minimum ratio of fresh wastewater depth per pulse/inlet spacing.

Research is needed to characterize settled bed behavior under pulsed flow.

- How does a settled bed form as suspended solids gradually settle for the cases of continuous and pulsed flows?
- What is the actual hydraulic residence time distribution in the bed for the case of continuous and pulsed flows?
- What are the failure modes for the pulsed system?
- What is the optimal pulsed height (volume of pulse/area of reactor)?
- How does the optimal pulsed height scale inlet spacing and bed depth? It will be difficult to conduct experiments at full scale and thus these experiments will require careful consideration of scaling effects. Full scale validation will be very helpful if we can develop a method.

All of this research will be aided by using transparent reactor walls to facilitate direct observation of the settled solids.

45.2 String Digester (SD)

Trickling filters are an old wastewater treatment technology that is much more energy efficient than the activated sludge process.

The measured hydraulic residence time for trickling filters is very short. This suggests that with proper design the ASD could be very compact. [Hinton and Stense \(1991\)](#) measured the residence time per unit length to be 30 seconds/meter. Thus for a 4 meter deep trickling filter the residence time would be 120 seconds. If this is accurate, then we may be able to achieve a compact design if we can pack stainless steel cables close together (order 4 mm spacing) AND achieve uniform flow distribution. In addition, [Hinton and Stense \(1991\)](#) used a hydraulic application rate of 4 m/hr⁻¹ (1.1 mm/s). This velocity confirms that a compact, well-designed ASD may be smaller than AquaClara sedimentation tanks that traditionally have operated at 1 mm/s.

Modular plastic trickling filter media are typically manufactured with the following specific surface areas:

- 223 m²/m³ as high density
- 138 m²/m³ as medium density
- 100 m²/m³ as low density

Vertical-flow media require an average hydraulic approach velocities greater than 1.8 m/h (0.5 mm/s) to maximize BOD5 removal efficiency. Shallow towers using cross-flow media have used hydraulic approach velocities in the range 0.4 to 1.1 m/h (0.1 to 0.3 mm/s) ([Daigger and Boltz, 2011](#))

[Crine et al. \(1990\)](#) found that the wetted area-to-specific-surface-area ratio ranged from 0.2 to 0.6 with the lowest values for high-density random pack trickling filter media. This confirms that conventional trickling filters are unable to efficiently wet all biofilm surfaces and thus the trickling filters must be substantially over-designed (by a factor of 2 to 5) to accommodate this poor wetting efficiency.

If we take the hydraulic approach velocity of 0.5 mm/s and divide by the wetted area-to-specific-surface-area ratio of 0.6 we obtain 0.83 mm/s, a velocity that is comparable to the upflow velocity in an AquaClara sedimentation tank. Thus a well designed String Digester could be quite compact.

There is extensive literature on design of trickling filters for removal of various nutrients and integration into multi-process treatment trains. Control of biofilm thickness seems to be a recurring issue and thus may be an important research area for the String Digester.

Can we make some sort of a diagnostic guide based on symptoms? Make the following into a giant table.

**CHAPTER
FORTYSIX**

PROBLEMS

Temperature fluctuations

Calcium from calcium hypochlorite combines with carbonate in the water to form low solubility calcium carbonate.

CHAPTER
FORTYSEVEN

SOLUTIONS

Drip chlorine into water rather than injecting it to eliminate formation of precipitate at the injection point

Use valves at the end of the pipeline to control transmission line flow rate rather than upstream control. Review the transmission line to ensure that all sections of the line have flow controlled by limiting available driving head. Air entrainment occurs when the available head exceeds the head required to transmit the target flow.

Table 47.1: Table of symptoms, problems, and solutions for AguaClara plant operation.

Observation	Problem	Solution
Air bubbles	Air entrainment in the transmission line and transport to a high pressure zone in the pipeline where the air is dissolved in the water	Use downstream head loss to prevent air entrainment in transmission line or add a unit process (TBD) that removes excess dissolved air
Bubbles in sedimentation tanks	anaerobic production in methane in settled sludge	clean sedimentation tank and check design to eliminate all settled sludge
Bubbles in EStaRS	water entering the plant is supersaturated with air and EStaRS filters operate at very low pressure (compared with OStaRS)	eliminate air entrainment in transmission line
Short filter runtimes	poor performance of floc/sed system or clogged slotted pipes or air lock	various solutions
Gradual increase in post backwash head loss in filters	slotted pipes are clogging	acid flush by pouring HCl (need the concentration) into the outlet pipes OR disassemble filter and clean with a pressure washer
Scale deposition in the distribution system	increase in temperature from groundwater conditions causes supersaturation of calcium carbonate	reduce the pH of the water slightly (perhaps 0.5 pH units) to eliminate the supersaturation
Clogging of chlorination system tubes and formation of precipitate at the injection point	dissolution of carbon dioxide from the atmosphere causes precipitation of calcium carbonate	Reduce exposure of the solution to the atmosphere OR drip the chlorine into the filtered water

47.1 High head loss after backwash in EStaRS

The 60 cm diameter EStaRS filters at El PODA had a very high head loss of 43 cm within a few minutes of ending backwash. The maximum available head is only 50 cm and thus filter runs lasted only a few hours. El PODA was the 2nd site where 60 cm diameter EStaRS filters were used. Given that the filter backwashed just fine without excess head required it appeared that the inlet system was performing as expected and did not have excessive head loss.

The key insight was that the top two layers of sand stopped producing filter water soon after beginning a filter run. A complete stoppage means that clogging ISNT the issue! Clogging would decrease the flow rate, but it wouldnt stop flow because the clogging would have to be absolute to stop the flow.

The failure was that the water traveling horizontally in the branches and trunk carries air bubbles to the vertically downward flowing pipe (see a photo here). The air accumulates in that pipe and the water falls through the air losing energy like a waterfall. Water completely stops flowing through the layers of the filter that exit through that trunk line when the height of the air is equal to the available energy driving water through the StaRS filter.

There are two improvements required at El PODA. First, the transmission line must be operated with downstream flow control to prevent air entrapment and compression. That will likely be sufficient to solve the problem of air in the filters. The second improvement is to add manual air release valves to the top of each of the trunk lines so that operators can expel air during filtration. Note that these air vents must have a valve that is closed during backwash because the filter is under negative pressure during backwash and would suck air into the filter through those vents if they were open.

EStaRS with the wing inlets make a possible failure mode. At the end of the backwash if the operator closes the valve in the siphon quickly, the water pressure rises quickly inside the filter. When the water rises in the three entry passages it is possible that the water carries the sand inside the branches with wings. And once there is sand inside the pipes, this sand may hinder the flow of water during the filter run.

The solution is in the operation of the system and maybe a change in the design. If the operators are manipulating pipe stubs to do the backwash, then the first step is to remove the pipe stubs before closing the valve. The other way around the problem would be to close the valve slowly to ensure that the water rises very slowly in the inlets.

47.2 Encrustation of filter pipes at Las Vegas

Remove the encrustation without disassembling the filter by pouring buckets of hydrochloric acid into pipes that are inserted in

1. Insert pipe stubs into the couplings in the filter inlet box to prevent any water from flowing through the filter
2. Remove the drain pipe stub so that any effluent from the filter goes to drain
3. Insert pipe stubs into the couplings in the filtered water effluent box
4. Fill a 20 L bucket with 16 L of water (mark the bucket level!)
5. Pour 4 L of 10% HCl into the bucket to reach a total of 20 L
6. Mix the diluted acid with a PVC pipe for a couple of minutes
7. Pour the 20 L of 2% HCl into the bottom outlet of the filter
8. Wait 5 minutes
9. Repeat previous 2 steps 5 times to pour a total of 100 L of acid into the bottom outlet pipe (note that the goal is for this total volume to exceed the outlet pipe, trunk, and branch pipes volume)
10. Pour 20 L of water into the bottom outlet pipe
11. Wait 5 minutes
12. Repeat previous 2 steps 5 times to push all (or at least most) of the acid through the outlet pipes and into the sand bed

Repeat steps 3 to 12 for the middle outlet pipe and then for the top outlet pipe.

With each addition of acid or water there should be a fresh batch of carbon dioxide bubbles released from the filter.

Turn the filter back on and dump all water to waste for at least 5 minutes and until the turbidity is less than 1 NTU.

47.3 Slime at Las Vegas

Iron bacteria slime showed up with application of hydrochloric acid at Las Vegas. The acid was needed to slightly reduce the pH to reduce the amount of encrustation in the distribution system. The addition of acid was correlated with the growth of a slime in the flocculator and sedimentation tanks. It was hypothesized that this slime was iron oxidizing bacteria (see Fig. 47.1).



Fig. 47.1: The slime at Las Vegas showed up in the flocculator and sedimentation tanks.

Iron oxidizing bacteria need oxygen and reduced iron. The Las Vegas water source is a stream that is clearly ground-water given its propensity to deposit calcium carbonate on everything in the stream. Thus the stream water is likely poor in oxygen.

We are adding oxygen at the LFOM. That oxygen can chemically oxidize the iron, but the rate of oxidation is a function of pH (see Fig. 47.2). When we decrease the pH it slows the oxidation of iron and thus keeps a higher concentration of reduced iron available for bacteria to oxidize. Thus the LFOM adds oxygen needed by the iron oxidizing bacteria and the acid prevents the iron from being chemically oxidized.

The rate of iron oxidation is strongly pH dependent At pH 7.0, 90% Fe+2 oxidation requires 1 hour at 21°C and 10 hours at 5°C. At pH 8.0, 90% Fe+2 oxidation occurs in 30 seconds. At pH 6.0 it requires 100 hours.

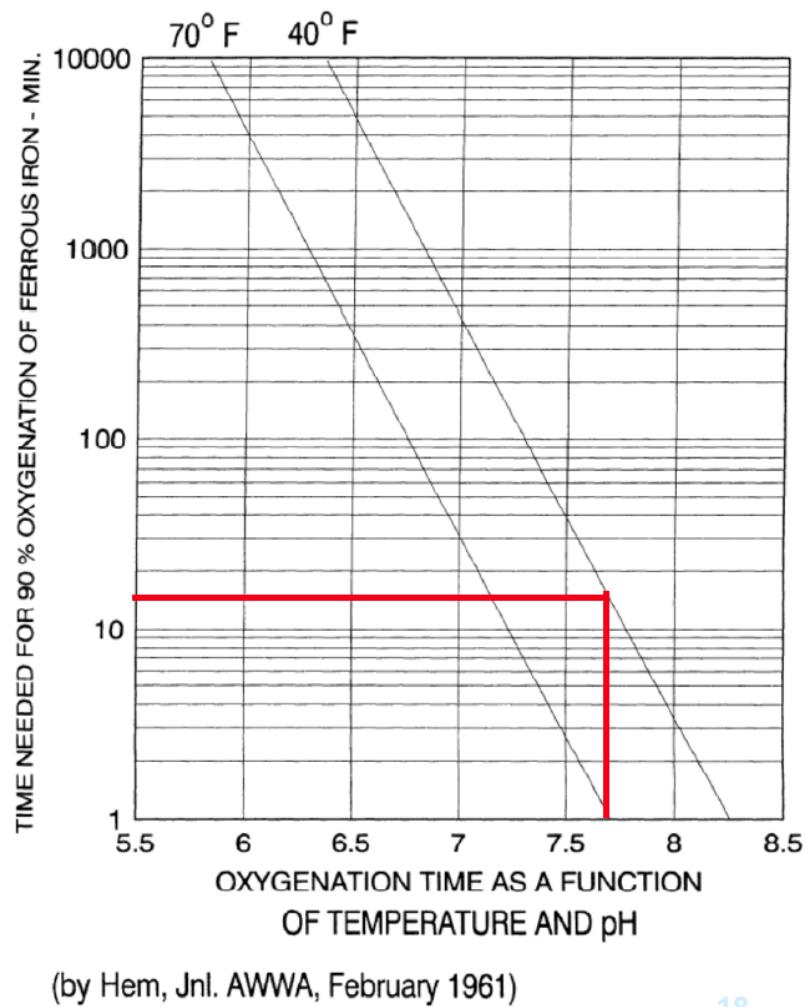


Fig. 47.2: The time required for oxidation is strongly dependent on pH. From Lenntech.

18

For several reasons, routine chemical disinfectants that effectively wipe out other bacteria are only modestly successful against iron bacteria. Iron bacteria build up in thick layers, each forming a slime around bacterial cells that keeps disinfectants from penetrating beyond the surface cells. Chemical reactions occur far slower at the cool temperatures common in wells, and bacterial cell need a long exposure to the chemical for the treatment to be effective. Even if chlorine kills all the bacterial cells in the water, those in the groundwater can be drawn in by pumping or drift back into the well. [Read more about chlorine and iron oxidizing bacteria](#)

CHAPTER
FORTYEIGHT

PROPOSED SOLUTIONS TO ELIMINATE IRON OXIDIZING SLIME BACTERIA

The solutions all revolve around oxidizing the iron chemically before the bacteria have a chance to oxidize the iron biologically.

- Move the acid addition point to the end of the plant. This will allow chemical oxidation and removal of the iron. Note that once the iron is oxidized it precipitates as $Fe(OH)_3$ and that is a wonderful coagulant. This is why the Las Vegas plant was shown to produce clean water even without addition of a coagulant!
- It might be possible to move the acid addition point to the exit of the flocculator if the flocculation process provides enough time for iron oxidation
- Create an aeration system upstream from the plant in the transmission line
- Move chlorination to the rapid mix. This might work, but given the chlorine resistance of the slime and the disadvantages of prechlorination for production of disinfection by-products, we dont recommend this.

CHAPTER
FORTYNINE

SYNTHESIS DESIGN CHALLENGES

1. Compare the hydraulic residence times for flocculation and sedimentation for AguaClara technologies and the leading commercially available clarification systems.
2. What design changes would you make to the Superpulsator to improve its performance? What improvements in performance and maintenance would your changes make? Would you recommend upgrading existing Superpulsator plants with your proposed changes?
3. Analyze the data in Jesus_de_Otoro_data.tsv. A useful way to plot water treatment plant performance data is to plot the turbidity (raw, settled, filtered) as a function of the rank of each of those data sets. Determine the fraction of the filtered and settled water samples that met the Honduran standard of 5 NTU and the USEPA standard of 0.3 NTU. (Switch this to POST data when possible.)

INTRODUCTION TO RST AND SPHINX FOR TEXTBOOK CONTRIBUTORS

50.1 What is RST?

RST stands for ReStructured Text. It is the standard markup language used for documenting python packages. Sphinx is the Python package that generates an html website from RST files, and it is what we are using to generate this site. To read more about why we chose RST over markdown or Latex, read the following section, [Why RST?](#).

50.1.1 Why RST?

In the beginning, we used markdown. As we tried to add different features to markdown (colored words, image sizes, citations), we were forced to use raw html and various pre-processors. With these various band-aid solutions came added complexity. Adding sections became cumbersome and awkward as it required ill-defined html. Additionally, providing site-wide style updates was prohibitively time-consuming and complex. Essentially, we were trying to pack too much functionality into markdown. In the search for an alternative, restructured text provided several advantages. Out of the box, RST supports globally-defined styles, figure numbering and referencing, Latex function rendering, image display customization and more. Furthermore, restructured text was already the language of choice for the AguaClara ecosystems documentation.

50.2 Setting up RST for Development

There are two ways to *quickly* view an RST file. The first is using an [Atom](#) plugin that renders the view alongside the source code. This is a good initial test to make sure the RST is proper RST and looks *mostly* correct. However, some functionality, such as any extensions provided by [Sphinx](#) wont run in the preview. In order to see the final html that will display on the website, youll need to use the second method, running sphinx locally to fully generate the html code. Once you are satisfied with your work and want to push it to the textbook, youll need to incorporate it to the master branch. To do so, refer to [Publishing online](#).

50.2.1 Installing the Atom Plugins

If you are using the Atom IDE to write RST, you can use the [rst-preview-pandoc](#) plugin to auto-generate a live RST preview within atom (much like the markdown-preview-plus preview page.) To get rst-preview working, youll need to install [language-restructuredtext](#) via atom and [Pandoc](#) via your command line (`$pip install pandoc`). If everything worked, you can use `ctrl + shift + e` to toggle a display window for the live-updated RST preview.

50.2.2 Building RST Locally with Sphinx

We use [Sphinx](#) to build RST locally and remotely. Follow these steps to get [Sphinx](#) and run it locally:

1. Get Python 3.5+ and pipenv (`$pip install pipenv`)
2. Install all dependencies and setup a virtual environment with `$pipenv install`
3. Whenever you want to issue commands, start your virtual environment with `$pipenv shell` - exit it with `$exit`
4. Generate all the html by navigating in the command line to the source directory `/Textbook` and creating the build in that directory with the command line `$make html`.
5. View the html generated in the `/Textbook/_build` directory by copying the full file path of `/Textbook/_build/html/index.html` and pasting it into your browser.

Note: Regarding 1. the master branch for the package implementing disqus in sphinx is broken, which is why we use a non-standard pip/online installation. If you already have the incorrect sphinx-disqus version installed, uninstall it with `pip uninstall sphinxcontrib-disqus` before installing the functioning version.

50.2.3 Publishing Online

We use [Travis](#) to ensure this site will always contain functional builds. To publish online, you need to:

1. Always test your build by first :ref:`building RST locally <heading_building_rst_locally>`, and then following the [testing online](#) instructions. Once you like how your build looks, follow the steps below to introduce it into the master branch.
2. Submit a [pull request to master](#). Youll need to ask for someone else to review your work at this stage- request reviewers. Every pull request **must** be reviewed by at least one other person.
3. [Travis](#) will build the site using [Sphinx](#), and if there arent any errors, Travis will report success to GitHub on the checks part of the pull request.
4. All your requested reviewers must now approve and comment on your commit before the merge is allowed.
5. Once the PR passes Travis and is approved by another author, feel free to merge to master.
6. **To release the master branch, (build the html, pdf, and latex, and upload the pdf to Pages) youll need to publish a [GitHub](#)**
 - Tag name: 0.1.5
 - Release title: Filtration section maintenance
 - Description: Added filter code from aquaclara 0.2.6. Also updated all broken external links.
7. Travis will rebuild the site and push the html to Pages, and the PDF and LaTeX to GitHub Releases under the tag name.

Important: If your changes to the master branch arent pushing to gh-pages, then check the status of the [Travis build here](#).

50.2.4 Building PDF Locally

To check errors with the PDF generation more quickly, it may be worthwhile to setup your local machine to build the PDF.

Now you should be able to run `./build_pdf` to run the `build_pdf` bash script within the root of the Textbook repo.

If you are on Windows, youll need to use a POSIX shell, such as git bash, Cygwin, etc

50.2.5 Testing Online

To test exactly what will be published, we have a test branch. The test branch is built by Travis and contains all the processed html that Travis produces in `_build/html`. This branch is populated when ANY COMMIT IS PUSHED. Meaning, the last commit to be pushed, if it passes the Travis tests, will be built and the output will be placed in the test branch. Also, if the `PDF=True` environment variable is triggered for a Travis build, the PDF will also be generated and placed in the test branch. To do this, use the Trigger Build option in Travis and put:

```
script:
  - PDF=True
```

The website output is viewable here.

50.2.6 Sharing Test Output

if you want to share what your latest branch developments look like without having whoever is viewing it actually have to build it, you can push a commit, and find the [rawgit URL with this site](#) by entering the URL of the git file within the test branch that youd like to share. Furthermore, if you want to point to the commit so that even if someone else pushes, the URL will still point to the code you intend it to, make sure to include the commit SHA within the rawgit URL like so: <https://rawgit.com/AquaClara/Textbook/e5693e0485702b95e11d4d6bdf76d07c42fdbf99/html/index.html>. That link will never change where it is pointing. To share the PDF output, follow the [testing online](#) instructions to build the PDF, and point to the commit with the PDF. Happy testing!

50.3 Brief Best Practices

When writing RST, there are often many ways to write the same thing. Almost always, the way with the fewest number of characters is the best way. Ideally, never copy and paste.

50.3.1 How do I write RST?

RST is friendly to learn and powerful. There are many useful cheatsheets, like [this one](#) and the next page on this site: Functionality in RST and AquaClara Convention, which is specifically for AquaClara and this textbook project. When you start writing RST, look at the cheat sheets all the time. Use `ctrl-f` all the time to find whatever you need.

Things not covered in most cheat sheets which are of critical importance:

- A document is referred to by its title, as defined between the `*****` signs at the top of the page, **NOT** the filename. So it is critical to have a title.
- Anything else youd like to add for the future

50.3.2 Example to Start From

This file is written in RST. You can start there! Just click on View page source at the top of the page.

Also, the next page contains the convention, and is where we specify all AquaClara RST best practices: Functionality in RST and AquaClara Convention. I recommend looking at the raw RST and the rendered html side by side.

50.4 Converting Markdown to RST

Ideally, use pandoc to do the conversion in the command line: `pandoc --from=markdown --to=rst --output=my_file.rst my_file.md`. Raw html will not be converted (because it is not actually markdown), and tables are converted poorly. You'll need to carefully review any page converted with pandoc.

**CHAPTER
FIFTYONE**

PARAMETER CONVENTION LIST

Table 51.1: Relevant Dimensions

Dimension	Abbreviation	Base Unit
Length	[L]	meter
Mass	[M]	kilogram
Time	[T]	second

If you would like to be able to `ctrl+f` some variables, click on View page source on the top right of this window. If you want to know what a Greek variable is but don't know what it's called, you can view the source text on the file where you found the variable. nu, mu, eta, who actually remembers what these all look like? The letter v should be nu for copyright infringement. Or is it the other way around?

Table 51.2: Parameter Guide

Parameter	Description	Units
Uppercase letters		
A	Area	[L] ²
B	Center-to-Center Spacing Between Two Objects	[L]
C	Concentration	[M]/[L] ³
D	Diameter	[L]
F	Force	[M][L]/[T] ²
G	Velocity Gradient/Fluid Deformation	1/[T]
G _{CS}	Camp Stein velocity gradient	1/[T]
H	Height	[L]
H _e	Height Between Flow Expansions in a Flocculator	[L]
K	Minor Loss coefficient	Dimensionless
L	Length	[L]
M	Fluid Momentum	[M][L]/[T] ²
P	Power	[M][L] ² /[T] ³
Q	Flow rate	[L] ³ /[T]
Re	Reynolds Number	Dimensionless
S	Spacing Between Two Objects	[L]
T	Object Thickness	[L]
V	Volume	[L] ³
W	Width	[L]
Lowercase letters		
f	Darcy-Weisbach friction factor	Dimensionless

Continued on next page

Table 51.2 – continued from previous page

Parameter	Description	Units
g	Acceleration due to Gravity	$\frac{[L]}{[T]^2}$
h	Head, Elevation	$[L]$
h_e	Minor Loss (expansion)	$[L]$
h_f	Major Loss (friction)	$[L]$
h_L	Total headloss	$[L]$
m	Mass	$[M]$
n	Number, Amount	Dimensionless
p	Pressure	$\frac{[M]}{[L][T]^2}$
r	Radius	$[L]$
t	Time	$[T]$
v	Velocity	$\frac{[L]}{[T]}$
z	Elevation	$[L]$
Greek lowercase letters		
ϵ	Surface roughness	$[L]$
ε	Energy Dissipation Rate	$\frac{[L]^2}{[T]^3}$
η_K	Kolmogorov Length Scale	$[L]$
λ_ν	Inner Viscous Length Scale	$[L]$
μ	Dynamic viscosity	$\frac{[M]}{[T][L]}$
ν	Kinematic viscosity	$\frac{[L]^2}{[T]}$
ϕ	Porosity	Dimensionless
ρ	Density	$\frac{[M]}{[L]^3}$
θ	Residence Time	$[T]$
Greek uppercase letters		
Λ	Distance Between Particles	$[L]$
Π	Dimensionless Proportionality Ratio	Dimensionless
$\Pi_{Fractal}$	volume based fractal dimension	Dimensionless
Π_{Error}	Linearity Error Ratio	Dimensionless
$\Pi_{\bar{\varepsilon}}^{\varepsilon_{Max}}$	$\frac{\varepsilon_{Max}}{\bar{\varepsilon}}$ Ratio in a Reactor	Dimensionless
$\Pi_{\bar{G}}^{G_{Max}}$	$\frac{G_{Max}}{\bar{G}}$ Ratio in a Reactor	Dimensionless
Π_{HS}	Height to Baffle Spacing in a Flocculator	Dimensionless
$\Pi_{K\nu}$	Ratio of Inner Viscous Length Scale to Kolmogorov Length Scale	Dimensionless
Π_{vc}	Vena Contracta Area Ratio	Dimensionless

PDF and LaTeX versions¹.

Notes

¹ PDF and LaTeX versions may contain visual oddities because it is generated automatically. The website is the recommended way to read this textbook. Please visit our [GitHub](#) site to submit an issue, contribute, or comment.

BIBLIOGRAPHY

- [BL13] MarkăM. Benjamin and DesmondăF. Lawler. *Water quality engineering: physical / chemical treatment processes*. Wiley, Hoboken, N.J., July 2013. ISBN 978-1-118-16965-0. URL: http://encompass.library.cornell.edu/cgi-bin/checkIP.cgi?access=gateway_standard%26url=http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=631668.
- [BB98] Markus Boller and Stefan Blaser. Particles under stress. *Water Science and Technology*, 37(10):9–29, May 1998. URL: <http://wst.iwaponline.com/content/37/10/9>.
- [DC08] MackenzieăLeo Davis and DavidăA Cornwell. *Introduction to Environmental Engineering*. McGraw-Hill Companies, Dubuque, IA, 4th edition, 2008. ISBN 978-0-07-242411-9 978-0-07-125922-4 978-7-302-15210-1. OCLC: 70708094.
- [GWSL16] Casey Garland, Monroe Weber-Shirk, and LeonardăW. Lion. Revisiting Hydraulic Flocculator Design for Use in Water Treatment Systems with Fluidized Floc Beds. *Environmental Engineering Science*, 34(2):122–129, August 2016. URL: <http://online.liebertpub.com/doi/abs/10.1089/ees.2016.0174>, doi:10.1089/ees.2016.0174.
- [Gra95] RobertăAlan Granger. *Fluid Mechanics*. Dover Publications, New York, 1995. ISBN 978-1-62198-654-6 1-62198-654-3.
- [ML00] G.ăL. McConnachie and J.ăLiu. Design of baffled hydraulic channels for turbulence-induced flocculation. *Water Research*, 34(6):1886–1896, April 2000. URL: <http://www.sciencedirect.com/science/article/pii/S0043135499003292>, doi:10.1016/S0043-1354(99)00329-2.
- [MCM+12] Jasmin Mertens, Barbara Casentini, Armand Masion, Rosemarie Pöthig, Bernhard Wehrli, and Gerhard Furrer. Polyaluminum chloride with high Al₃₀ content as removal agent for arsenic-contaminated well water. *Water Research*, 46(1):53–62, January 2012. URL: <http://www.sciencedirect.com/science/article/pii/S0043135411006294>, doi:10.1016/j.watres.2011.10.031.
- [PCWSL16] WilliamăH. Pennock, FeliceăC. Chan, MonroeăL. Weber-Shirk, and LeonardăW. Lion. Theoretical Foundation and Test Apparatus for an Agent-Based Flocculation Model. *Environmental Engineering Science*, July 2016. URL: <http://online.liebertpub.com/doi/abs/10.1089/ees.2015.0558>, doi:10.1089/ees.2015.0558.
- [SWSL15] Siwei Sun, Monroe Weber-Shirk, and LeonardăW. Lion. Characterization of Flocs and Floc Size Distributions Using Image Analysis. *Environmental Engineering Science*, 33(1):25–34, November 2015. URL: <http://online.liebertpub.com/doi/10.1089/ees.2015.0311>, doi:10.1089/ees.2015.0311.
- [SWSL13] KarenăA. Swetland, MonroeăL. Weber-Shirk, and LeonardăW. Lion. Influence of Polymeric Aluminum Oxyhydroxide Precipitate-Aggregation on Flocculation Performance. *Environmental Engineering Science*, 30(9):536–545, September 2013. URL: <http://online.liebertpub.com/doi/abs/10.1089/ees.2012.0199>, doi:10.1089/ees.2012.0199.
- [SWSL14] KarenăA. Swetland, MonroeăL. Weber-Shirk, and LeonardăW. Lion. Flocculation-Sedimentation Performance Model for Laminar-Flow Hydraulic Flocculation with Polyaluminum Chloride and Aluminum Sulfate Coagulants. *Journal of Environmental Engineering*, 140(3):04014002, March 2014. URL: <http://ascelibrary.org/doi/10.1061/%28ASCE%29EE.1943-7870.0000814>, doi:10.1061/(ASCE)EE.1943-7870.0000814.

[WZL+15] Ning Wei, Zhongguo Zhang, Dan Liu, Yue Wu, Jun Wang, and Qunhui Wang. Coagulation behavior of polyaluminum chloride: Effects of pH and coagulant dosage. *Chinese Journal of Chemical Engineering*, 23(6):1041–1046, June 2015. URL: <http://www.sciencedirect.com/science/article/pii/S1004954115000804>, doi:10.1016/j.cjche.2015.02.003.

[BPMWSCIWSCUWS16] BP-MWS, CIWS, and CUWS. Drinking Water Quality Report 2016. Technical Report, Bolton Point Municipal Water System, City of Ithaca Water System, Cornell University Water System, Ithaca, NY, 2016. URL: https://energyandsustainability.fs.cornell.edu/file/AWQR_2016%20final.pdf.

[Haa98] Johannes Haarhoff. Design of Around-the-End Hydraulic Flocculators. *Journal of Water Supply: Research and Technology - Aqua*, 47(3):142–152, June 1998. URL: <https://doi.org/10.2166/aqua.1998.20>.

[HvdW01] Johannes Haarhoff and JeremiaäJ. vanäder Walt. Towards Optimal Design Parameters for Around-the-End Hydraulic Flocculators. *Journal of Water Supply: Research and Technology - Aqua*, 50(3):149–159, 2001.