Spring 2021

Simulation of Water Loading in Filter Medium

Anthony Mole
atm86@zips.uakron.edu

Simulation of Water Loading in Filter Medium

Anthony Mole

Undergraduate in Chemical Engineering

Honors Research Project

23 April 2021

Abstract

The loading of water droplets into a wire mesh coalescer used for diesel fuel was investigated in an independent research project. FlexPDE was used to simulate water droplets plugging a filter medium by functionally decreasing the permeability in selected areas of the filter. Variables such as the function describing the permeability, the wavelength of the permeability function, the magnitude of the permeability in the drop area, and the geometry of the filter were changed in order to determine their impact on the overall permeability of the filter. The results from the initial simulations are inconclusive, as there is variation in calculated permeability that does not match a specific trend. The variability is hypothesized to be due to the capacity limits associated with the student version of FlexPDE. The code itself will produce results, however a finer grid is needed to be able to encompass the small spatial changes in permeability. More research should be done with the professional version of FlexPDE in the future to refine the grid size and obtain accurate simulations to model the plugged filter.

Introduction

Understanding filter systems is an important aspect of a chemical engineer's job as many applications in the world involve filter systems of some sort. Filters are used to separate or extract unwanted materials from a fluid, and can be solid-gas, solid-liquid, liquid-gas, or liquid-liquid separation. For this study, the liquid-liquid separation of water from diesel was investigated, however the same approach can be applied for other forms of separation because of the way the filter was modeled.

Diesel is usually relatively pure when it is produced in a refinery. However, small water droplets from atmospheric condensation or precipitation can contaminate the fuel during transfer between tanks and when transported in tanks. If contaminated fuel is injected into diesel engines untreated, the water droplets can cause the injector ports to fail and even explode. Therefore, the diesel fuel is treated to remove the water. The most common methods of treatment include coalescing filters and gravity settling, or by adsorbent media. The reduction of water in the fuel significantly reduces the probability of water droplet damage to engine components.

The filter performance itself can depend on the amount of water in the diesel fuel. As more and more water droplets are absorbed by the filter, they begin to plug up the pores of the filter, ultimately effecting the overall permeability and effectiveness of the filter. This phenomenon is known as water loading.

The finite element analysis software, FlexPDE, was used to perform the water loading simulation. FlexPDE solves systems of a set of non-linear continuous partial differential equations by dividing the volume into nodes and meshes, integrating the partial differential equations into algebraic ones, and solving the system for each node. It is important to make sure the solutions are reasonable at each step of the process, so FlexPDE performs error calculations to drive the solutions to converge at each mesh node. This software is powerful at giving accurate solutions for fluid flow and more.

Equations

The equations that describe fluid flow can be found by performing a conservation of mass and momentum analysis. When doing this, you obtain the mass continuity equation and the momentum balance equation shown below.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{1}$$

$$\rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} \right) = -\nabla P + \mu \nabla^2 \boldsymbol{v} + \rho \boldsymbol{g} \tag{2}$$

Where ρ is the fluid density, v is the fluid velocity, P is the pressure, g is gravity, and μ is the viscosity of the fluid. The momentum balance equation shown is known as Navier Stoke equation and assumes that the fluid is incompressible (i.e. the density, ρ, is constant) and behaves like a Newtonian fluid (i.e. the viscosity, μ, is constant). This equation also only applies to laminar flow situations, so the Reynolds number needs to be monitored throughout the process to make sure the flow is laminar. These assumptions are justified for a diesel fuel filter system, as the velocity through the filters is usually small.

The relationship that models flow through a filter was developed by Darcy and is known as Darcy's law. The equation relates the flow rate and viscosity of a fluid through a filter area to the pressure drop across the length of the filter and is shown below.

$$\frac{Q}{A} = \frac{k}{\mu} \frac{\Delta P}{L} \tag{3}$$

Where Q is the volumetric flow rate, A is the cross-sectional area of the filter, μ is the viscosity, ΔP is the pressure drop across the filter, L is the thickness of the filter, and k is the permeability of the filter and has the units of m^2.

When modeling, these equations will be expanded out to encompass the spatial dimensions that best fit the problem, cartesian or cylindrical. However, an additional equation is needed to relate pressure to the velocity in FlexPDE. This is done by loosely relating the pressure of the fluid to the density of the fluid at a given point in space. Assuming a linear relationship to density and applying the continuity equation leads to the following expressions.

$$P = P_0 + C * (\rho - \rho_0) \tag{4}$$

$$\frac{dP}{dt} = -C\rho_0 (\nabla \cdot \boldsymbol{v}) \tag{5}$$

Where P_0, ρ_0, and C are constants. In FlexPDE, the time derivative of a variable can be approximated with the negative Laplacian of that variable. Using this identity results in the following equations.

$$\frac{dP}{dt} = -\nabla^2 P \tag{6}$$

$$\nabla^2 P = M(\nabla \cdot \boldsymbol{v}) \tag{7}$$

Where M is a constant, commonly referred to as a penalty factor. In the simulations, a penalty factor of 10,000 was used.

Methodology

A method needs to be introduced on how to model the water droplets plugging the filter. When the water droplet plugs a pore of the filter, no fluid can pass through that pore and must go around it. Many pores in a small region will become plugged at the same time and act as one relatively larger pore being plugged by a water droplet with diameter, d_p. Models of pore scale plugging can be created in FlexPDE, however the intense curvature in these regions would force FlexPDE to create ever smaller nodes and meshes to model the flow around and through these spaces. The speed and memory of the computer would limit such models to small volumes of filter media, and the practical usefulness of the results would be limited.

In this study, a different approach was applied to smooth out the phenomenon of water plugging in the pores over a larger region of filter media instead of at the scale of the individual pores. In this approach, the permeability of the filter was modeled to vary in the spatial dimensions using a sinusoidal function where the peaks and troughs of the curve represent regions that are plugged or not plugged with water drops. While this approximation is not exactly what physically happens at the individual pore scale, it gives insight as to how the droplets effect the pressure drop across the filter system.

To generate many localized regions with variable permeability throughout the filter, a sinusoidal function was used in each coordinate direction. Different sinusoidal functionalities were investigated to represent the effect of water on the permeability in different ways. The general form of each function is shown below for the variation in the x-direction along with a plot of how the permeability changes in the x-direction. Similar functions were applied for the y and z components.

$$f_{1x} = sin^2(\frac{2\pi}{\lambda_x}x) \tag{8}$$

$$f_{2x} = (1-(1-sin^2(\frac{2\pi}{\lambda_x}x))^2)^2 \tag{9}$$

$$f_{3x} = (1-(1-sin^4(\frac{2\pi}{\lambda_x}x))^4)^4 \tag{10}$$

Where λ_x is the wavelength in the x direction and x is the spatial variable. When modeling, the permeability of the filter will be described as with these functions in all three dimensions, essentially distributing water droplets throughout the filter.

$$k_{perm} = k_0 * (1 - f_{ix} f_{iy} f_{iz}) + k_{min} \tag{11}$$

Where subscript i represents the function type (1,2, or 3) corresponding to Eqs. (8), (9), or (10). k_{perm} is the permeability of the filter at a local (x,y,z) position used in calculating the local fluid flow. k_0 is a constant permeability, and k_{min} is the permeability of the filter when it is fully loaded with water. Filters generally do not become completely plugged, so the k_{min} in Eq. (4) ensures the permeability does not go to zero. The dry permeability of the filter is therefore $k_{max} = k_0 + k_{min}$. In the simulations, k_0 is set at 10^{-8} and k_{min} is varied. The wavelength of the sinusoidal function can roughly be related to the water droplet size. The water droplet is assumed to have a position where the value of the sinusoidal function is less than 0.5. For f1 this occurs when $x/\lambda_x = 0.125$, for f2 this occurs at $x/\lambda_x = 0.1184$, and for f3 this occurs at $x/\lambda_x = 0.1422$. This corresponds to water droplet sizes of $d_p = 0.25\lambda_x$ for f1, $d_p = 0.2632\lambda_x$ for f2, and $d_p = 0.2156\lambda_x$ for f3. There are 2 water droplets per wavelength because of the $\sin^{2n}$ functionality. One half of a wave can be seen in Figure 1 for each functionality.

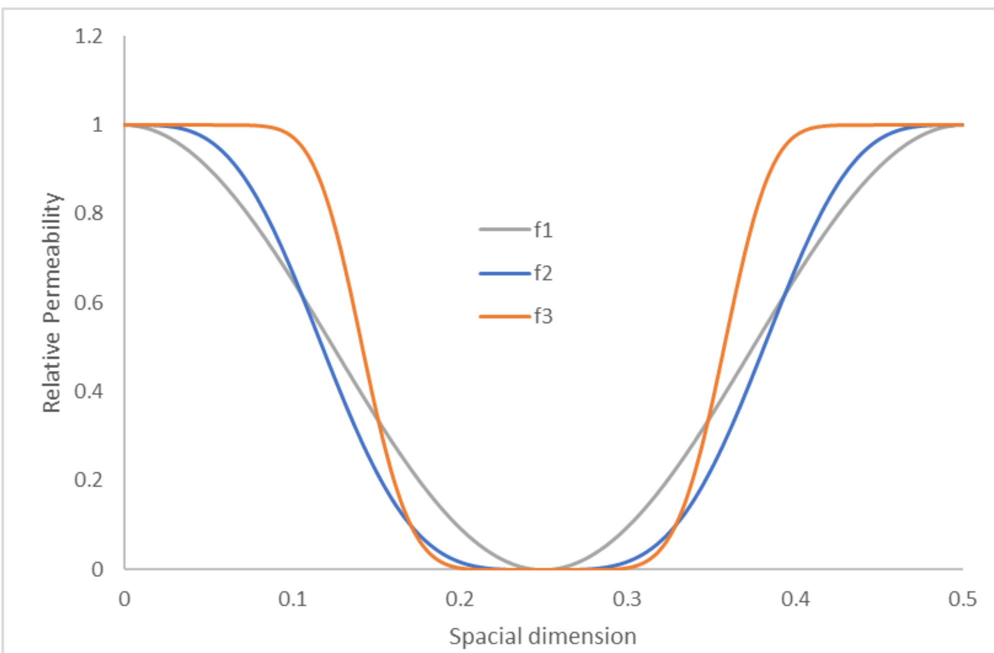

Figure 1. *Three sinusoidal functionalities used to represent a single water droplet plugging the filter over half the wavelength. In this case, λ = 1. The 0.5 values occur at 0.125 for f1, 0.1184 for f2, and 0.1422 for f3.*

The effective permeability of the filter can be found in two ways. First, the volume integral average can be taken over the filter to get an average permeability. Secondly, Darcy's Law can be used by setting a flow rate in FlexPDE and observing the pressure drop across the filter to calculate the permeability. For simple geometries and permeability functions, both methods will be used to further verify FlexPDE simulations. For more complex geometries and permeability functions, the volume integral average method is not trivial, and the results will only rely on the FlexPDE simulations.

Models

When simulating the pressure drop across a filter for a flowing fluid, it is important to have open channels on either side of the filter so that large variations at the filter surface are not used as the value for pressure. Three filter models were created using the open channel filter system idea in different geometries and configurations.

Model 1 represents the fluid flowing in a square duct with a filter in the middle. The fluid enters on the left side and exits on the right. There are no slip boundary conditions along the sides of the duct and the outlet pressure is set to 1 bar.

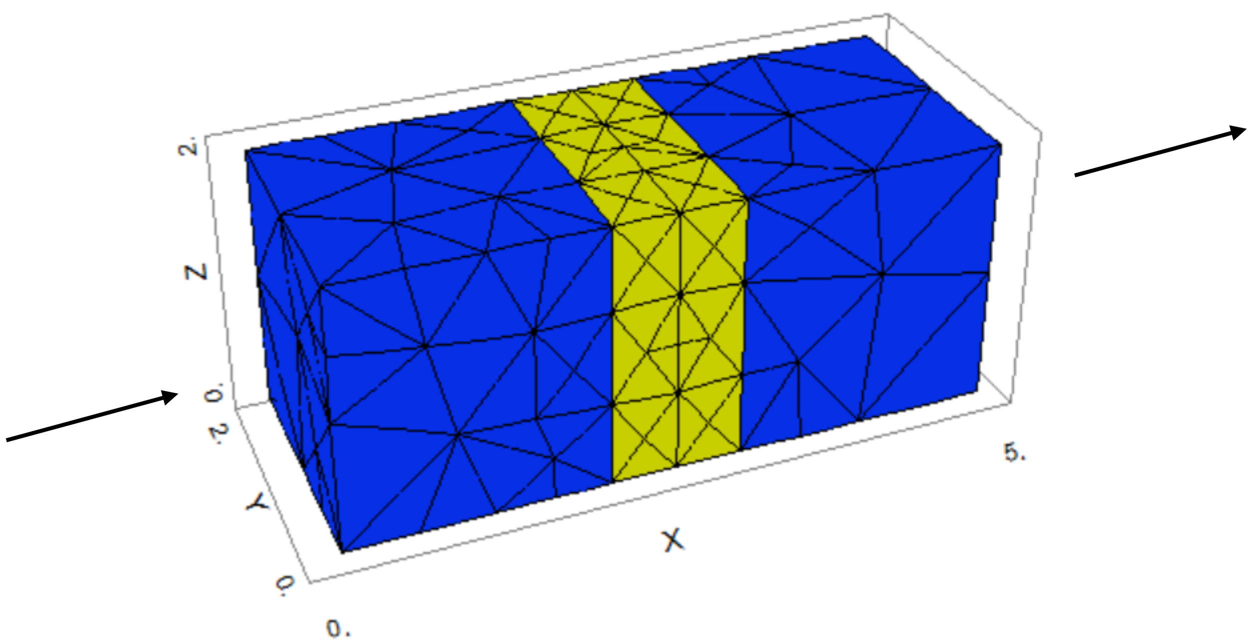

Figure 2. *Illustration of Model 1. The model is a square duct with a width of 2 and a length of 5 with the filter located in the center of the duct represented by the yellow section. The thickness of the filter is 1. The arrows indicate the direction of flow.*

Model 2 represents fluid flowing in a pipe with a filter in the middle. The fluid enters through the bottom and exits through the top. There are no slip boundary conditions on the side of the pipe and the outlet pressure is set to 1 bar. The functionality of the permeability is the same for the first two models.

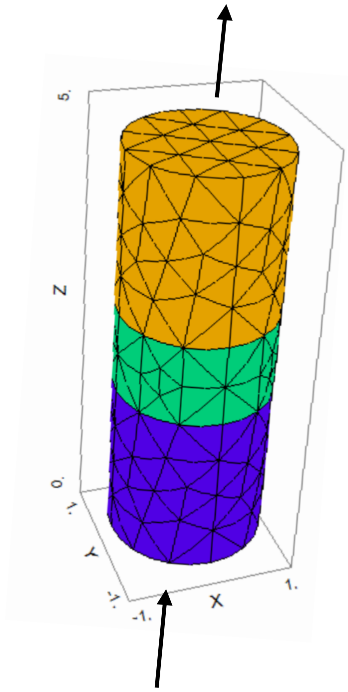

***Figure 3**. Illustration of Model 2. The model has a diameter of 2 and a length of 5 with filter location in the center of the pipe represented by the green section. The thickness of the filter is 1. The arrows indicate the direction of the flow.*

Model 3 is similar to the first but with a different filter configuration. Instead of one filter, there are three variable filters. Each variable filter has stripes of filter where water loading can occur and stripes of filter where no water loading will occur. The first and third filters are oriented such that the stripes are vertical while the second filter is oriented such that the stripes are horizontal. This pattern results in a unique flow path that could reduce pressure drop while maintaining a larger flow.

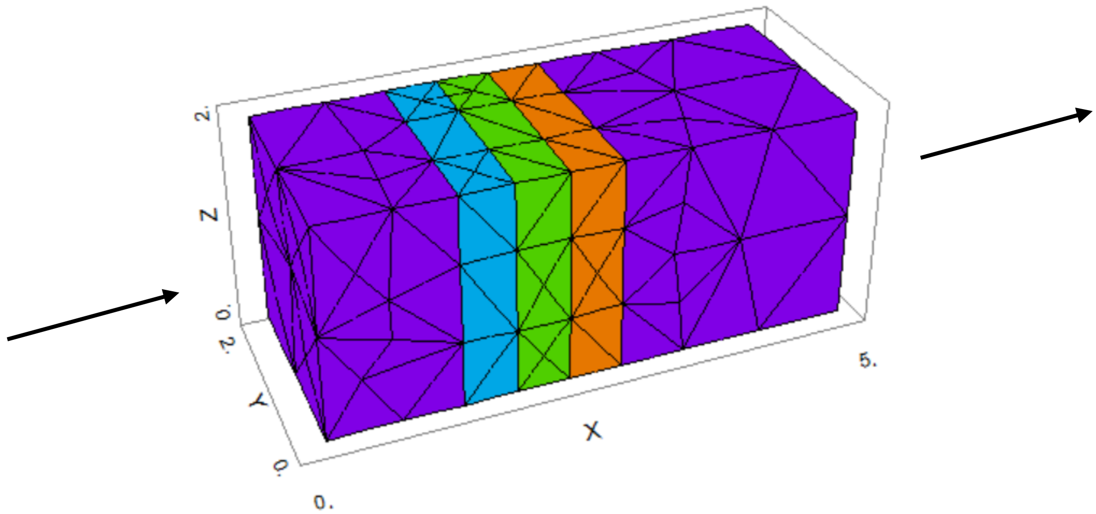

Figure 4. *Illustration of Model 3. The model is a square duct with a width of 2 and a length of 5. The filter is made up of 3 regions, the first and third oriented 90 degrees from the second.*

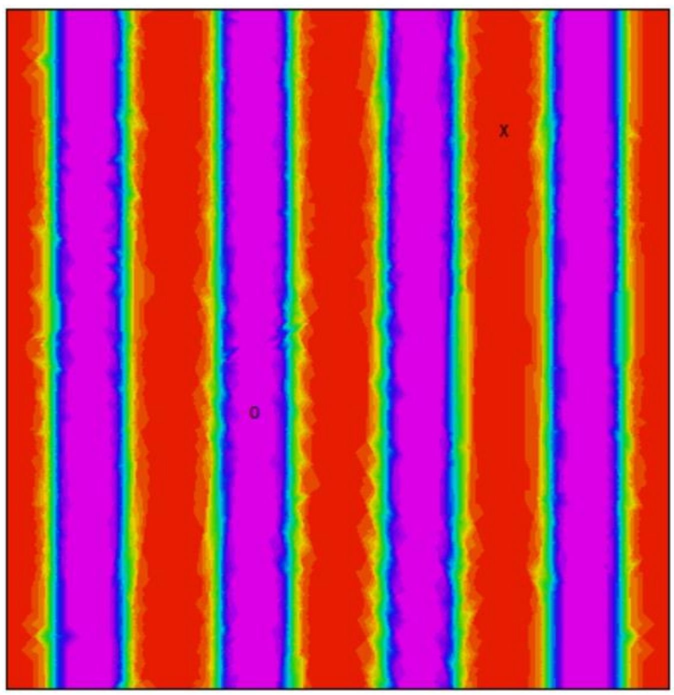

Figure 5. *Stripes of permeable filter used in Model 3. The red regions indicate where the filter has a maximum permeability while the purple regions indicate where the filter has a minimum permeability. Functionality 3 was used to generate the stripes with a wavelength of 1.*

The student version of FlexPDE has node and mesh limitations, making the third model difficult to model by using sinusoidal functions for drops because of all the new regions that would need to be created. It is currently modeled with just one lower permeability for the stripes that is meant to represent all the droplets plugging the filter in that stripe. Therefore, the functionality and diameter of the water droplets cannot be observed with this model. This model will still be ran and compared with a filter with no water loading, a constant permeability.

Each simulation has a specific label corresponding to the parameters of the simulation. The model being used is represented by MX where X is the model number, either 1, 2, or 3. The function that is being used to describe a spatial dimension permeability is labeled fX where X is the function number. 1 corresponds to f1, 2 corresponds to f2, and 3 corresponds to f3. The magnitude of the permeability reduction caused by the water droplet is represented by mX where X is magnitude number. 1 corresponds to a k_{min} of 10^{-10} and 2 corresponds to a k_{min} of 10^{-9}. Finally, the wavelength of the sinusoidal functions is represented by lX where X is the wavelength. 1 corresponds to a λ_x of 0.4, 2 corresponds to a λ_x of 0.5, and 3 corresponds to a λ_x of 0.667. On top of the variations, a control simulation will be ran for each model that contains a filter of the same thickness and constant maximum permeabilities corresponding to no water loading at the two different k_{min} values. The variation of parameters leads to a total of 20 simulations for each of the first 2 models and 4 simulations for Model 3. Source code for an example of each model can be found in the Appendix with highlights that show which parameters are changing in the code.

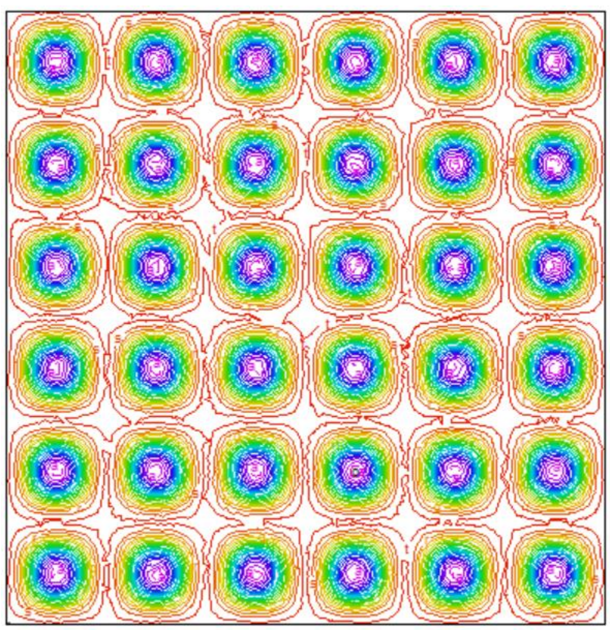

Figure 6. *Permeability cross section of M1f1m1l3. Model 1 with functionality 1, wavelength 3, and magnitude 1. Corresponds to a wavelength of 0.667 and a* k_{min} *of* 10^{-10}*. The blue/purple regions are areas of decreased permeability and represent a water droplet plugging the filter.*

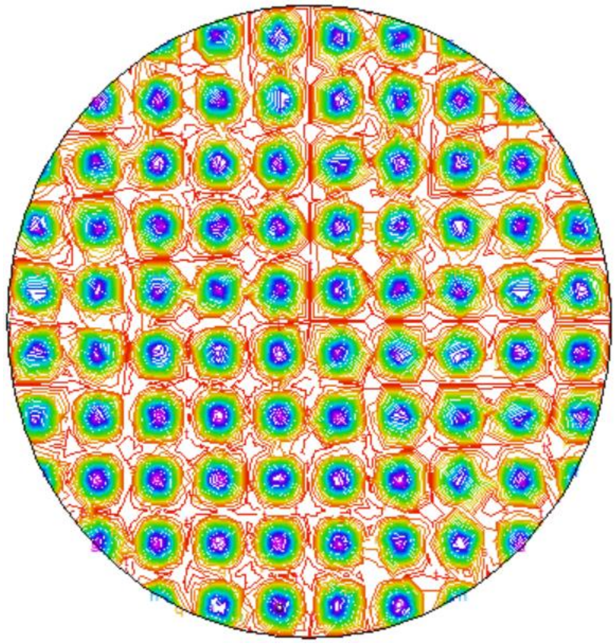

Figure 7. *Permeability cross section of M2f1m1l1. Model 2 with functionality 1, wavelength 1, and magnitude 1. Corresponds to a wavelength of 0.4 and a k_{min} of 10^{-10}. The blue/purple regions are areas of decreased permeability and represent a water droplet plugging the filter.*

Results and Discussion

After the simulations were performed, the data pertaining to the permeability of the filter, such as the flow rate and pressure drop, were recorded in Tables 1, 2, and 3. The k_{eff} (vol_int) was found by taking the volume average of the permeability function across the entire filter. The k_{eff} (Darcy) was calculated by using the pressure drop and flow rate obtained in the simulation to solve for the permeability through Darcy's Law. The data was then converted into figures, with each figure belong to a specific model and permeability functionality, shown in Figures 8-13.

The figures are meant to show the relationship between the permeability of the filters and the size and distribution of the water droplets. To do this, the parameters were put into dimensionless form. The effective permeability of the filter was divided by the maximum permeability so that the scale would run from 0 to 1. The drop diameter, d_p, which was calculated for each functionality, was divided by the dimension of the channel, D. For Model 1, the dimension of the channel is the width and for Model 2, the dimension of the channel is the diameter. For both models, the dimension D is 2.

Dimensionless parameters for constant values were chosen. The dimensionless density and viscosity of the system were set to 10 and 0.3, respectively. The penalty factor chosen for the pressure equation was 10,000. The constant permeability k_0 was set to 10^{-9}. The starting inlet

dimensionless velocity was set to 0.001. These values were manipulated until the resulting flow field gave laminar flow with reasonable pressure drop across the filter.

Table 1. *List of results for simulations of Model 1 while adjusting the functionality of the permeability function, the magnitude of* k_{min} *and the wavelength of the permeability function. Magnitude m1 corresponds to a* k_{min} *of* 10^{-10} *while magnitude m2 corresponds to a* k_{min} *of* 10^{-9}*. Wavelength L1 corresponds to a wavelength of 0.4, wavelength L2 corresponds to a wavelength of 0.5, and wavelength L3 corresponds to a wavelength of 0.667. The control simulations were simply filters with a constant permeability of* k_{max}*.*

Function	Magnitude	Wavelength	RMS error	deltaP	keff (vol_int)	keff (Darcy)	keff/kmax
1	1	1	0.011	30344	8.84E-09	7.86E-09	0.78
1	1	2	0.009	27129	8.90E-09	8.79E-09	0.87
1	1	3	0.007	29711	8.89E-09	8.03E-09	0.79
1	2	1	0.012	26940	9.75E-09	8.85E-09	0.80
1	2	2	0.01	25302	9.79E-09	9.43E-09	0.86
1	2	3	0.009	26576	9.79E-09	8.97E-09	0.82
2	1	1	0.019	31073	8.61E-09	7.68E-09	0.76
2	1	2	0.013	24971	8.74E-09	9.55E-09	0.95
2	1	3	0.011	28871	8.71E-09	8.26E-09	0.82
2	2	1	0.015	29976	9.56E-09	7.96E-09	0.72
2	2	2	0.009	23463	9.64E-09	1.02E-08	0.92
2	2	3	0.007	26305	9.62E-09	9.07E-09	0.82
3	1	1	0.012	29743	9.35E-09	8.02E-09	0.79
3	1	2	0.012	24754	9.35E-09	9.63E-09	0.95
3	1	3	0.007	29763	9.38E-09	8.01E-09	0.79
3	2	1	0.01	27379	1.024E-08	8.711E-09	0.79
3	2	2	0.016	23240	1.025E-08	1.026E-08	0.93
3	2	3	0.006	26898	1.033E-08	8.867E-09	0.81
control	1	control	0.008	29335	1.01E-08	8.130E-09	0.80
control	2	control	0.008	26934	1.10E-08	8.855E-09	0.80

Table 2. *List of results for simulations of Model 2 while adjusting the functionality of the permeability function, the magnitude of k_{min} and the wavelength of the permeability function. Magnitude m1 corresponds to a k_{min} of 10^{-10} while magnitude m2 corresponds to a k_{min} of 10^{-9}. Wavelength L1 corresponds to a wavelength of 0.4, wavelength L2 corresponds to a wavelength of 0.5, and wavelength L3 corresponds to a wavelength of 0.667. The control simulations were simply filters with a constant permeability of k_{max}.*

Function	Magnitude	Wavelength	RMS error	deltaP	keff (vol_int)	keff (Darcy)	keff/kmax
1	1	1	0.274	45134	8.85E-09	6.64E-09	0.66
1	1	2	0.423	49025	8.92E-09	6.12E-09	0.61
1	1	3	0.352	40244	8.85E-09	7.45E-09	0.74
1	2	1	0.212	39721	9.76E-09	7.55E-09	0.69
1	2	2	0.332	41376	9.74E-09	7.25E-09	0.66
1	2	3	0.358	36724	9.73E-09	8.16E-09	0.74
2	1	1	0.507	53069	8.63E-09	5.65E-09	0.56
2	1	2	0.34	49492	8.89E-09	6.06E-09	0.60
2	1	3	0.491	38787	8.69E-09	7.73E-09	0.77
2	2	1	0.525	45588	9.59E-09	6.58E-09	0.60
2	2	2	0.43	43701	9.77E-09	6.86E-09	0.62
2	2	3	0.388	36238	9.57E-09	8.27E-09	0.75
3	1	1	0.633	49476	9.35E-09	6.06E-09	0.60
3	1	2	0.459	47922	9.35E-09	6.26E-09	0.62
3	1	3	0.718	37244	9.27E-09	8.05E-09	0.80
3	2	1	0.265	41410	1.024E-08	7.24E-09	0.66
3	2	2	0.434	42249	1.028E-08	7.10E-09	0.65
3	2	3	0.458	34987	1.011E-08	8.57E-09	0.78
control	1	control	0.161	38337	1.01E-08	7.82E-09	0.77
control	2	control	0.165	35201	1.10E-08	8.52E-09	0.77

Table 3. *List of results for simulations of Model 3 while adjusting the magnitude of k_{min}. Magnitude m1 corresponds to a k_{min} of 10^{-10} while magnitude m2 corresponds to a k_{min} of 10^{-9}. The control simulations were simply filters with a constant permeability of k_{max}.*

Function	Magnitude	RMS error	deltaP	keff (vol_int)	keff (Darcy)	keff/kmax
3	1	0.023	171458	5.83E-09	2.09E-09	0.21
3	2	0.008	83859	6.73E-09	4.27E-09	0.39
control	1	0.002	45773	1.01E-08	7.82E-09	0.77
control	2	0.002	42028	1.10E-08	8.51E-09	0.77

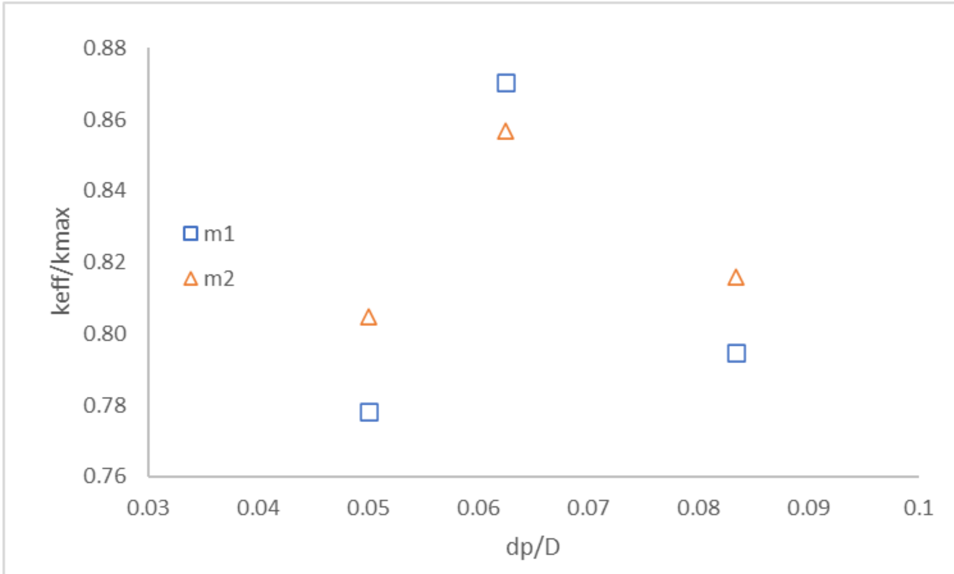

Figure 8. *Relationship between k_{eff}/k_{max} and d_p/D for Model 1 with functionality 1 at different magnitudes of k_{min} based on the results from the simulations. The magnitude m1 refers to a k_{min} value of 10^{-10} and m2 refers to a k_{min} value of 10^{-9}. The greatest permeability ratio is seen around a d_p/D value of 0.062.*

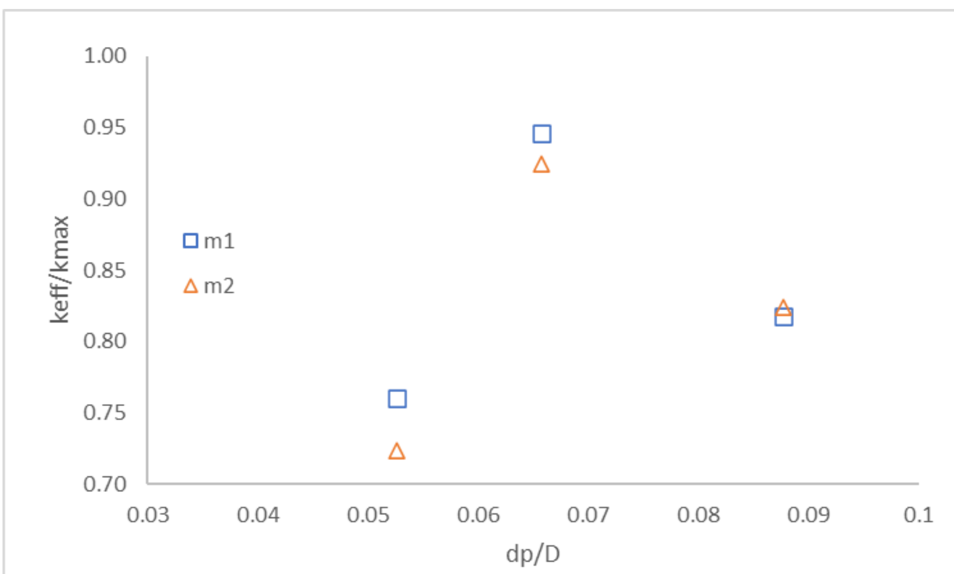

Figure 9. *Relationship between k_{eff}/k_{max} and d_p/D for Model 1 with functionality 2 at different magnitudes of k_{min} based on the results from the simulations. The magnitude m1 refers to a k_{min} value of 10^{-10} and m2 refers to a k_{min} value of 10^{-9}. The greatest permeability ratio is seen around a d_p/D value of 0.066.*

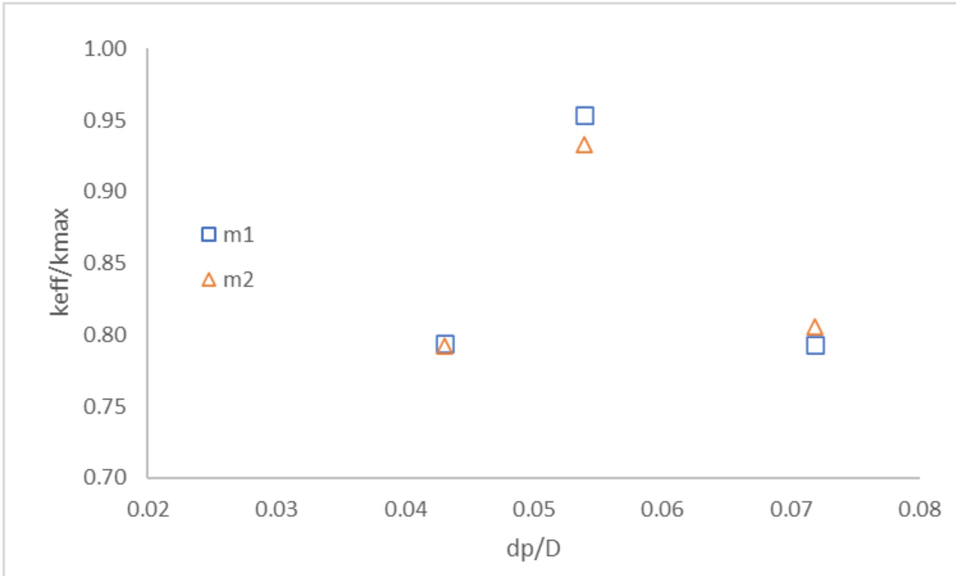

Figure 10. *Relationship between k_{eff}/k_{max} and d_p/D for Model 1 with functionality 3 at different magnitudes of k_{min} based on the results from the simulations. The magnitude m1 refers to a k_{min} value of 10^{-10} and m2 refers to a k_{min} value of 10^{-9}. The greatest permeability ratio is seen around a d_p/D value of 0.054.*

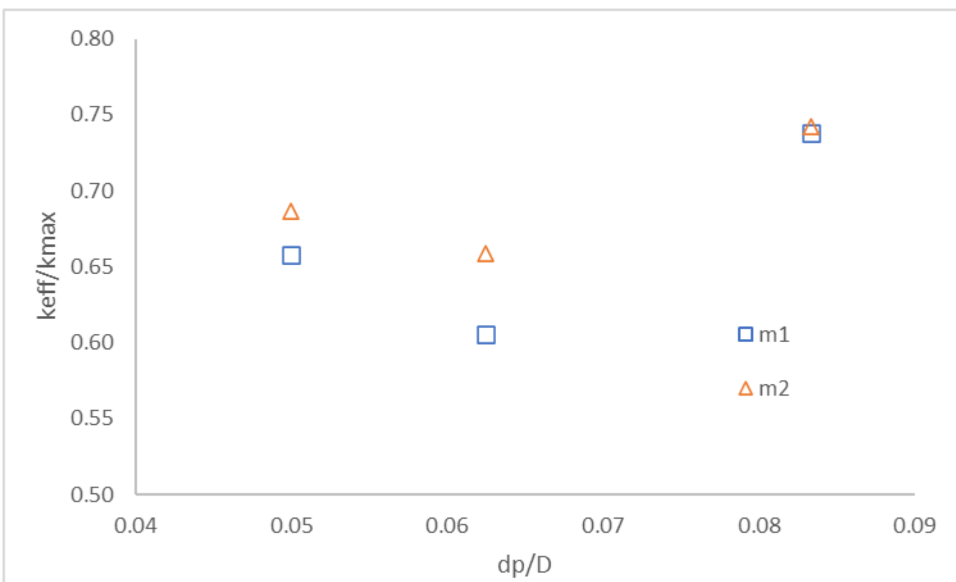

Figure 11. *Relationship between k_{eff}/k_{max} and d_p/D for Model 2 with functionality 1 at different magnitudes of k_{min} based on the results from the simulations. The magnitude m1 refers to a k_{min} value of 10^{-10} and m2 refers to a k_{min} value of 10^{-9}. The greatest permeability ratio is seen around a d_p/D value of 0.083.*

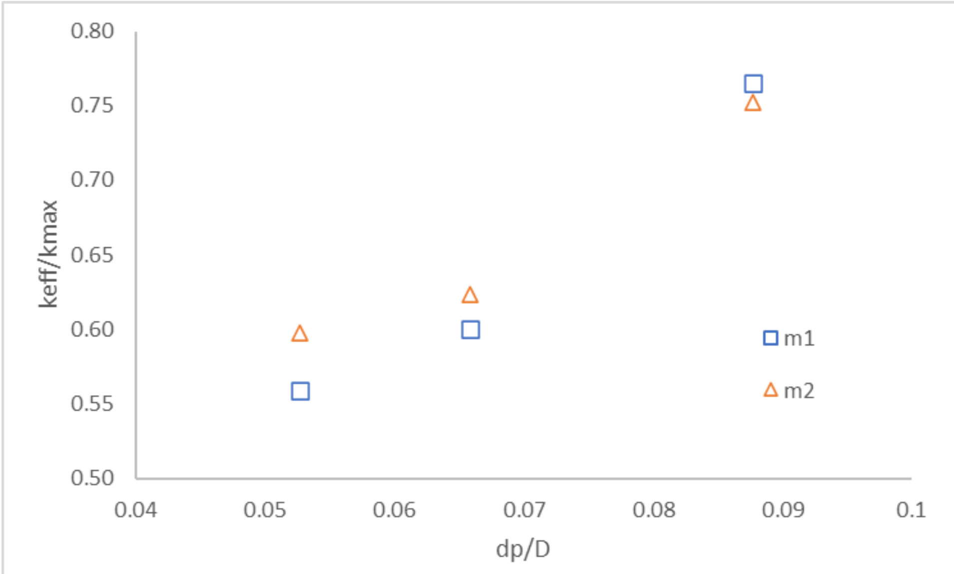

Figure 12. *Relationship between k_{eff}/k_{max} and d_p/D for Model 2 with functionality 2 at different magnitudes of k_{min} based on the results from the simulations. The magnitude m1 refers to a k_{min} value of 10^{-10} and m2 refers to a k_{min} value of 10^{-9}. The greatest permeability ratio is seen around a d_p/D value of 0.088.*

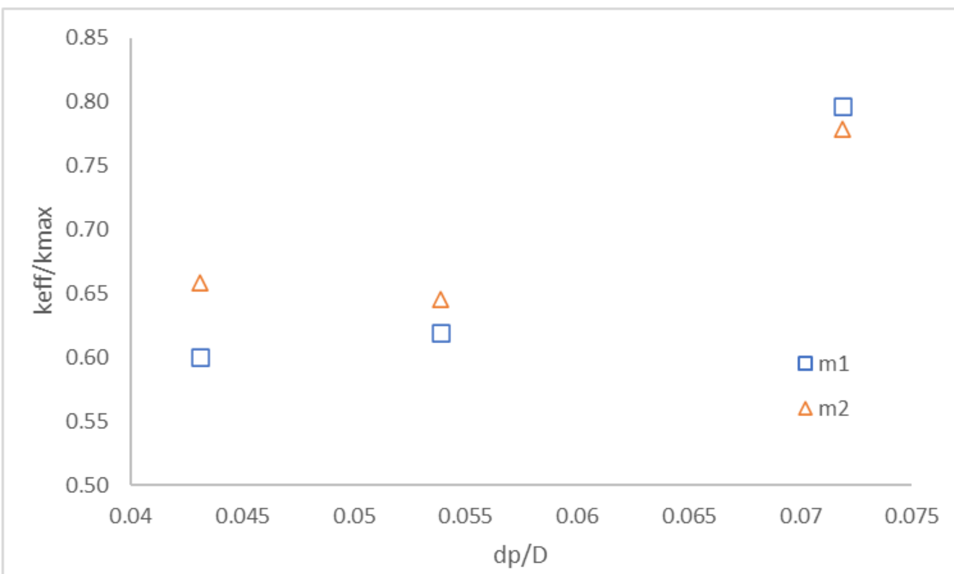

Figure 13. *Relationship between k_{eff}/k_{max} and d_p/D for Model 2 with functionality 3 at different magnitudes of k_{min} based on the results from the simulations. The magnitude m1 refers to a k_{min} value of 10^{-10} and m2 refers to a k_{min} value of 10^{-9}. The greatest permeability ratio is seen around a d_p/D value of 0.072.*

The results from the simulations do not show much of a trend and often times the calculated value of the permeability of the filter does not match the volume integral average value. The simulations also showed a lot of variability in the permeability of the filter when different diameters of water droplets were simulated in it. The variation and error could be due to the limitations of the student version of FlexPDE that was used to perform the simulations. Even though the results do not show any trend, it can be seen from the figures that the effective permeability decreases when the filter is functionally plugged with water droplets.

The most concerning result from the simulations is that the volume integral permeabilities do not match the Darcy's Law permeabilities. In principle these values should be the same, but the lack of a finer grid size and limitations in computer power have given pressure drop values that do not represent the true permeability of the filter systems. The discrepancy could also be due to the constant parameters chosen for the simulation.

One important observation about the fluid flow is that the equations only work in laminar flow situations. Therefore, the Reynolds number was monitored in each model and the value stayed low enough where turbulent flow would not happen. The laminar flow behavior is also evidenced by the parabolic flow field of the open channels, shown below.

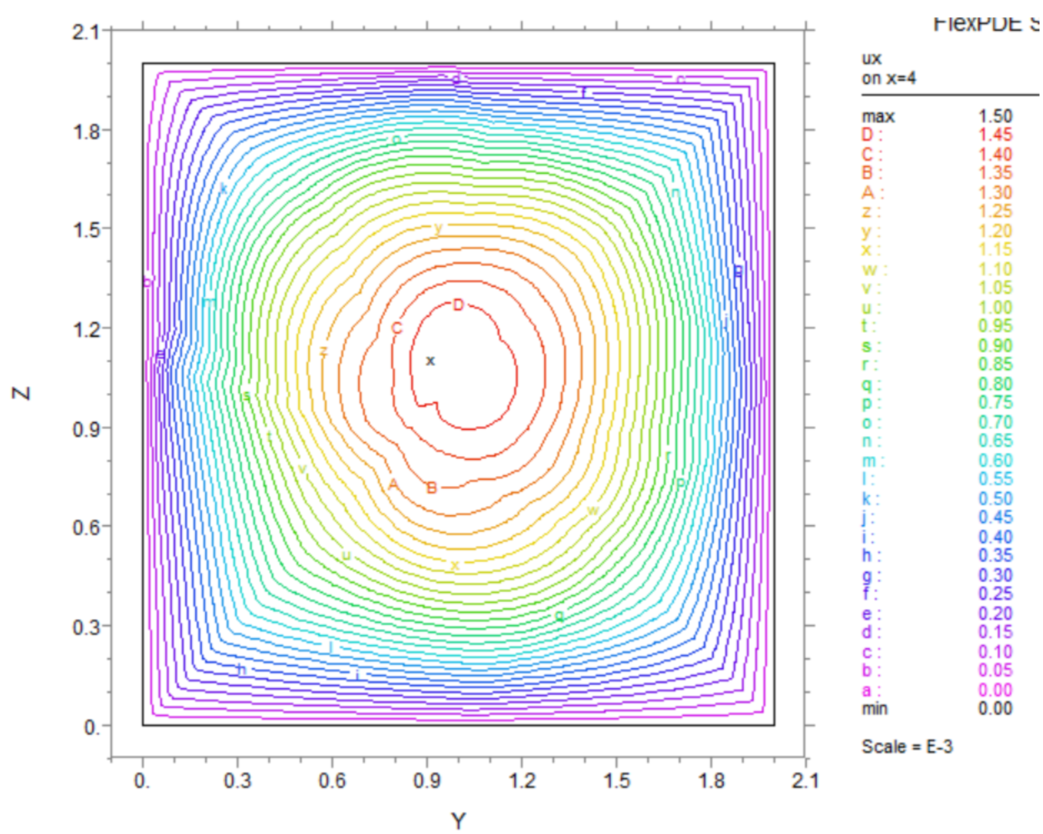

Figure 14. *Parabolic flow field in Model 1. The x-component velocity is taken at the cross-section x=4, after the filter. The red lines indicate the maximum velocity is in the center of the duct and the purple lines show the no slip boundaries on the sides.*

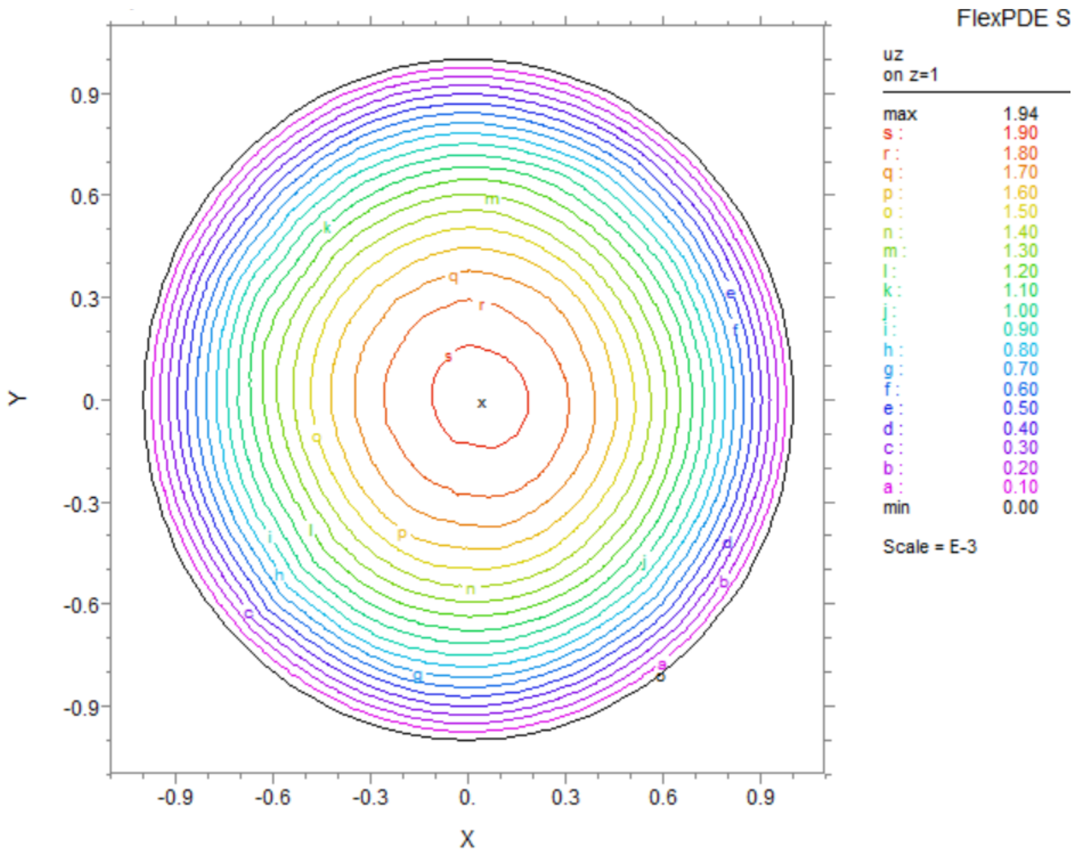

Figure 15. *Parabolic flow field in Model 2. The z-component velocity is taken at the cross-section z=1, before the filter. The red lines indicate the maximum velocity is in the center of the pipe and the purple lines show no slip boundaries on the sides.*

The simulations were able to predict and model fluid flow on a larger scale through the channels, evidenced by the parabolic flow field. However, at the relatively smaller scale of the plugged pores, the simulations failed to give precise values that represent the true nature of water loaded filters.

Conclusions and Recommendations

The loading of water droplets into a filter medium was investigated. Models were generated using FlexPDE in three different configurations. The first model was a single layer of filter medium in a rectangular geometry, the second model was a single layer of filter medium in a cylindrical geometry, and the third was a triple layer of filter media in a rectangular geometry.

The permeabilities of the single layer models were modeled to have spatially varying values using three sinusoidal functionalities. The peaks of the sinusoidal waves correspond to a lower permeability of the filter in that area to simulate the presence of water droplets. The magnitude of the peaks was altered to observe the effect on the overall permeability. The wavelengths of the functions were also changed to explore different amounts and distributions of water loading. The permeability of the triple layer model was arranged as spatially varying stripes. The magnitude of the permeability decrease in the stripe was altered to observe the effect on the overall permeability.

The data from these simulations were converted to dimensionless numbers and graphed to determine the effect that the size and distribution of water droplets had on the effective permeability of the filter in each geometry. The results were inconclusive and showed significant variation in the pressure drop of the system and the effective permeability of the filter.

The simulations were conducted using the student version of FlexPDE. There are limitations on the number of nodes and cells allowed in a single run. This means that the small changes in permeability seen in many of the models is not captured by the simulation. Therefore, future work should consider using a professional version of FlexPDE to obtain better convergence. Example code for each model is provided in the Appendix for future work.

References

Li, Yalong. *Solutions of Potential Fields Using Flexible Finite Element Methods with Applications in Flow Through Porous Media and Electrospinning*. May 2017.

McCabe and Smith. *Unit Operations of Chemical Engineering*. 7th Edition. 2005

Appendices

Appendix A: Example source code for Model 1. Simulation M1f2m1l3

```
TITLE 'Flow through duct with variable filter'     { the problem identification }
COORDINATES cartesian3  { coordinate system, 1D,2D,3D, etc }
VARIABLES        { system variables }
  ux(.1) uy(.1) uz(.1) p(.1)          { choose your own names }
SELECT         { method controls }
!Ngrid = 5
DEFINITIONS    { parameter definitions }
L = 5
Wid = 2
H = 2
filterloc = 2.5
filterwid = 1

dens = 10
visc = .3
k0 = 1e-8
kmin = 1e-10
lamday = 2/3
lamdaz = 2/3
lamdax = 2/3
kperm = k0*(1-    (1-(1-(sin(2*pi*y/lamday))^2)^2)^2     *     (1-(1-(sin(2*pi*z/lamdaz))^2)^2)^2
*     (1-(1-(sin(2*pi*x/lamdax))^2)^2)^2     )+kmin
keff = vol_integral(kperm,'filter')/(H*Wid*filterwid)
p0 = 100000
Pin = surf_integral(P,'start')/(H*Wid)
```

```
Pout = surf_integral(P,'end')/(H*Wid)
deltaP = Pin-Pout
M = 10000
fluid_part = 1

speed = sqrt(ux^2+uy+uz^2)

Qin = surf_integral(ux, 'start')
Qout = surf_integral(ux, 'end')

Va = integral(ux)/integral(1)
Vmax = globalmax(ux)

Re = Va * Wid * dens / visc
Remax = Vmax * Wid * dens / visc

INITIAL VALUES
ux = .001 uy = 0 uz = 0 p = 100000
EQUATIONS        { PDE's, one for each variable }
  ux:  dx(P) - visc*(div(grad(ux)))+dens*(ux*dx(ux)+uy*dy(ux)+uz*dz(ux)) + (1-fluid_part)*ux*visc/kperm= 0 {navier stokes in 3D cartesian, SS}
  uy:  dy(P) - visc*(div(grad(uy)))+dens*(ux*dx(uy)+uy*dy(uy)+uz*dz(uy))  +(1-fluid_part)*uy*visc/kperm= 0  {no graivty term, incompressible}
  uz:  dz(P) - visc*(div(grad(uz)))+dens*(ux*dx(uz)+uy*dy(uz)+uz*dz(uz))  +(1-fluid_part)*uz*visc/kperm= 0
  p:  fluid_part*(div(grad(p)) - M*(dx(ux)+dy(uy)+dz(uz))) + (1-fluid_part)*(div(kperm*grad(p))) = 0

! CONSTRAINTS    { Integral constraints }
```

```
Extrusion
surface 'bottom' z=0
layer 'everything'
surface 'top' z=H

BOUNDARIES       { The domain definition }

  REGION 'domain'  1     { For each material region }
   surface 'bottom' value(ux) = 0 value(uy) = 0 value(uz) = 0
   surface 'top' value(ux) = 0 value(uy) = 0 value(uz) = 0
  START(0,0)   { Walk the domain boundary }
    value(uy) = 0 value(ux) = 0 value(uz) = 0
    line to (L,0)
    load(uy) = 0 load(ux) = 0 load(uz) = 0 value(p) = p0 line to (L,Wid)
    value(uy) = 0 value(ux) = 0 value(uz) = 0 load(p) = 0  line to (0,Wid)
    load(uy) = 0 value(ux) = 0.001 load(uz) = 0 line to close

 Region 'filter' 2
  fluid_part = 0
   surface 'bottom' value(ux) = 0 value(uy) = 0 value(uz) = 0
   surface 'top' value(ux) = 0 value(uy) = 0 value(uz) = 0
    start (filterloc - filterwid/2,0)
    line to (filterloc+filterwid/2,0)
    line to (filterloc+filterwid/2,Wid)
    line to (filterloc-filterwid/2,Wid)
    line to close

feature 'start' start(0,0) line to (0,Wid)
```

```
feature 'filter cross section' start (filterloc,0) line to (filterloc,Wid)
feature 'end' start (L,0) line to (L,Wid)

! TIME 0 TO 1    { if time dependent }
MONITORS         { show progress }
PLOTS            { save result displays }

contour(ux) on x=.5
contour(ux) on x=1
contour(ux) on x=2
contour(ux) on x=3
contour(ux) on x=4

contour(p) painted on z=H/2

contour(kperm) on x = filterloc-filterwid/3
contour(kperm) on x = filterloc-filterwid/5
contour(kperm) on x = filterloc
contour(kperm) on x = filterloc+filterwid/5
contour(kperm) on x = filterloc+filterwid/3

vector(ux,uy) on z=H/2
vector(ux,uy) on z=H/2 zoom(filterloc-filterwid,0,2,Wid)

Summary
report Qin report Qout
report Vmax
report Re report Remax
```

```
report Pin report Pout report deltaP
report keff

END
```

Appendix B: Example source code for Model 2. Simulation M2f1m2l2

```
TITLE 'Vertical Pipe with filter'     { the problem identification }
COORDINATES cartesian3  { coordinate system, 1D,2D,3D, etc }
VARIABLES        { system variables }
  ux uy uz p             { choose your own names }
SELECT         { method controls }
!ngrid = 10
DEFINITIONS
L = 5
R = 1
filterstart = 2
filterend = 3
filterlen = filterend-filterstart

dens = 10
visc = .3
k0 = 1e-8
kmin = 1e-9
lamday = .5
lamdaz = .5
lamdax = .5
```

kperm = k0*(1- (sin(2*pi*x/lamdax))^2 * (sin(2*pi*y/lamday))^2 * (sin(2*pi*z/lamdaz))^2) +kmin

keff = vol_integral(kperm,'filter')/(pi*R^2*filterlen)

p0 = 100000

Pin = surf_integral(p,'bottom')/(pi*R^2)

Pout = surf_integral(p,'top')/(pi*R^2)

deltaP = Pout-Pin

Qin = surf_integral(uz,'bottom')

Qout = surf_integral(uz,'top')

Qfilter = surf_integral(uz,'filter top')

Re = dens*Qout/(pi*R^2)*2*R/visc

Ref = dens*Qfilter/(pi*R^2)*2*R/visc

M = 10000

fluid_part = 1

! INITIAL VALUES

EQUATIONS { PDE's, one for each variable }

ux: dx(P) - visc*(div(grad(ux)))+dens*(ux*dx(ux)+uy*dy(ux)+uz*dz(ux)) + (1-fluid_part)*ux*visc/kperm= 0 {navier stokes in 2D cartesian, SS}

uy: dy(P) - visc*(div(grad(uy)))+dens*(ux*dx(uy)+uy*dy(uy)+uz*dz(uy)) +(1-fluid_part)*uy*visc/kperm= 0 {no graivty term, incompressible}

uz: dz(P) - visc*(div(grad(uz)))+dens*(ux*dx(uz)+uy*dy(uz)+uz*dz(uz)) +(1-fluid_part)*uz*visc/kperm= 0

p: fluid_part*(div(grad(p)) - M*(dx(ux)+dy(uy)+dz(uz))) + (1-fluid_part)*(div(kperm*grad(p))) = 0

! CONSTRAINTS { Integral constraints }

extrusion

surface 'bottom' z = 0

```
layer 'under filter'
surface 'filter bottom' z = filterstart
layer 'filter'
surface 'filter top' z = filterend
layer 'above filter'
surface 'top' z = L

BOUNDARIES
  REGION 1
    layer 'filter' fluid_part = 0
    surface 'bottom' value(uz) = .001
    surface 'top' value(p) = p0


    START'pipe'(R,0)
    value(ux) = 0 value(uy) = 0 value(uz) = 0
    arc(center = 0,0) angle = 360 close

! TIME 0 TO 1    { if time dependent }
MONITORS         { show progress }
PLOTS            { save result displays }

contour(uz) on z=.5
contour(uz) on z=1
contour(uz) on z=2.5
contour(uz) on z=4

contour(p) on x=0 painted
vector(uy,uz) on x=0
```

```
vector(uy,uz) on x=R/2
contour(kperm) on z = (filterstart+filterend)/2

Summary
report keff
report pin
report pout report deltaP
report Re report Ref
report Qin report Qout report Qfilter

END
```

Appendix C: Example source code for Model 3. Simulation M3m1

```
TITLE 'Flow through duct with variable filter'     { the problem identification }
COORDINATES cartesian3  { coordinate system, 1D,2D,3D, etc }
VARIABLES        { system variables }
  ux(.1) uy(.1) uz(.1) p(.1)         { choose your own names }
SELECT        { method controls }
!Ngrid = 5
DEFINITIONS    { parameter definitions }
L = 5
Wid = 2
H = 2

filterloc = 1.5
filterwid = .5
```

dens = 10

visc = .3

k0 = 1e-8

kmin = 1e-10

lamday = 1

lamdaz = 1

lamdax = 1

kperm = k0

keff1 = vol_integral(kperm,'filter 1')/(H*Wid*filterwid)

keff1b = vol_integral(kperm,'filter 1')/vol_integral(1,'filter 1')

keff2 = vol_integral(kperm,'filter 2')/(H*Wid*filterwid)

keff3 = vol_integral(kperm,'filter 3')/(H*Wid*filterwid)

keff = (keff1 + keff2 + keff3)/3

p0 = 100000

Pin = surf_integral(P,'start')/(H*Wid)

Pout = surf_integral(P,'end')/(H*Wid)

flowin = surf_integral(ux,'start')/(H*Wid)

flowout = surf_integral(ux,'end')/(H*Wid)

M = 10000

fluid_part = 1

Va = integral(ux)/integral(1)

Vmax = globalmax(ux)

Re = Va * Wid * dens / visc

Remax = Vmax * Wid * dens / visc

```
INITIAL VALUES

ux = .001 uy = 0 uz = 0 p = 100000

EQUATIONS       { PDE's, one for each variable }

  ux:  dx(P) - visc*(div(grad(ux)))+dens*(ux*dx(ux)+uy*dy(ux)+uz*dz(ux)) + (1-fluid_part)*ux*visc/kperm= 0 {navier stokes in 3D cartesian, SS}

  uy:  dy(P) - visc*(div(grad(uy)))+dens*(ux*dx(uy)+uy*dy(uy)+uz*dz(uy))  +(1-fluid_part)*uy*visc/kperm= 0  {no graivty term, incompressible}

  uz:  dz(P) - visc*(div(grad(uz)))+dens*(ux*dx(uz)+uy*dy(uz)+uz*dz(uz))  +(1-fluid_part)*uz*visc/kperm= 0

  p:  fluid_part*(div(grad(p)) - M*(dx(ux)+dy(uy)+dz(uz))) + (1-fluid_part)*(div(kperm*grad(p))) = 0


! CONSTRAINTS    { Integral constraints }


Extrusion

surface 'bottom' z=0

layer 'everything'

surface 'top' z=H


BOUNDARIES       { The domain definition }


  REGION 'domain'  1     { For each material region }

   surface 'bottom' value(ux) = 0 value(uy) = 0 value(uz) = 0

   surface 'top' value(ux) = 0 value(uy) = 0 value(uz) = 0

  START(0,0)   { Walk the domain boundary }

    value(uy) = 0 value(ux) = 0 value(uz) = 0

    line to (L,0)

    load(uy) = 0 load(ux) = 0 load(uz) = 0 value(p) = p0 line to (L,Wid)

    value(uy) = 0 value(ux) = 0 value(uz) = 0 load(p) = 0  line to (0,Wid)
```

```
load(uy) = 0 value(ux) = 0.001 load(uz) = 0 line to close

Region 'filter 1' 2
 fluid_part = 0
kperm = k0*(1-  (1-(1-(sin(2*pi*z/lamdaz))^4)^4)^4  ) +kmin
  surface 'bottom' value(ux) = 0 value(uy) = 0 value(uz) = 0
  surface 'top' value(ux) = 0 value(uy) = 0 value(uz) = 0
   start (filterloc - filterwid/2,0)
   line to (filterloc+filterwid/2,0)
   line to (filterloc+filterwid/2,Wid)
   line to (filterloc-filterwid/2,Wid)
   line to close

Region 'filter 2' 3
 fluid_part = 0
 kperm = k0*(1-  (1-(1-(sin(2*pi*y/lamday))^4)^4)^4  ) +kmin
 surface 'bottom' value(ux) = 0 value(uy) = 0 value(uz) = 0
 surface 'top' value(ux) = 0 value(uy) = 0 value(uz) = 0
  start (filterloc+filterwid/2,0)
  line to (filterloc+1.5*filterwid,0)
  line to (filterloc+1.5*filterwid,Wid)
  line to (filterloc+filterwid/2,Wid)
  line to close

Region 'filter 3' 4
 fluid_part = 0
 kperm = k0*(1-  (1-(1-(sin(2*pi*z/lamdaz))^4)^4)^4  ) +kmin
 surface 'bottom' value(ux) = 0 value(uy) = 0 value(uz) = 0
```

```
  surface 'top' value(ux) = 0 value(uy) = 0 value(uz) = 0
   start (filterloc+1.5*filterwid,0)
   line to (filterloc+2.5*filterwid,0)
   line to (filterloc+2.5*filterwid,Wid)
   line to (filterloc+1.5*filterwid,Wid)
   line to close


feature 'start' start(0,0) line to (0,Wid)
feature 'end' start (L,0) line to (L,Wid)

! TIME 0 TO 1    { if time dependent }
MONITORS         { show progress }
PLOTS            { save result displays }

contour(ux) on x=.5
contour(ux) on x=1
contour(ux) on x=2
contour(ux) on x=3
contour(ux) on x=4

contour(p) painted on z=H/2

contour(kperm) on x=filterloc as 'filter 1 permeability' painted
contour(kperm) on x=filterloc+filterwid as 'filter 2 permeability' painted
contour(kperm) on x=filterloc +2*filterwid as 'filter 3 permeability' painted

vector(ux,uy) on z=H/2
```

```
vector(ux,uy) on z=H/2 zoom(filterloc-filterwid,0,2,Wid)

Summary

report Va report Vmax
report Re report Remax
report flowin report flowout
report Pin report Pout
report keff1 report keff1b report keff2 report keff3 report keff

END
```