Activated Carbon Model Simulations

Razvan Carbunescu, Sarah Johnston, Mona Crump,
Brett McCullough, Daniel Guidry

1 Introduction

This project is an attempt to understand the role a Granulated Activated Carbon (GAC) Filter has in relieving the stress of a Biofilter. The purpose is to understand how the GAC filter works in conjunction with the Biofilter in many different phases. The mathematics behind the process of a GAC filter are complicated and the goal is to find a method to understand these filters and provide models of their effectiveness and efficiency.

2 GAC Filter Description

Activated carbon is a natural material resulting from a mixture of bituminous coal, lignite, wood, coconut shell etc., activated by steam and other means, and each one have different adsorption properties. Activated carbon surface properties are both hydrophobic and oleophilic. This means that the carbon molecules dispel water but are attracted to oil. When flow conditions are suitable, dissolved chemicals in either liquid or gas flowing over the carbon surface adhere to the carbon in a thin film while the liquid or gas passes on. This process is called adsorption. Adsorption is a surface occurrence, in which molecules of adsorbate (toluene) are attracted and held to the surface of an adsorbent (carbon) until equilibrium is reached between adsorbed molecules and those still freely distributed in the carrying gas or liquid. This physical adsorption is a type of adsorption that is dependent mainly on surface attraction in which factors such as temperature, pressure and impurity may alter the equilibrium. As a result of the adsorption process, activated carbon is an effective method in removing organic compounds such as volatile organic compounds (carbon based VOC's). Other chemicals that activated carbon cannot reduce are non-carbon based anions (-) and cations (+) such as arsenic, fluorides, some heavy metals, nitrate, etc. The electronic forces, known as Van der Waal's forces, responsible for adsorption are related to those which cause like molecules to bind together, producing the phenomenon of condensation and surface tension. Van der Waal's force refers to positive and negative forces between molecules such as dipole-dipole forces, dipole-induced dipole forces, as well as London forces. Some of the advantages of GAC vs other forms of carbon are: on a large scale such as municipal water treatment pools (gravity filters) for taste, odor and chemical reduction GAC is cheaper, very effective and can be re-used. Just one gram of activated carbon has a surface area of approximately 500 m. This surface area is typically determined by nitrogen gas adsorption. This high surface area is due to extreme microporosity and macroporisity. These innumerable microscopic cracks and pores in the activated carbon, also known as microporisity, exponentially increases the surface area. While openings into the carbon structure may be of various shapes, the term pore, implying a cylindrical opening, is widely used. Microporosity is helpful in adsorbing lower molecular weight, lower boiling point organic vapors, as well as in removing trace organics in liquid or gas to non-detectable levels. Larger pore openings make up the macroporosity, which is useful in adsorbing very large molecules and aggregates of molecules, such as "color bodies" in raw sugar solutions. Activity level is often expressed as total surface area per unit weight, usually in square meters per gram. This total exposed surface will typically be in the range of 600-1200 m2/g. At the higher end of this range, one might better visualize one pound, about a quart in volume, of granular activated carbon with a total surface area of 125 acres. Granular vapor phase carbons were first widely used in WWI military gas masks. In the years between World Wars, this carbon was also used commercially in solvent recovery systems. Granular liquid phase carbons achieved their first prominent applications following WWI, in sugar de-colorization and in the purification of antibiotics. Today, there are hundreds of applications of both liquid and gas phase activated carbon.

3 Biofilter Description

Biofiltration uses microorganisms to remove VOC's (volatile organic compounds) from contaminated air. Although, a steady input of contaminants into the biofilter is ideal, it is not realistic due to operational shutdowns. During periods of increased contaminant loading, the microorganisms can undergo shock. Also, during periods of little or no loading, the microorganisms can undergo starvation conditions. Both of these situations can reduce the efficiency of the biofilter. Two main disadvantages of feeding biofilters are that contaminants are expensive, which would result in an increased operational cost. Also, if the contaminants are not completely degraded in the biofilter, then the facility would have an increase in contaminant mass emissions. Using GAC filters in conjunction with biofilters will alleviate the adverse effects of starvation during periods of operational shutdown when no contaminants would otherwise be supplied to the biofilter.

EXPERIMENT WITH DR. MOE

We are working with Dr. Moe, who is with the LSU Civil and Environmental Engineering Department, to develop mathematical models that coincide with his experimental results. In his research, Dr. Moe analyzed how different loading ratios and time intervals affected the ability of the GAC filter to stabilize the contaminant loading. This question can arise in cases of unplanned outages in manufacturing or other variations of schedule that may not have been taken into account during the design of the filtration system. A GAC filter is able to stabilize loads passively due to its ability to both adsorb contaminants during periods of increased loading and desorb previously adsorbed contaminants during periods of reduced loading. The GAC filter will initially adsorb practically all contaminants until it has reached a quasi-steady state breakthrough. It then begins to have an output in correlation with its input and it also acts to attenuate changes in the contaminant loading.

His results below show that toluene can be temporarily accumulated in a GAC filter during intervals when concentrations are high and then desorb within a short time interval when concentrations are low. It can be observed that smaller cycle times and smaller loading ratios provide for a more constant load input to the biofilter.

4 Model developmnent

Assumptions

1. Plug flow conditions exist in the bed
2. No angular concentration gradients within a particle
3. Intraparticule mass flux described by surface and pore diffusion
4. Local adsorbtion equilibrium exists between the solute adsorbed onto the GAC paricle and the solute in the intraaggregate stagnant fluid.
5. fluid enters from the top of the filter and leaves from its bottom at the same uniform rate ρv=constant

Based on these assumptions the non-dimensionalized governing equations are given below.

4.1 Mobile Phase

$$\frac{1}{(1+D_{g_t})}\frac{\partial \overline{C_i}}{\partial T}(\bar{z},T) = -\frac{\partial \overline{C_i}}{\partial \bar{z}}(\bar{z},T) - 3S_{t_i}\left(\overline{C_i}(\bar{z},T) - \overline{C}_{p,i}(\bar{r}=1,\bar{z},T)\right) \tag{4.1}$$

The transformed initial and boundary conditions are:

$$\overline{C_i}(\overline{z},T=0) = 0 \tag{4.2}$$

$$\overline{C_i}(\overline{z}=0,T\geq 0) = 1 \tag{4.3}$$

4.2 Pore Phase

The intraparicule phase mass balance:

$$\frac{1}{\overline{r}^2}\frac{\partial}{\partial \overline{r}}\left[\overline{r}^2\left(\left(Ed_{s,i}+D_iEd_{p,i}\right)\frac{\partial \overline{Y}_i}{\partial \overline{r}}\left(\bar{r},\bar{z},T\right)+Ed_{p,i}\left(1-D_i\right)\frac{\partial \overline{C}_{p,i}}{\partial \overline{r}}\left(\bar{r},\bar{z},T\right)\right)\right]$$
$$=\frac{D_{g_i}}{\left(1+D_{g_t}\right)}\frac{\partial \overline{Y}_i}{\partial T}\left(\bar{r},\bar{z},T\right) \tag{4.4}$$

The transformed initial and boundary conditions are:

At time $T=0$, there is no solute in the adsorbent, The IC is given by:

$$\overline{Y_i}\left(\overline{z},\overline{r},T=0\right)=0 \tag{4.5}$$

Due to symmetry at the center of the particle, the first BC is given by:

$$\frac{\partial \overline{Y}_i}{\partial \overline{r}}\left(\bar{r}=0,\bar{z},T\right)=0 \tag{4.6}$$

From the mass balance at the adsorbent phase we can derive the following second BC:

$$\frac{D_{g_i}}{\left(1+D_{g_i}\right)}\frac{\partial}{\partial T}\int_0^1 \overline{Y}_i\left(\bar{r},\bar{z},T\right)\bar{r}^2\,d\bar{r}=S_{t_i}\left[\overline{C}_i\left(\bar{z},T\right)-\overline{C}_{p,i}\left(\bar{r}=1,\bar{z},T\right)\right] \tag{4.7}$$

4.3 The nonlinear coupling equation

$$\overline{C}_{p,i}\left(\bar{r},\bar{z},T\right)=\frac{\overline{Y}_i\left(\bar{r},\bar{z},T\right)Y_{e,i}-\frac{\epsilon_p}{\rho_a}\overline{C}_{p,i}\left(\bar{r},\bar{z},T\right)C_{o,i}}{C_{o,i}\sum_{j=1}^m\left[\overline{Y}_j\left(\bar{r},\bar{z},T\right)Y_{e,j}-\frac{\epsilon_p}{\rho_a}\overline{C}_{p,j}\left(\bar{r},\bar{z},T\right)C_{o,j}\right]}$$
$$\cdot\left[\frac{\sum_{k=1}^m n_k\left[\overline{Y}_k\left(\bar{r},\bar{z},T\right)Y_{e,i}-\frac{\epsilon_p}{\rho_p}\overline{C}_{p,k}\left(\bar{r},\bar{z},T\right)C_{o,i}\right]}{n_iK_i}\right]^{n_i} \tag{4.8}$$

4.4 Nomenclature

The unknown functions are :
$\overline{C_i}\left(\overline{z},T\right)$: adsorbate concentration in bulk phase
$\overline{C}_{p,i}\left(\bar{r},\bar{z},T\right)$: adsorbate concentration in adsorbent pores
$\overline{Y_i}\left(\overline{z},\overline{r},T\right)$: Total adsorbent phase concentration
and $\bar{r},\bar{z}$ and T are respectively the radial, the axial and the time coordinates.

4.5 Numerical Method

In regards to the pore phase equation we attempted to understand the mathematics behind this equation. It just so happens that the mobile phase model is a symmetric equation with three variables, being the rate of the fluid into the biomass filter, the time at which the particular pore in question is assessed, and the size of the carbon pore in question. We first took the partial derivatives of each of these variables and then we proceeded to plug in the points of our model trials, with these being three, five, and seven grid points. These points are known as collocation points. Once you do this you can arrange each of these solutions together to form three different matrices. This is known as orthogonal collocation. Doing this procedure by hand is very expensive and time consuming, so we have worked together to create a program in Matlab that solves for these matrices for us. However, we still find that it is very crucial to understand the mathematical methods used to determine these matrices. The next step from here will be to further our knowledge of the mobile phase of our model and determine exactly how this phase is deduced mathematically.

The method of orthogonal collocation [1] is applied to solve the above equations. This method uses Legendre polynomials as trial functions and the collocations points are taken as the roots of those polynomials. The spatial derivatives are expressed in matrix form in terms of the dependent variables evaluated at the collocation points. This process leads to a set of ordinary differential equations where the independent variable is the time T. Finally, to integrate these equations we used the following Matlab solver:

```
[T, Y] = ode15i (odefun, tspan, y0, yp0, options)
```

4.6 Development of Computer Program and Design of Interface

One part of the research that we have developed and improved is the approximations used for the partial in the form of the weights, first order derivative and Laplacian matrices and doing this in a way that can allow for further better approximations if necessary. Unlike up to this moment the program has been changed to select and input the desired number of axial and radial points. Through a MATLAB program we now calculate the desired matrices without having to input them from a particular paper. This has the following advantages: allowing for a much easier change from one set of parameters to another, allows for the user to easily upgrade to as high as an accuracy as needed without having to do any extra research to find the desired matrices and this change also allows for a greater accuracy in the matrices as these are calculated now to double precision within MATLAB rather than 5-6 digits accuracy when hardcoded. The only current drawback of using this technique is that the values of the Legendre Roots have to be specified for the spherical geometry with weighting function w = 1-x2 but we have included in the code the values of the first 7 roots of the Legendre polynomials and for simulating the carbon itself this number is more than enough for any practical purposes. The code used was validated against the tables provided for both symmetrical and non-symmetrical results from "SPEEDUPTM ION EXCHANGE COLUMN MODEL" (1) which was the paper initially used to obtain the hardcoded matrices. Results were also compared against the examples provided in "The Method of Weighted Residuals and Variational Principles" (2) Chapter 5.1 page103 from which the procedure to generate the matrices was taken. The full MATLAB Code for generating the matrices is attached to the end of the report - Appendix A. The Legendre Roots for the axial points used in the scheme are now calculated using a MATLAB program that auto generates the nth degree Legendre Polynomial. The program uses the generation function described in "Computational Algorithm for Higher Order Legendre Polynomial and Guassian Quadrature Method"(3) to generate the Legendre polynomials. The roots are then calculated for the symmetric planar geometry and through manipulation of the solutions the correct roots can be produced for symmetric planar geometry, symmetric spherical geometry and non-symmetric planar geometry(used for the axial points). All roots generated are for Legendre Polynomials with a unity weight function. The roots were against verified against "SPEEDUPTM ION EXCHANGE COLUMN MODEL" (1) for the first roots and then against "Multiple Root Finder for Legendre and Chebyshev polynomials via Newton's Method"(4). The full MATLAB Code for generating the Legendre roots is attached to the end of the report - Appendix B. While these accomplishments improve the quality of the simulation most of them can remain hidden from the user of the code so during our time in this project we also worked to improve on the basic interface to the GAC filter simulation using MATLAB's built in GUIDE system(4). The new GUI will present the option to change the basic GAC filter properties and approximations without having to manually edit the .m file. The GUI also helps in easily identifying variables and values needed for them as it starts with a default test case already loaded. The new GUI also presents the results incorporated within the interface rather than as a separate graph appearing in a more complete way. The GUI also presents tabs for the selectrion of different input conditions than constant loading like intermittent loading, input according to a function or input read in from an Excel file. The GUI now also allows printing the results to a Excel file for further analysis. A view of the GUI is provided at the end of the report - Appendix C.

4.7 Results and Simulations

When doing the simulations of the filtering process, it is important to realize how many grid points to take in order for the graph to look like the model graph. The results were compared to the results of Dr. Moe. If the results match the data provided by Dr. Moe, then it can be assured that the program is working correctly. It can be seen that with three grid points, the graph clearly shares some characteristics as the model, but it does not align with it as expected, so more points are taken. With five points, it can be seen that the graph is very close to the model, yet there is still a bump. So now seven points are taken, and the graph produced from the results lines up almost perfectly with the model. There are no deviations from the model to show that the simulations are giving improper data. Therefore, it is sufficient to say taking seven points can give an accurate reading of results of the current specifications of the filtering process. As expected, eight grid points gives the same graph. Professor Moe's experimental results have compared almost identically. Thus from this, the program seems to be working efficiently. It has been seen that the program works with constant loading. The program was used in a simulation with intermittent loading. This means that instead of a constant stream of the pollutant through the filter, the contaminate was passed through the filter for 8 hours, then off for 16 hours. This was done continually for many days. What resulted was a graph that shows the output reaching an almost steady rate after 5 days. It can be seen that it is not perfectly constant, but rather fluctuates. What is constant is the amount of output between which it fluctuates. This type of intermittent loading kept the filter from becoming saturated. Rather, with the continual on/off loading pattern, the filter could be used without ever having to be replaced since it never reached a level of 100 percent pollutant output. Therefore, the filter could be used for practical application. It is especially useful for use with the bio-filter, which needs a certain amount of pollutant consistently. The other type of loading would saturate the filter, and the organisms in the bio-filter would die from exposure to excess pollutants. When comparing the data with that of Dr. Moe, it again can be seen that the results mirror his. This means that everything that has been done up to this point is still accurate.

4.8 Conclusion

In conclusion, the project has turned out to be a success. There is a very good understanding of what a GAC filter is and how it interacts with a biomass filter to work efficiently together. The procedure of the mathematics was very complicated but has now become very clear and understandable. The work put in to form a GUI has been completed and the GUI is functioning correct. Also, the results of the simulations have corresponded well with Dr. Moe's experiments. So, all in all the project has been a success and was a great learning experience.

4.9 Bibliography

1. Mathematical Modeling of Multicomponent Adsorption in Batch and Fixed-Bed Reactors, by Gary Friedman, Michigan Technical University 1984.

2. SPEEDUPTM ION EXCHANGE COLUMN MODEL T.Hang, R.A.Dimenna, 2000

3. The Method of Weighted Residuals and Variational Principles Bruce A. Finlayson, 1972

4. Computational Algorithm for Higher Order Legendre Polynomial and Guassian Quadrature Method Asif M. Mughal, Xiu Ye, Kamran Iqbal

5. Multiple Root Finder for Legendre and Chebyshev polynomials via Newton's Method Victor Barrera-Figueroa, Jorge Sosa-Pedroza, Jose Lopez-Bonilla, 2006,

6. MATLAB, The language of Technical Computing, Creating Graphic User Interfaces Version 6, The Mathworks

7. Moe, William. Activated Carbon Load Equalization of Gas-Phase Toluene: Effect of Cycle Length and Fraction of Time in Loading. Environmental Science and Technology 41 (2007): 5478-5484.

8. Nabatilan, Marilou M. and William Moe. Use of GAC Adsorption Columns to Mitigate the Adverse Effects of Various Shutdown Conditions on Biofilter Performance. Research Paper. Louisiana State University, Baton Rouge, Louisiana.

4.10 Appendix A

```
function [W,A,B] = GACMatrices(n,X,a,s)
%
% This function will calculate the matrices needed for the GAC calculations
% The function should be used in the following manner:
%
% [W,A,B] = GACMatrices(n,X,a,s);
%
% where:
% n     -   is the number of points
% x[j] -   are the roots of the polynomials
% a     -   is the nature of the geometry
%           ( 1 - planar    2 - cylindrical   3 - spherical)
% s - whether the function is symmetrical about the origin

if(s == 1)
% adding 1 as the last point
X(n+1) = 1;
n = n+1;
end

if(s == 0)
% adding 0 and 1 as the first and last point
for i=n:-1:1
X(i+1)=X(i);
end
X(1) = 0;
X(n+2) = 1;
n = n+2;
end

if( s == 1 )
for i=1:n
for j=1:n
% setting Q to simply be the function
Q(j,i) = X(j)^(2*i-2);

% setting C to be the first derivative
C(j,i) = (2*i-2) * X(j)^(2*i-3);

if (a == 1)
% setting D to be the laplacian
```

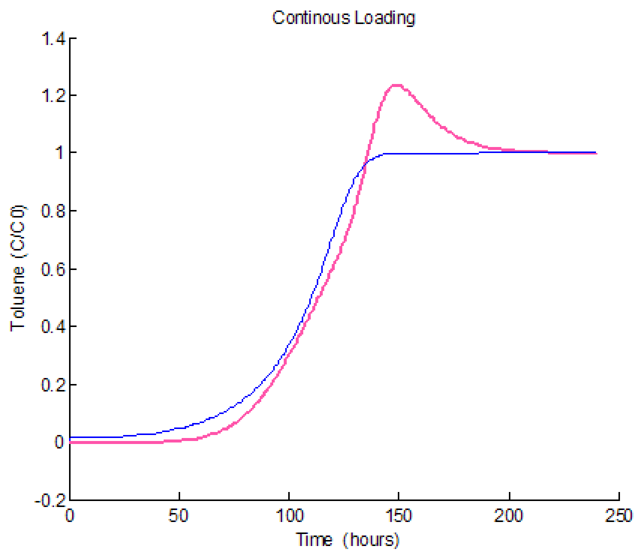
```
D(j,i) = (2*i-2)*(2*i-3) * X(j)^(2*i-4);
end
if (a == 2)
% setting D to be the laplacian
D(j,i) = (2*i-2)^2 * X(j)^(2*i-4);
end
if (a == 3)
% setting D to be the laplacian
D(j,i) = (2*i-2)*(2*i-1) * X(j)^(2*i-4);
end
end

% Calculating F
F(i)=1 / (2*i - 2 + a);
end
end

if( s == 0 )
for i=1:n
for j=1:n
% setting Q to simply be the function
Q(j,i) = X(j)^(i-1);

% setting C to be the first derivative
C(j,i) = (i-1) * X(j)^(i-2);

if (a == 1)
% setting D to be the laplacian
D(j,i) = (i-1)*(i-2) * X(j)^(i-3);
end
if (a == 2)
% setting D to be the laplacian
D(j,i) = (i-1)^2 * X(j)^(i-3);
end
if (a == 3)
% setting D to be the laplacian
D(j,i) = (i-1)*(i) * X(j)^(i-3);
end
end

% Calculating F
F(i)=1 / (a + i - 1);
```

```
end
end


%inverting Q
Qi = inv(Q);

%calculating A as the product of C and the inverse of Q
A = C*Qi;

%calculating B as the product of D and the inverse of Q
B = D*Qi;

%calculating W as the product of F and the inverse of Q
W = F*Qi;

%Quick fix for initial row problem for non-symmetrical roots
if ( s == 0)
for i=1:n
A(1,i) = -1 * A(n,n+1-i);
B(1,i) = B(n,n+1-i);
end
end
```

4.11 Appendix B

```
function [X] = LegendreRoots(n,a,s)
% given n this will give the roots of the n-th order polynomial
%
%    [X] = LegendreRoots(n,a,s)
%
% Input:
% n - order of Legendre Polynomial
%  a - is the nature of the geometry
%          ( 1 - planar    2 - cylindrical   3 - spherical)
%  s -  whether the function is symmetrical about the origin
% Output:
% X - an array of the roots of Legendre Polynomials



if (s == 1)
if (a==1)
n=n*2;
end

if (a==3)
n=n*2+1;
end
end

if (s == 0)
if(a == 1)
% nothing to do for a=1
end
end


A = zeros(1,n+1);
bm = floor(n/2);
for m=0:bm
A( n-2*m + 1)= power(-1,m) * factorial(n) * factorial(2*n - 2*m) / (factorial(2*n) * fa
end
for i=1:n+1
P(i) = A(n+2-i);
end
Y = roots(P);
```

```
Y = sort(Y);

if (s == 1)
if (a == 1)
for i=(bm+1):n
X(i- bm) = Y(i);
end
end

if (a == 3)
for i=(bm+2):n
X(i- bm -1) = Y(i);
end
end
end

if (s == 0)
if(a == 1)
Y = Y.*0.5;
X = Y + 0.5*ones(n,1);
X = X';
end
end
```



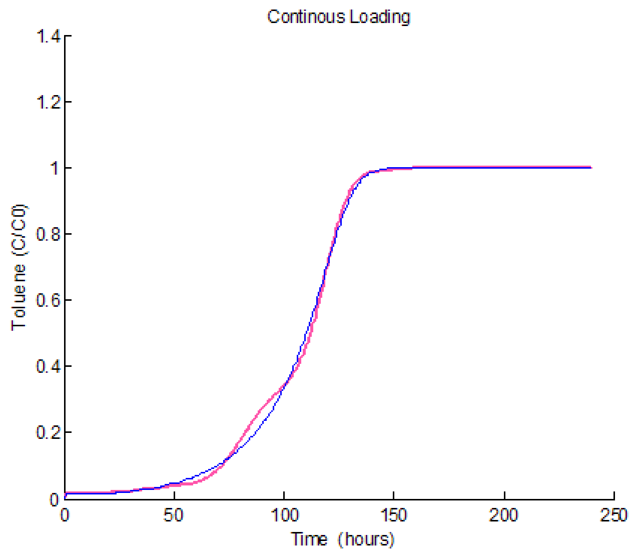
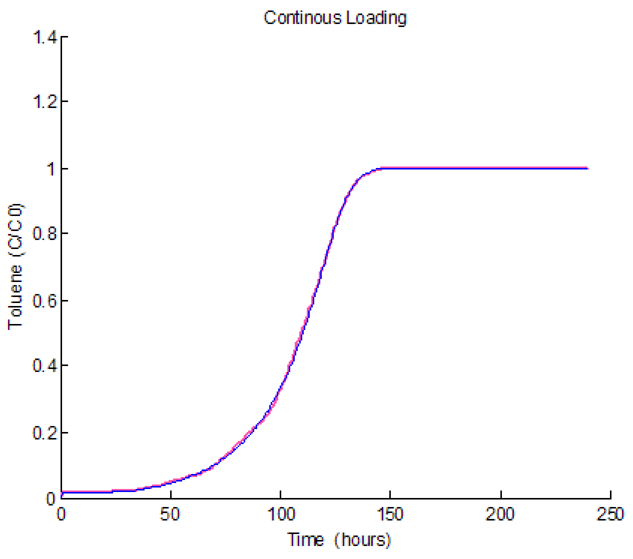


Figure 1: Comparison with the University of Michigan Model



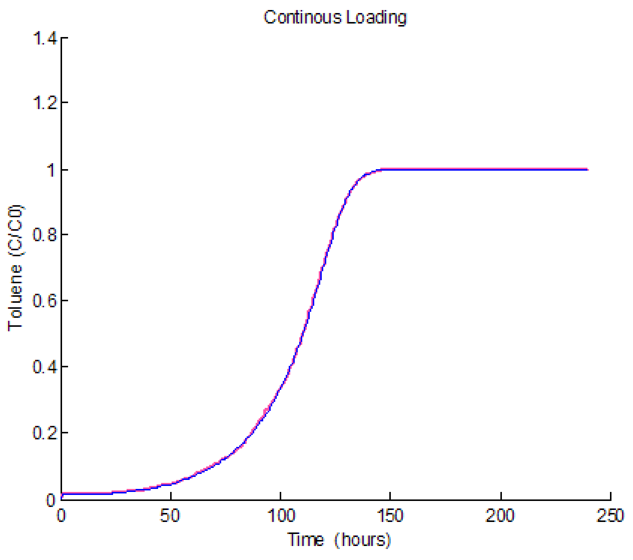


Figure 2: Comparison with the University of Michigan Model

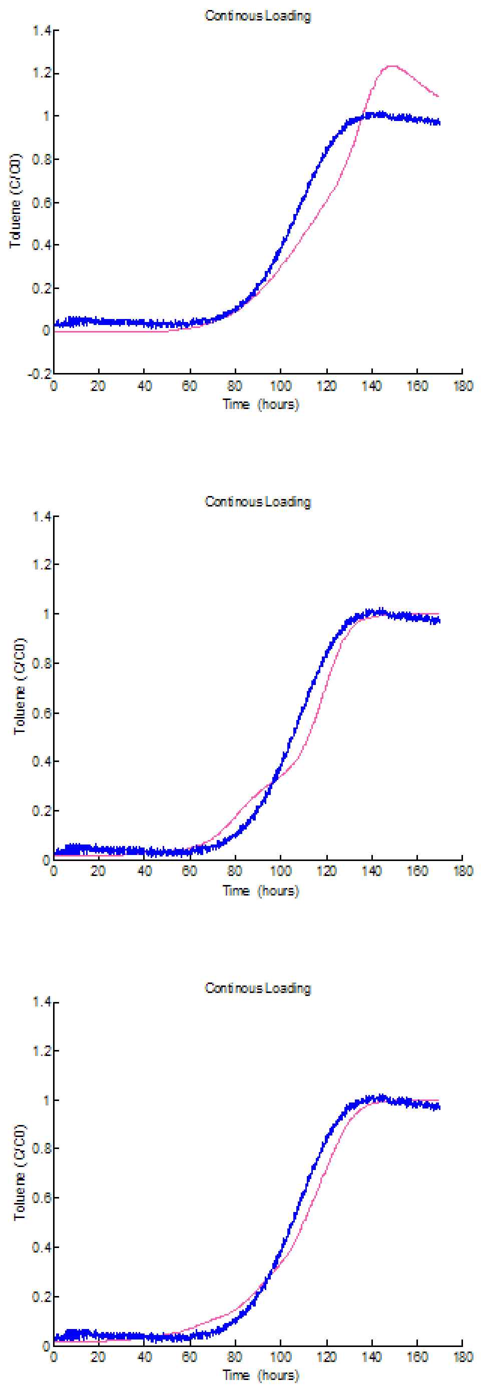


Figure 3: Comparison with experimental data

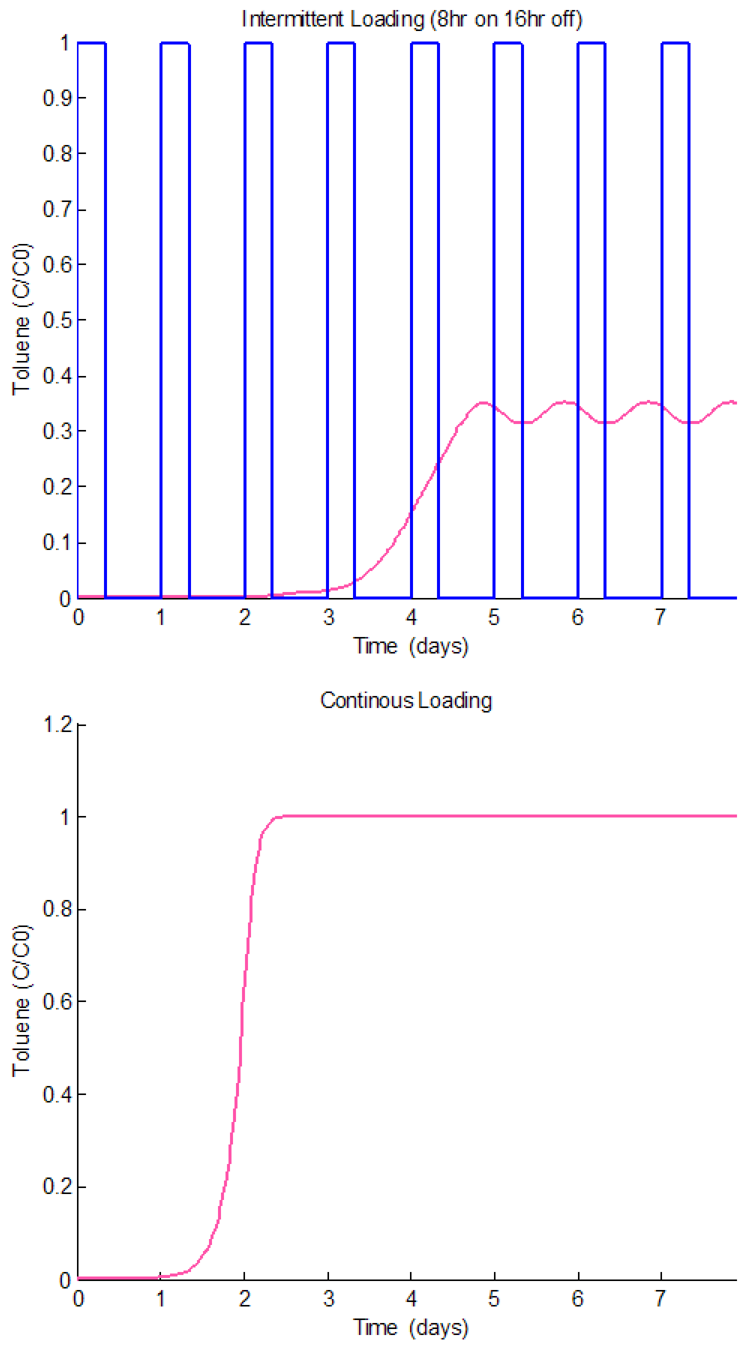
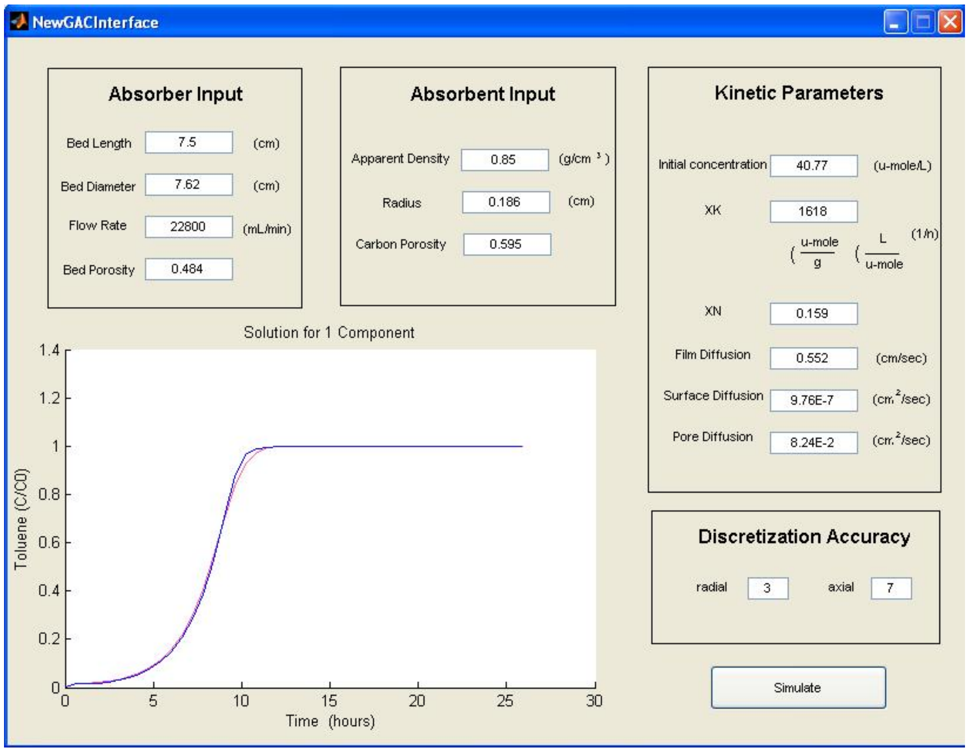


Figure 4: Comparison of Intermittent and continous Loading


NewGACInterface

Absorber Input

Bed Length	7.5	(cm)
Bed Diameter	7.62	(cm)
Flow Rate	22800	(mL/min)
Bed Porosity	0.484	

Absorbent Input

Apparent Density	0.85	(g/cm ^{3})
Radius	0.186	(cm)
Carbon Porosity	0.595	

Kinetic Parameters

Initial concentration	40.77	(u-mole/L)
XK	1618	$(\frac{\text{u-mole}}{\text{g}})(\frac{\text{L}}{\text{u-mole}})^{(1/n)}$
XN	0.159	
Film Diffusion	0.552	(cm/sec)
Surface Diffusion	9.76E-7	(cm ^{2}/sec)
Pore Diffusion	8.24E-2	(cm ^{2}/sec)

Discretization Accuracy

radial	3	axial	7

Simulate

Figure 5: Interface