Computational Characterization of Protein A – Antibody Binding

Submitted in partial fulfillment of the degree requirements for the degree of

Master of Science in Chemical Engineering

Virginia Louise Lane

B.S., Chemical Engineering, University of Florida
B.S., Microbiology and Cell Science, University of Florida
B.M., Music Performance, University of Florida

Carnegie Mellon University
Pittsburgh, PA

December, 2018

Acknowledgements

I would like to thank my advisor, Todd M. Przybycien, without whose continuous help, this research and thesis would not have been possible.

I would also like to thank Justin Weinberg for help in continuing on his initial research.

This research was made possible by funding from the Carnegie Mellon University Chemical Engineering Department and the Row W. Weiland Graduate Fellowship in Chemical Engineering.

Abstract

Protein A affinity chromatography is a key step in the downstream processing of monoclonal antibodies (mAbs). Because this step also accounts for over 50% of downstream process costs, improving its separation efficiency could have a significant impact on production costs, ultimately making mAb treatments cheaper and more widely available. Much research has previously been done on the large difference in binding affinity of antibodies and non-antibody proteins to protein A. However, we assert that it may be possible to exploit subtle differences in binding affinity between different antibody subspecies to the wild type protein A ligand, if properly understood. Previous adsorption experiments using media with wild-type protein A ligands have shown an unexpected binding and displacement phenomenon when polyclonal antibodies are in solution with monoclonal antibodies. This phenomenon has not been previously explained or examined in the literature, and we posit that it may have implications which could be exploited for future separations of antibody subspecies and variants. Here, we develop a computational model to simulate competitive antibody binding and displacement behavior on protein A media with wild-type ligands. Our computational results have successfully captured observed binding and displacement effects between competitive components at the single chromatography bead level. We have also developed a new isotherm to define the relationship between protein A- antibody complexation and pH. This provides a connection between antibody elution pH and binding strength that can be utilized in future simulations to predict relative antibody species binding strengths and, in turn, the performance of displacement-mode separations.

Table of Contents

List of Figures

List of Tables

Introduction

Monoclonal antibodies are an important type of biotherapeutic in western medicine. With a market of USD 100 billion in 2017 and an estimated market of USD 140 billion by 2024, their impact on the pharmaceutical industry is substantial and growing (Market Analysis, 2016). Monoclonal antibodies (mAbs) work by a variety of mechanisms, and have been used to treat a wide range of prevalent diseases, including lymphoma, leukemia, colorectal cancer, psoriasis, arthritis, and multiple sclerosis (Buss et al., 2010). A mAb is an IgG-class antibody that binds a specific target and is produced and secreted into the cell culture media by a cell line, or cells that are all cloned from a single cell. Because of this, a given mAb ideally consists of antibody molecules identical in their chemical and physical structure and binding specificity. For mAbs to be used clinically, they must have a certain level of purity to be both effective and safe. If not eliminated, impurities present in the cell culture media such as host cell proteins, DNA, RNA, antibody aggregates, antibody fragments, and adventitious agents such as viruses can wreak havoc in vivo (Carta & Jungbauer, 2010). Another concern, is that even though a given cloned mAb should be a homogenous population of molecules, there is still potential for heterogeneity. This includes glycosylation variants and chemical degradation products, such as deamination products, that give rise to charge variants.

In the industrial production of mAbs, the entire downstream process is devoted to ensuring purity of the target antibody solution. The key step, or capture step, in this downstream purification process, is typically protein A affinity chromatography (González-Valdez et al., 2014). Protein A is a cell wall protein found in the bacterium *Staphylococcus aureus*, and is known for its strong binding affinity to many forms of human IgG-class (hIgG) antibodies (Sjöquist, 1972). This is significant as IgG is the most prevalent antibody class, making up 90% of blood serum (Murphy,

2012). Because of its strong and specific binding, protein A has been utilized in affinity chromatography processes to purify antibodies. Protein A ligands can be immobilized within the porous bead-based media used in chromatography columns. The immobilized protein A ligands bind to antibodies when antibody-containing harvested cell culture fluids are sent through the column at a neutral pH, allowing impurities to be washed out as the antibodies stay bound in the column. Elution of the bound antibodies then takes place as the pH is reduced via a linear or step gradient sent through the column and antibodies are washed out based on their decreased binding affinity. In addition to being the capture step in mAb purification, protein A chromatography is also the most expensive step, as the protein A resin accounts for over 50% of the downstream process cost (Dransart, et al., 2018). Any improvement in this key step could therefore lead to significant savings in mAb production costs and ultimately make mAb treatments cheaper and more widely available.

Much research has previously been done on the strong, specific binding affinity between antibodies and protein A relative to non-antibody proteins (González-Valdez et al., 2014; Hahn et al., 2003; Huang et al., 2011). The wild type protein A ligand has five similar binding domains (A, B, C, D, and E), each of which recognize the Fc portion of many antibody types, including three of the four human IgG subclasses (Ghose et al., 2005). Protein A binding domain-IgG association constants are on the order of 10^7 to $10^8\ M^{-1}$ (or 66 to 667 $\frac{mL}{mg}$), equivalent to a binding energy of -11 to -12 $\frac{kcal}{mol}$ at standard conditions (Pabst, Thai, & Hunter, 2018). Several commercial resins have exploited the binding strength and stability of the B domain by engineering multimers of the B binding domains as chromatographic ligands. Most notably, these engineered ligands eliminate the secondary, weaker binding observed between wild-type ligands and the Fab portion of the antibody in addition to the Fc portion (Ghose et al., 2005).

We assert that interesting and useful separations may be possible by exploiting observed differences in binding strength among mAb variants with the wild type protein A ligand, if properly understood. Previous single resin bead uptake experiments using wild type protein A ligands have shown an unexpected binding and displacement phenomenon when polyclonal antibodies, with a range of binding strengths as indicated by a distribution of elution pHs, are in solution with monoclonal antibodies (Weinberg et al., 2017). This phenomenon had not been previously explained or examined in the protein A chromatography literature, and we posit that it may have implications which could be exploited in future separations aimed at resolving mAb variants. Another potential use for this phenomenon would be to aid in the analysis of variants within monoclonal antibody process streams. It is clinically important, and of interest to the Food and Drug Administration (FDA), to understand the homogeneity of a therapeutic molecule when it is produced. While absolute homogeneity is not required, it is important that an approved biotherapeutic has a consistent variant distribution, as a change in distribution may indicate a change in biological activity or may alter the spectrum or intensity of side effects (Khawli et al., 2010). An understanding of the previously seen displacement phenomenon, may lead to the ability to resolve mAbs into subfractions based on differences in charge, sequence and or glycosylation pattern and to purify them within a specific range. Cation exchange and anion exchange are the typical modes of chromatography used to fractionate solutions based on small differences in charge and cation exchange chromatography has been used in a displacement mode for analytical-scale separations of mAb populations into main components and lumped acidic and basic charge variant components (Khawli et al., 2010; Zhang, Bourret, & Cano, 2011). Isoelectric focusing methods can be used to evaluate charge profiles and there is current research for the improvement of ion exchange chromatography for better resolution of these charge differences (Parekh et al., 2018).

However, these modes work best overall for smaller molecules having a small charge difference, while typical IgG-class antibodies have a size of approximately 150 kDa (Murphy, 2012), and the small charge differences within a mAb population such as those resulting from deamination, are more difficult to exploit at process scales.

Here, we develop a computational model to fully describe observed competitive antibody binding behavior in protein A chromatography media. This model simulates competitive binding and displacement between multiple antibody components with different binding strengths, within a single protein A resin bead. Model simulations were compared to competitive, single bead antibody uptake experimental results from prior work (Weinberg et al., 2017). In support of these simulations, we also developed an isotherm to define the relationship between protein A- antibody complexation and pH. This provides a direct connection between observed antibody subspecies/variant elution pH in gradient elution experiments and subspecies/variant binding strength. Ultimately, this model can be incorporated within the general rate model for chromatography to predict how the observed displacement phenomenon could be exploited in future chromatographic separations based on protein A affinity media.

Theory and Simulation

In order to understand and eventually exploit the binding and displacement phenomenon described above, our goal was to more completely characterize the binding interaction between IgG and protein A, and to use that characterization to model multicomponent antibody mass transport in the context of protein A affinity chromatography. Using displacement mode chromatography to experimentally characterize this phenomenon would unfortunately use a considerable amount of antibody protein. And, as protein A chromatography media and antibodies are very expensive, we chose to start with a computational approach that could be expanded to guide subsequent chromatographic experimentation.

Modeling Diffusion in a Single Sphere

We began with the simple case of diffusion into a sphere (Crank, 1975). This geometry poised us for subsequent simulations of mass transport within porous, adsorptive, spherical chromatography media beads. This also provides a test case where an analytical solution is available, permitting debugging of our coding and assessment of the numerical accuracy of the partial differential equation solver we chose to use. The system considered was a single sphere and an infinite surrounding bath of protein solution was assumed. The microscopic mass balance equation for the spherical volume (Equation 1), the initial condition (Equation 2), and the boundary conditions (Equation 3, 4) used for this case are shown below:

$$\varepsilon \frac{\partial c(r,t)}{\partial t} = \varepsilon D_{pore} \nabla^2 c(r,t), \tag{1}$$

$$c(r,0) = c_1 = 0, \tag{2}$$

$$c(R,t) = c_0, \tag{3}$$

and

$$\left.\frac{\partial c}{\partial r}\right|_{r=0} = 0. \tag{4}$$

Here, c is concentration of the protein within the liquid phase, D_{pore} is pore diffusivity of the solute within the bead pore space, ε is the porosity of the sphere, t is time, r is radial position, and R is the outer radius of the sphere. The sphere was assumed to have a uniform initial concentration c_1, and to be placed in an infinite bath of protein solution which maintained the surface of the sphere at a constant concentration, c_0.

The well-known analytical solution (Equation 5) for this case is available from (Crank, 1975):

$$\frac{c - c_1}{c_0 - c_1} = 1 + \frac{2R}{\pi r}\sum_{n=1}^{\infty}\frac{(-1)^n}{n}\ sin\frac{n\pi r}{R}\ exp\frac{-D_{pore}n^2\pi^2 t}{R^2}. \tag{5}$$

As r approaches 0, the concentration at the center of the sphere is shown below:

$$\frac{c - c_1}{c_0 - c_1} = 1 + 2\sum_{n=1}^{\infty}(-1)^n\ exp\frac{-D_{pore}n^2\pi^2 t}{R^2}. \tag{6}$$

Modeling Single Component Adsorption and Diffusion in a Sphere

The next step, was to build complexity into the model by incorporating solute binding to the porous spherical particle into the microscopic balance to mimic an adsorptive chromatographic media bead. Using the same boundary conditions as the first case, this modified diffusion equation is shown below:

$$\varepsilon\frac{\partial c(r,t)}{\partial t} + \frac{\partial q(r,t)}{\partial t} = \varepsilon D_{pore}\nabla^2 c(r,t). \tag{7}$$

Here, q is the concentration of complexed or absorbed protein per unit volume of wetted particle, representing the sum of the volumes of the pore liquid and the solid portion of the bead. This modified transport equation gives a material balance over the total volume of the spherical media bead, including the solute within the liquid phase of the pores and the solute adsorbed to the porous solid phase.

In order to numerically solve this equation, we must express the adsorbed phase accumulation $\frac{\partial q}{\partial t}$ in terms of the solute concentration, c. Two different cases may be taken into consideration. The first takes place under the assumption that the solute in complexed, or adsorbed, form is in equilibrium with solute in the pore liquid. Using the chain rule, $\frac{\partial q}{\partial t}$ can be expanded as the following equality:

$$\frac{\partial q}{\partial t} = \frac{\partial q}{\partial c} \cdot \frac{\partial c}{\partial t}. \tag{8}$$

The $\frac{\partial c}{\partial t}$ term is already present in the diffusion equation and at equilibrium conditions, an adsorption isotherm expression, such as the Langmuir isotherm shown below (Equation 9), can be used to define the relationship between c and q in $\frac{\partial q}{\partial c}$ (Pabst et al., 2018):

$$q = \frac{q_{max} K_a c}{1 + K_a c}, \tag{9}$$

giving

$$\frac{\partial q}{\partial c} = \frac{K_a c}{(1 + K_a c)^2}. \tag{10}$$

Here, K_a is the binding affinity constant and q_{max} is the maximum possible concentration of absorbed protein per unit of wetted porous solid volume. The Langmuir isotherm, studied

extensively in previous literature, has been shown to provide an accurate correlation between the liquid and adsorbed phase antibody concentrations in protein A affinity chromatography, despite the questionable validity of the assumptions underpinning this isotherm in this context (Bowes & Lenhoff, 2011; Lee et al., 2012; Pabst et al., 2018; Weinberg et al., 2017).

Since it is not clear whether adsorption in the current system can be assumed to be in equilibrium, the second case considers adsorption as a kinetic process. The following mass action kinetic equation (Equation 11) can be used to express $\frac{\partial q}{\partial t}$:

$$\frac{\partial q}{\partial t} = k_f c(q_{max} - q) - k_r q, \tag{11}$$

where k_f is the rate of the forward, or complex-forming/adsorption, reaction and $k_r = \frac{k_f}{K_a}$ is the rate of the reverse, or decomplexation, reaction. Equation 11 is thermodynamically consistent with the Langmuir isotherm expressed in Equation 9. This kinetic equation is coupled with the microscopic mass balance in Equation 7, giving a system of two partial differential equations to be solved simultaneously for q and c. Since the kinetic case gives an additional partial differential equation to solve that is first order in time, it is also necessary to add the corresponding initial condition for the solid phase:

$$q(r, 0) = \frac{K_a q_{max} c_1}{1 + K_a c_1} = 0. \tag{12}$$

Desorption can also be simulated with the above equilibrium and kinetic models. In this case, the particle is initially loaded with solute with some initial distribution in the liquid and solid phases, $c = c(r, 0)$ and $q = q(r, 0)$, respectively, the external boundary condition is changed to provide an external sink, $c_0 = 0$ or $c_0 > c(R, 0)$, and the binding equilibrium constant, K_a, or forward rate

constant, k_f, is set to some small value (or to zero) for the equilibrium or kinetic cases, respectively.

Dimensionless Adsorption and Diffusion Modeling Equations

To create redundancy and double check the equations used in the computational simulations, the dimensionless forms of the above equations were also generated. In order to characterize the system in dimensionless terms, the following substitutions were made:

$$c^* = \frac{c - c_1}{c_0 - c_1}, \tag{13}$$

$$r^* = \frac{r}{R}, \tag{14}$$

$$t^* = \frac{t}{t_{reference}} = \frac{t}{R^2/D_{pore}}, \tag{15}$$

and

$$q^* = \frac{q}{q_{max}}. \tag{16}$$

When combined with the previous dimensional modified diffusion equation, the new dimensionless microscopic mass balance is given by Equation 17:

$$\frac{\partial c^*}{\partial t^*} + \frac{q_{max}}{(c_0 - c_1)}\frac{\partial q^*}{\partial t^*} = \nabla^2 c^* \tag{17}$$

The dimensionless initial condition (Equation 18) and boundary conditions (Equation 19, 20) are also shown below:

$$c^*(r^*, 0) = c_1 = 0, \tag{18}$$

$$c^*(1, t^*) = 1, \tag{19}$$

and

$$\left.\frac{\partial c^*}{\partial r^*}\right|_{r^*=0} = 0. \tag{20}$$

Evaluating the equilibrium case in dimensionless terms yields the following chain rule (Equation 21) and Langmuir isotherm (Equation 22).

$$\frac{\partial q^*}{\partial t^*} = \frac{\partial q^*}{\partial c^*} \cdot \frac{\partial c^*}{\partial t^*} \tag{21}$$

$$q^* = \frac{K_a(c^*(c_0 - c_1) + c_1)}{1 + K_a(c^*(c_0 - c_1) + c_1)} \tag{22}$$

Also considering the kinetic case, the following equation dimensionless kinetic equation and corresponding initial condition are obtained:

$$\frac{\partial q^*}{\partial t^*} = k_f \frac{R^2}{D}(c^*(c_0 - c_1) + c_1) * (1 - q^*) - k_r \frac{R^2}{D}, \tag{23}$$

and

$$q^*(r^*, 0) = \frac{K_a c_1}{1 + K_a c_1} = 0. \tag{24}$$

These dimensionless equations were numerically solved in the same way as the dimensional equations and the results compared to expectations.

Modeling Noncompetitive Multicomponent Binding

In a polyclonal antibody solution, antibodies are not a homogeneous population of species. This is the case for solutions taken from blood serum. At 90%, IgG-class antibodies are the main antibody type produced in the blood (Murphy, 2012). There are three subclasses of IgG which bind to protein A, IgG_1, IgG_2, and IgG_4, and one that does not, IgG_3 (Burton, 1985; Ghose et al., 2005);

these subclasses are numbered in order of decreasing relative abundance in normal human serum. Extrapolating from this information and observing protein A pH gradient elution profiles in the literature (Weinberg et al., 2017), it can be determined that polyclonal antibody solutions taken from blood serum could contain at least four main groupings of components, an unretained component and three retained components. The retained components may bind competitively with each other, as they have differing affinities for protein A as determined by differences in their elution pH, and there are a finite number of protein A binding sites in the media. As a stepping stone to the consideration of competitive binding, the adsorptive behavior of multiple antibody components in the system were first simulated noncompetitively. Four components were simulated initially, to represent the mAb and three distinct groupings of binding species in the hIgG polyclonal antibody solution studied by Weinberg et al., 2017. We note that the mapping of the three observed binding groupings onto distinct $IgG_{1,2,4}$ subspecies is indeterminate from these experimental results at this point. In the model, each component, $i = 1$ (mAb), and 2,3,4 (hIgG), has its own set of independent microscopic mass balances. Equations 25, 26, and 27 represents the equilibrium case:

$$\varepsilon \frac{\partial c_i}{\partial t} + \frac{\partial q_i}{\partial t} = \varepsilon D_{pore} \nabla^2 c_i, \tag{25}$$

$$\frac{\partial q_i}{\partial t} = \frac{\partial q_i}{\partial c_i} \cdot \frac{\partial c_i}{\partial t}, \tag{26}$$

and

$$q_i = \frac{q_{max,i} K_{a,i} c_i}{1 + K_{a,i} c_i}. \tag{27}$$

Each component also had its own corresponding initial and boundary conditions, which describe adsorption or uptake into the sphere that were dependent on the species' prevalence in solution. These can be seen below where x_i is the mass fraction of binding species in the polyclonal solution:

$$c_i(r,0) = c_1 = 0, \tag{28}$$

$$\left.\frac{\partial c_i}{\partial r}\right|_{r=0} = 0, \tag{29}$$

$$c_{i=1}(R,t) = c_{0,mAb}, \tag{30}$$

and

$$c_{i=2,3,4}(R,t) = c_{0,\Sigma hIgG} * x_i. \tag{31}$$

Here, $c_{0,\Sigma hIgG}$ represents the total solution concentration of polyclonal antibody. For the kinetic case, the following model was used:

$$\frac{\partial q_i}{\partial t} = k_{f,i}c_i\left(q_{max,i} - q_i\right) - k_{r,i}q_i. \tag{32}$$

The boundary and initial conditions for the liquid phase are the same as above, while the solid phase initial condition is given below:

$$q_i(r,0) = \frac{q_{max,i}K_{a,i}c_{1,i}}{1 + K_{a,i}c_{1,i}} = 0 \tag{33}$$

Modeling Competitive Binding

Moving forward, the equations describing competitive binding in the equilibrium and kinetic cases are the same as those for noncompetitive binding, with the exception that the Langmuir isotherm and the dynamic binding equation must be modified to account for the competition for the finite

number of available binding sites between the components in solution. The competitive form of the Langmuir isotherm is given by (Carta & Jungbauer, 2010).

$$q_i = \frac{q_{max,i} K_{a,i} c_i}{1 + \sum_{j=1}^{N} K_{a,j} c_j} \tag{34}$$

Note that in Equation 34, each solute has its own maximum binding capacity. This is a commonly used, but thermodynamically inconsistent empirical modification of the fundamental competitive Langmuir isotherm equation wherein each solute competes for the same binding sites and has the same maximum binding capacity, q_{max}.

For the kinetic formulation, the following form is frequently used (T. Hahn et al., 2015; Thornton & Johnson, 1996),

$$\frac{\partial q_i}{\partial t} = k_{r,i} K_{a,i} c_i q_{max,i} \left(1 - \sum_{j=1}^{N} \frac{q_j}{q_{max,j}} \right) - k_{r,i} q_i, \tag{35}$$

where, the forward and reverse binding rate constants are connected by the effective equilibrium constant,

$$K_{eff,i} = \frac{k_{f,i}}{k_{r,i}}. \tag{36}$$

Note that this kinetic formulation is also thermodynamically inconsistent.

The equations to be solved for competitive binding are a system of i coupled partial differential equations for the equilibrium case and $2i$ coupled equations for the kinetic case. These equations must be solved simultaneously in each case. The competitive binding model can also be converted to its dimensionless form. These equations are as follows:

$$q_i^* = \frac{K_{a,i}\frac{(c_i^* - c_{1,i})}{(c_{0,i} - c_{1,i})}}{1 + \sum_{j=1}^{N} K_{a,j}\frac{(c_j^* - c_{1,j})}{(c_{0,j} - c_{1,j})}}, \tag{37}$$

and

$$\frac{\partial q_i^*}{\partial t^*} = \frac{k_{r,i}K_{eff,i}t_{ref}(c_i^* - c_{1,i})}{q_{max,i}(c_{0,i} - c_{1,i})}\left(1 - \sum_{j=1}^{N} q_j^* q_{max,j}\right) - q_i^* k_{r,i} t_{ref}. \tag{38}$$

Reproduction of Experimental Results

The first step in validating the above competitive computational model, was to computationally reproduce the experimental results where the displacement phenomenon was first observed. In these experiments, a resin bead with pre-adsorbed mAb was placed into a polyclonal solution of hIgG antibodies where displacement of the bound mAb by antibodies in solution was observed. This situation was simulated by breaking up the model into two phases – phase I comprised adsorption of the mAb, and phase II comprised the subsequent displacement of the mAb due to the competitive binding of the hIgG. For phase I, the mAb is the only component present. The partial differential equations describing the microscopic mass balances are as above with the initial and boundary conditions are as follows:

$$c_i(r, 0) = 0 \ \ \forall i, \tag{39}$$

$$\left.\frac{\partial c_i}{\partial r}\right|_{r=0} = 0 \ \ \forall i, \tag{40}$$

$$c_{i=1}(R, t) = c_{0,mAb}, \tag{41}$$

and

$$c_{i=2,3,4}(R, t) = 0. \tag{42}$$

For the kinetic case, the initial conditions for the solid phase are:

$$q_i(r,0) = \frac{q_{max,i} K_{a,i} c_{1,i}}{1 + K_{a,i} c_{1,i}} = 0 \ \ \forall i. \tag{43}$$

After phase I is solved, the outputs of $c(r, t_{final})$ and $q(r, t_{final})$ at the final time point t_{final} become the initial conditions for the mAb component in phase II. Phase II is solved using the same microscopic balances with the initial and boundary conditions as follows:

$$c_{i=1}(r,0) = c_{1,mAb,phase\ I}\left(r, t_{final}\right), \tag{44}$$

$$c_{i=2,3,4}(r,0) = 0, \tag{45}$$

$$\left.\frac{\partial \mathrm{c_i}}{\partial \mathrm{r}}\right|_{\mathrm{r=0}} = 0 \ \ \forall \mathrm{i}, \tag{46}$$

$$c_{i=1}(R,t) = 0, \tag{47}$$

and

$$c_{i=2,3,4}(R,t) = c_{0,\Sigma hIgG} * x_i. \tag{48}$$

For the kinetic case, the additional initial conditions for the solid phase are below.

$$q_{i=1}(r,0) = q_{1,mAb,phase\ I}(r, t_{final}) \tag{49}$$

and

$$q_{i=2,3,4}(r,0) = \frac{q_{max,i} c_{1,i} K_{a,i}}{1 + c_{1,i} K_{a,i}} = 0. \tag{50}$$

The computational outputs and experimental images are compared in the results section.

Microscopic Binding Behavior

To fully characterize the system, it is important to understand the binding interaction between protein A and hIgG at the molecular level. Complexation occurs between protein A binding subunits and the C_H2-C_H3 cleft of hIgG-class antibodies. The contact area between the C_H2-C_H3 cleft and the protein a binding domain is approximately 1,234 $Å^2$, comprising mostly hydrophobic surface area which becomes buried upon binding. This cleft also contains three histidine residues at amino acid positions 435, 310, and 433 (Deisenhofer, 1981). Protein A also has a singular histidine residue within each of its five antibody binding domains, and at amino acid position 137 in the fraction B (FB) domain in particular (Deisenhofer, 1981). These histidine residues control the binding interaction between hIgG and Protein A as protonation of these residues make it energetically unfavorable for these complexes to form (DeLano et al., 2000; Huang et al., 2011). This behavior is demonstrated by the conditions under which protein A affinity chromatography operates: antibody is loaded onto the column at neutral pH where the histidine residues are deprotonated and the pH is subsequently lowered, protonating the histidine residues in the uncomplexed state, to effect elution (Carta & Jungbauer, 2010). The binding interaction is further illustrated in the figures below which were created from the 1FC2 crystal structure from Deisenhofer (1981) found in the Protein Data Bank (Berman, et al., 2000) using RASMOL (Sayle & Milner-White, 1995).

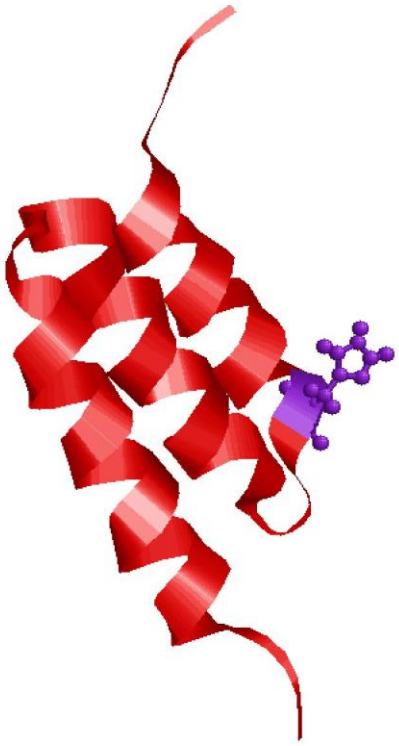


Figure 1: Protein A FB domain.
Backbone ribbon structure shown in red with histidine residue 137 and its imidazole side chain highlighted in purple.

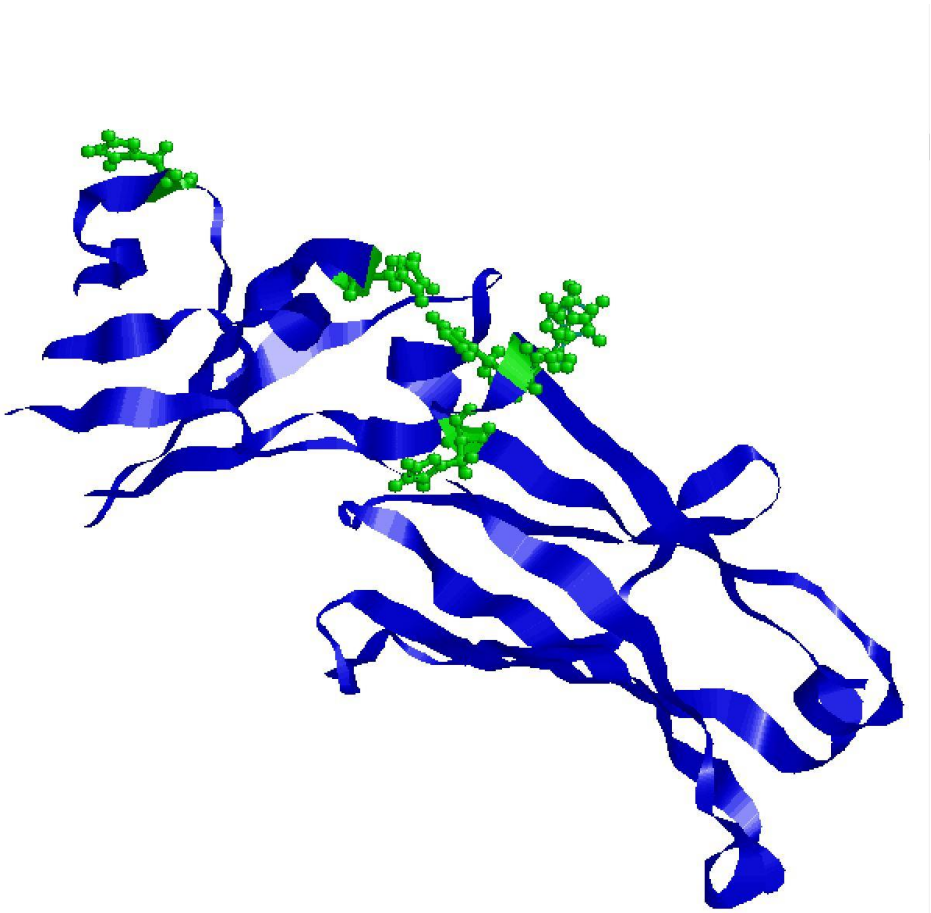


Figure 2: hIgG C_H2-C_H3 domain.
Backbone ribbon structure shown in blue with histidine residues 310, 433, and 435, and their imidazole side chains highlighted in green.

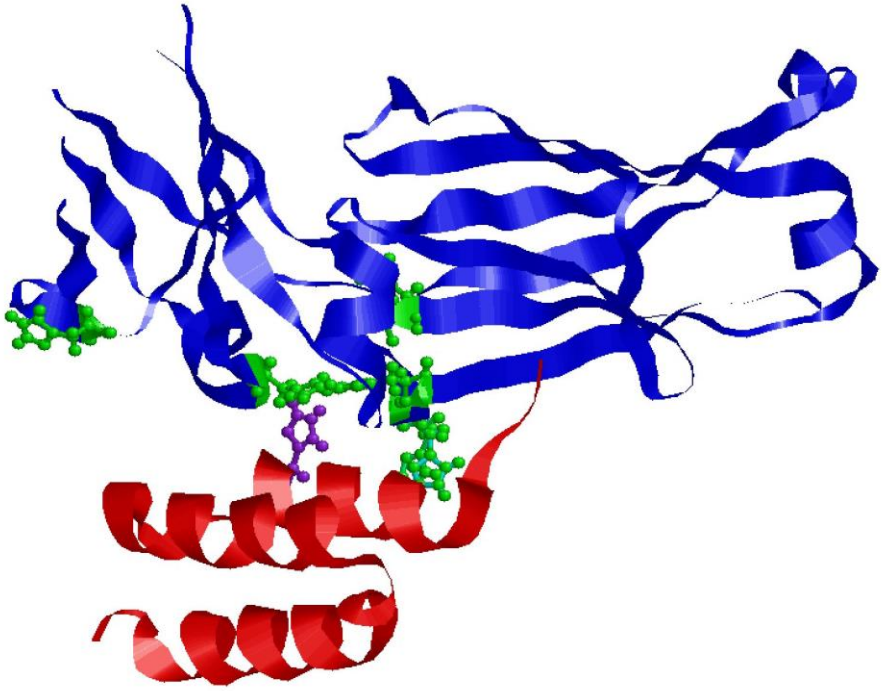


Figure 3: Protein A FB domain – hIgG domain complex structure.
Color scheme same as figures 1 and 2.

Here, we present two different models to account for the impact of histidine side chain protonation on the formation of protein A – hIgG complexes. The first model assumes that a complex ("AG") cannot form if a histidine residue within the binding domain of either the protein A ligand ("A") or the IgG-class antibody ("G") is protonated:

$$A + G \overset{K_{bind}}{\longleftrightarrow} AG,$$

$$G^+ \overset{K_a''}{\longleftrightarrow} H^+ + G,$$

and

$$A^+ \overset{K_a'}{\longleftrightarrow} H^+ + A.$$

The second model takes into consideration the possibility that a complex may still form if a histidine residue in either the protein A ligand or the IgG-class antibody, but not both, is protonated:

$$A + G \overset{K_{bind}}{\longleftrightarrow} AG,$$

$$G^+ \overset{K_a''}{\longleftrightarrow} H^+ + G,$$

$$A^+ \overset{K_a'}{\longleftrightarrow} H^+ + A,$$

$$A + G^+ \overset{K_{bind}''}{\longleftrightarrow} AG^+,$$

and

$$G + A^+ \overset{K_{bind}'}{\longleftrightarrow} A^+G.$$

These two mechanistic models were used to develop new pH-dependent isotherms to describe protein A- antibody binding interactions.

Modeling pH Effect on the Binding Isotherm

Binding between protein A and IgG-class antibodies is clearly dependent on the pH of the system. Further, there are antibody-to-antibody differences in the strength of the binding interaction that are more pronounced in the wild type protein A ligand, as compared to an engineered protein A ligand, because of the secondary binding interaction with the Fab region of the antibody. These variations in binding strength are reflected by differences in the depth of the acidic pH required for elution, with lowered required elution pHs indicating stronger binding. Even though the basic Langmuir isotherm represents observed binding behavior well, the equilibrium binding constant must be manually adjusted to represent binding and elution behavior as formulated in Equations 9

and 34. We desire a new isotherm which incorporates pH explicitly into the relationship between the concentration of complex state q and concentration of antibody species in solution, c, to connect elution pH to binding strength. There exist empirical isotherms which explicitly consider pH effects on binding behavior. Two such adsorption relationships are shown below in Equation 51 (Nilsson & Andersson, 2017) and Equation 52 (Sandoval et al., 2012):

$$q_i = \frac{q_{max,i} K_i \left(pH_{ref,i}/pH\right)^n c_i}{1 + K_i \left(pH_{ref,i}/pH\right)^n c_i} \tag{51}$$

and

$$q_i = \frac{q_{max,i} K_i \cdot 10^{\gamma_i \cdot pH} c_i}{1 + K_i \cdot 10^{\gamma_i \cdot pH} c_i}. \tag{52}$$

However, we are interested in a relationship that connects complexation and decomplexation behavior to pH in a fundamental manner. Therefore, we chose to develop our own pH-based isotherm, grounded in the mechanistic binding models described above.

For model 1 of binding, the associated material balances, (incorporating corresponding solution and solid phase concentration definitions) and equilibrium equations are shown below:

$$[A_{total}] = [AG] + [A] + [A^+] = q + [A] + [A^+] = q_{max}, \tag{53}$$

$$[G_{total}] = [AG] + [G] + [G^+] = q + c, \tag{54}$$

$$K_{bind} = \frac{[AG]}{[A][G]}, \tag{55}$$

$$K''_a = \frac{[G][H^+]}{[G^+]}, \tag{56}$$

and

$$K'_a = \frac{[A][H^+]}{[A^+]}. \tag{57}$$

Combining the above equations gives a new isotherm of the same form as the Langmuir isotherm:

$$q = \frac{q_{max} K_{bind,0} c}{\left(1 + \frac{[H^+]}{K'_a}\right)\left(1 + \frac{[H^+]}{K''_a}\right) + cK_{bind}}. \tag{58}$$

This new isotherm gives q as a function of c, the total solution concentration of protonated and deprotonated antibody, as well as pH, q_{max}, the histidine residue dissociation constants, and the intrinsic antibody binding affinity constant, $K_{bind,o}$. In model 1, it was assumed that a complex between protein A and IgG could only form if neither of these components were protonated. Model 2 considers the possibility that a complex could still form if either the protein A or IgG component was protonated. Therefore, the material balances must be modified and two new complexation equilibria must be added to our first model. The modified/additional equations for model 2 are shown below:

$$[A_{total}] = [AG] + [A^+G] + [AG^+] + [A] + [A^+] = q + [A] + [A^+] = q_{max}, \tag{59}$$

$$[G_{total}] = [AG] + [A^+G] + [AG^+] + [G] + [G^+] = q + c, \tag{60}$$

$$K'_{bind} = \frac{[AG^+]}{[A][G^+]}, \tag{61}$$

and

$$K''_{bind} = \frac{[GA^+]}{[G][A^+]}. \tag{62}$$

Again, combining the previous equations and material balances, yields the following equation:

$$q = \frac{q_{max}\left(K_{bind,o} + K'_{bind}\frac{[H^+]}{K'_a} + K''_{bind}\frac{[H^+]}{K''_a}\right)c}{\left(1 + \frac{[H^+]}{K'_a}\right)\left(1 + \frac{[H^+]}{K''_a}\right) + \left(K_{bind,o} + K'_{bind}\frac{[H^+]}{K'_a} + K''_{bind}\frac{[H^+]}{K''_a}\right)c}. \quad (63)$$

As expected, this isotherm is similar in form to the previous isotherm developed in model 1, however, the equilibrium binding constant term has been expanded to account for the possibility of the formation of two additional types of complexes.

Connecting pH to Free Binding Energy

The newly developed isotherms can be used to connect pH to the free energy of binding using the following equation.

$$\Delta G_{bind,i} = \Delta G_{bind,o,i} + \Delta G_{pH} = -RTlnK_{bind,o,i} + \Delta G_{pH} = -RT\,ln\,K_{eff,i} \quad (64)$$

Here, ΔG is free energy, R is the ideal gas constant, T is temperature, and K_{eff} is the effective or observed binding constant at a given pH. The binding free energy has been parsed into an intrinsic binding contribution and a pH-dependent contribution. This effective binding constant is taken from the isotherm when it is in the following general form:

$$q_i = \frac{q_{max,i}K_{eff,i}c_i}{1 + K_{eff,i}c_i} \quad (65)$$

Thus, $K_{eff,i}$ for model 1 and model 2, respectively, is given by:

$$K_{eff,i} = \frac{K_{bind,o,i}}{\left(1 + \frac{[H^+]}{K'_a}\right)\left(1 + \frac{[H^+]}{K''_a}\right)} \quad (66)$$

and

$$K_{eff,i} = \frac{\left(K_{bind,o,i} + K'_{bind,i}\frac{[H^+]}{K'_a} + K''_{bind,i}\frac{[H^+]}{K''_a}\right)}{\left(1+\frac{[H^+]}{K'_a}\right)\left(1+\frac{[H^+]}{K''_a}\right)}. \quad (67)$$

As it is energetically costly to bury a charged group within a hydrophobic patch (Israelachvili, 2011) as allowed by the additional complexation equilibria in model 2, we'll focus exclusively on model 1 in subsequent sections as the more likely scenario.

Solving the PDE System

MATLAB was chosen as the platform for the computational model as it had the ability to solve parabolic partial differential equations and also allowed for the convenient graphical visualization of the results. The MATLAB routine used to solve the equations was *pdepe*. This routine uses the finite difference method for solution, a discretization method which solves partial differential equations by approximating the spatial derivatives with difference equations. The resulting set of ordinary differential equations in time are solved within *pdepe* using the *ode15s* routine, an implicit, backward differentiation routine for stiff equations. Relative tolerance, the error relative to the size of each state, was left at its default value of 10^{-3}, meaning that the computed state is within 0.1% accuracy. Absolute tolerance, the acceptable error as the measured state approaches zero, must satisfy the following equation for the ith state, e_i (MathWorks, 2018):

$$e_i \leq max\left((relative\ tolerance) * |r_i|, (absolute\ tolerance)_i\right), \quad (68)$$

where r_i is the radial position of component i. Absolute tolerance was left at its default value of $e_i = 10^{-6}$. The spatial grid points for radial position in the MATLAB computational model were defined as vector r and could be manually adjusted through the *parameters.m* script (see Appendix V). Figures and plots shown in the results section were modeled using a minimum of 70 radial grid points. Spacing between radial points was even. Computational scripts for the more complex kinetic simulations included a split in the r vector and defined the radial positions closer to the edge separately to allow for an increased number of points near the edge. This was necessary to decrease the error along the edge of the sphere without overburdening the solver. Model parameter values were taken from relevant literature or estimated and are listed with their sources in Appendix II.

Results and Discussion

Diffusion in a Porous Sphere

Using the MATLAB generated solution of Equation 1, two-dimensional contour plots of solute solution phase concentration within a porous sphere at different solute uptake times were produced. This visualization approach was chosen to emulate the cross-sectional, confocal laser-scanning microscopy (CLSM) images obtained experimentally for single chromatography beads exposed to a fluorescently-labelled antibody solution. The unit of concentration was mg antibody per mL pore liquid. The simulations showed that it took less than a minute for a solute with the typical pore diffusivity of an antibody (See Appendix II) to diffuse to the center of a non-adsorptive sphere the size of an average affinity chromatography resin bead (Figure 4).

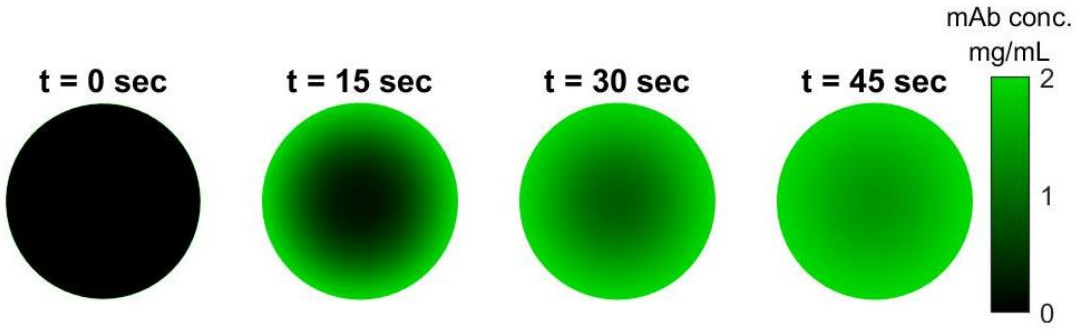


Figure 4: Simulation of mAb diffusion into a porous sphere.
Solution phase concentration profiles at several sequential time intervals. Key simulation parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

Here, and in subsequent similar figures, the concentration is highest where the color is brightest; a scale bar is given in each case. Experimental results for the uptake of mAb into protein A media resin beads in an infinite bath showed much longer time scales for the antibody to penetrate to the center of similarly-sized resin beads (Weinberg et al., 2017), as expected for adsorptive media.

Analytical Solution for Diffusion

To check the accuracy of MATLAB's *pdepe* solver, the simulations in Figure 4 above were compared to the analytical solution given in Equations 5 and 6. The analytical series solution was approximated using the first one thousand terms ($n = 1{,}000$) using MATLAB. The cross-sectional solute solution phase concentration contour plots for the analytical solution are shown below in Figure 5. The percent error between the analytical solution and MATLAB simulation is shown below in Figure 6 for selected time points.

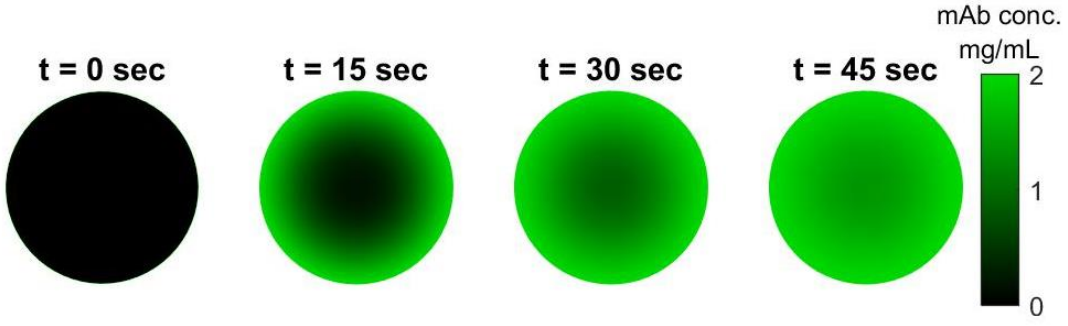


Figure 5: Analytical solution for mAb diffusion into a porous sphere from analytical solution. Key analytical solution parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

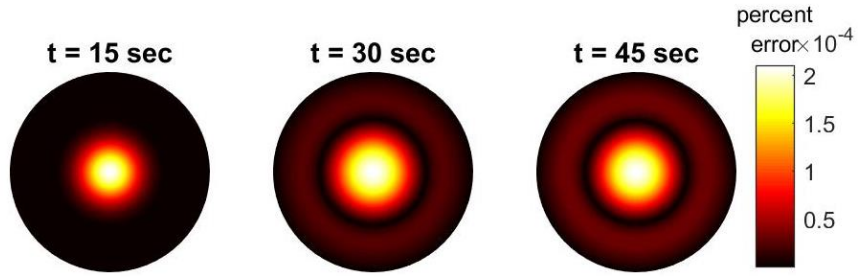


Figure 6: Percent difference between MATLAB simulation and analytical solution. This is for diffusion in a porous sphere.

The difference between the MATLAB simulation and the analytical solution is a 0.0002% error at its greatest point. Therefore, we were satisfied that the MATLAB *pdepe* solver would give reasonable solutions for the microscopic mass balances of interest.

Adsorption and Diffusion in a Single Chromatographic Bead, Equilibrium Case

Using the adsorption and diffusion transport equation introduced in Equation 7, a series of contour plots of the solution and solid phase solute concentrations during uptake were generated from the MATLAB simulations. The images represent the cross section of a protein A chromatography media sphere at increasing times. The first case considered was the equilibrium adsorption case. Figure 7 shows antibody movement through the porous media in the liquid phase of the system, c, in milligrams of antibody per milliliter of pore liquid and Figure 8 shows the corresponding adsorbed antibody concentration, q, in milligrams of antibody per milliliter of wetted bead.

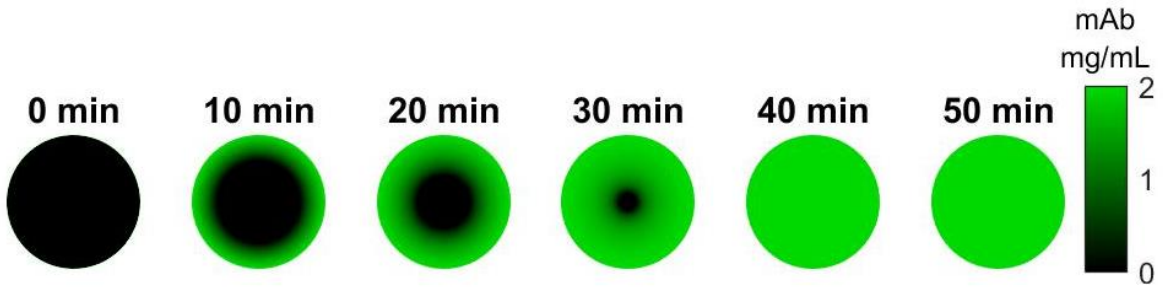


Figure 7: Solution phase antibody concentration profiles.
Diffusion and adsorption for the equilibrium binding case. Key parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

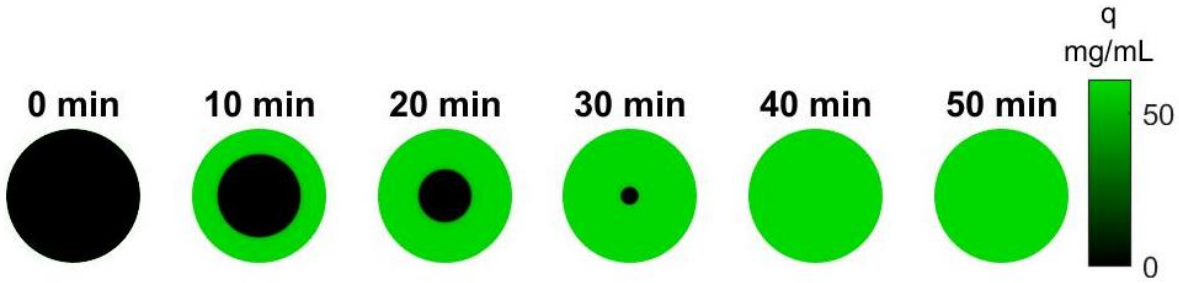


Figure 8: Solid phase antibody concentration profiles.
Diffusion and adsorption for the equilibrium binding case. Key parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

Compared to the pure diffusion case, when adsorption is taken into account, the time scales for the progress of both the solution and solid phase concentration fronts are much more consistent with that of experimental antibody uptake data (Weinberg et al., 2017). These timescales that include adsorption are on the order of minutes rather than seconds when only diffusive transport is considered. It can also be seen from Figure 8, that the results for the solid phase concentration resemble the behavior of a shrinking core model. There is a relatively sharp, defined advancing "front" when observing the concentration profiles of the protein A – antibody complexes. These figures show that the front of the solid phase concentration moves more quickly towards the center of the sphere than the front of the liquid phase concentration. This is understandable since this is an equilibrium with a strong binding isotherm; as soon as antibody in solution at any appreciable concentration reaches an open binding spot, the complex will form.

Analytical Solution for Shrinking Core Model

The shrinking core model represents a system with infinitely strong binding where the antibody is localized in an outer shell of the particle at each time point until saturation. Pore diffusion drives the transport in the solution phase. This results in progressively thicker shells of adsorbed solute

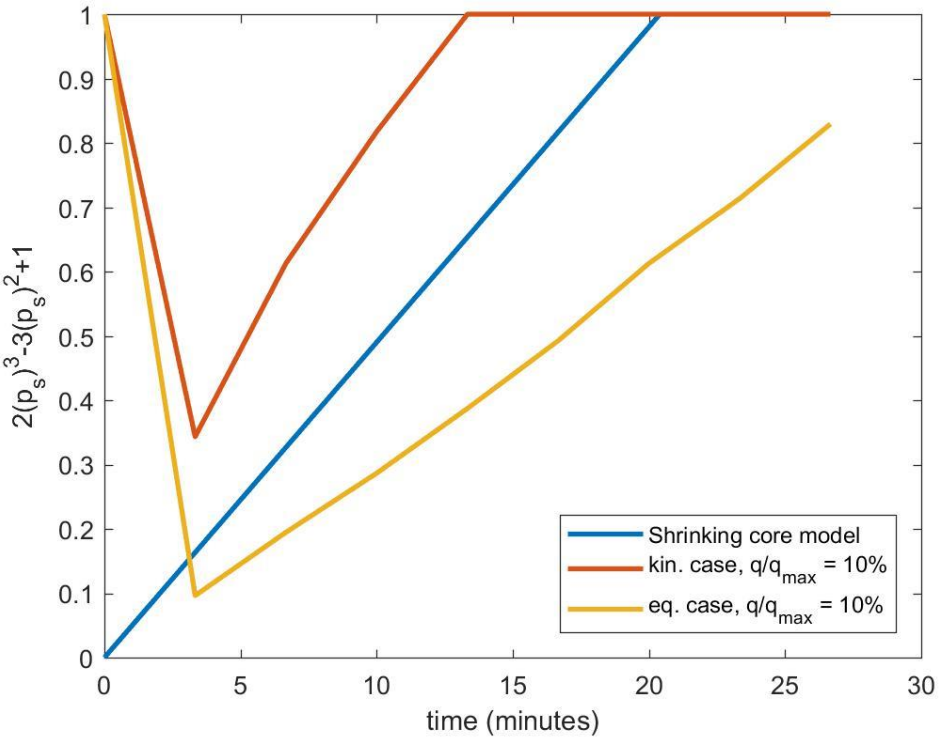
during uptake and an antibody-free core that shrinks over time (Carta & Jungbauer, 2010). Since there is an analytical solution for the shrinking core model from Martin et al., 2005, it was necessary to perform a comparison with simulated results. The shrinking core model is shown below:

$$2\rho_s^3 - 3\rho_s^2 + 1 = \frac{6D_0c_0t}{q_{max}R^2}\frac{D_{pore}}{D_0} \tag{69}$$

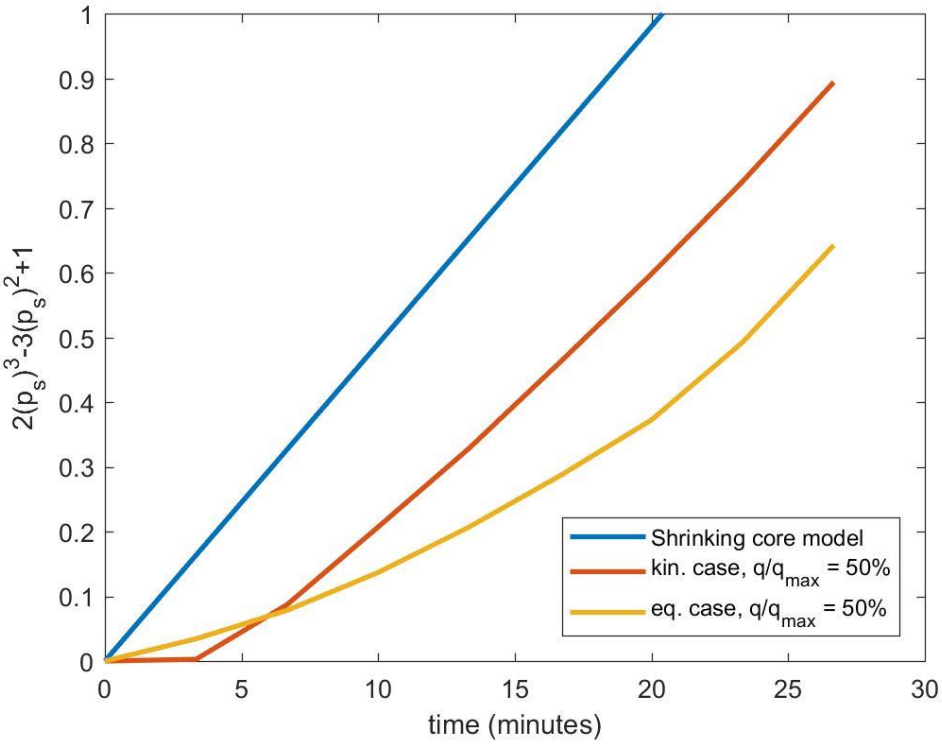
where

$$D_{pore} = \frac{\varepsilon D_0}{\tau}. \tag{70}$$

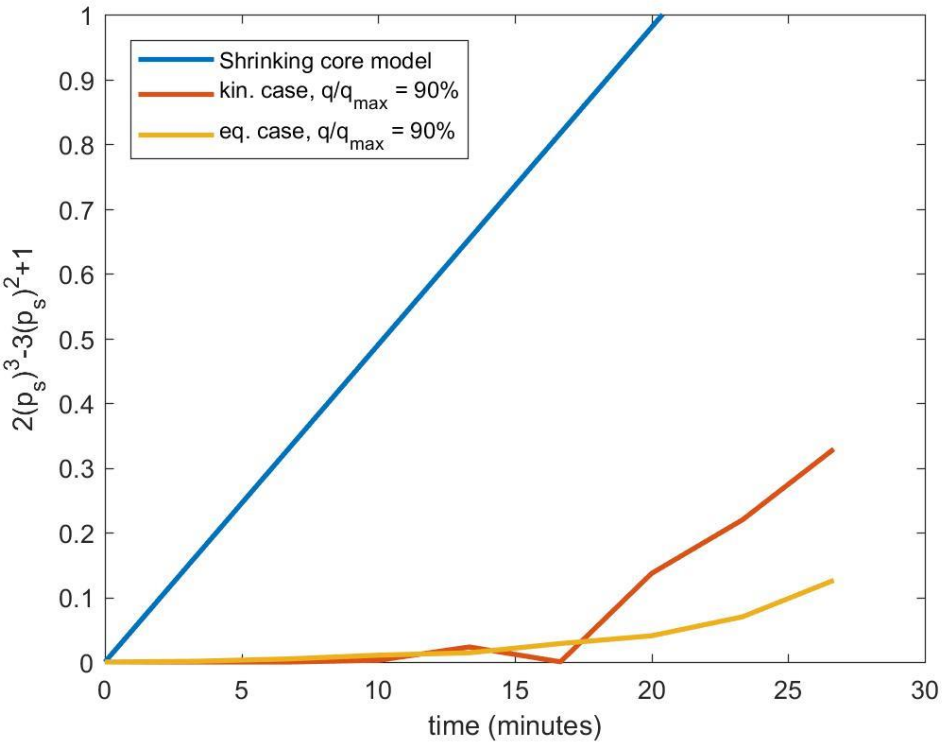
Here, $\rho_s = \frac{r}{R}$ is the dimensionless front position within the resin particle, c_0 is the concentration in the bulk solution, D_{pore} is the effective pore diffusivity, D_0 is the bulk diffusivity of hIgG molecules, τ is resin particle tortuosity, and t is time of adsorption(Martin et al., 2005; Perez-Almodovar & Carta, 2009). These parameters can be found in Appendix II. Figure 9 below shows the moving front of antibody from the shrinking core model (SCM), $2\rho_s^3 - 3\rho_s^2 + 1$, compared to the position of moving front of adsorbed antibody for the kinetic and equilibrium cases at 10%, 50% and 90% of q_{max}.


2(p_s)^3-3(p_s)^2+1
time (minutes)
Shrinking core model
kin. case, q/q_max = 10%
eq. case, q/q_max = 10%

(A)


2(p_s)^3-3(p_s)^2+1
time (minutes)
Shrinking core model
kin. case, q/q_max = 50%
eq. case, q/q_max = 50%

(B)



(C)

Figure 9: Moving front of antibody in shrinking core model compared to model over time.
Moving front, $2\rho_s^3 - 3\rho_s^2 + 1$, of shrinking core model shown in blue, kinetic case shown in red, and equilibrium case shown in yellow, for (A) $\frac{q}{q_{max}} = 10\%$ (B) $\frac{q}{q_{max}} = 50\%$, and (C) $\frac{q}{q_{max}} = 90\%$.

Adsorption and Diffusion in a Single Chromatographic Bead, Kinetic Case

We also simulated the kinetic binding case for the microscopic mass balances given by Equations 7 and 11. The simulation results for the antibody concentration in the solution and adsorbed phases are shown below in Figures 10 and 11, respectively.

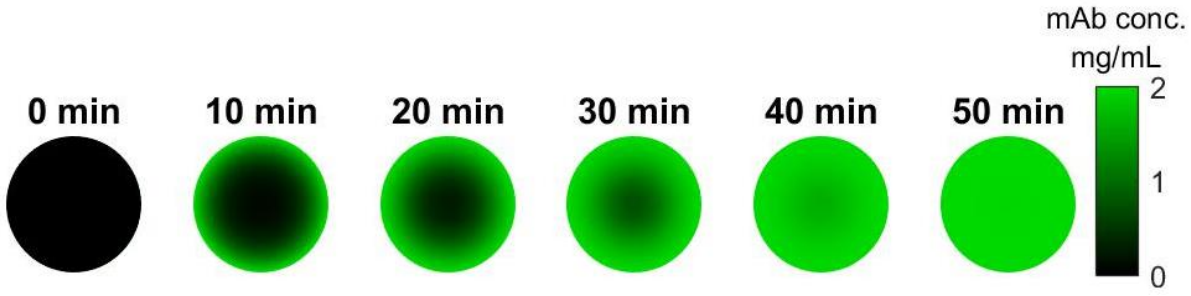


Figure 10: Solution phase antibody concentration profiles.
Diffusion and adsorption for the kinetic binding case. Key parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

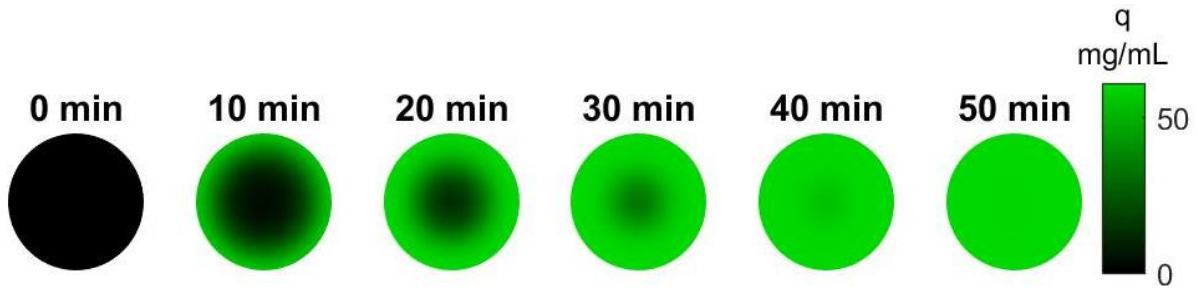


Figure 11: Solid phase antibody concentration profiles.
Diffusion and adsorption for the kinetic binding case. Key parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

In comparing the kinetic and equilibrium binding cases, it can be seen that antibodies in the kinetic case take longer to reach the center of the sphere and the advancing antibody fronts are more diffuse than the equilibrium case for both the solution and adsorbed phase concentrations. This observation is reasonable as the kinetic binding case will always have slower dynamics than the corresponding instantaneous, equilibrium binding case. And, as a result, Figure 11 does not exemplify the shrinking core model. Figure 12 below shows a plot of radial position vs. time where the value of $\frac{q}{q_{max}} = 50\%$ and $\frac{c}{c_0} = 50\%$.. This shows the progress of the binding front for antibody movement towards the center of the sphere.

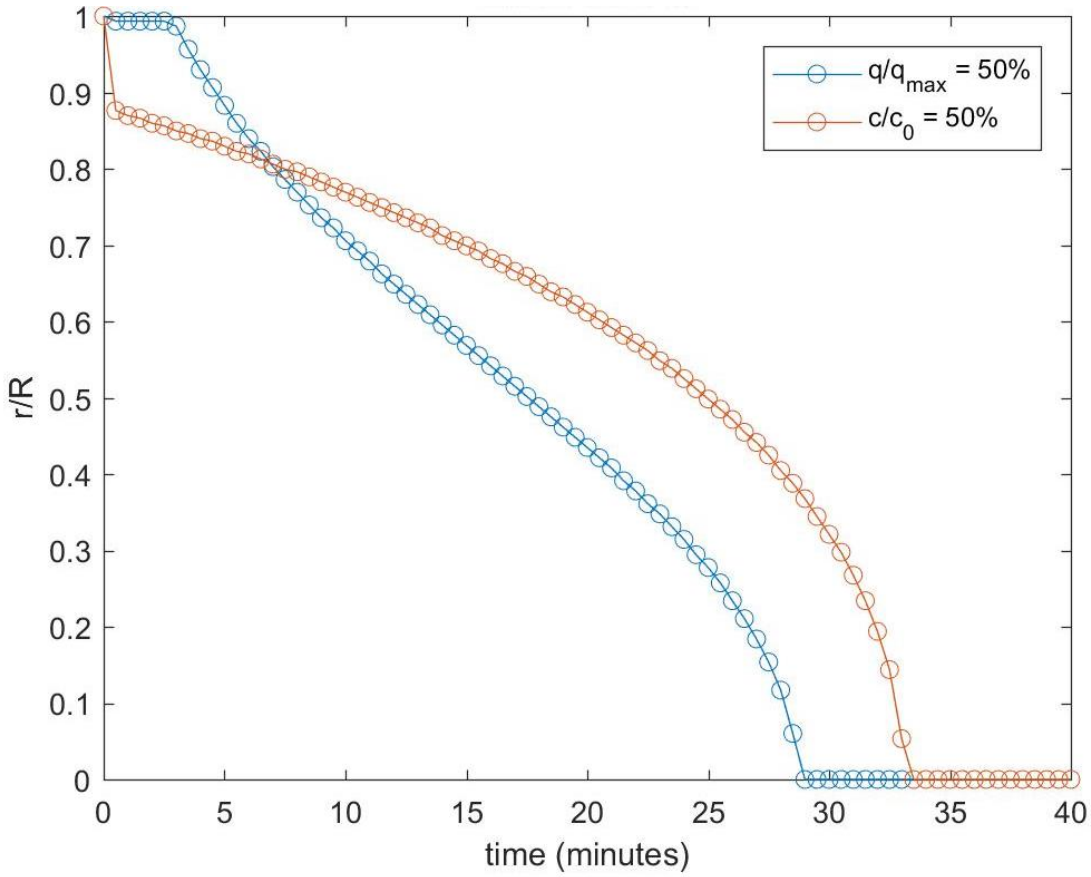


Figure 12: Progress of the position where q/q_{max} = 0.5 and c/c_0 = 0.5 versus time. Key parameters R, ε, D_{mAb}, and c_0 are found in Appendix II.

Estimation of Component Isotherm Parameters

The previous simulations were all run using the diffusion and adsorption behavior of a single, representative mAb component. In order to add multiple components to the model to represent a polyclonal antibody solution, it was important to distinguish the relative quantities of the different components and to estimate their individual binding constants and binding capacities. This data came from experimental pH gradient elution profiles for a mAb solution and a polyclonal hIgG solution (Weinberg et al., 2017).

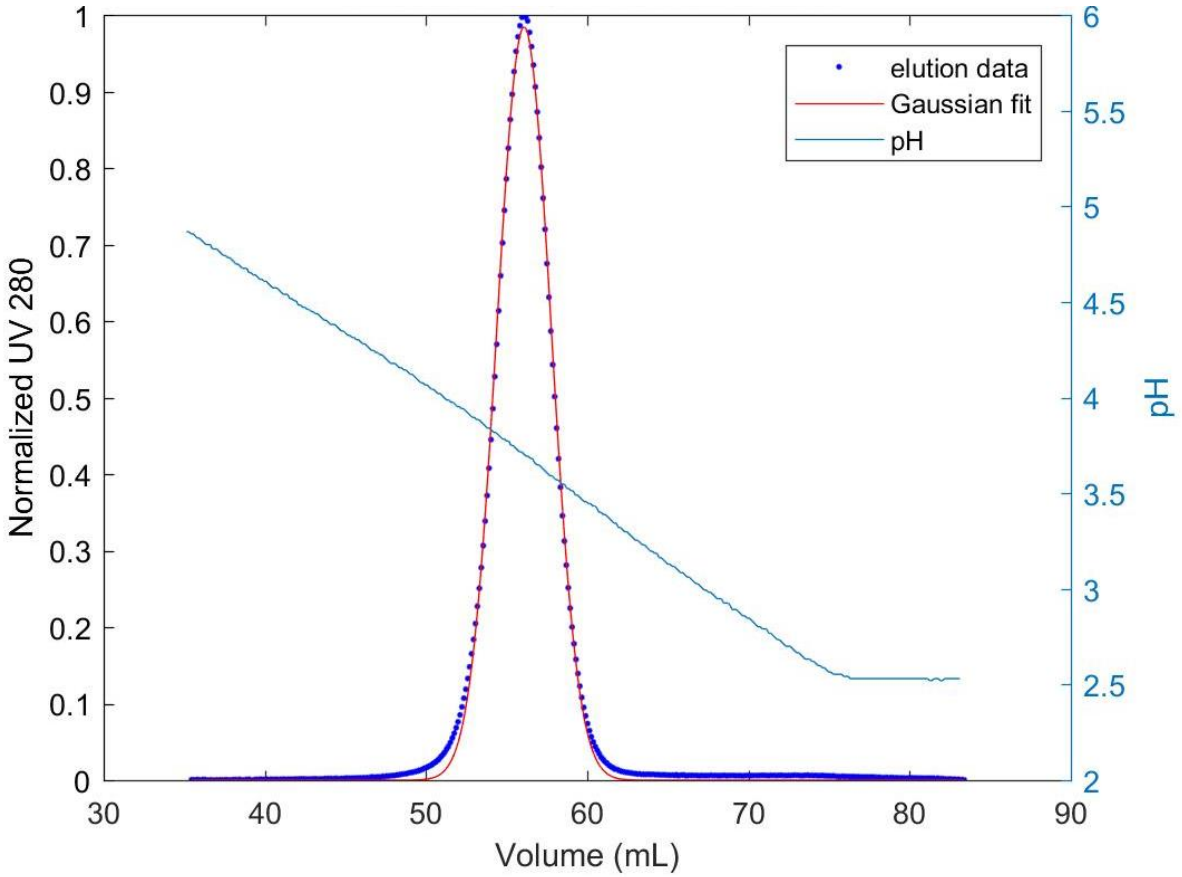


Figure 13: Gradient elution behavior for a mAb on CaptivA PriMab protein A media.
The experimental elution profile and pH gradient are shown along with a Gaussian fit to the elution peak. Data taken from (Weinberg et al., 2017).

Figure 13 above shows the fit of the gradient elution profile of a mAb solution from a CaptivA PriMab protein A column with a single Gaussian curve with a constant baseline contribution. This is consistent with the elution behavior of a single solute species. For the hIgG elution data shown below in Figure 14, it can be seen from the shoulders of in the elution envelop, that a polyclonal hIgG solution which has been eluted from a CaptivA PriMab protein A chromatography column has at least three main components. The observed hIgG elution envelope is well represented by three overlapping Gaussian peaks with a constant baseline contribution. This analysis is consistent with what is known from the literature about hIgG having three subclasses which bind to Protein A, although the connection between a given subclass and component peak is not necessarily one to one.

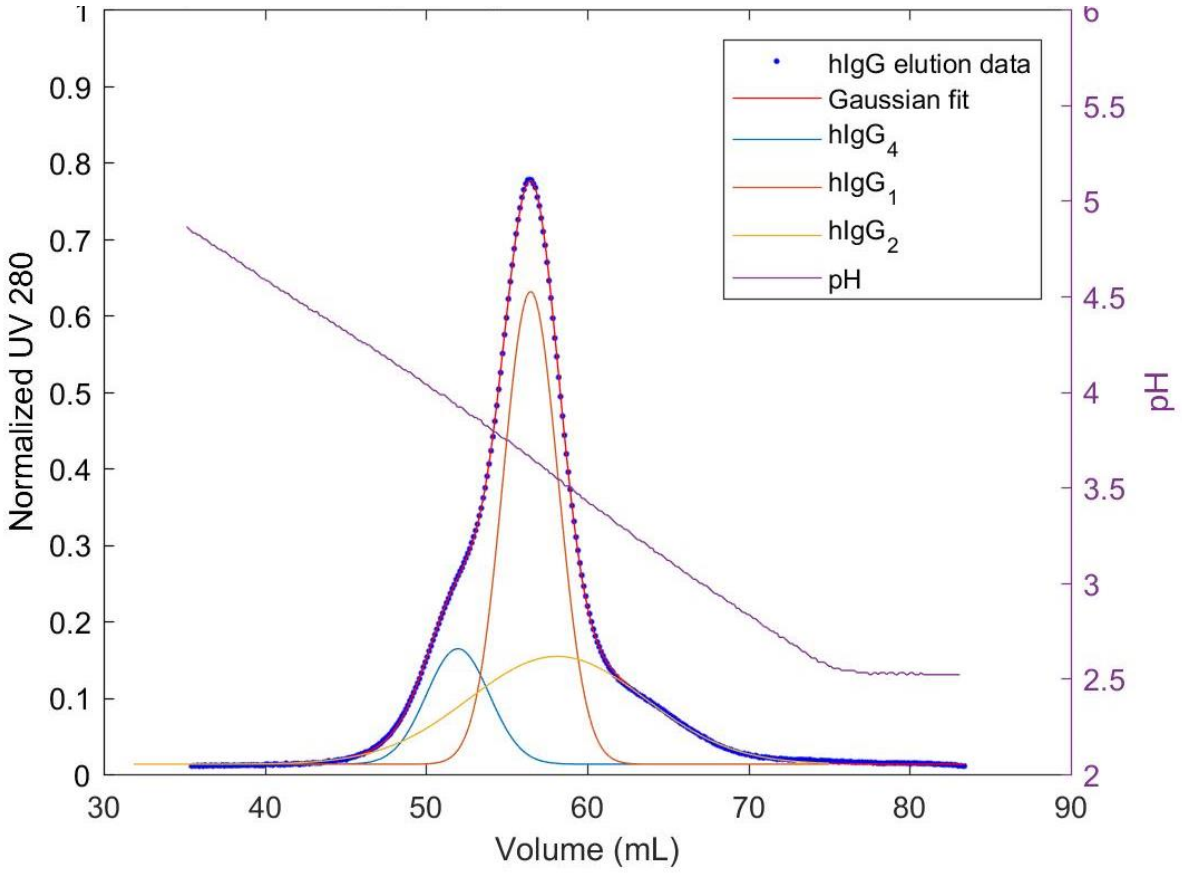


Figure 14: Gradient elution behavior for polyclonal hIgG on CaptivA PriMab protein A media
The experimental elution profile and pH gradient are shown along with a Gaussian fit to the elution envelope and the three underlying peaks from the fit. Data taken from (Weinberg et al., 2017).

In total, hIgG is known to have four different subclasses, IgG_1, IgG_2, IgG_3, and IgG_4, which have sequence and structural differences in the Fc region (Murphy, 2012). IgG_3, however, is missing critical histidine residues within the C_H2-C_H3 cleft that are present in the other three subclasses, and therefore does not bind to protein A. The concentrations of these individual subclasses as found in the blood serum of an adult human (Murphy, 2012) are shown below in Table 1.

Table 1: Abundancy of hIgG subclasses.

	$hIgG_1$	**$hIgG_2$**	**$hIgG_3$**	**$hIgG_4$**
Serum level in adult human (mg/mL)	9	.5	1	3
Relative abundancy excluding $hIgG_3$	.72	.04	--	.24

Figure 14 above, also shows the individual Gaussian curves for each component which make up the overall Gaussian fit. The area under each of the Gaussian curves was used to estimate relative abundancy for each hIgG component and can be seen below in Table 2.

Table 2: Estimated Isotherm Component Parameters, based on interactions with CaptivA PriMab protein A media

Component or relative binding strength	**mAb**	**Weak**	**Moderate**	**Strong**
Relative abundancy from elution profile	--	.36	.50	.14
Elution pH	3.7	3.92	3.66	3.55
$\boldsymbol{K_{obs,\ pH7}(M^{-1})}$ **estimate**	--	1.17E7	1.17E7	1.17E7
$\boldsymbol{K_o(M^{-1})}$ **estimate**	1.12E7	4.17E6	1.39E7	2.23E7
$\boldsymbol{K_{eff,\ pH7}(M^{-1})}$ **estimate**	9.27E6	3.45E6	1.14E7	1.92E7
$\boldsymbol{q_{max}}$ **estimate (mg/mL)**	61	15.1	50.1	84

The large, central component peak of the elution envelope comprises the majority of the polyclonal solution and it is tempting to assign this peak, representing a component with moderate binding strength, to the most abundant IgG_1 subclass. Coincidentally, the mAb elutes at a pH nearly identical to this central hIgG peak and is known to be a $hIgG_1$-type antibody. The subclass assignments of the weaker, binding pH eluting component and the stronger, lower pH eluting component of the hIgG are less clear as the relative abundancies in Tables 1 and 2 do not match well. We further note that the smallest peak, which elutes at the lowest pH, is quite broad.

The fitted experimental pH gradient elution profiles for mAb and polyclonal hIgG were also used to estimate binding constants for each species grouping. If we assume that the total free energy of binding at the peak elution pH is the same for each component,

$$\Delta G_{bind,i@elute\,pH} = \Delta G_{bind,j@elute\,pH}, \tag{71}$$

then the intrinsic binding free energy difference between components can be connected to their respective elution pHs by,

$$\Delta\Delta G_{bind,o,(i-j)} = -RTln\left\{\frac{\left(1+10^{\left(pK'_a-pH_{elute,i}\right)}\right)\left(1+10^{\left(pK''_{a,i}-pH_{elute,i}\right)}\right)}{\left(1+10^{\left(pK'_a-pH_{elute,j}\right)}\right)\left(1+10^{\left(pK''_{a,j}-pH_{elute,j}\right)}\right)}\right\}. \quad (72)$$

And further, if

$$pH_{elute} + 1 < min(pK'_a, pK''_a) \quad (73)$$

and

$$pK''_{a,i} \approx pK''_{a,j} \quad (74)$$

and assuming histidine sidechain pKas on the order of 6.0 (Nelson & Cox), then the difference in intrinsic binding energy between two antibody species may be approximated from the difference in their elution pHs by

$$\Delta\Delta G_{bind,o,\,(i-j)} \approx 4.606RT\left(pH_{elute,i} - pH_{elute,j}\right) \sim \frac{2.7\frac{kcal}{mol}}{\Delta pH}. \quad (75)$$

Thus, if this analysis holds, there is a direct and simple connection between antibody elution pH and relative intrinsic binding strength free energy and, further, lower elution pHs ($\Delta pH < 0$) clearly correspond to stronger binding, as expected. This analysis assumes that the observed pHs of the gradient elution peak maxima are not shifted by mass transfer limitations. Further, if we have available a measurement or estimate of the effective binding constant of a given antibody species at a given pH and we have gradient elution pH data available for this species and others on the same media, we can use Equations 64 and 72 to make estimates of the absolute values of the intrinsic binding constants each of the species.

We estimated the intrinsic binding equilibrium constants and effective binding constants at pH 7 for the three hIgG species and the mAb using model 1. Weinberg (2017) reported estimates of the

association equilibrium constant for hIgG on CaptivA PriMab media ranging from 78 to 103 mL wetted bead per mg hIgG, obtained by fitting a Langmuir isotherm to equilibrium batch uptake data. Using an average value of 90 mL/mg and an average hIgG molecular weight of 150 kDa, we estimate an observed lumped binding affinity constant of $1.17 \times 10^{-7}\ M^{-1}$ for hIgG at pH 7. We assume that the corresponding observed free energy of binding represents a weighted sum of the binding energies of the three underlying hIgG components resolved in the pH gradient elution experiment and use the relative abundances of these components as the weights to relate the observed binding constant to the intrinsic binding constants:

$$ln\, K_{obs,pH7} = \sum_{i=2}^{4} x_i\, ln\, K_{eff,pH7} = \sum_{i=2}^{4} x_i\, ln \left\{ \frac{K_{bind,o,i}}{\left(1+10^{(pKa'-7)}\right)\left(1+10^{(pKa''-7)}\right)} \right\}. \qquad (76)$$

Further since we know the elution pHs of each of the components, we can use Equations 64 and 72 to recast Equation 76 in terms of a single intrinsic binding constant value, e.g. $K_{bind,o,3}$, and solve for this value. Once this is done, Equation 72 may be used to find the remaining intrinsic binding constants, $K_{bind,o,2}$ and $K_{bind,o,4}$. These values are reported in Table 2. Once the intrinsic binding constants are estimated, the effective binding constants at pH 7 can be calculated from Equation 66. These values are also reported in Table 2. A similar procedure was used to estimate the mAb intrinsic binding constant from its elution pH relative to one of the hIgG species, e.g. species 3, and then, in turn, the mAb effective binding constant at pH 7 could be determined.

The maximum binding capacities on the CaptivA PriMab media at pH 7 were also determined by Langmuir isotherm fits to batch uptake data (Weinberg, 2017). A lumped value of 84 mg/mL wetted beads was fit for the hIgG; a value of 61 mg/mL wetted beads was fit for the mAb. To simulate the observed competitive displacement behavior, we assumed that the binding capacities for the three observed hIgG components was proportional to the magnitudes of the effective

binding equilibrium constants at pH 7; q_{max} was set at 84 mg/mL for the strongest binding component and scaled appropriately for the moderate and weak components as reported in Table 2.

Noncompetitive Binding Behavior in a Single Chromatographic Bead

Using the component isotherm parameters that were estimated above, it becomes possible to use the combined adsorptive and diffusive transport model to simulate the uptake of multiple components. Using Equations 7 through 11, it is possible to see how each component with its individual parameters, moves through the spherical system without competitive binding effects. The boundary conditions and initial solution concentration were assumed to be the same for each component. The solution and solid phase concentrations for the noncompetitive binding kinetic case is shown below in Figures 15 and 16 respectively.

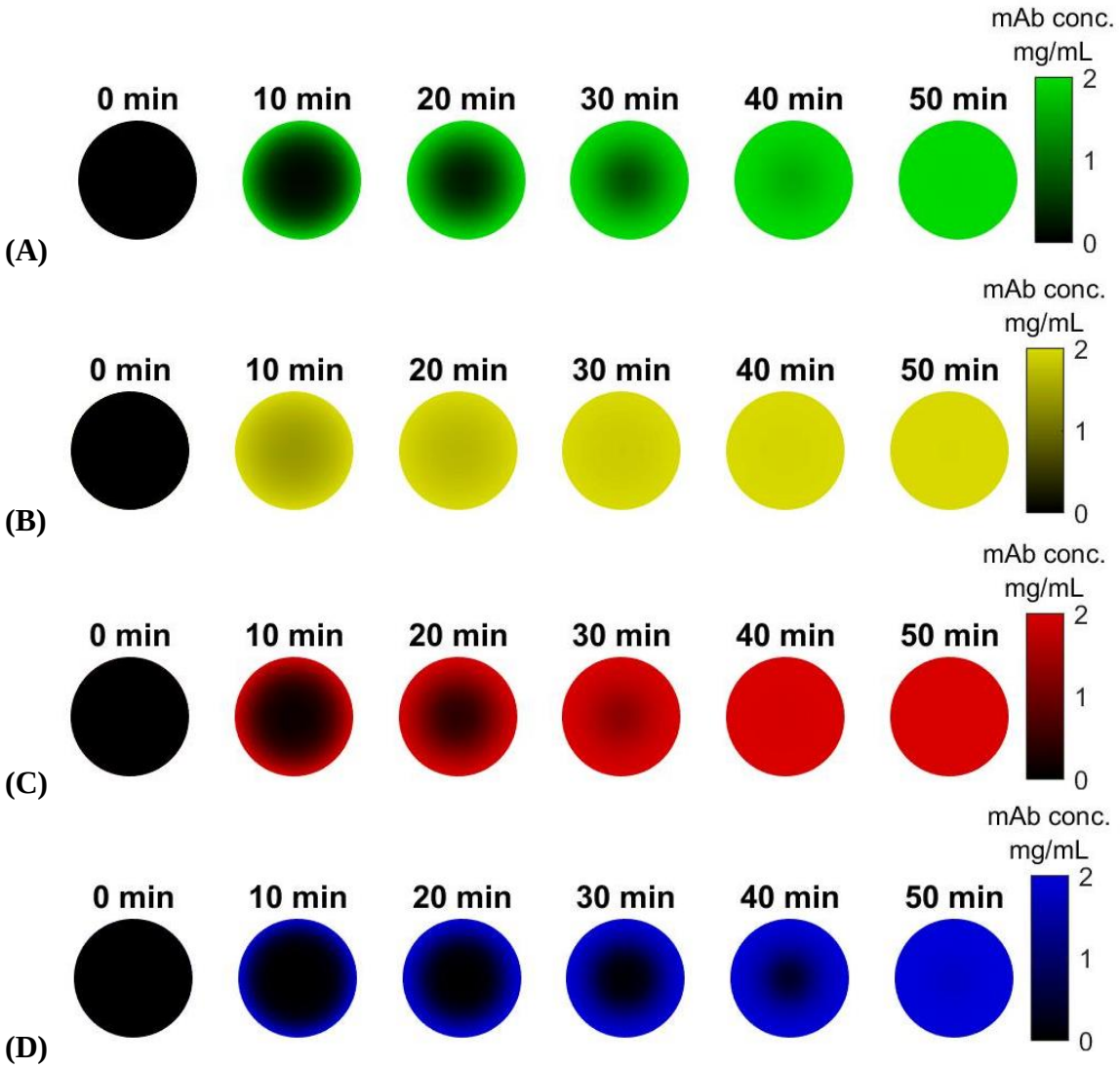


Figure 15: Noncompetitive solution phase binding profiles for individual components.
Kinetic case, solid phase antibody concentration profiles for (A) mAb, (B) weak binding hIgG, (C) moderate binding hIgG, (D) strong binding hIgG.

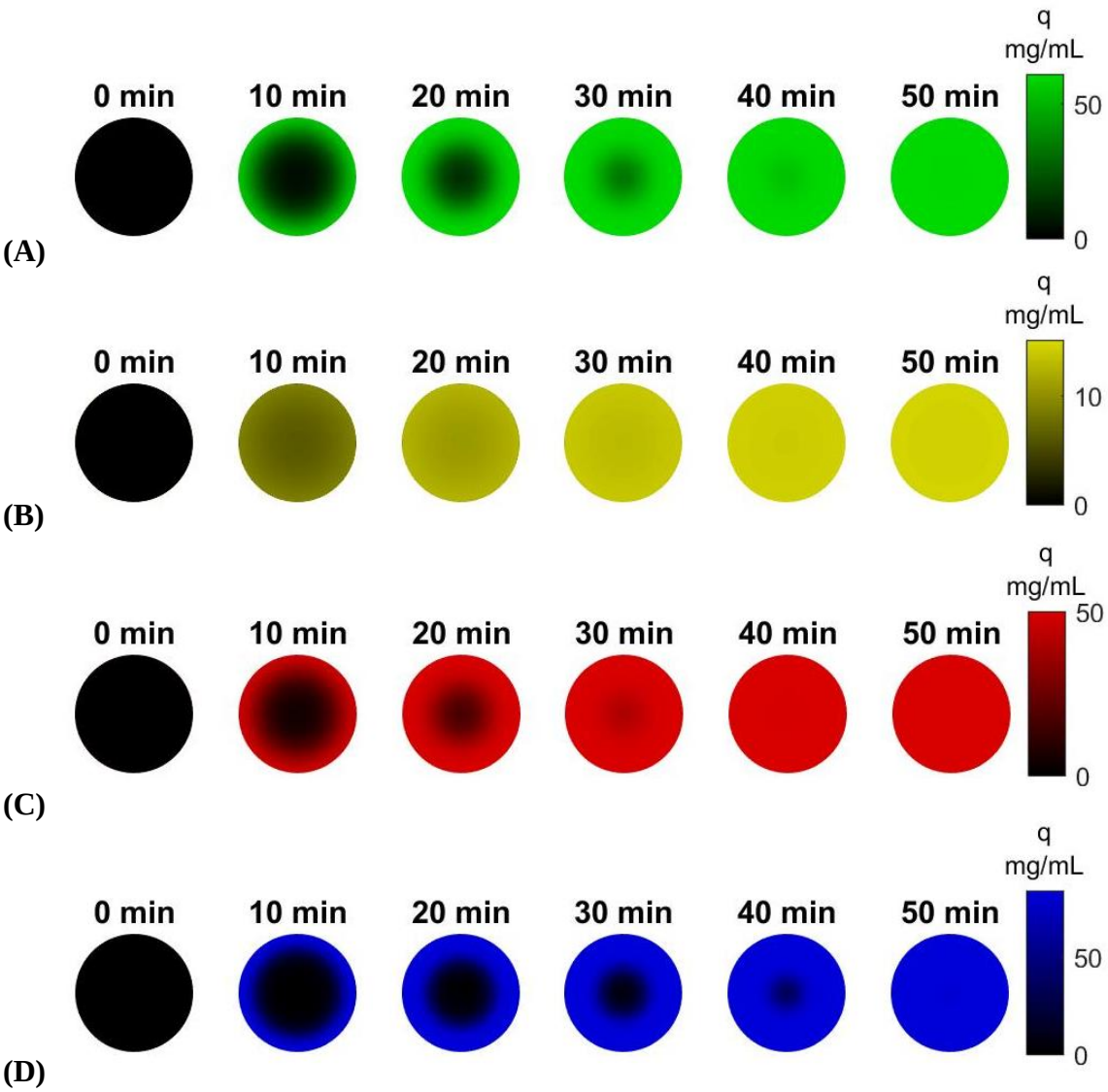


Figure 16: Noncompetitive solid phase binding profiles for individual components.
Kinetic case solid phase antibody concentration profiles for (A) mAb, (B) weak binding hIgG, (C) moderate binding hIgG, (D) strong binding hIgG.

It can be seen from these images that components with weaker binding strength move to the center of the sphere more quickly than components with a stronger binding strength. This is expected, as weaker binding components bind less often and therefore have more freedom to move. Eventually, each component reaches an equilibrium value where q is equal to q_{max}.

Competitive Binding Behavior in a Single Chromatographic Bead

The next step is to consider these components in the combined adsorptive and diffusive transport model under competitive binding conditions. It is necessary to incorporate Equations 34 and 35 into the model to account for competitive binding effects. Boundary conditions were taken to be the same as in the previous noncompetitive case, and the surrounding solution was assumed to have an equal initial concentration for the mAb component and hIgG polyclonal solution. Initial concentrations for the individual components of the hIgG polyclonal solution were set to the relative abundancy of each component multiplied by the polyclonal solution initial concentration. Constants and parameters for mAb and IgG not previously estimated were taken from the literature (Weinberg et al., 2017) and can be found in Appendix II. Computational results for the solid phase concentrations are shown below in Figure 17.

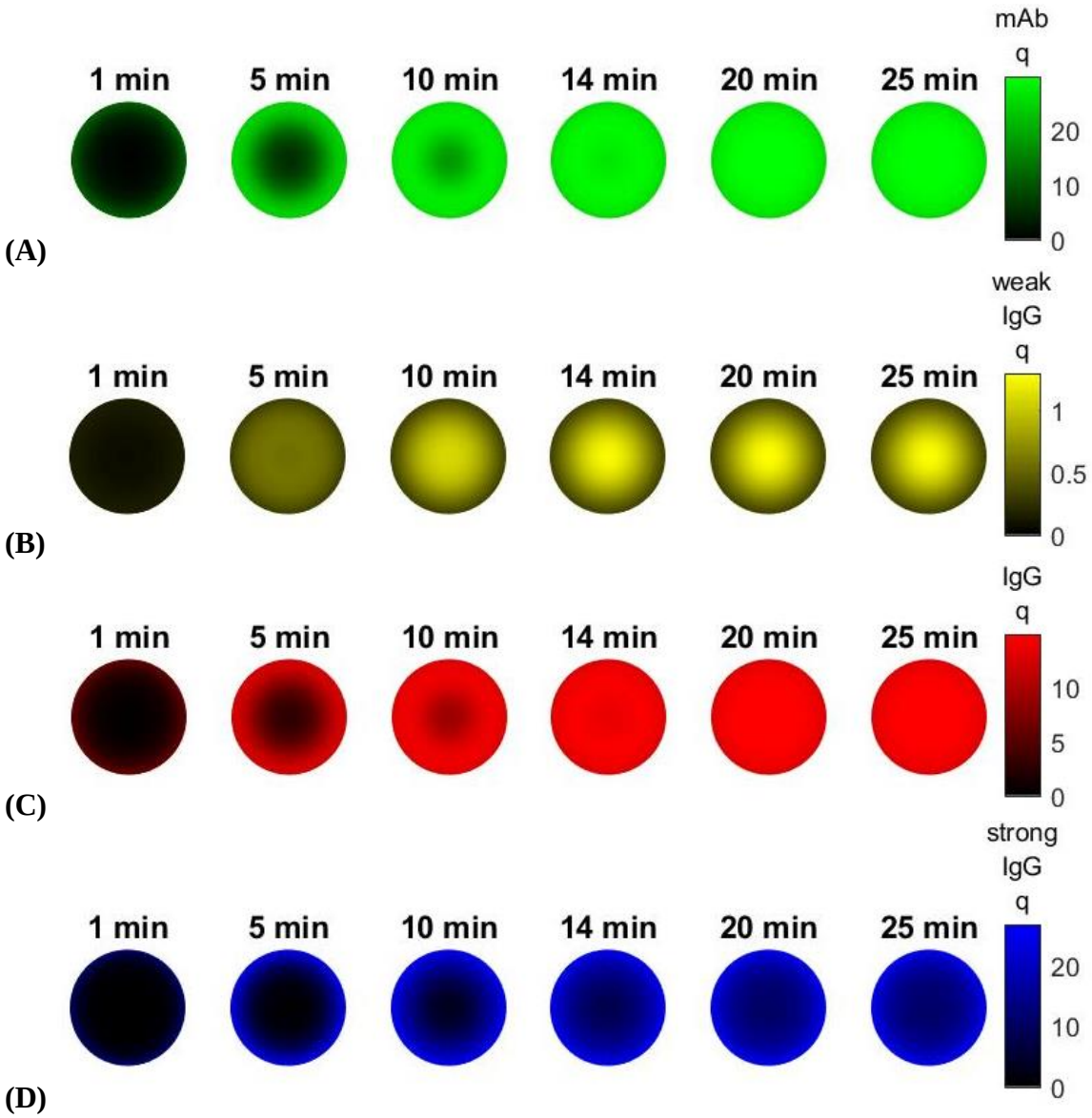
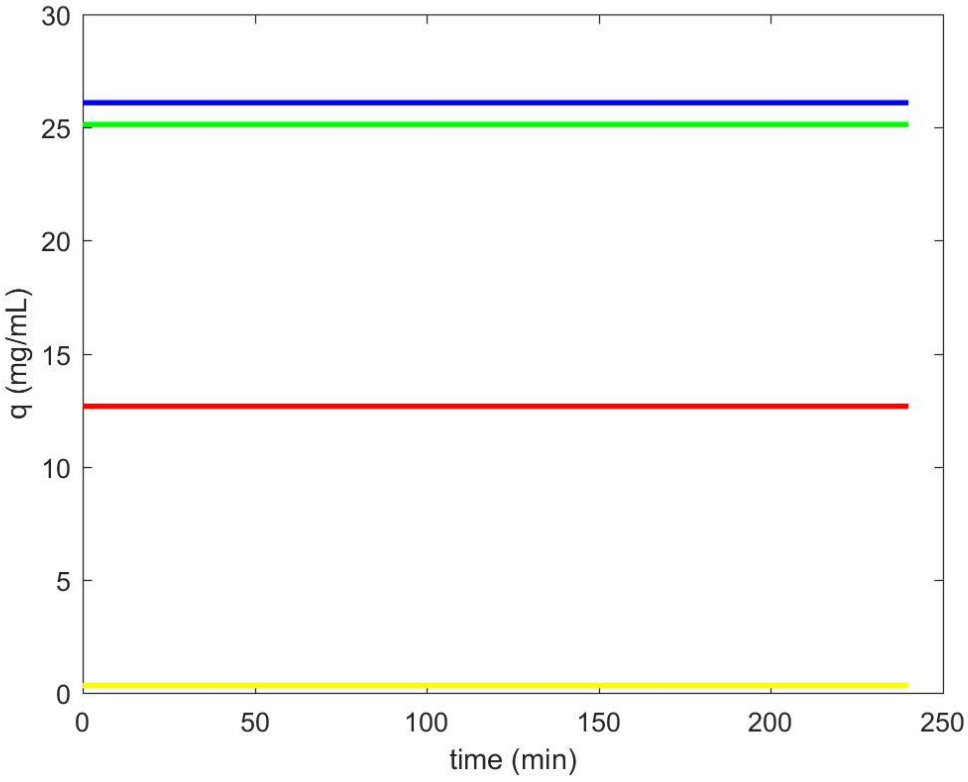


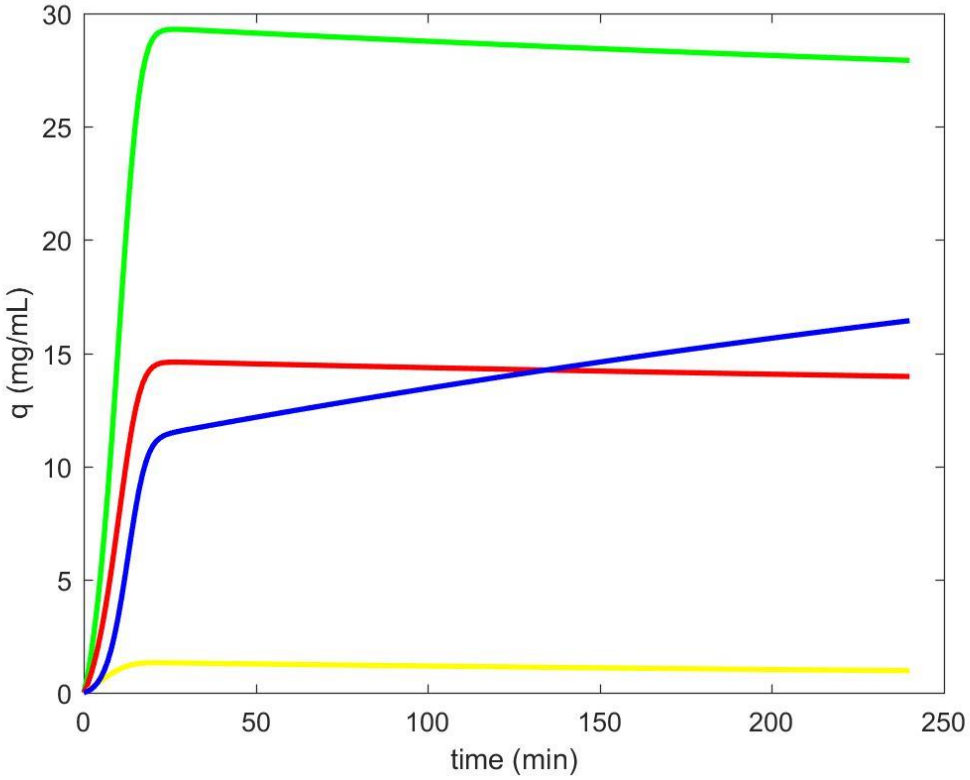
Figure 17: Competitive solid phase binding profiles for individual components.
Kinetic case solid phase antibody concentration profiles for (A) mAb, (B) weak binding hIgG, (C) moderate strength binding hIgG, (D) strong binding hIgG.

Figure 17, with competitive binding, is compared to Figure 16, with noncompetitive binding, to understand how competitive binding effects impact the adsorption behavior of each component. The clearest observation, is that when species bind in competition, the q values for all components decrease significantly. This is most obvious for the components with weaker binding strength. As expected from the noncompetitive model, components with weaker binding strengths moved more quickly through the porous media in both the liquid and solid states. The concentration of protein

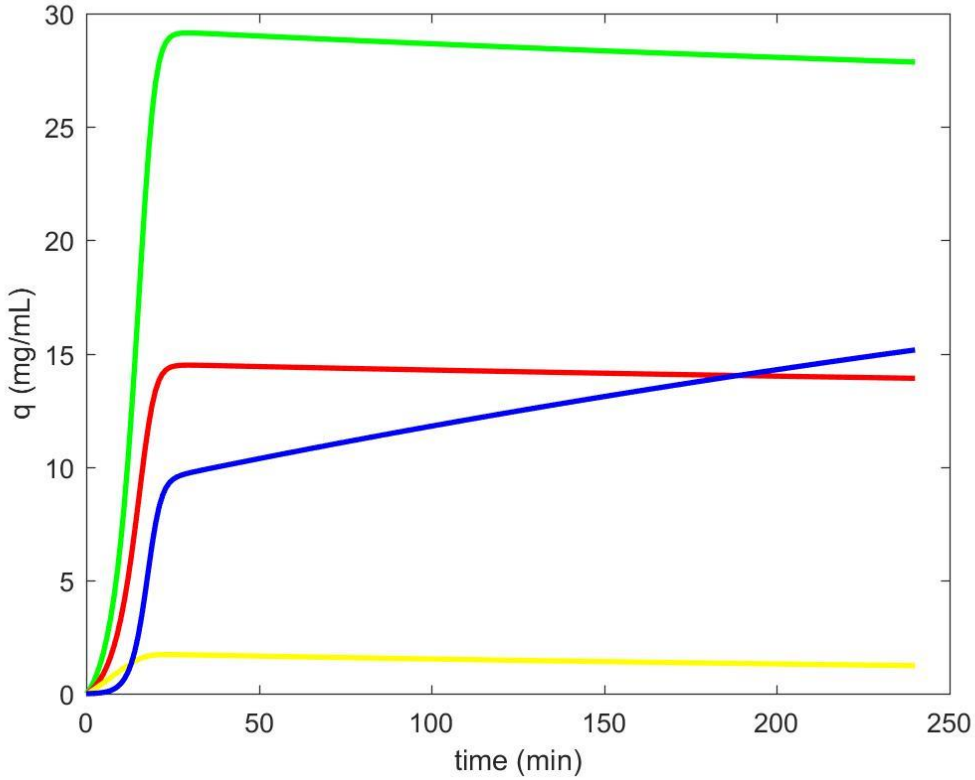
A – antibody complexes, seen in Figure 17, showed that antibodies with a weaker binding affinity bound preferentially towards the center of the bead, as these weaker complexes were quickly replaced with complexes having stronger binding affinities. This can be seen more clearly, quantitatively in Figure 18 below, where changes in component concentration are shown as a function of time at different radial positions of the sphere.



(A)



(B)



(C)

Figure 18: q vs. t profiles at select radial positions.

Competitive, kinetic case, solid phase binding profiles for individual components. Selected radial positions: (A) $r = R$, (B) $r = R/2$, and (C) $r = 0$.

Computational Reproduction of Experimental Results

After successfully modeling competitive binding effects between four components, it was necessary to check the computational model by comparing it to experimental results. This was done by computationally reproducing the experimental results of Weinberg et al., (2017). In these experiments, chromatography resin beads with embedded protein A ligands were soaked in a solution of mAb at a concentration of 2 $\frac{mg}{mL}$. They were subsequently washed and soaked in a solution of polyclonal hIgG antibodies also at a concentration of 2 $\frac{mg}{mL}$. After selected times, the resin beads were washed and antibody binding was observed using CLSM imaging. A flow chart of this process can be seen below in Figure 19.

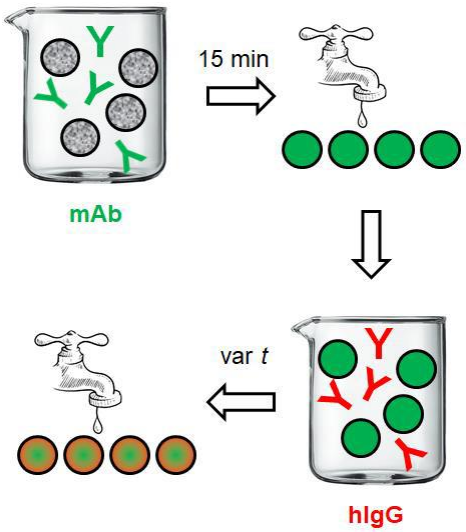


Figure 19: Flow chart of experimental processing showing antibody displacement.
(Przybycien & Lane, 2018)

To qualitatively compare the computational simulations to experimental results, we started with simulations of a two component competitive binding model. These two components were the mAb

and the hIgG polyclonal antibody, treated as the sum of the weak-, moderate- and strong-binding species.

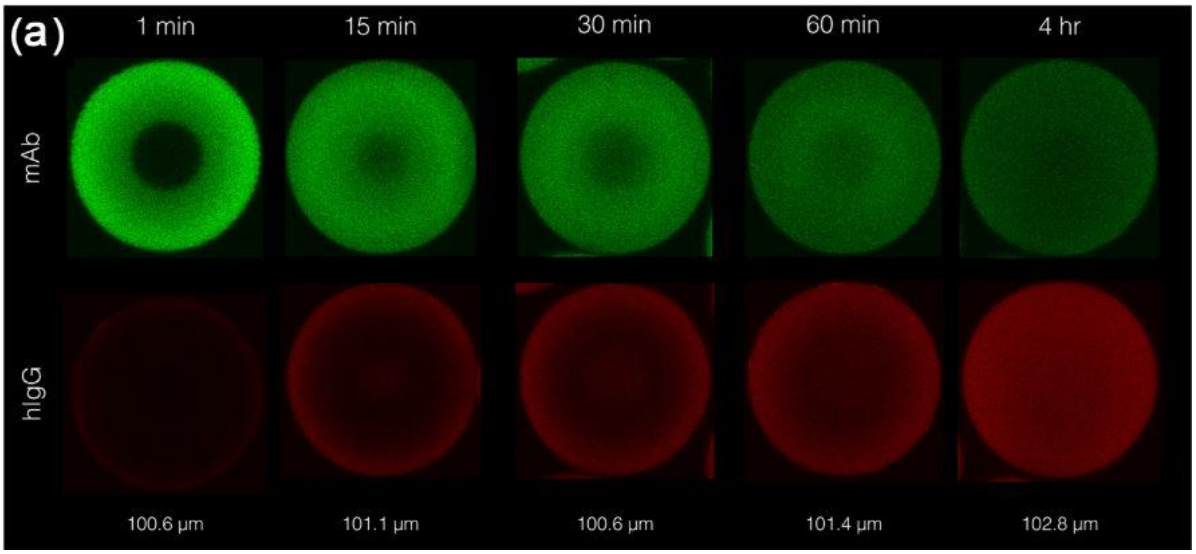


Figure 20: Representative CLSM images of sequential mAb-hIgG adsorption experiments.
Images taken at selected times with CaptivA PriMab protein A media, from Weinberg et al., 2017.

Figure 20 above shows the images taken from Weinberg et al., (2017) that were repeated by the computational simulations. Below in Figures 21 and 22 are the computationally reproduced images for these two components.

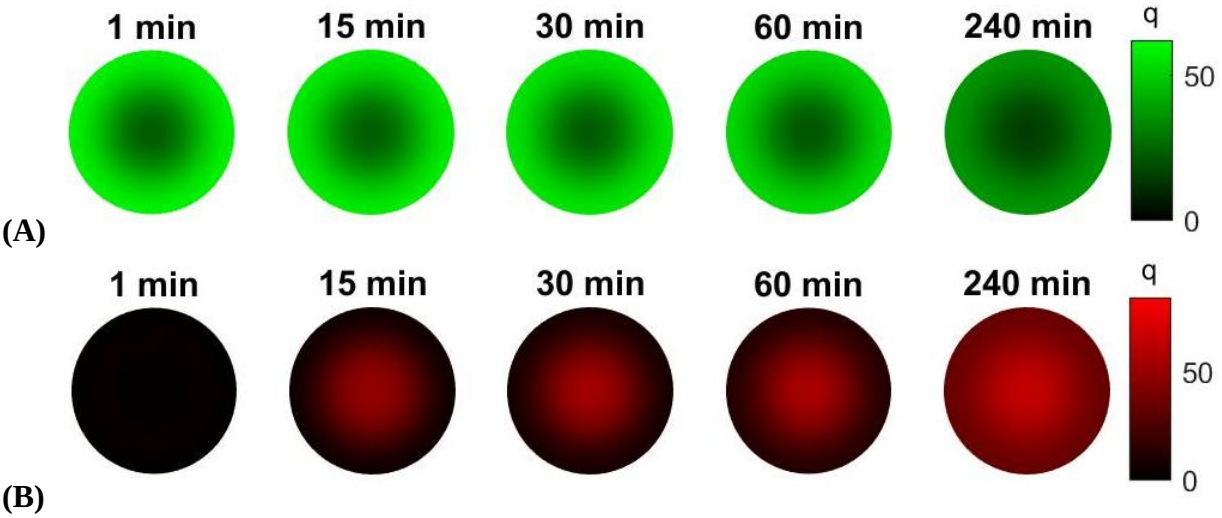


Figure 21: Kinetic case, competitive solid phase binding profiles for two components.
(A) mAb component is shown in green, and (B) hIgG polyclonal solution is shown in red. q_{max} values and other key parameters can be found in Appendix II.

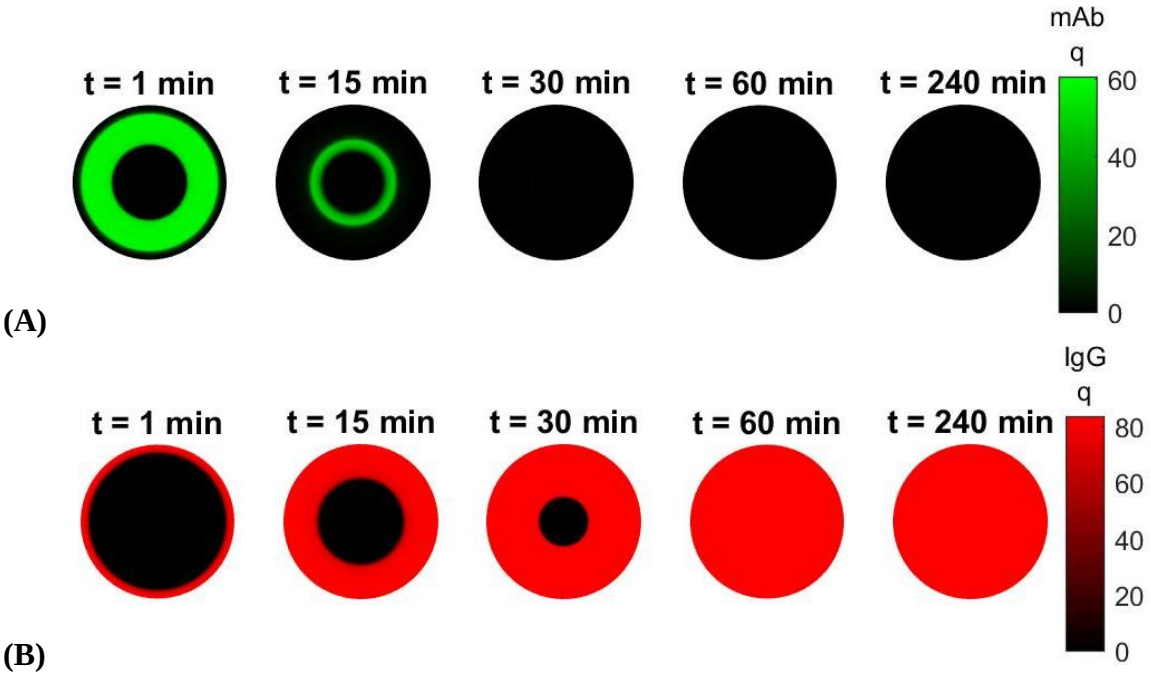


Figure 22: Equilibrium case, competitive solid phase binding profiles for two components. (A) mAb component is shown in green, and (B) hIgG polyclonal solution is shown in red. q_{max} values and other key parameters can be found in Appendix II.

Qualitatively, it is noticeable that the computational images for the equilibrium follow the same pattern as the experimental ones. This suggests that the assumed kinetics for the kinetic case are too slow. This will be further analyzed below with a quantitative analysis of the experimental images.

Computationally, the components of the polyclonal hIgG solution can also be broken up into the three previously defined subclasses. Although it is not possible to compare this quantitatively to the experimental images, it is interesting to observe the computationally predicted movement of these individual components. Figures 23 and 24 below show qualitatively how the components individually displace mAb that has already adsorbed to a resin bead under the above experimental conditions for the kinetic and experimental cases respectively.

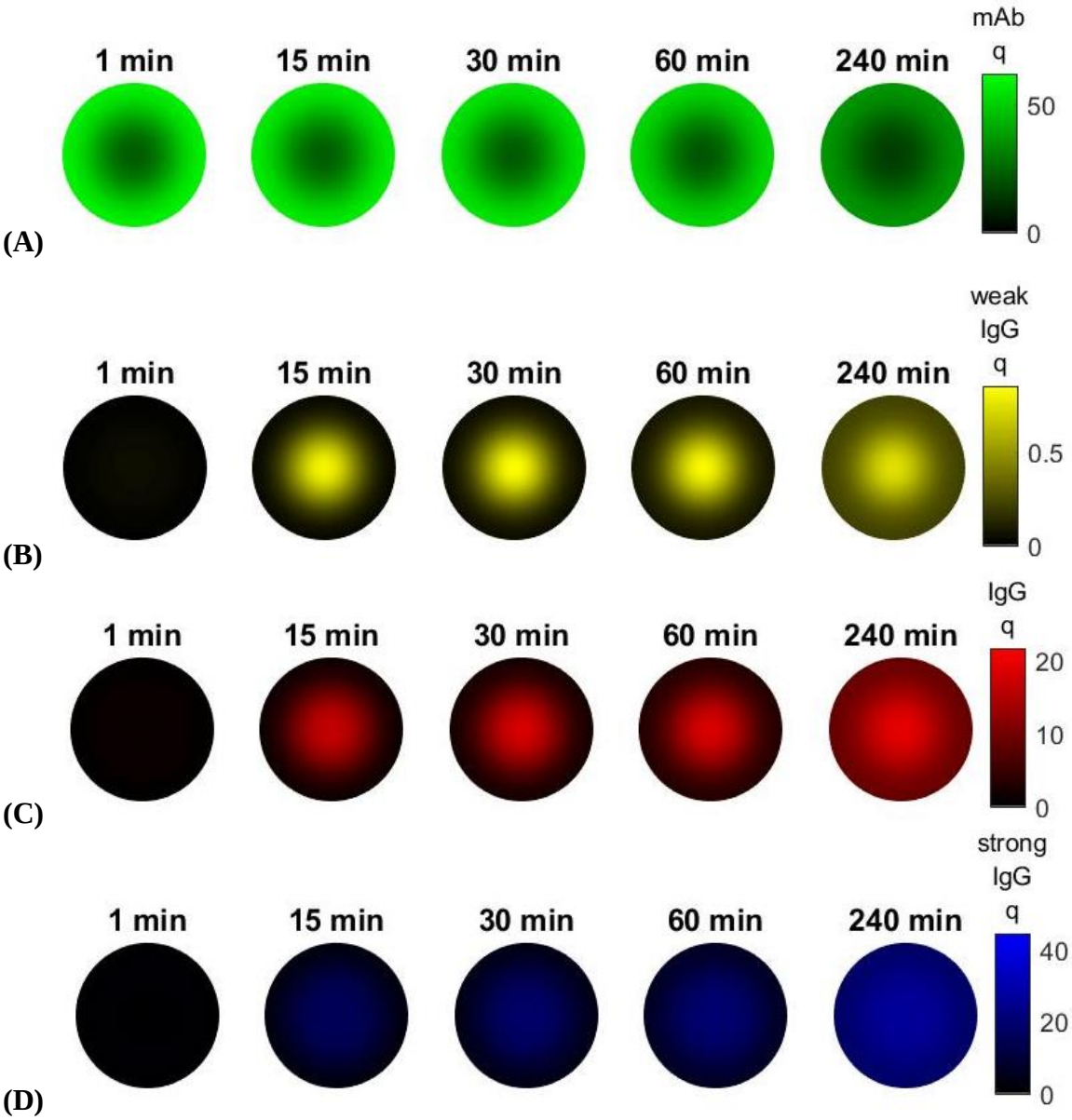


Figure 23: Kinetic case, competitive solid phase binding profiles for four components.
(A) mAb component shown in green, (B) weak binding hIgG component shown in yellow, (C) moderate strength binding hIgG component shown in red, (D) strong binding hIgG component shown in blue. q_{max} *values and other key parameters can be found in Appendix II.*

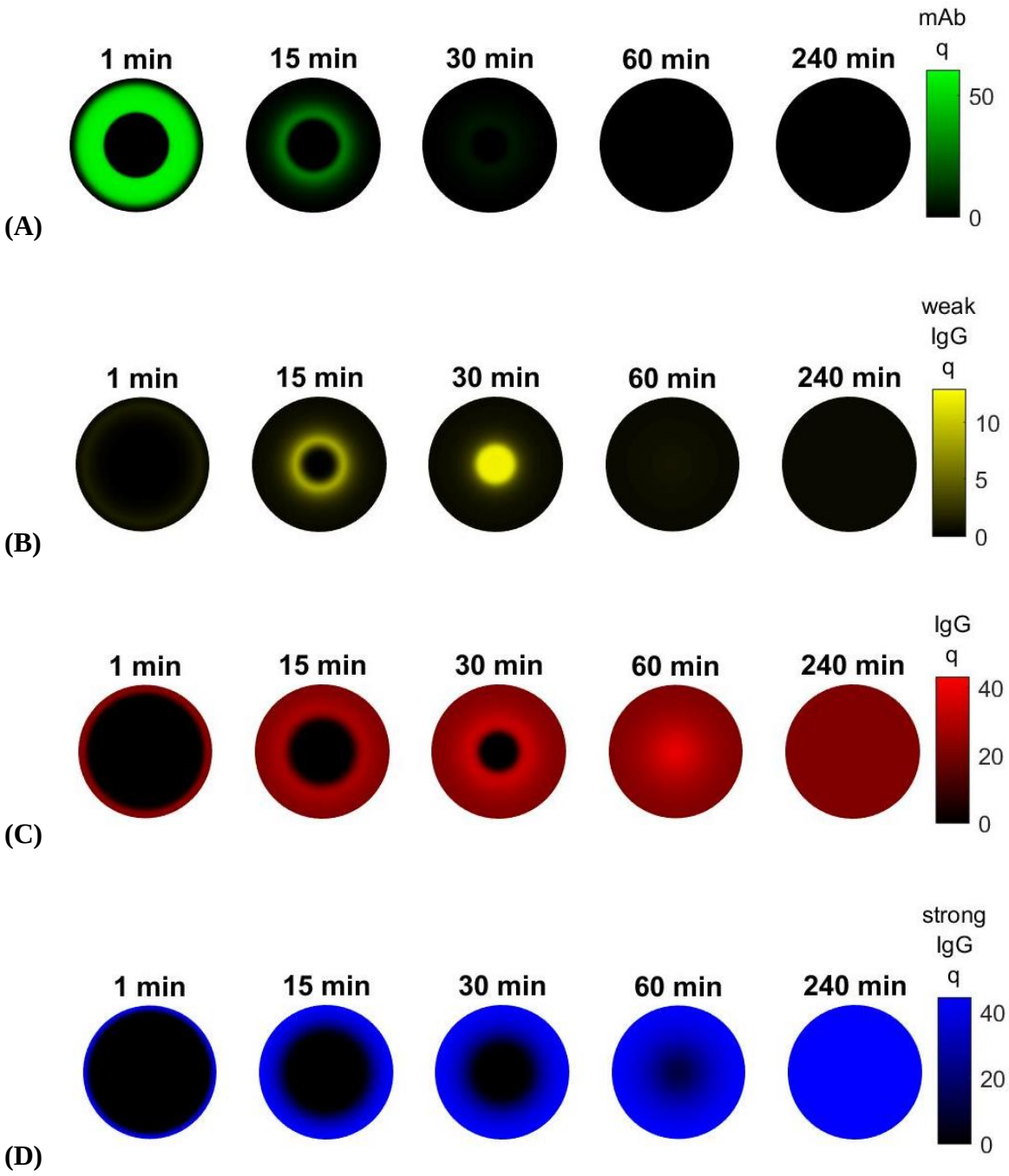
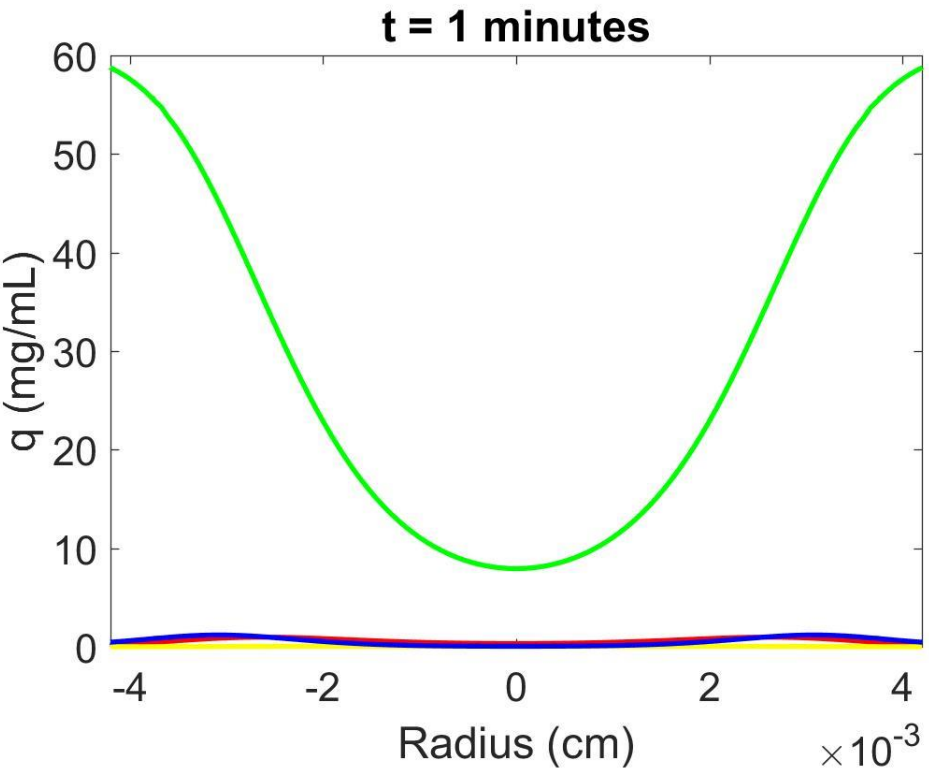
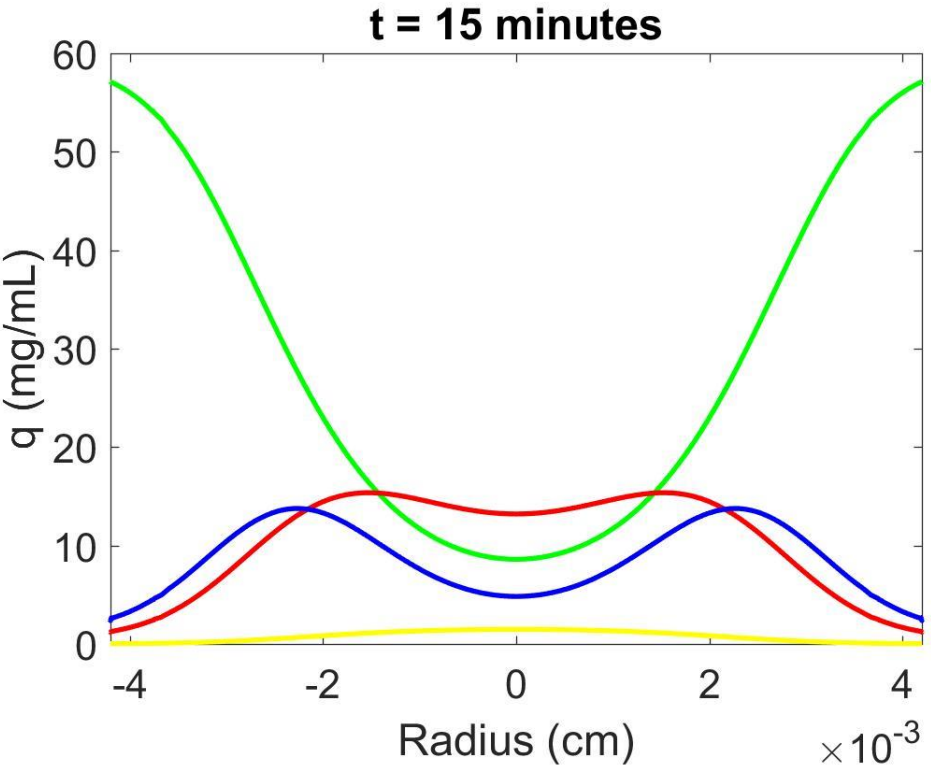


Figure 24: Equilibrium case, competitive solid phase binding profiles for four components.
(A) mAb component shown in green, (B) weak binding hIgG component shown in yellow, (C) moderate strength binding hIgG component shown in red, (D) strong binding hIgG component shown in blue. q_{max} values and other key parameters can be found in Appendix II.

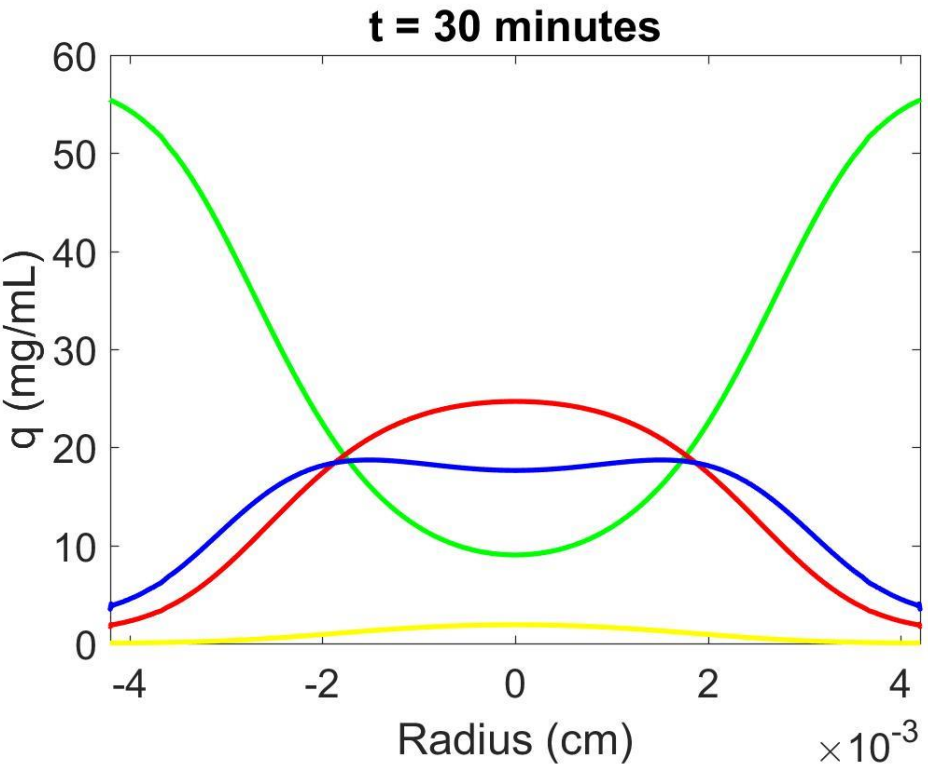
Figures 25 and 26 below show quantitatively how the components are predicted to bind as a function of radial position at selected times.


t = 1 minutes
q (mg/mL)
Radius (cm)
×10-3

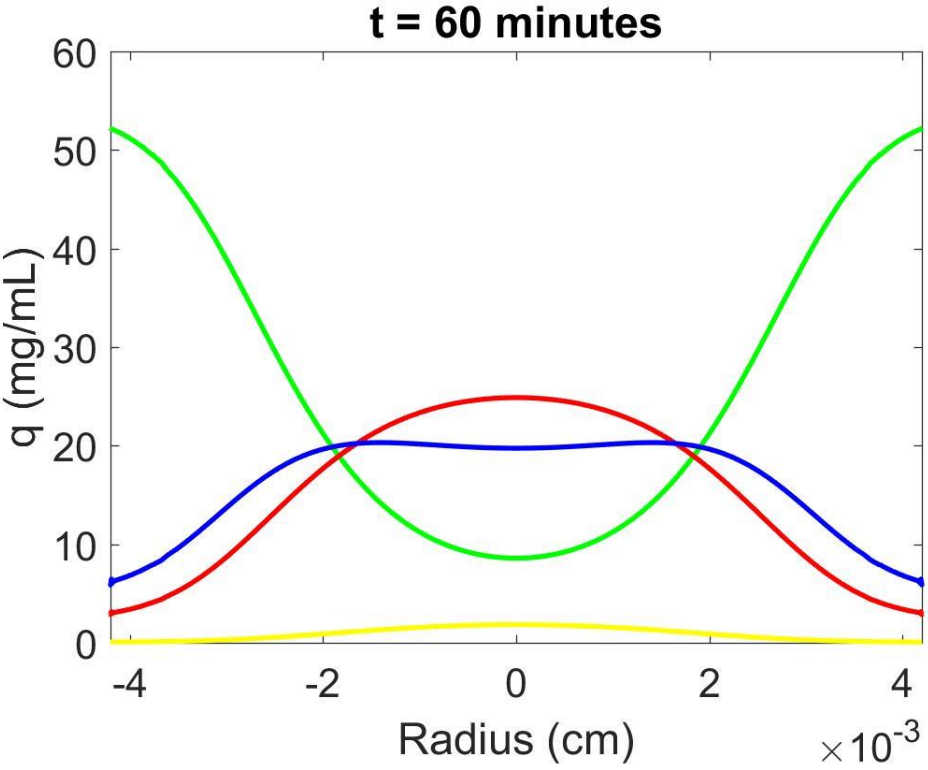
(A)


t = 15 minutes
q (mg/mL)
Radius (cm)
×10-3

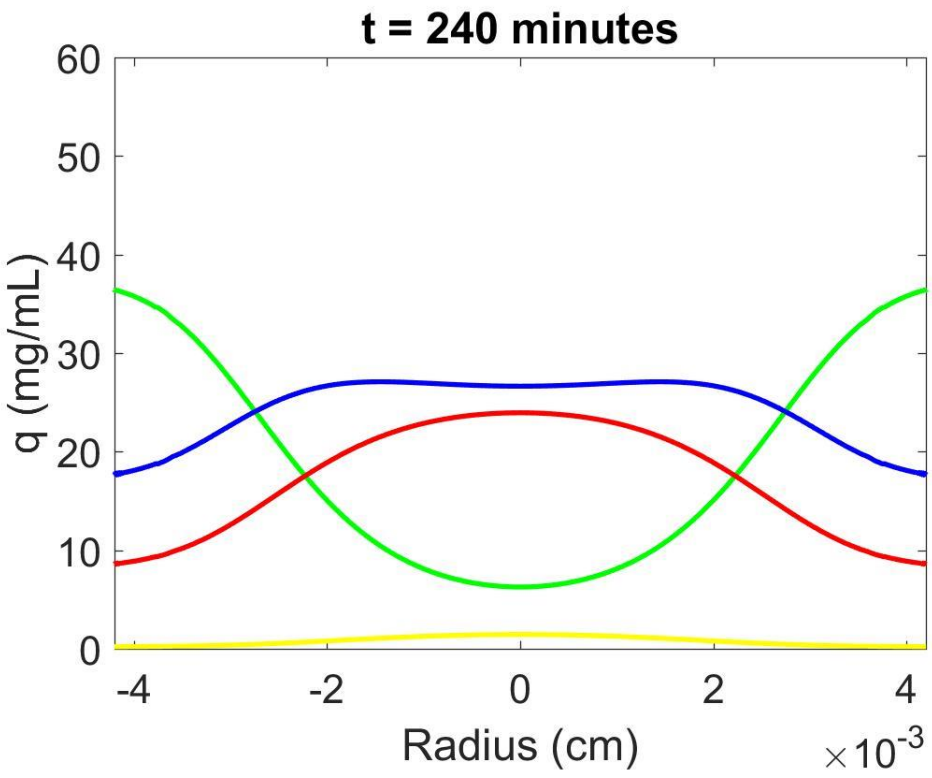
(B)


t = 30 minutes
q (mg/mL)
Radius (cm)
×10-3

(C)

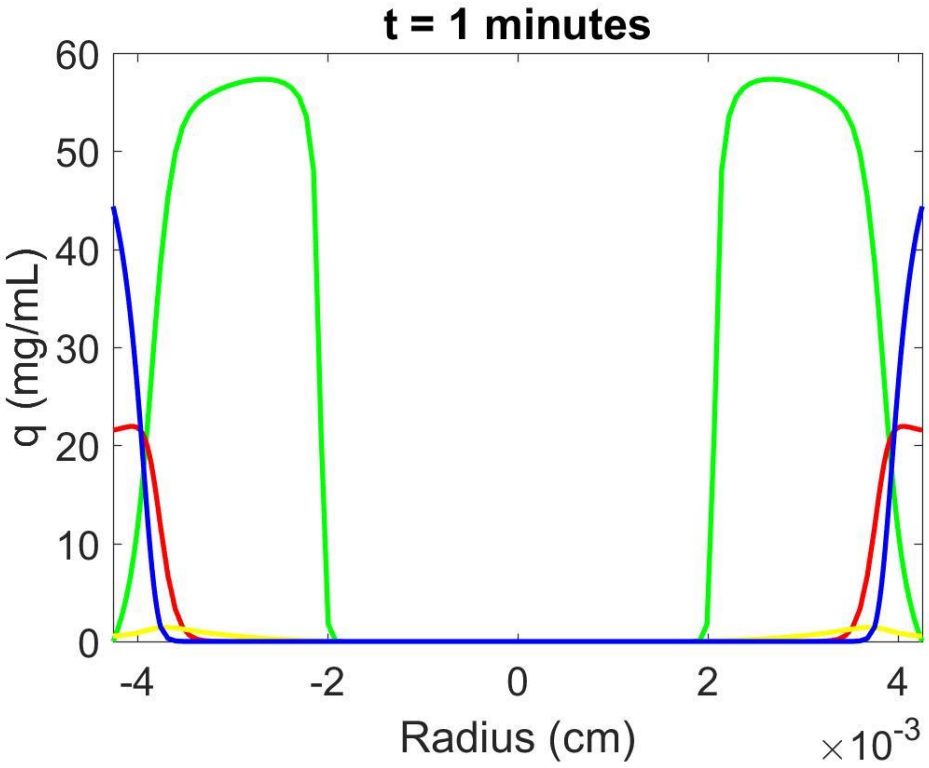

t = 60 minutes
q (mg/mL)
Radius (cm)
×10-3

(D)

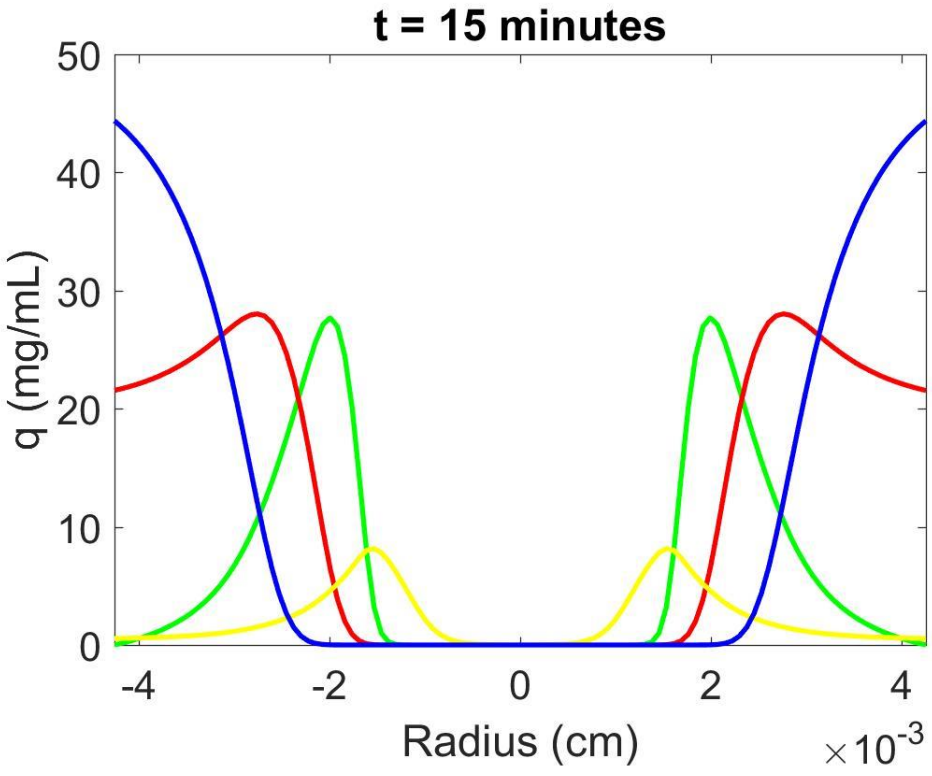


(E)

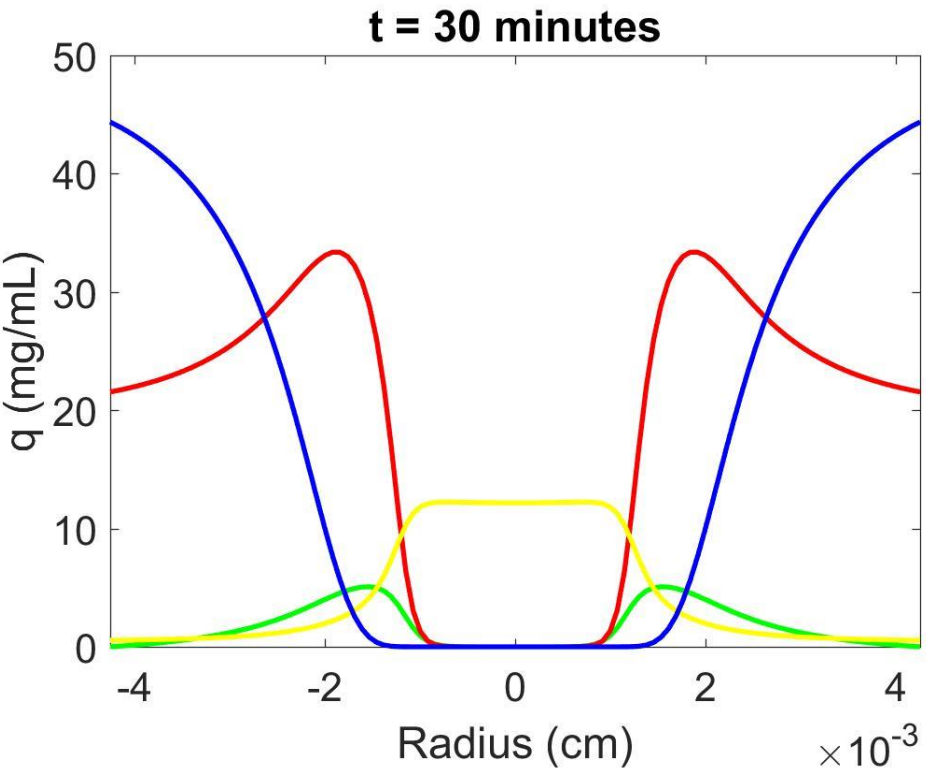
Figure 25: Kinetic case, q vs. radial position for four competitive components at select times. mAb component shown in green, weak binding hIgG component shown in yellow, moderate strength binding hIgG component shown in red, strong binding hIgG component shown in blue. Select times are (A) 1 minute (B) 15 minutes (C) 30 minutes (D) 60 minutes (E) 4 hours. q_{max} values and other key parameters can be found in Appendix II.


t = 1 minutes
q (mg/mL)
Radius (cm)
×10^-3

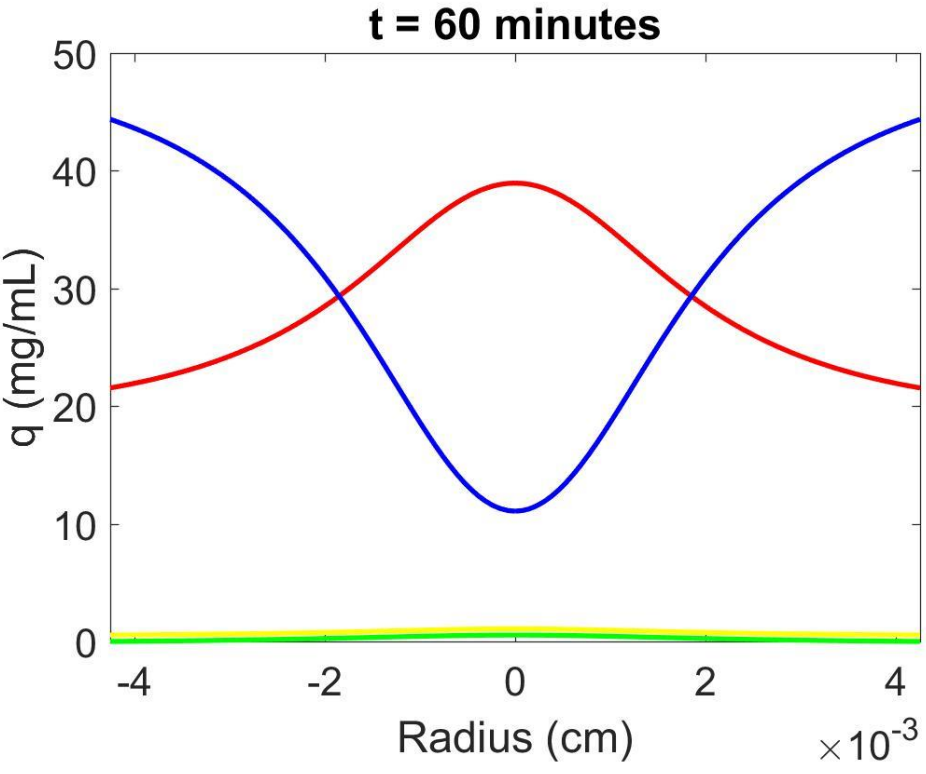
(A)


t = 15 minutes
q (mg/mL)
Radius (cm)
×10^-3

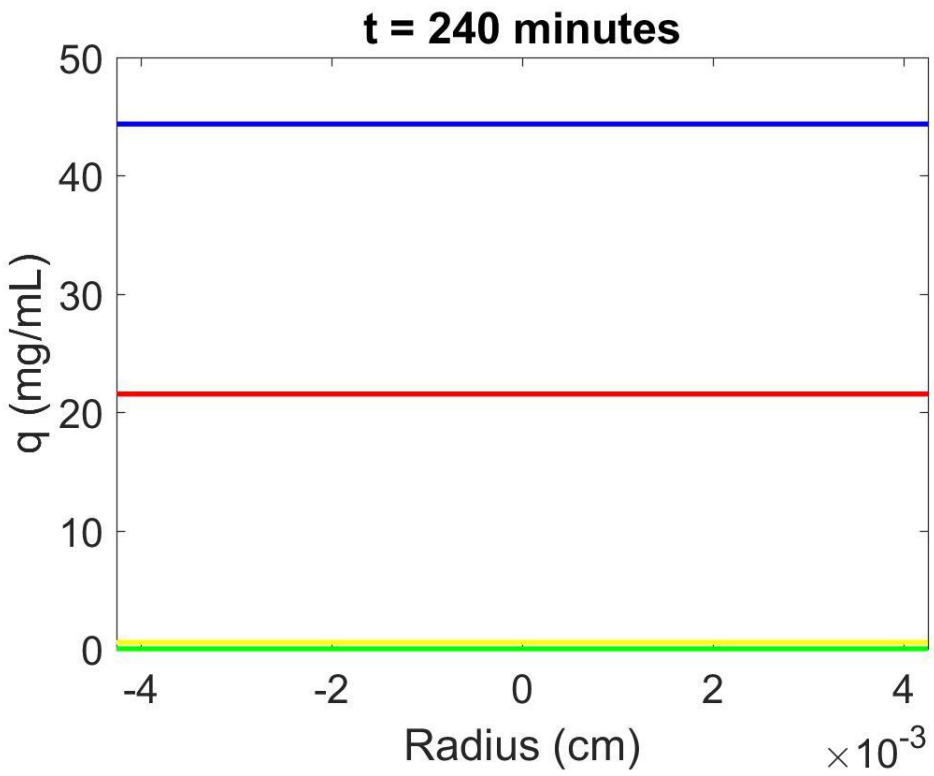
(B)


t = 30 minutes
q (mg/mL)
Radius (cm)
×10-3

(C)


t = 60 minutes
q (mg/mL)
Radius (cm)
×10-3

(D)



(E)

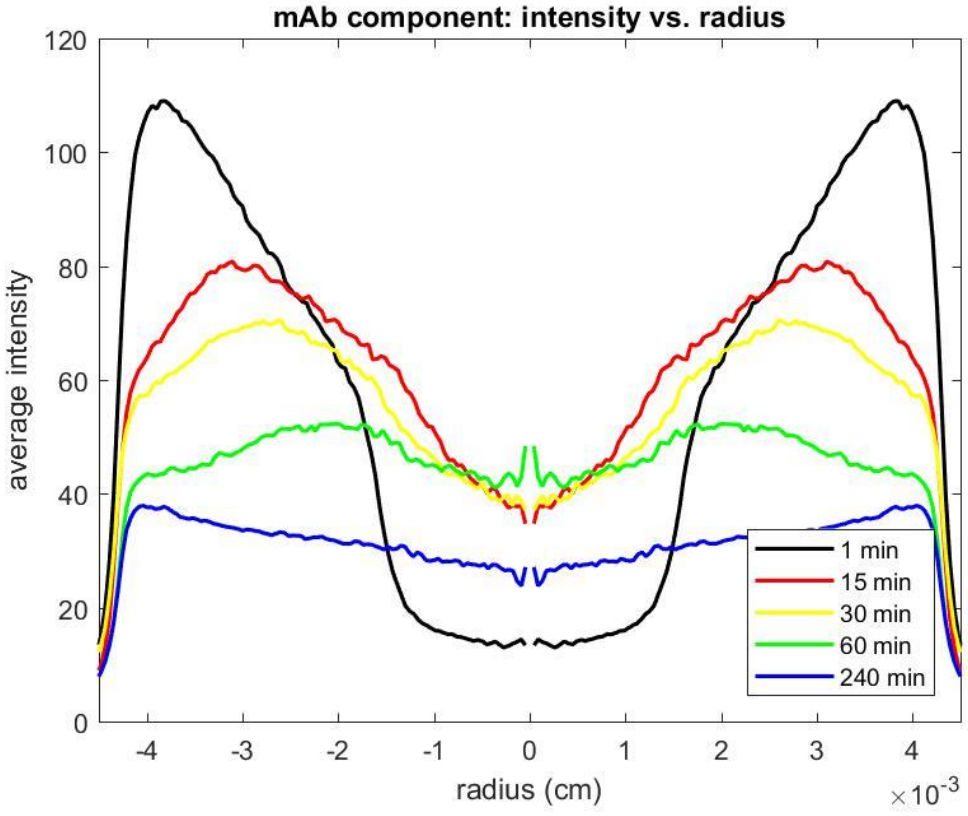
Figure 26: Equilibrium case, q vs. radial position for four competitive components. mAb component shown in green, weak binding hIgG component shown in yellow, moderate strength binding hIgG component shown in red, strong binding hIgG component shown in blue. Select times are (A) 1 minute (B) 15 minutes (C) 30 minutes (D) 60 minutes (E) 4 hours. q_{max} values and other key parameters can be found in Appendix II.

In Figures 23 through 26 above, it is seen that q of the mAb component decreases over time and its distribution becomes more even. The hIgG component with moderate strength is quicker to bind than the strong binding component, however, over time the stronger binding component displaces more of the other components. The mAb component is displaced from the outside in as the other components absorb to the center of the sphere. The weakest binding hIgG component moves to the center of the sphere quickest but is quickly displaced by stronger binding components. The moderate binding hIgG component fills in to the center where there are more available binding spots and binding is most energetically favorable. The strongest binding hIgG component moves the slowest, but displaces the other components in its path.

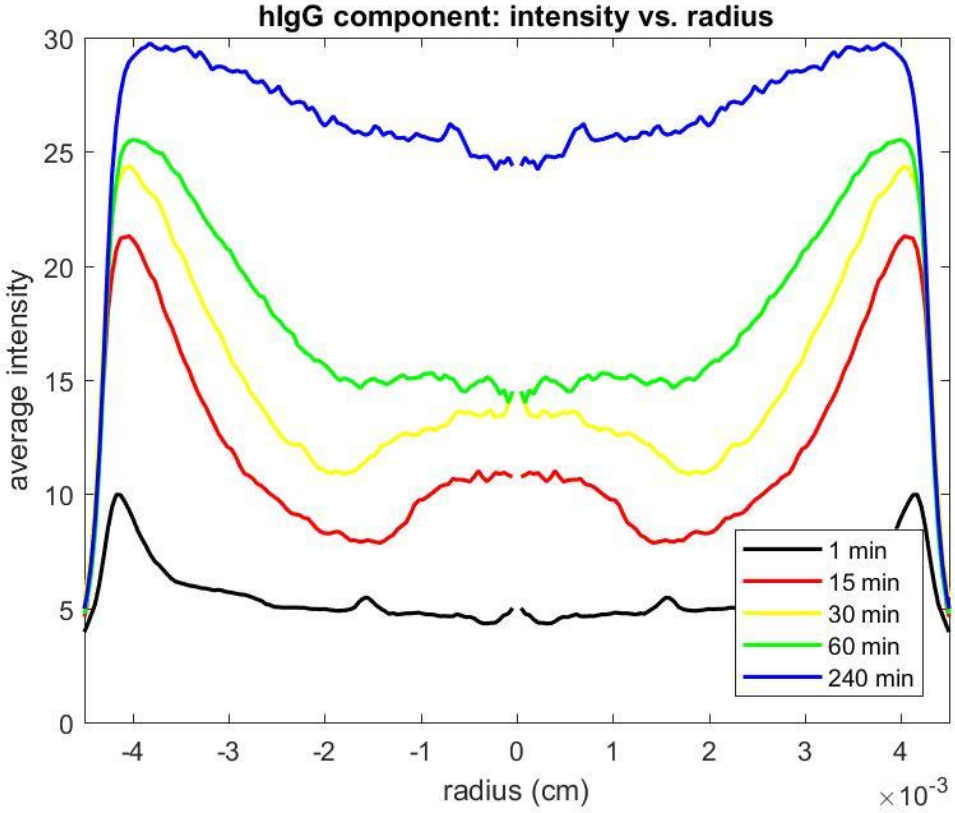
It should be noted that these computational results are not an exact replica of the experimental images shown. In comparing the kinetic and equilibrium cases, it can be concluded that the kinetics of binding play an important role in protein A affinity chromatography and must be considered in the development of an accurate model. However, the current kinetic parameters used are seemly too slow and require tuning for a more accurate model.

Quantitative Analysis of Experimental Images

A qualitative comparison of experimental images and computational simulations shows that the model captures the behavior of the given chromatography system. However, to affirm this conclusion and for further understanding, it is also necessary to execute a quantitative comparison. The circular CLSM images seen above in Figure 21 were imported into MATLAB and analyzed to estimate relative species concentration versus radial position, as color intensity of CLSM images is proportional to adsorbed protein concentration. After the image of each spherical cross section was imported into MATLAB, the *imfindcircles* command was used to map each circle. Sensitivity for circle detection was modified for each individual image so only one circle was found, and *Rmin* and *Rmax* parameters were set to 95 and 105 respectively. Then, the pixels inside the circle were analyzed by intensity. Pixel intensities were grouped by distance from the center pixel at the origin, and these groups were averaged to obtain a quantitative radial profile. These profiles of intensity vs. radial position are seen below in Figure 27.


mAb component: intensity vs. radius
average intensity
radius (cm)
×10^-3
1 min
15 min
30 min
60 min
240 min

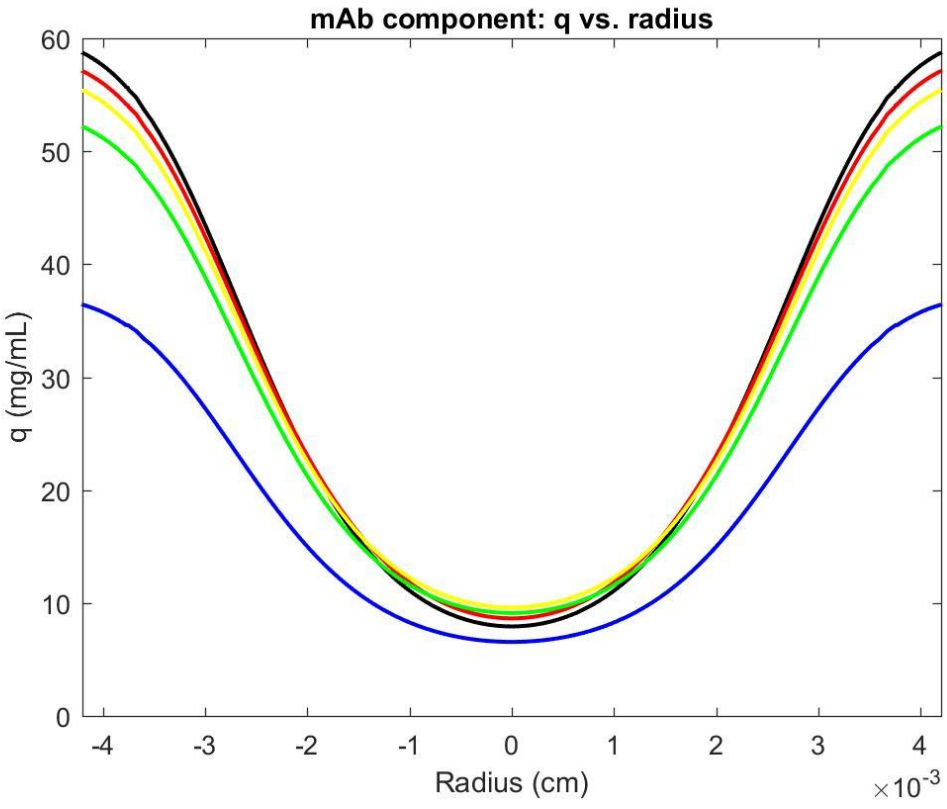
(A)


hIgG component: intensity vs. radius
average intensity
radius (cm)
×10^-3
1 min
15 min
30 min
60 min
240 min

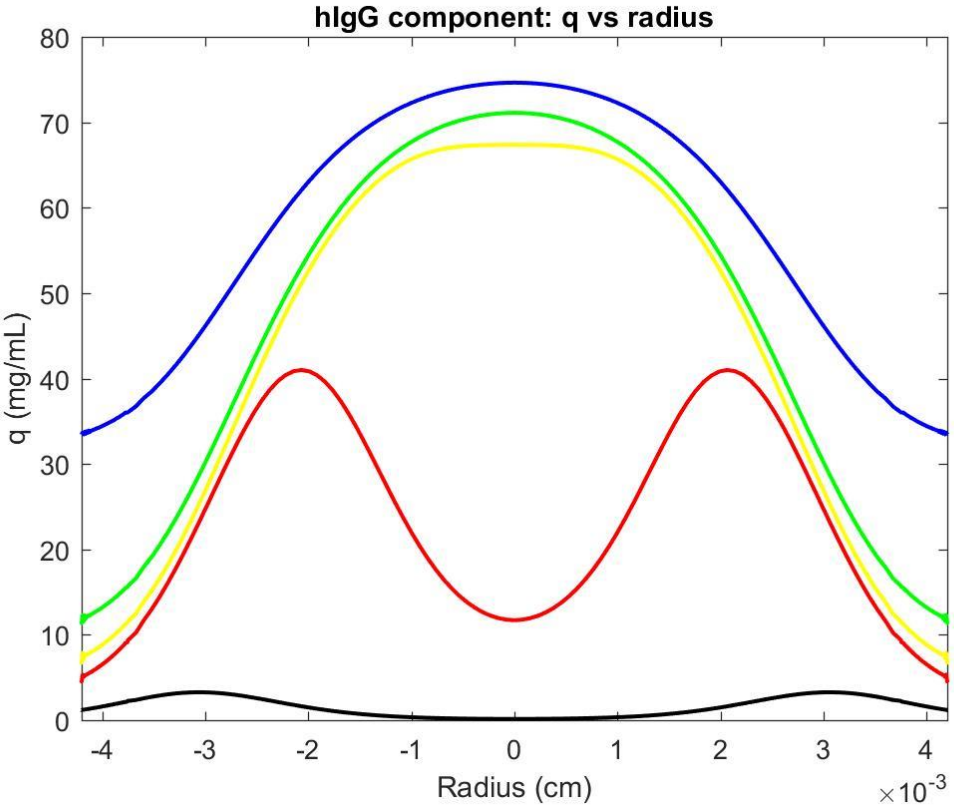
(B)

Figure 27: Intensity vs. radius profiles
for (A) mAb and (B) hIgG polyclonal solution at selected times. Black is 1 min, red is 15 min, yellow is 30 min, green is 60 min, and blue is 4 hours.

It should be noted that the mAb and hIgG intensity-concentration proportionalities shown in Figure 27 above are not identical. The relative concentrations for these two components cannot be related to each other based on the intensity scalings. When the computational graphs of concentration vs. radial position are compared with these plots, it can be seen that both the mAb and hIgG components follow the same general trends. Plots of simulated concentration versus radial position for the same time points can be seen below in Figures 28 and 29, for the kinetic and equilibrium case respectively. The plots of the hIgG component depict the sum of the three simulated binding components, strong, moderate and weak, of the polyclonal solution.

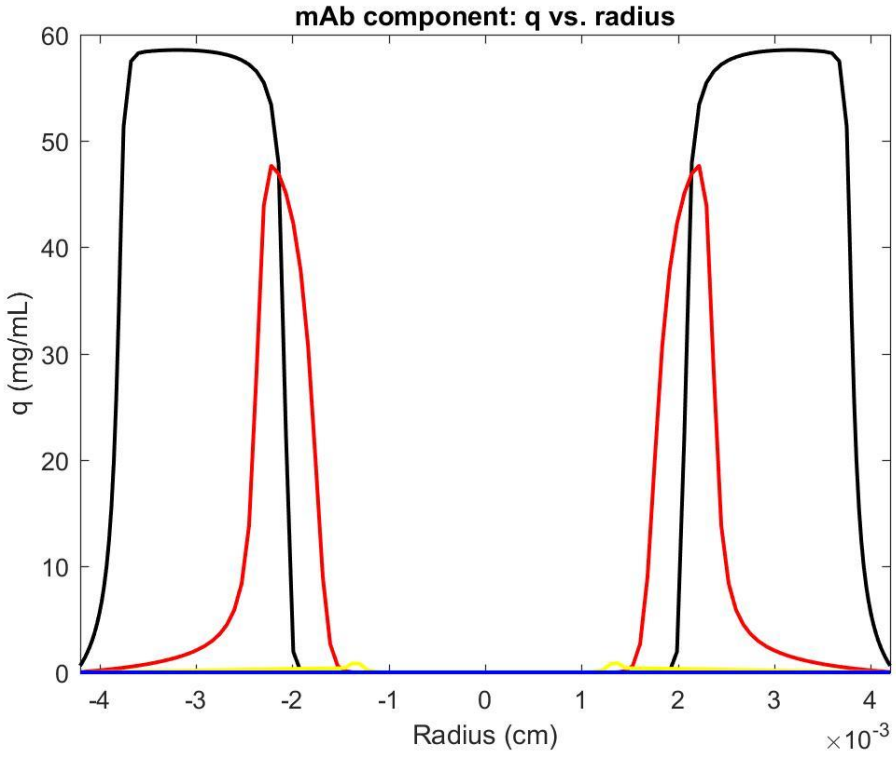


(A)

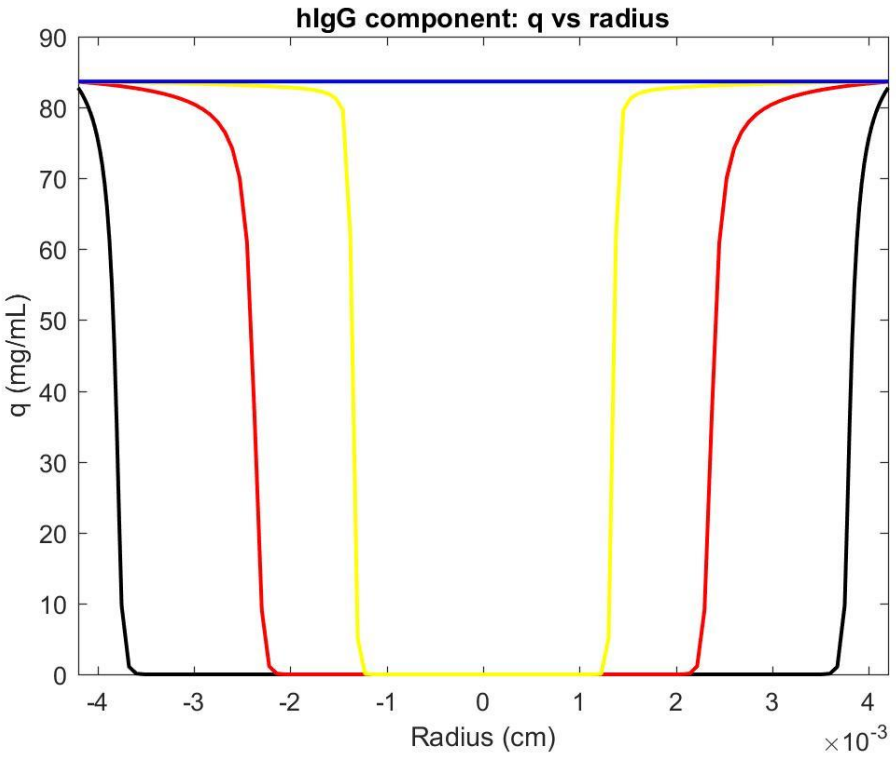


(B)

Figure 28: Kinetic case, computational q vs. radius profiles
for (A) mAb and (B) hIgG polyclonal solution at selected times. Black is 1 min, red is 15 min, yellow is 30 min, green is 60 min, and blue is 4 hours.



(A)



(B)

Figure 29: Equilibrium case, computational q vs. radius profiles for (A) mAb and (B) hIgG polyclonal solution at selected times. Black is 1 min, red is 15 min, yellow is 30 min, green is 60 min, and blue is 4 hours.

The experimental case shows results that are consistent with the shrinking core model (Weinberg et al., 2017). The data follows the same trends, but is very different from the kinetic case as it takes the kinetic case time to reach equilibrium, making the profiles more diffuse. Again, it should be noted that the computational model for the kinetic case does not produce results that match well with the same as the experimental ones, although they follow the same trends. This could possibly be contributed to the estimation of kinetic binding parameters, specifically the k_f and k_r constants. Given that the equilibrium simulations are a better qualitative match with the experimental results, it is suggested that the kinetic constants must be faster than the estimates used here. More experimental work would ultimately be needed to tune up these parameters and improve accuracy of the model.

Conclusion

The ability to model binding and displacement as seen above is integral to our computational approach. The simulations presented in this paper show promising results for the ability to model antibody movement and displacement in a protein A chromatography system. Qualitative behavior of model is correct with given literature parameter estimates when compared to initial experimental results. It should be noted, however, that the kinetics of the computational model are not yet correctly captured when the model is compared to experimental results. This is likely an effect of the approximate k_f and k_r constants used. The input of more exact parameters will increase the accuracy of the given model and its predictions.

Displacement Phenomenon for Future Separations

Both computational and experimental results have shown that competitive binding between different species is significant, and this has implications for future separations. Because of the

strong binding of protein A to antibodies, the observation of a binding and displacement phenomenon was unexpected. Although it is not clear yet whether these differences in binding affinities can be exploited for better separations, the model presented here is the first step towards exploitation if it is possible. This presented theory and model provides a basis to simulate protein A affinity chromatography on the column scale by incorporating the newly developed pH isotherm into the general rate model which would give the ability to model break through curves with binding and displacement at the column level. Ultimately, it could be possible to further resolve charge variants of monoclonal antibodies and assess charge heterogeneity profiles.

Another benefit to the model is that it allows for the comparison of different types of chromatography media. The parameters used in this model are from experiments with CaptivA PriMab chromatography media, however, parameters for MabSelect and MabSelect SuRe are also available in the literature (Weinberg et al., 2017). Currently, the necessary information is available to computationally compare these different types of media and future work in this area could lead to interesting findings.

Future Work

Moving forward, the immediate next step would be to incorporate the transport of protons and the pH-dependent isotherm into the computational model, so that we can describe absorption and desorption behavior as a function of external pH. Further steps include the incorporation of bead microscopic balances as part of multicomponent general rate model to predict chromatographic performance, in which case, assumptions such as an infinite bath for the beads can no longer be assumed. As the kinetics of the computational model were slightly off, it is also necessary to do further experimental tests to determine more accurate k_f and k_r constants. Ultimately, it would be

advantageous to computationally simulate elution profiles for antibody solutions, so it could be understood how changes in parameters and operating conditions affect separations performance.

References

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., . . . Bourne, P. E. (2000). The Protein Data Bank. *28*, 235-242. Nucleic Acids Research. doi:10.1093/nar/28.1.235

Bowes, B. D., & Lenhoff, A. M. (2011). Protein adsorption and transport in dextran-modified ion-exchange media. II. Intraparticle uptake and column breakthrough. *Journal of Chromatography A*, *1218*(29), 4698–4708. https://doi.org/10.1016/j.chroma.2011.05.054

Burton, D. R. (1985). Immunoglobulin G: Functional sites. *Molecular Immunology*, *22*(3), 161–206. https://doi.org/10.1016/0161-5890(85)90151-8

Buss, N. A. P. S., Henderson, S. J., Mcfarlane, M., Shenton, J. M., & Haan, L. De. (2010). Monoclonal antibody therapeutics : history and future. *Current Opinion in Pharmacology*, *12*(5), 615–622. https://doi.org/10.1016/j.coph.2012.08.001

Carta, G., & Jungbauer, A. (2010). *Protein Chromatography: Process Development and Scale-Up*. Wiley-VCH.

Crank, J. (1975). *The Mathematics of Diffusion* (2 ed.). Oxford University Press.

Deisenhofer, J. (1981). Crystallographic Refinement and Atomic Models of a Human Fc Fragment and Its Complex with Fragment B of Protein A from Staphylococcus aureus at 2.9- and 2.8-Å Resolution. *Biochemistry*, *20*(9), 2361–2370. https://doi.org/10.1021/bi00512a001

DeLano, W. L., Ultsch, M. H., De Vos, A. M., & Wells, J. A. (2000). Convergent solutions to

binding at a protein-protein interface. *Science*, *287*(5456), 1279–1283. https://doi.org/10.1126/science.287.5456.1279

Dransart, B., Wheeler, A., Tran, C., Abalos, R., Hong, T., Kluck, B., . . . Sanaie, N. (2018). *Evaluation of a Next-Generation Protein A Chromatography Resin for the Purification of Monoclonal Antibodies.* BioProcess International. Retrieved from http://www.bioprocessintl.com/2018/january-2018-ge-special-report/evaluation-of-a-next-generation-protein-a-chromatography-resin-for-the-purification-of-monoclonal-antibodies/

Ghose, S., Allen, M., Hubbard, B., Brooks, C., & Cramer, S. M. (2005). Antibody variable region interactions with protein A: Implications for the development of generic purification processes. *Biotechnology and Bioengineering*, *92*(6), 665–673. https://doi.org/10.1002/bit.20729

González-Valdez, J., Yoshikawa, A., Weinberg, J., Benavides, J., Rito-Palomares, M., & Przybycien, T. M. (2014). Toward improving selectivity in affinity chromatography with PEGylated affinity ligands: The performance of PEGylated protein A. *Biotechnology Progress*, *30*(6), 1364–1379. https://doi.org/10.1002/btpr.1994

Hahn, R., Schlegel, R., & Jungbauer, A. (2003). Comparison of protein A affinity sorbents. *J.Chromatogr.B Analyt.Technol.Biomed.Life Sci.*, *790*(1570-0232), 35–51. Retrieved from file://o/Referenzmanager/Immunglobulin.PDFs/Hahn 2003 .pdf

Hahn, T., Huuk, T., Heuveline, V., & Hubbuch, J. (2015). Simulating and Optimizing Preparative Protein Chromatography with ChromX. *Journal of Chemical Education*, *92*(9), 1497–1502. https://doi.org/10.1021/ed500854a

Huang, B., Liu, F.-F., Dong, X.-Y., & Sun, Y. (2011). Molecular Mechanism of the Affinity Interactions between Protein A and Human Immunoglobulin G1 Revealed by Molecular Simulations. *The Journal of Physical Chemistry B, 115*(14), 4168–4176. https://doi.org/10.1021/jp111216g

Israelachvili, J. N. (2011). *Intermolecular and Surface Forces* (3 ed.). Elsevier.

Khawli, L. A., Goswami, S., Hutchinson, R., Kwong, Z. W., Yang, J., Wang, X., … Motchnik, P. (2010). Charge variants in IgG1: Isolation, characterization, in vitro binding properties and pharmacokinetics in rats. *MAbs*, *2*(6), 613–624. https://doi.org/10.4161/mabs.2.6.13333

Lee, S. H., Hoshino, Y., Randall, A., Zeng, Z., Baldi, P., Doong, R. A., & Shea, K. J. (2012). Engineered synthetic polymer nanoparticles as IgG affinity ligands. *Journal of the American Chemical Society*, *134*(38), 15765–15772. https://doi.org/10.1021/ja303612d

Market Analysis. (2016). *Monoclonal Antibodies (mAbs) Market Analysis By Source (Chimeric, Murine, Humanized, Human), By Type of Production, By Indication (Cancer, Autoimmune, Inflammatory, Infectious, Microbial, Viral Diseases), By End-use, And Segment Forecasts 2018-2024.* San Fransisco, CA: Grand View Research.

Martin, C., Iberer, G., Ubiera, A., & Carta, G. (2005). Two-component protein adsorption kinetics in porous ion exchange media. *Journal of Chromatography A, 1079*(1–2 SPEC. ISS.), 105–115. https://doi.org/10.1016/j.chroma.2005.03.007

MathWorks. (2018). *Types of solvers*. Retrieved from MathWorks: https://www.mathworks.com/help/simulink/ug/types-of-solvers.html#f11-44943

Murphy, K. (2012). *Janeway's Immunobiology* (8 ed.). London and New York: Garland Science.

Nelson, D. L., & Cox, M. M. (n.d.). *Lehninger Principles of Biochemistry* (5 ed.). New York: W. H. Freeman and Company.

Nilsson, B., & Andersson, N. (2017). *Preparative Chromatography for Separation of Proteins: Simulation of Process Chromatography* (1st ed.). (A. Staby, A. S. Rathore, & S. Ahuja, Eds.) Lund: John Wiley & Sons.

Pabst, T. M., Thai, J., & Hunter, A. K. (2018). Evaluation of recent Protein A stationary phase innovations for capture of biotherapeutics. *Journal of Chromatography A, 1554*, 45–60. https://doi.org/10.1016/j.chroma.2018.03.060

Parekh, B. S., Srivastava, A., Sundaram, S., Ching-Heish, M., Goldstein, J., Barry, M., & Zhou, Q. (2018). Correlating charge heterogeneity data generated by agarose gel isoelectric focusing and ion exchange chromatography methods. *Journal of Chromatography B: Analytical Technologies in the Biomedical and Life Sciences*, *1073*(December 2017), 1–9. https://doi.org/10.1016/j.jchromb.2017.11.043

Perez-Almodovar, E. X., & Carta, G. (2009). IgG adsorption on a new protein A adsorbent based on macroporous hydrophilic polymers. I. Adsorption equilibrium and kinetics. *Journal of Chromatography A*, *1216*(47), 8339–8347. https://doi.org/10.1016/j.chroma.2009.09.017

Przybycien, T., & Lane, V. (2018). Competitive adsorption of antibodies in protein A chromatography. *Recovery of Biological Products Conference XVIII.*

Sandoval, G., Andrews, B. A., & Asenjo, J. A. (2012). Elution relationships to model affinity chromatography using a general rate model. *Journal of Molecular Recognition*, *25*(11), 571–579. https://doi.org/10.1002/jmr.2223

Sayle, R., & Milner-White, E. J. (1995, September). RasMol: Biomolecular graphics for all. *20(9)*, 374. Trends in Biochemical Sciences (TIBS).

Sjöquist, J. (1972). Protein A from Staphylococcus aureus. *FEBS Letters*, *28*(1), 73–76.

Thornton, A. C., & Johnson, A. L. (1996). CADET: A software support tool for constraint processes in embodiment design. *Research in Engineering Design - Theory, Applications, and Concurrent Engineering*, *8*(1), 1–13. https://doi.org/10.1007/BF01616553

Weinberg, J., Zhang, S., Crews, G., Carta, G., & Przybycien, T. (2018). Chemical modification of protein A chromatography ligands with polyethylene glycol. I: Effects on IgG adsorption equilibrium, kinetics, and transport. *Journal of Chromatography A*, *1546*, 77–88. https://doi.org/10.1016/j.chroma.2018.03.011

Weinberg, J., Zhang, S., Crews, G., Healy, E., Carta, G., & Przybycien, T. (2017). Polyclonal and monoclonal IgG binding on protein A resins—Evidence of competitive binding effects. *Biotechnology and Bioengineering*, *114*(8), 1803–1812. https://doi.org/10.1002/bit.26286

Weinberg, J. (2017). Competitive IgG Adsorption on Protein A Chromatography Resins and Improving Resin Performance with PEGylated Ligands. Carnegie Mellon University.

Zhang, T., Bourret, J., & Cano, T. (2011). Isolation and characterization of therapeutic antibody charge variants using cation exchange displacement chromatography. *Journal of Chromatography A*, *1218*(31), 5079–5086. https://doi.org/10.1016/j.chroma.2011.05.061

Appendix I: Abbreviations

A	Abbreviation used for protein A ligand
AG	Abbreviation used for protein A – hIgG complex
CLSM	Confocal laser scanning microscope
Fab	Antigen-binding fragment of antibody
FB	Fraction B domain of protein A ligand
Fc	Fragment crystallizable region of antibody
G	Abbreviation used for hIgG antibody
hIgG	Human Immunoglobulin G
IgG	Immunoglobulin G
mAb	Monoclonal antibody
PDE	Partial differential equation
SCM	Shrinking Core Model
USD	United States Dollars

Appendix II: Parameter Tables

Table 3: Process Parameters

Parameter	Value	Units	Description	Source
c	--	$\frac{mg\ antibody}{mL\ solute}$	Concentration of antibody in liquid phase	MATLAB output
c^*	--	--	Dimensionless concentration of antibody in liquid phase	MATLAB output
$c_{0,mAb}$	2	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of mAb antibody in solution surrounding chromatography media	(Weinberg et al., 2017)
$c_{0,\Sigma hIgG}$	2	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of polyclonal antibody concentration in solution surrounding chromatography media	(Weinberg et al., 2017)
$c_{0,\ "weak"\ hIgG}$	.277	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of "weak" binding hIgG antibody in solution surrounding chromatography media	Calculated using elution profiles
$c_{0,\ "moderate"\ hIgG}$	.994	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of "moderate" binding hIgG antibody in solution surrounding chromatography media	Calculated using elution profiles
$c_{0,\ "strong"\ hIgG}$	.728	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of "strong" binding hIgG antibody in solution surrounding chromatography media	Calculated using elution profiles
$c_{1,mAb}$	0	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of mAb in chromatography media sphere	(Weinberg et al., 2017)
$c_{1,\Sigma hIgG}$	0	$\frac{mg\ antibody}{mL\ solute}$	Initial concentration of polyclonal antibody solution in chromatography media sphere	(Weinberg et al., 2017)
Dm	90	μm	Average diameter of CaptivA PriMab chromatography spheres	(Weinberg et al., 2017)
D_0	3.7×10^{-7}	$\frac{cm^2}{second}$	Bulk diffusivity of hIgG molecules	(Perez-Almodovar & Carta, 2009)

$\boldsymbol{D_{pore,mAb}}$	7.5 × 10^{-8}	$\frac{cm^2}{minute}$	Diffusivity constant for mAb in CaptivA PriMab, including hindrance, porosity, and tortuosity	(Weinberg et al., 2018)
$\boldsymbol{D_{pore,\Sigma hIgG}}$	7.5 × 10^{-8}	$\frac{cm^2}{minute}$	Diffusivity constant for hIgG in CaptivA PriMab, including hindrance, porosity, and tortuosity	(Weinberg et al., 2018)
$\boldsymbol{\varepsilon}$	.65	--	Accessible interstitial porosity of CaptivA PriMab	(Weinberg et al., 2018)
$\boldsymbol{\Delta G_{bind,mAb}}$	9.50	$\frac{kcal}{mol}$	Free energy of binding for mAb from $K_{eff,mAb}$	Calculated
$\boldsymbol{\Delta G_{bind,"weak"hIgG}}$	8.91	$\frac{kcal}{mol}$	Free energy of binding for weak binding hIgG from $K_{eff,\ "weak"\ hIgG}$	Calculated
$\boldsymbol{\Delta G_{bind,"moderate"hIgG}}$	9.62	$\frac{kcal}{mol}$	Free energy of binding for moderate strength binding hIgG from $K_{eff,\ "moderate"\ hIgG}$	Calculated
$\boldsymbol{\Delta G_{bind,"strong"hIgG}}$	9.93	$\frac{kcal}{mol}$	Free energy of binding for strong binding hIgG from $K_{eff,\ "strong"\ hIgG}$	Calculated
$\boldsymbol{\Delta G_{bind,i,o}}$	9.64	$\frac{kcal}{mol}$	Intrinsic free energy of binding from $K_{bind,i.o}$	Calculated
$\boldsymbol{\Delta\Delta G_{bind,o,(i-j)}}$	2.7	$\frac{kcal}{mol\ \Delta pH}$	Change in intrinsic binding pH related to difference in elution pH	Calculated
$\boldsymbol{K_{a,mAb}}$	61.81	$\frac{mL}{mg}$	Association constant for mAb and protein A at neutral pH	(Weinberg, 2017)
$\boldsymbol{K_{a,\Sigma hIgG}}$	128.0	$\frac{mL}{mg}$	Association constant for polyclonal hIgG and protein A at neutral pH	(Weinberg, 2017)
$\boldsymbol{K_{a,\ "weak"\ hIgG}}$	22.95	$\frac{mL}{mg}$	Association constant for weak binding hIgG and protein A at neutral pH	Free energy calculation
$\boldsymbol{K_{a,\ "moderate"\ hIgG}}$	76.31	$\frac{mL}{mg}$	Association constant for moderate strength binding hIgG and protein A at neutral pH	Free energy calculation
$\boldsymbol{K_{a,\ "strong"\ hIgG}}$	128.0	$\frac{mL}{mg}$	Association constant for strong binding hIgG and protein A at neutral pH	Free energy calculation
$\boldsymbol{K'_a}$	--	M^{-1}	Binding constant for the protonation of protein A ligand	--
$\boldsymbol{K''_a}$	--	M^{-1}	Binding constant for the protonation of hIgG antibody	--

K_{bind}	--	M^{-1}	Binding constant for the AG complex	--
K'_{bind}	--	M^{-1}	Binding constant for the $A^{+}G$ complex	--
K''_{bind}	--	M^{-1}	Binding constant for the AG^{+} complex	--
$K_{eff,mAb}$	9.27×10^{6}	M^{-1}	Effective Langmuir binding constant for mAb	Calculated from elution profile
$K_{eff,\ "weak"\ hIgG}$	3.45×10^{6}	M^{-1}	Effective Langmuir binding constant for weak binding hIgG	Calculated from elution profile
$K_{eff,\ "moderate"\ hIgG}$	1.14×10^{7}	M^{-1}	Effective Langmuir binding constant for moderate strength binding hIgG	Calculated from elution profile
$K_{eff,"strong"\ hIgG}$	1.92×10^{7}	M^{-1}	Effective Langmuir binding constant for strong binding hIgG	Calculated from elution profile
$K_{obs,pH7,hIgG}$	1.17×10^{7}	M^{-1}	Intrinsic equilibrium binding constant for polyclonal hIgG solution. Assuming that both histidine residues are uncharged	Calculated from elution profile
$K_{o,mAb}$	1.12×10^{7}	M^{-1}	Intrinsic equilibrium binding constant for mAb. Assumes that both histidine residues are uncharged	Calculated from elution profile
$K_{o,\ "weak"\ hIgG}$	4.17×10^{6}	M^{-1}	Intrinsic equilibrium binding constant for weak binding component of hIgG. Assumes that both histidine residues are uncharged	Calculated from elution profile
$K_{o,\ "moderate"\ hIgG}$	1.39×10^{7}	M^{-1}	Intrinsic equilibrium binding constant for moderate strength binding component of hIgG. Assumes that both histidine residues are uncharged	Calculated from elution profile
$K_{o,"strong"\ hIgG}$	2.23×10^{7}	M^{-1}	Intrinsic equilibrium binding constant for strong binding component of hIgG. Assumes that both histidine residues are uncharged	Calculated from elution profile
k_f	--	$M^{-1}s^{-1}$	Rate of forward (complex association) reaction $k_f = K_a k_r$	Calculated

$\boldsymbol{k_r}$	.002	s^{-1}	Rate of reverse (complex dissociation) reaction	Estimate from Table 6.4 (Carta & Jungbauer, 2010)
$\boldsymbol{r}$	--	cm	Radial position	MATLAB input
$\boldsymbol{r^*}$	--	--	Dimensionless radial position	MATLAB input
$\boldsymbol{pH_{elute,mAb}}$	3.7	--	Elution pH for mAb	Calculated from elution profile
$\boldsymbol{pH_{elute,\ "weak"\ hIgG}}$	3.92	--	Elution pH for weak binding hIgG	Calculated from elution profile
$\boldsymbol{pH_{elute,\ "moderate"\ hIgG}}$	3.66	--	Elution pH for moderate strength binding hIgG	Calculated from elution profile
$\boldsymbol{pH_{elute,\ "strong"\ hIgG}}$	3.55	--	Elution pH for strong binding hIgG	Calculated from elution profile
$\boldsymbol{pK_a'}$	6.0	--	$pK_a' = -\log(K_a')$ Assumed from the typical pK_a value for a histidine residue	(Nelson & Cox)
$\boldsymbol{pK_a''}$	6.0	--	$pK_a'' = -\log(K_a'')$ Assumed from the typical pK_a value for a histidine residue	(Nelson & Cox)
$\boldsymbol{q}$	--	$\frac{mg\ antibody}{mL\ wetted\ particle}$	Concentration of adsorbed antibody	MATLAB output
$\boldsymbol{q^*}$	--	--	Dimensionless concentration of adsorbed antibody	MATLAB output
$\boldsymbol{q_{max,\ mAB}}$	61	$\frac{mg\ antibody}{mL\ wetted\ particle}$	Maximum concentration of adsorbed mAb on CaptivA PriMab media	(Weinberg et al., 2017)
$\boldsymbol{q_{max,\ \Sigma hIgG}}$	84	$\frac{mg\ antibody}{mL\ wetted\ particle}$	Maximum concentration of adsorbed polyclonal hIgG antibodies on CaptivA PriMab media	(Weinberg et al., 2017)
$\boldsymbol{q_{max,\ "weak"\ hIgG}}$	15.1	$\frac{mg\ antibody}{mL\ wetted\ particle}$	Maximum concentration of adsorbed weak binding hIgG antibodies on CaptivA PriMab media	Scaled from relative effective binding constants @pH7

$q_{max,\ "moderate"\ hIgG}$	50.1	$\frac{mg\ antibody}{mL\ wetted\ particle}$	Maximum concentration of adsorbed moderate strength binding antibodies on CaptivA PriMab media	Scaled from relative effective binding constants @pH7
$q_{max,\ "strong"\ hIgG}$	84	$\frac{mg\ antibody}{mL\ wetted\ particle}$	Maximum concentration of adsorbed strong binding antibodies on CaptivA PriMab media	Assumed the same as $q_{max,\ \Sigma hIgG}$
R	45	μm	Average outer radius of CaptivA PriMab chromatography spheres	(Weinberg et al., 2017)
R_{min}	95	--	Minimum radius input for *imfindcircles* MATLAB command	MATLAB input
R_{max}	105	--	Maximum radius input for *imfindcircles* MATLAB command	MATLAB input
t	--	$minutes$	Time point	MATLAB input
t^*	--	--	Dimensionless time point	MATLAB input
$t_{reference}$	1	$minutes$	Reference time point for dimensionless equations	MATLAB input
τ	2.8	--	Resin particle tortuosity factor	(Perez-Almodovar & Carta, 2009)
$x_{"weak"\ hIgG}$	.1386	--	Fraction of weak binding hIgG in polyclonal hIgG solution	Calculated from elution profile
$x_{"moderate"\ hIgG}$	.4972	--	Fraction of moderate strength binding hIgG in polyclonal hIgG solution	Calculated from elution profile
$x_{"strong"\ hIgG}$	.3642	--	Fraction of strong binding hIgG in polyclonal hIgG solution	Calculated from elution profile

Appendix III: Additional Figures

Adsorption and Diffusion in a Single Chromatography Sphere, Equilibrium Case

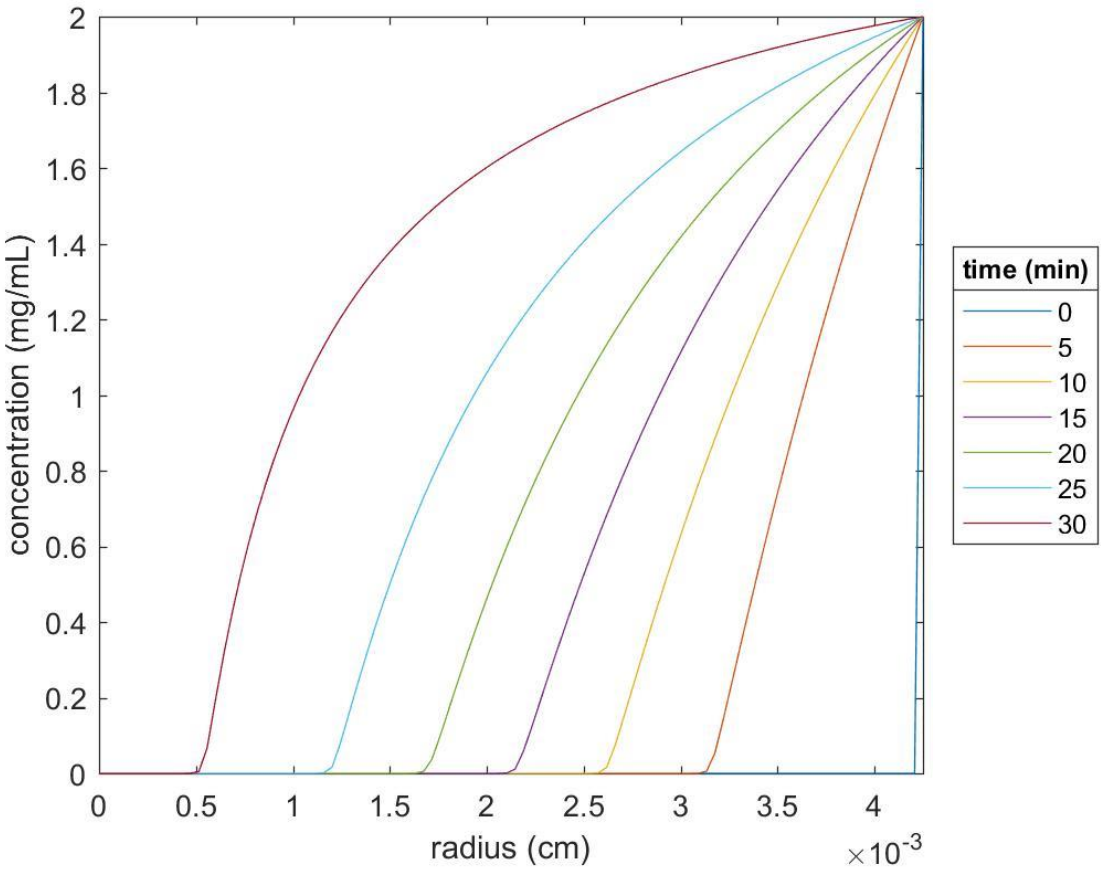


Figure 30: Concentration vs. radius profile at selected times, one component, equilibrium case. Key parameters can be found in Appendix II.

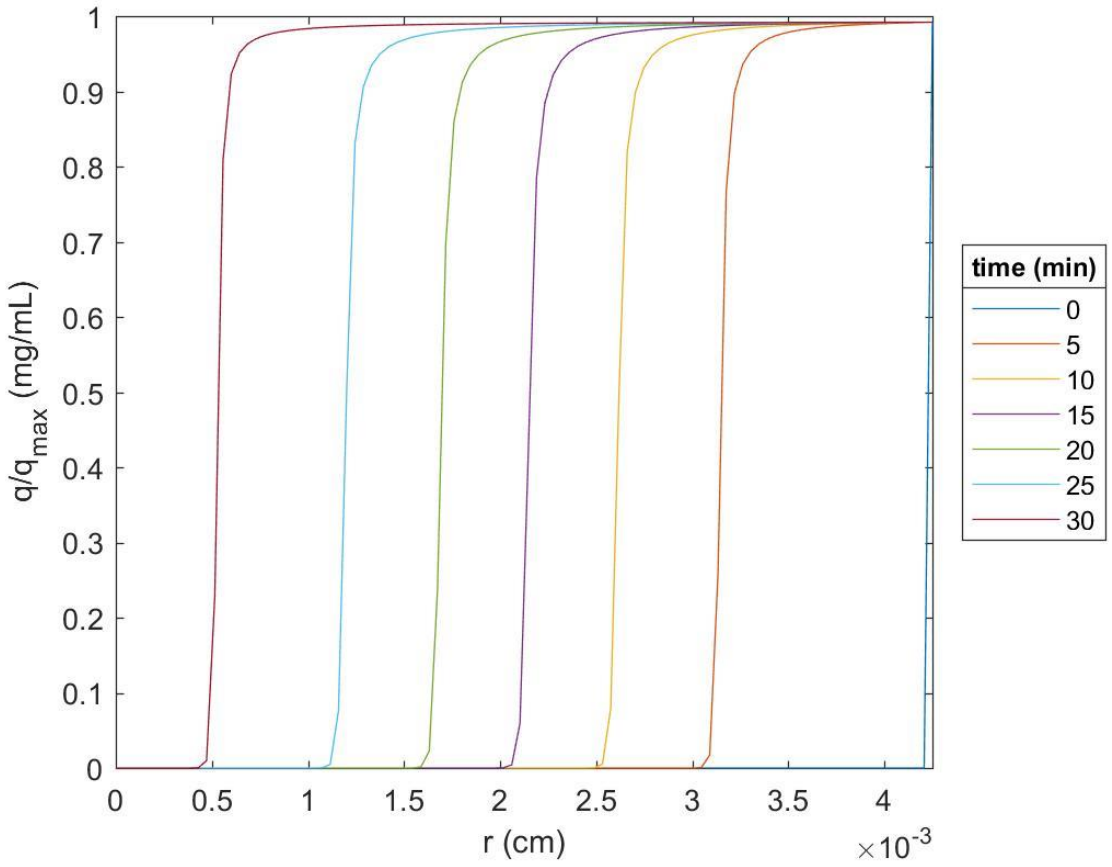


Figure 31: q vs. radius profile at selected times, one component, equilibrium case. Key parameters can be found in Appendix II.

Adsorption and Diffusion in a Single Chromatography Sphere, Kinetic Case

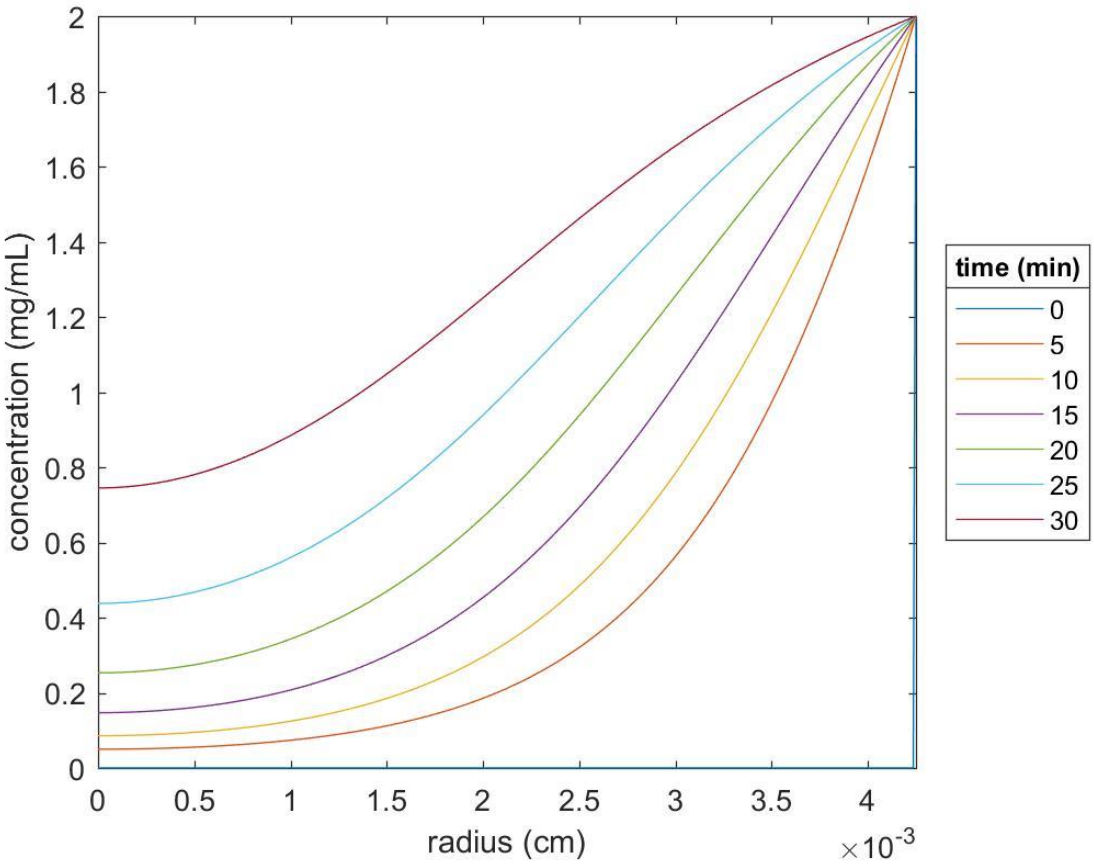


Figure 32: Concentration vs. radius profile at selected times, one component, kinetic case. Key parameters can be found in Appendix II.

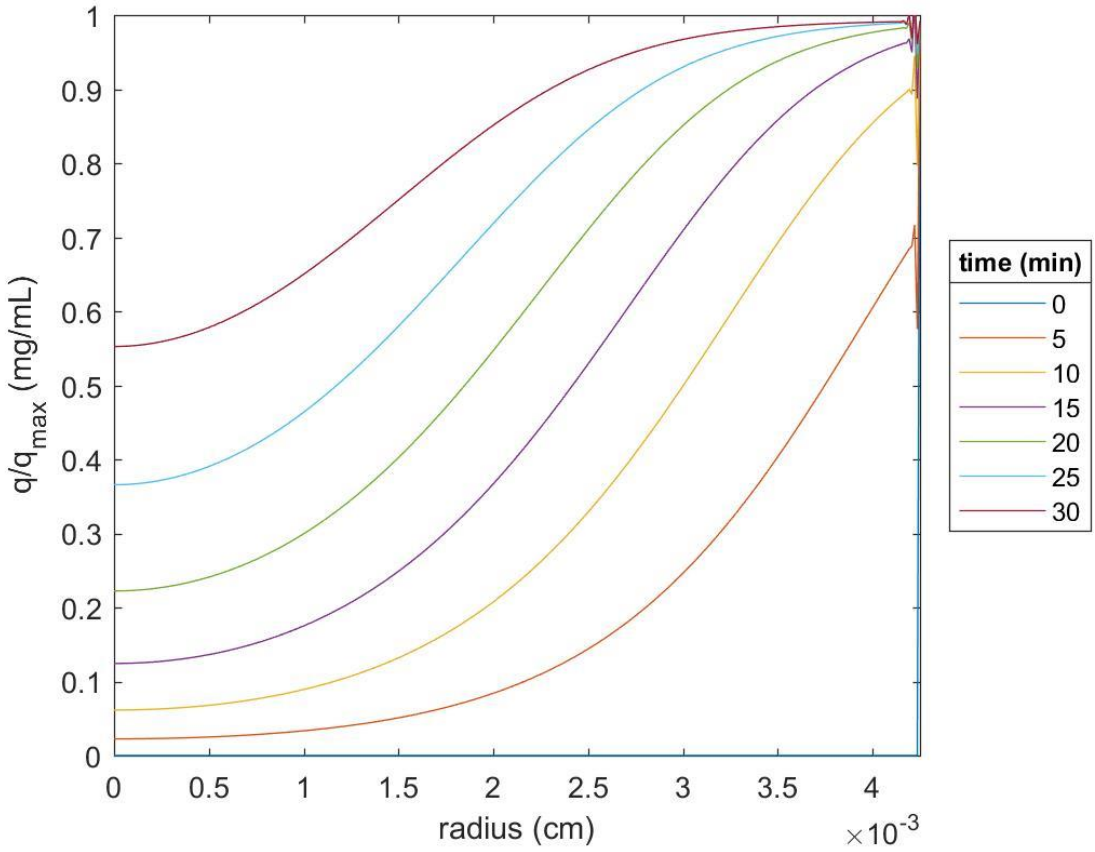
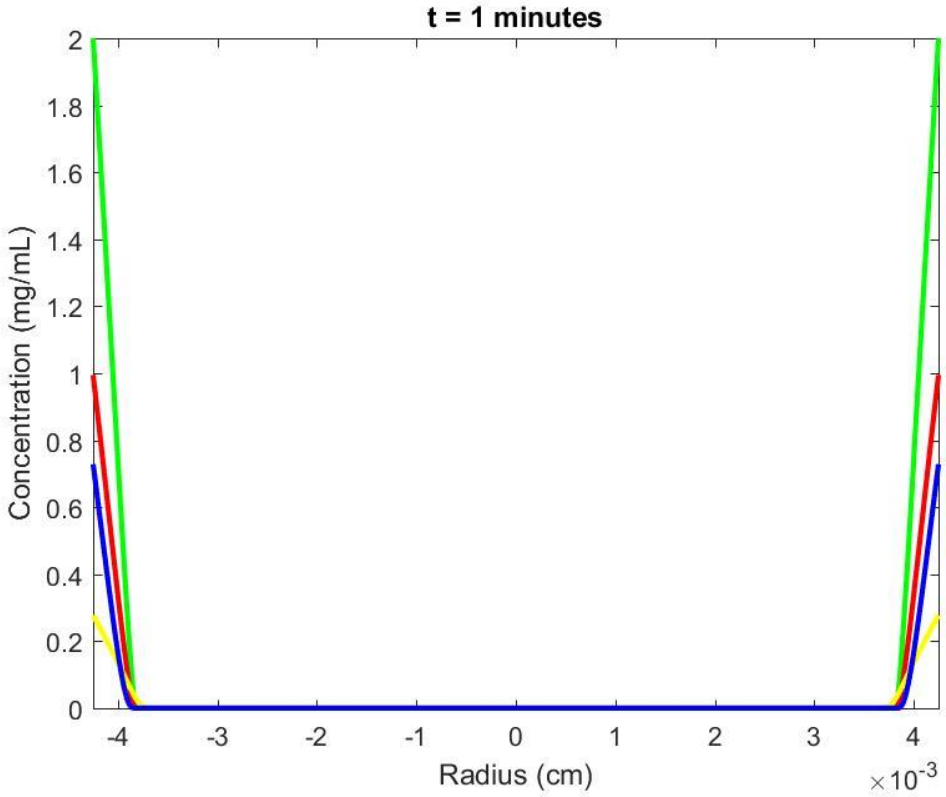
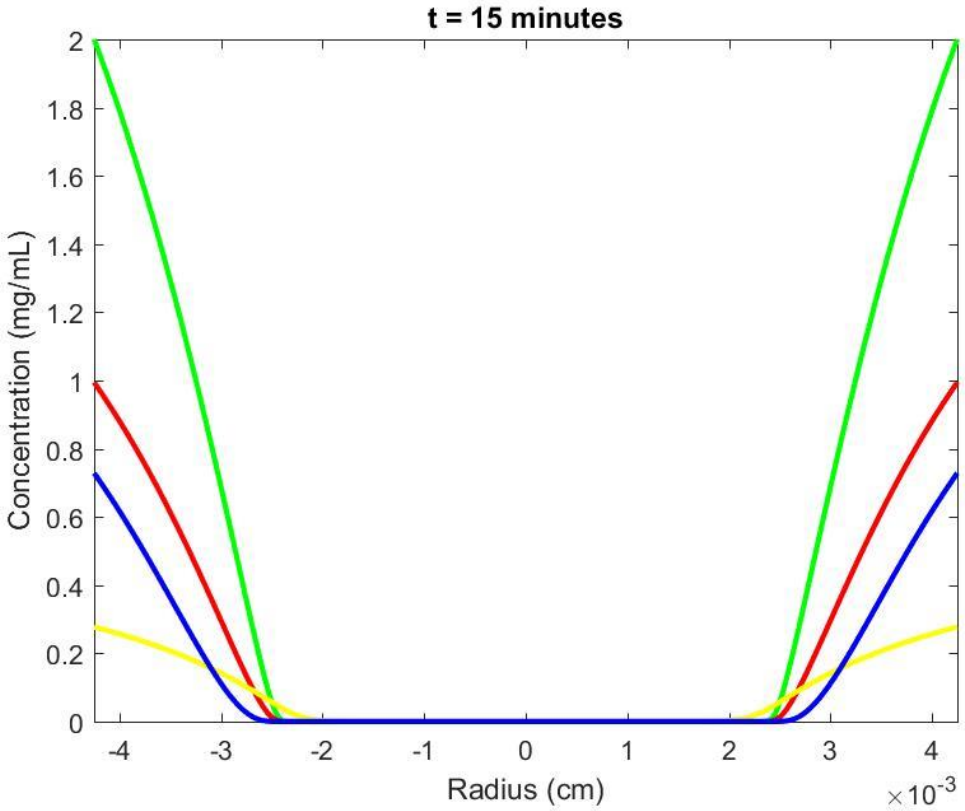


Figure 33: q vs. radius profile at selected times, one component, kinetic case. Key parameters can be found in Appendix II.

Competitive Binding Behavior in a Single Chromatographic Bead, Equilibrium Case

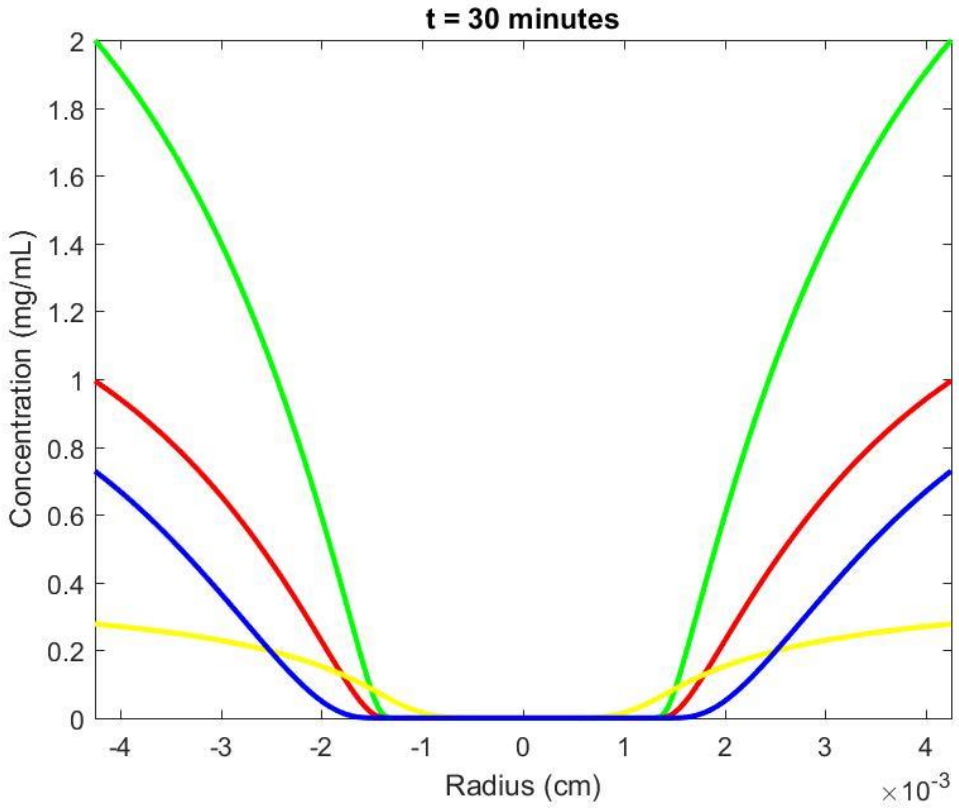


(A)

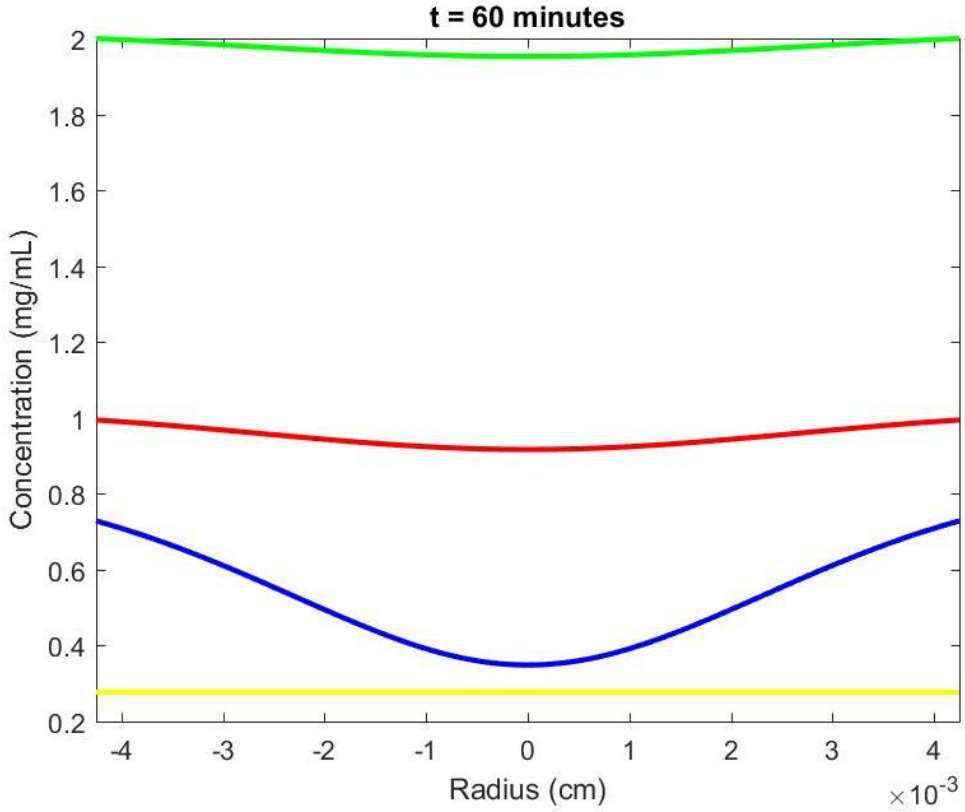


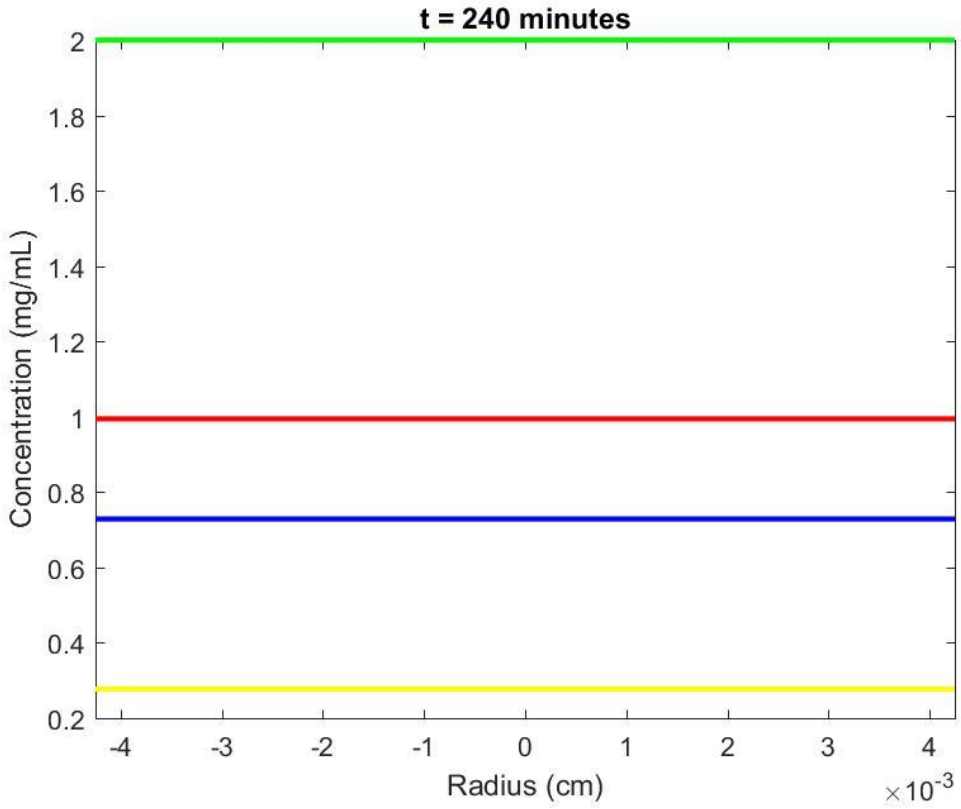
(B)

(C)


t = 30 minutes
Concentration (mg/mL)
Radius (cm)
×10^-3

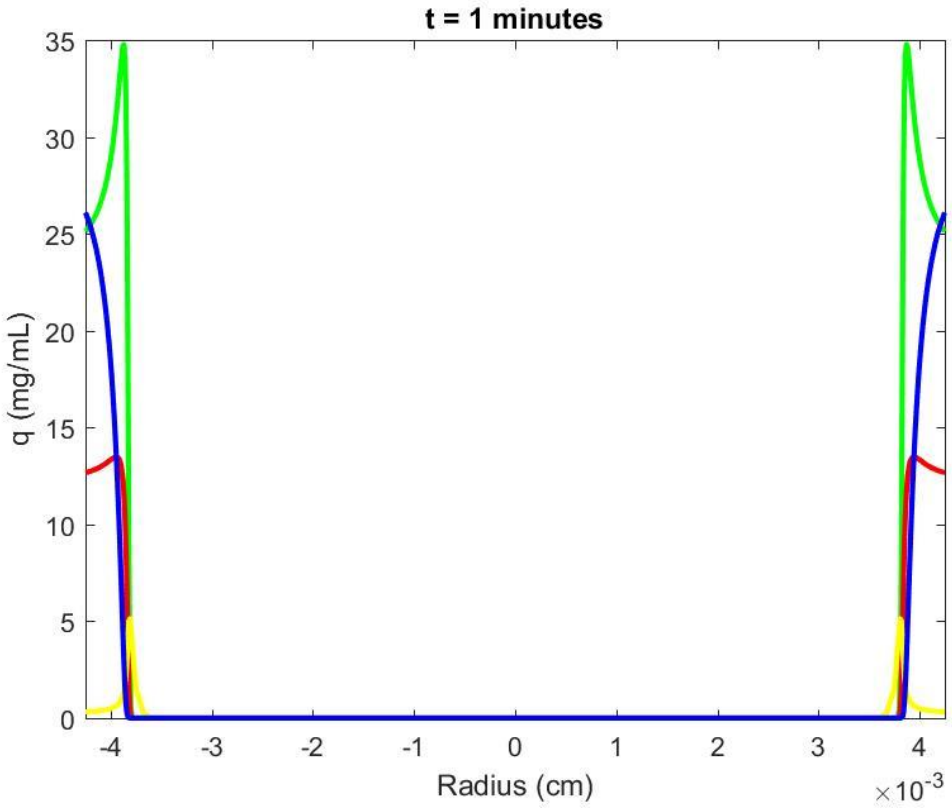
(D)


t = 60 minutes
Concentration (mg/mL)
Radius (cm)
×10^-3

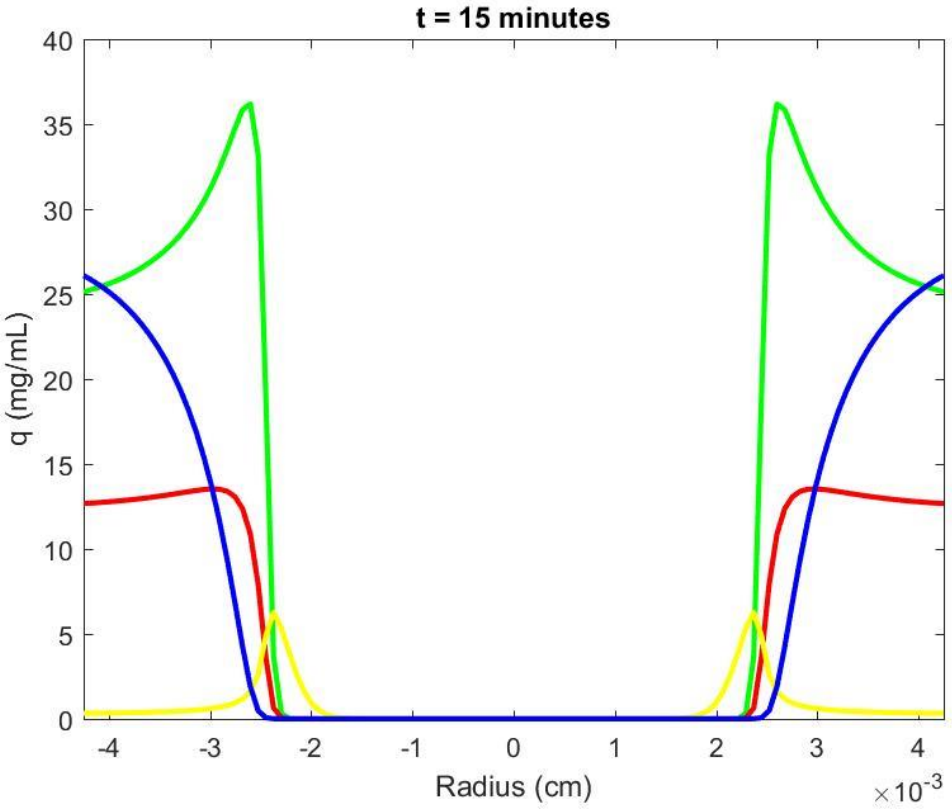


(E)

Figure 34: Equilibrium case, concentration vs. radial position for four competitive components $c_1 = 0$ for all components. mAb component shown in green, weak binding hIgG component shown in yellow, moderate strength binding hIgG component shown in red, strong binding hIgG component shown in blue. Select times are (A) 1 minute (B) 15 minutes (C) 30 minutes (D) 60 minutes (E) 4 hours. q_{max} values and other key parameters can be found in Appendix II.

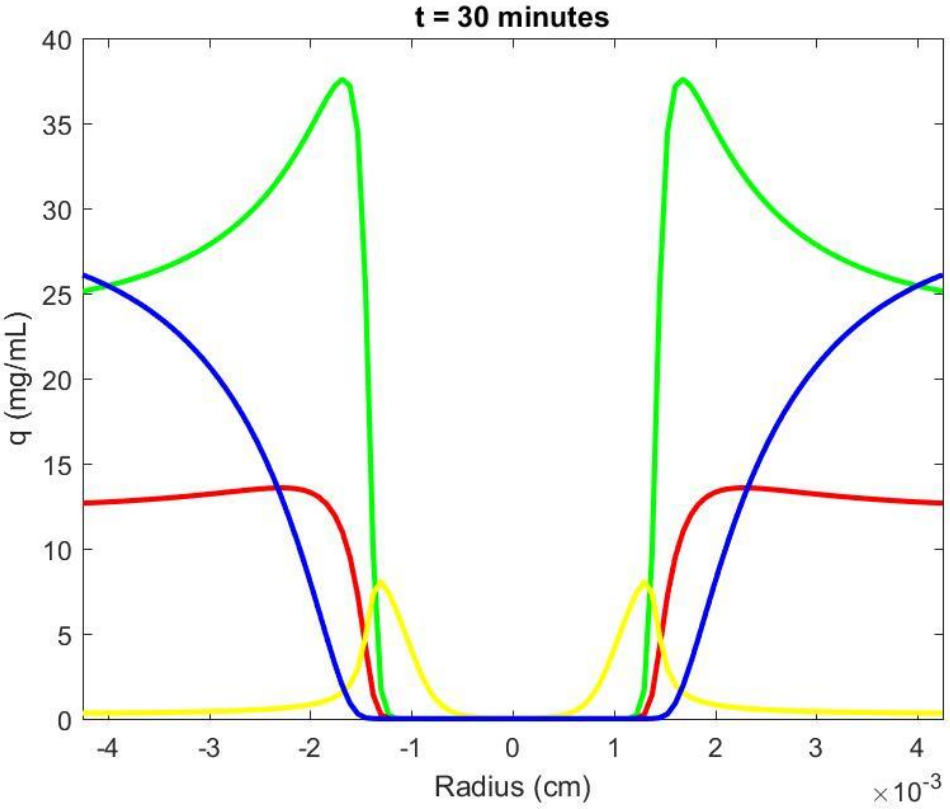

t = 1 minutes
q (mg/mL)
Radius (cm)
×10^-3

(A)

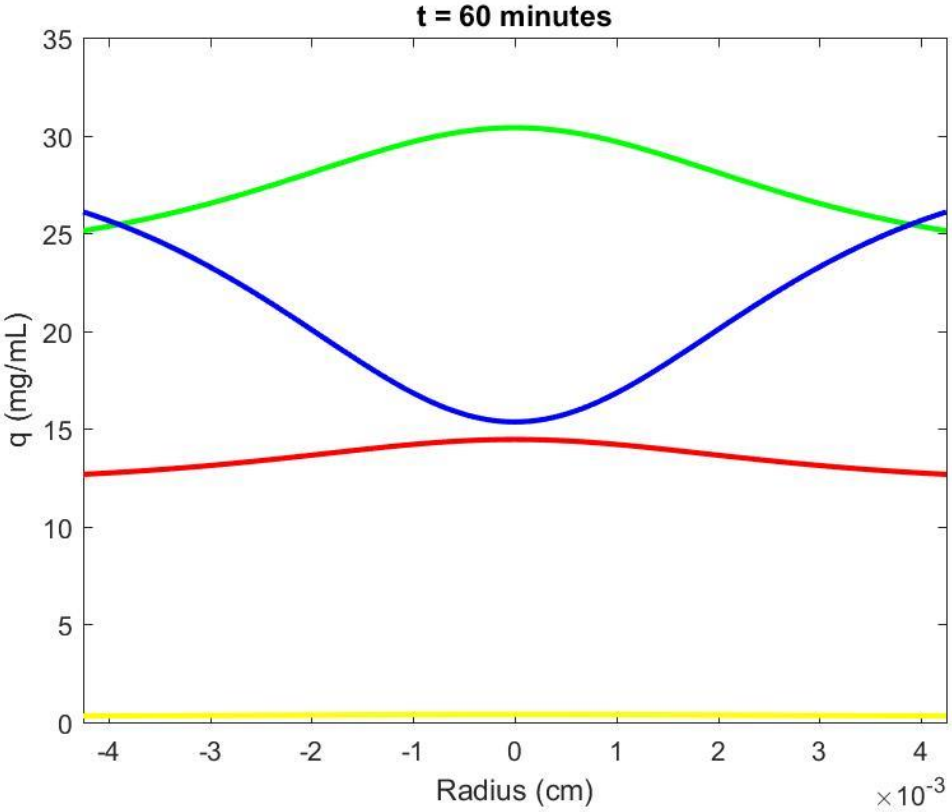

t = 15 minutes
q (mg/mL)
Radius (cm)
×10^-3

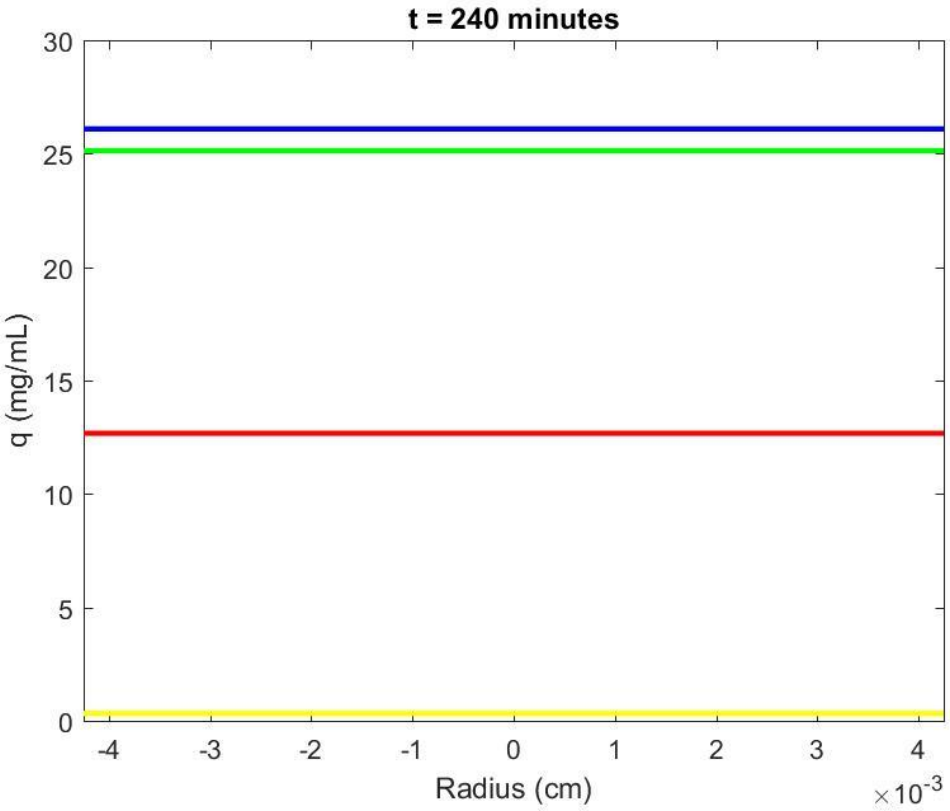
(B)

(C)


t = 30 minutes
q (mg/mL)
Radius (cm)
×10^-3

(D)

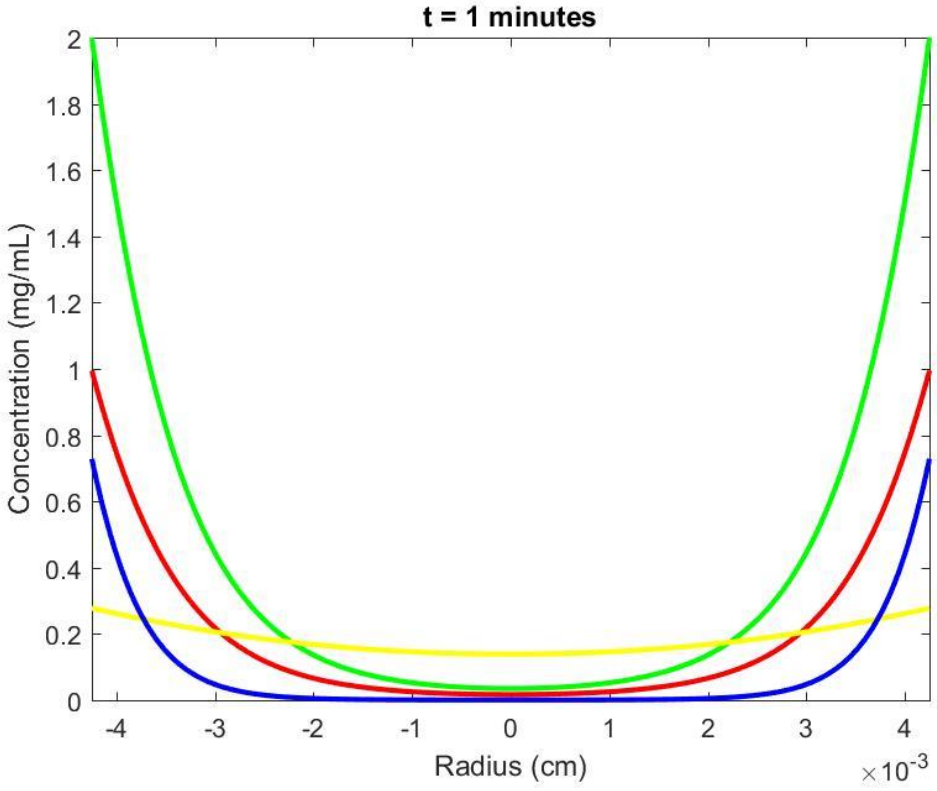

t = 60 minutes
q (mg/mL)
Radius (cm)
×10^-3



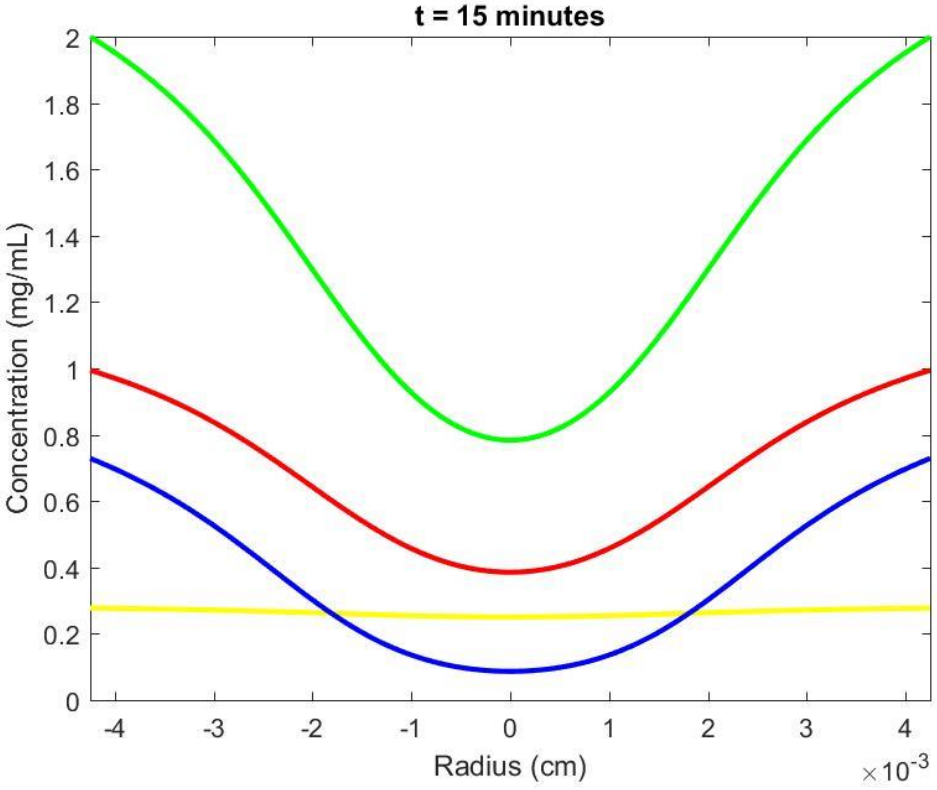
(E)

Figure 35: Equilibrium case, q vs. radial position for four competitive components. $c_1 = 0$ *for all components. mAb component shown in green, weak binding hIgG component shown in yellow, moderate strength binding hIgG component shown in red, strong binding hIgG component shown in blue. Select times are (A) 1 minute (B) 15 minutes (C) 30 minutes (D) 60 minutes (E) 4 hours.* q_{max} *values and other key parameters can be found in Appendix II.*

Competitive Binding Behavior in a Single Chromatographic Bead, Kinetic Case

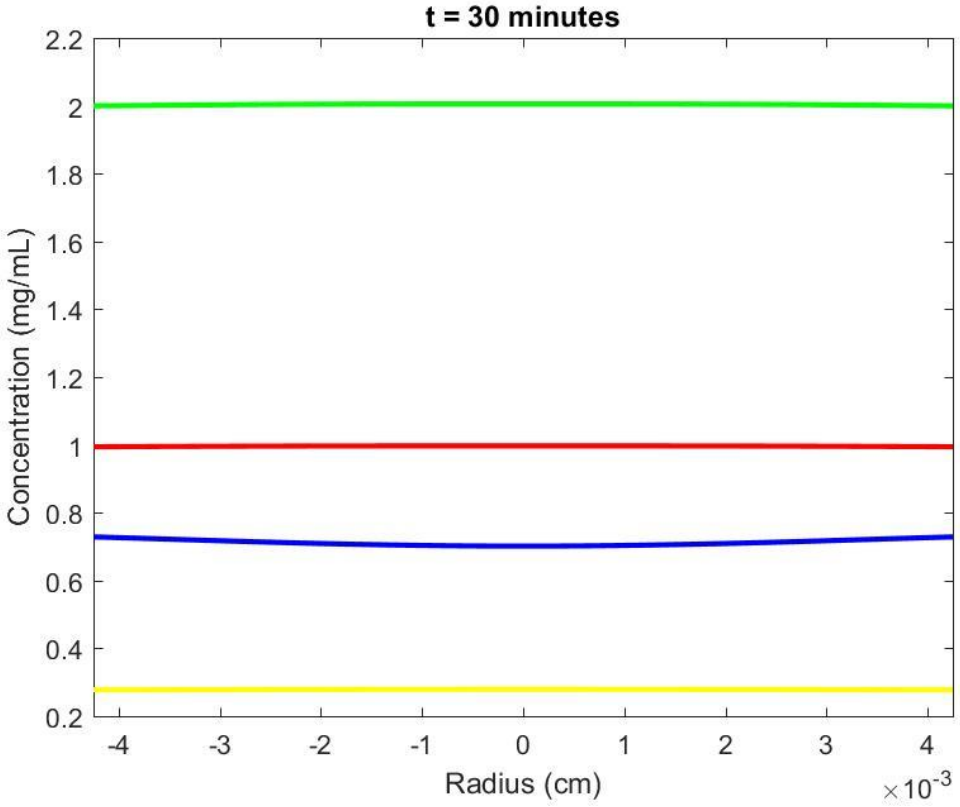


(A)

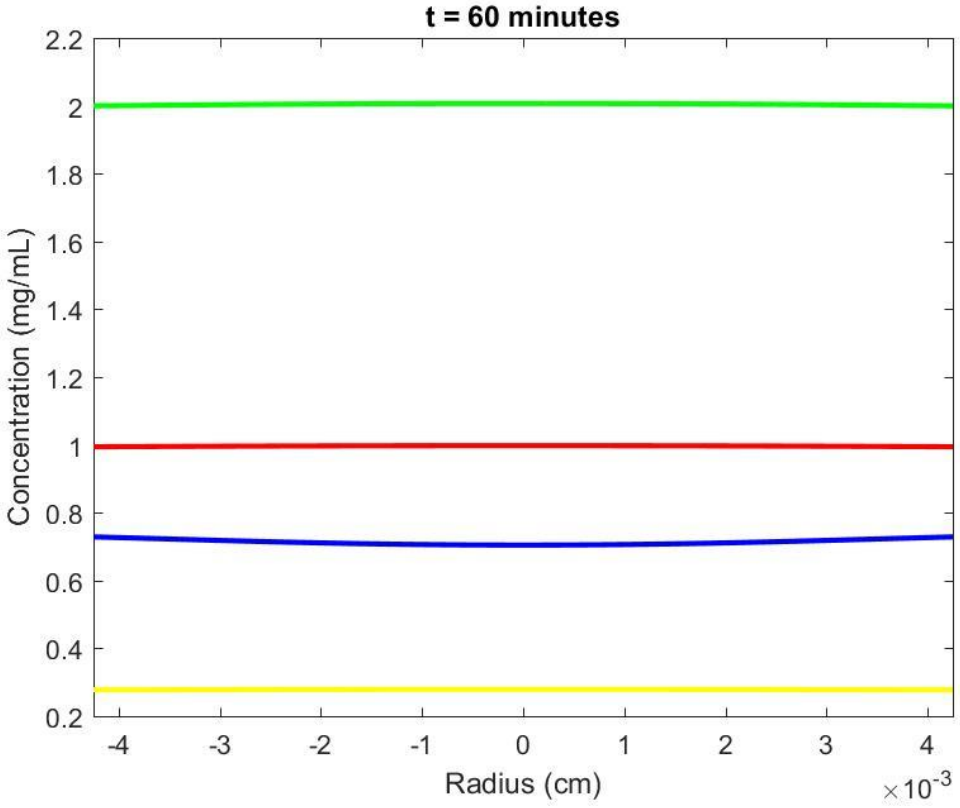


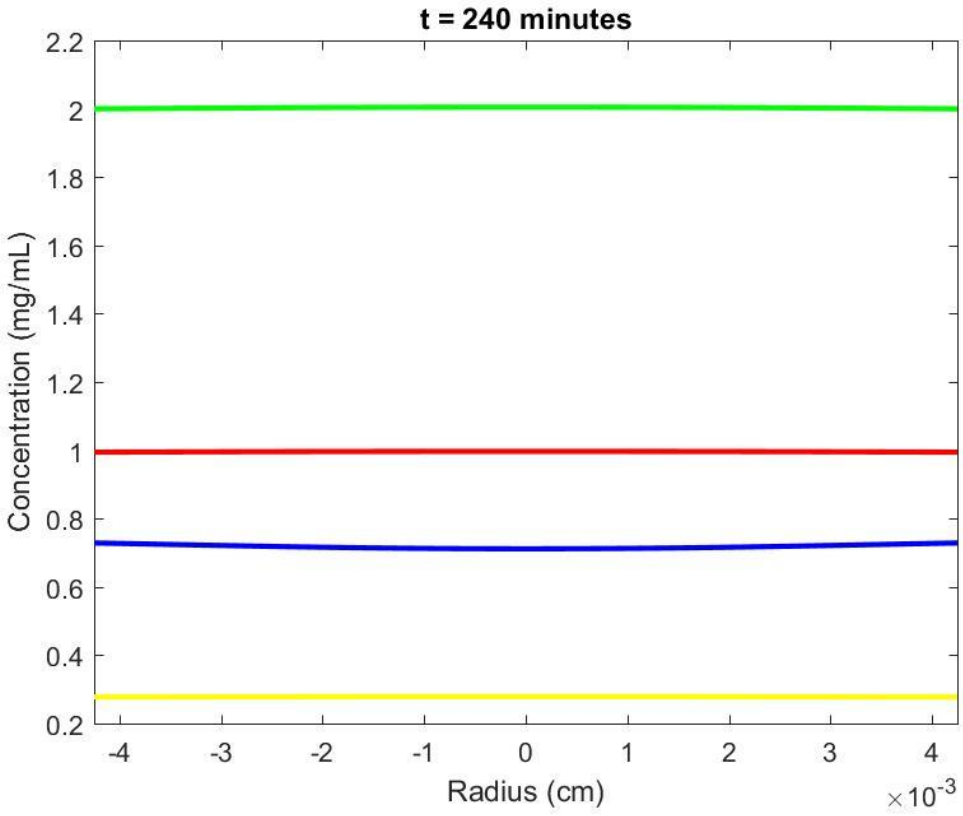
(B)

(C)


t = 30 minutes
Concentration (mg/mL)
Radius (cm)
×10^-3

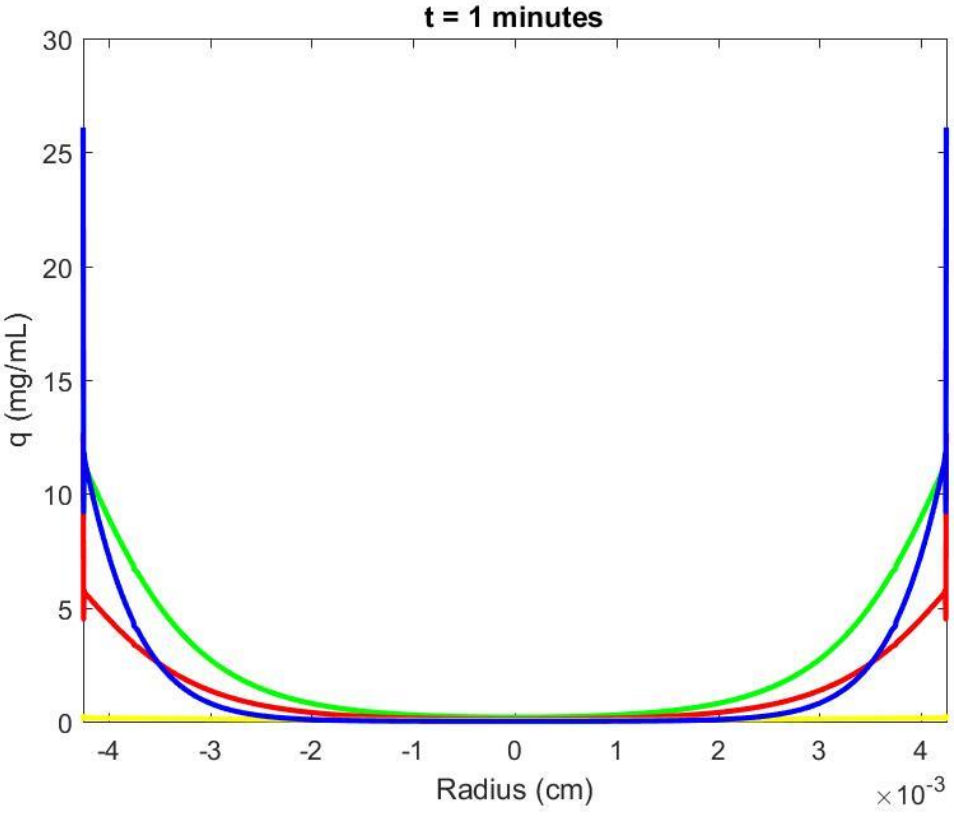
(D)


t = 60 minutes
Concentration (mg/mL)
Radius (cm)
×10^-3

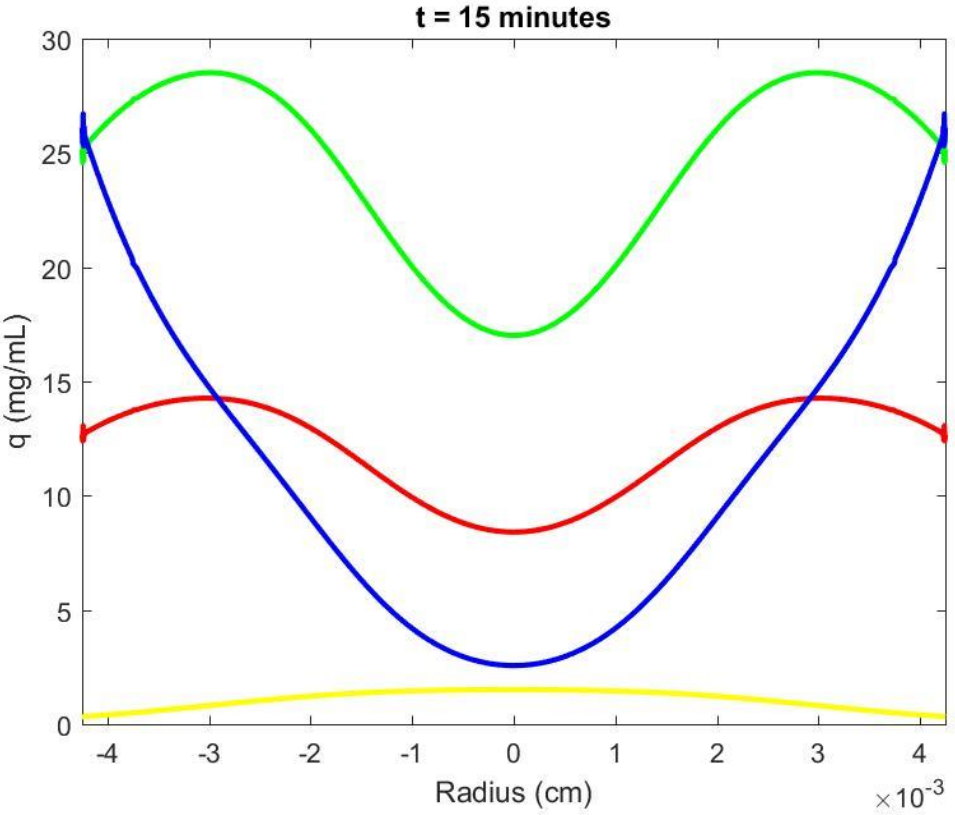


(E)

Figure 36: Kinetic case, concentration vs. radial position for four competitive components. $c_1 = 0$ *for all components. mAb component shown in green, weak binding hIgG component shown in yellow, moderate strength binding hIgG component shown in red, strong binding hIgG component shown in blue. Select times are (A) 1 minute (B) 15 minutes (C) 30 minutes (D) 60 minutes (E) 4 hours.* q_{max} *values and other key parameters can be found in Appendix II.*

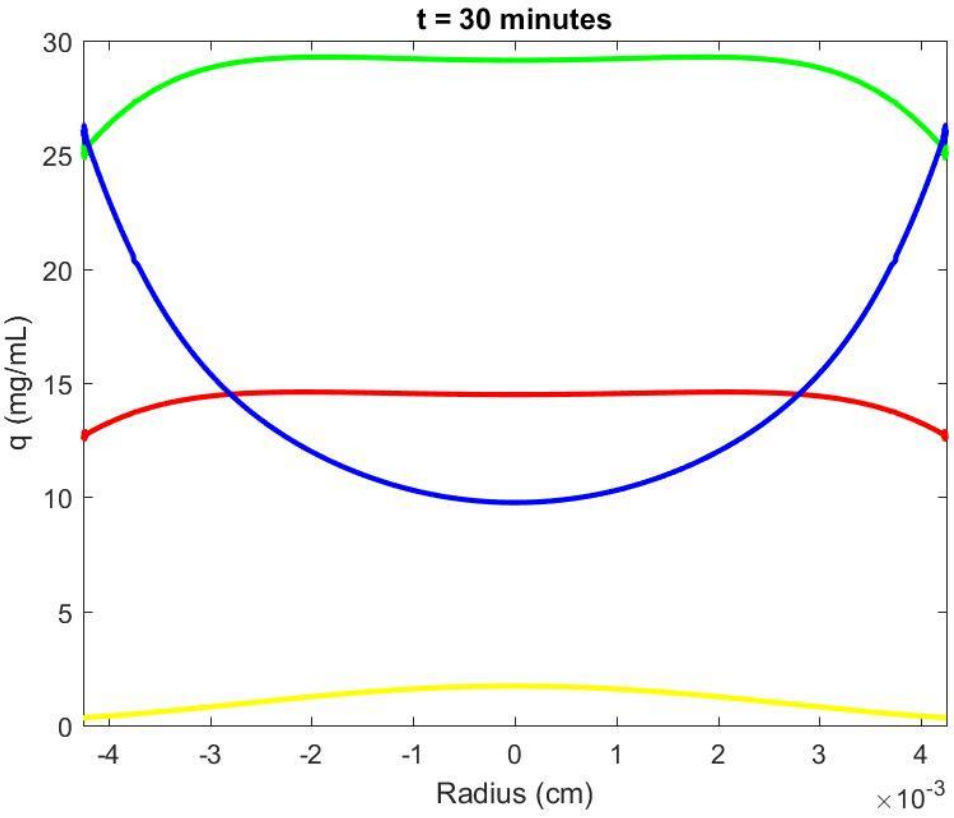

t = 1 minutes
q (mg/mL)
Radius (cm)
×10^-3

(A)

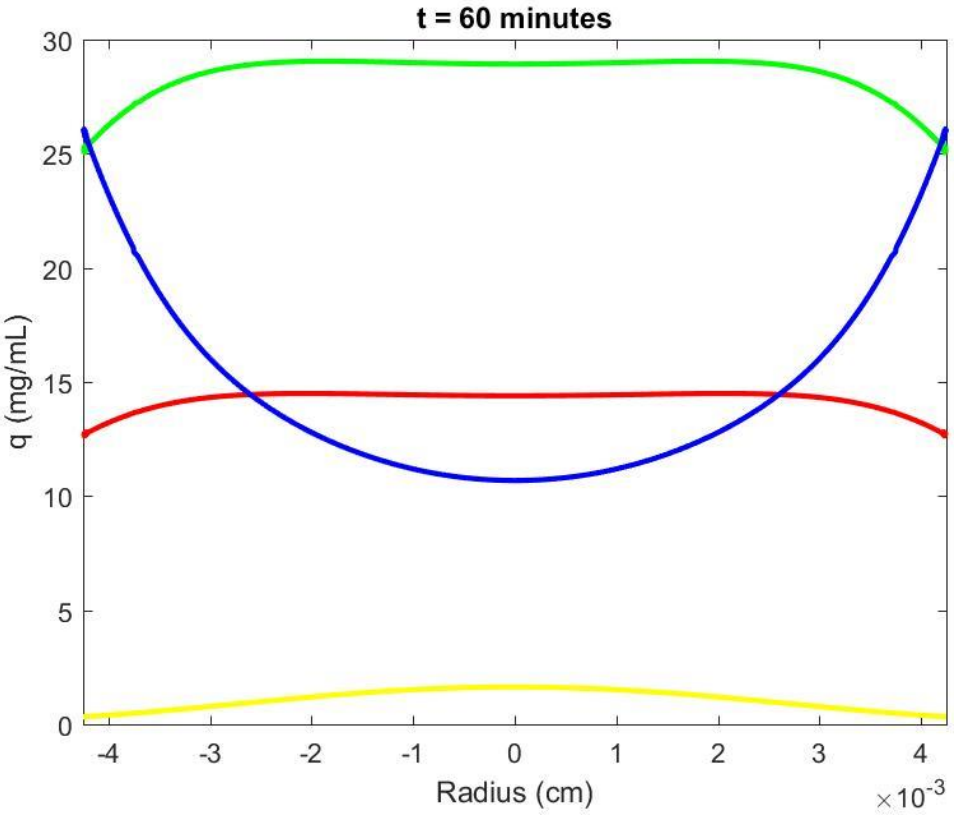

t = 15 minutes
q (mg/mL)
Radius (cm)
×10^-3

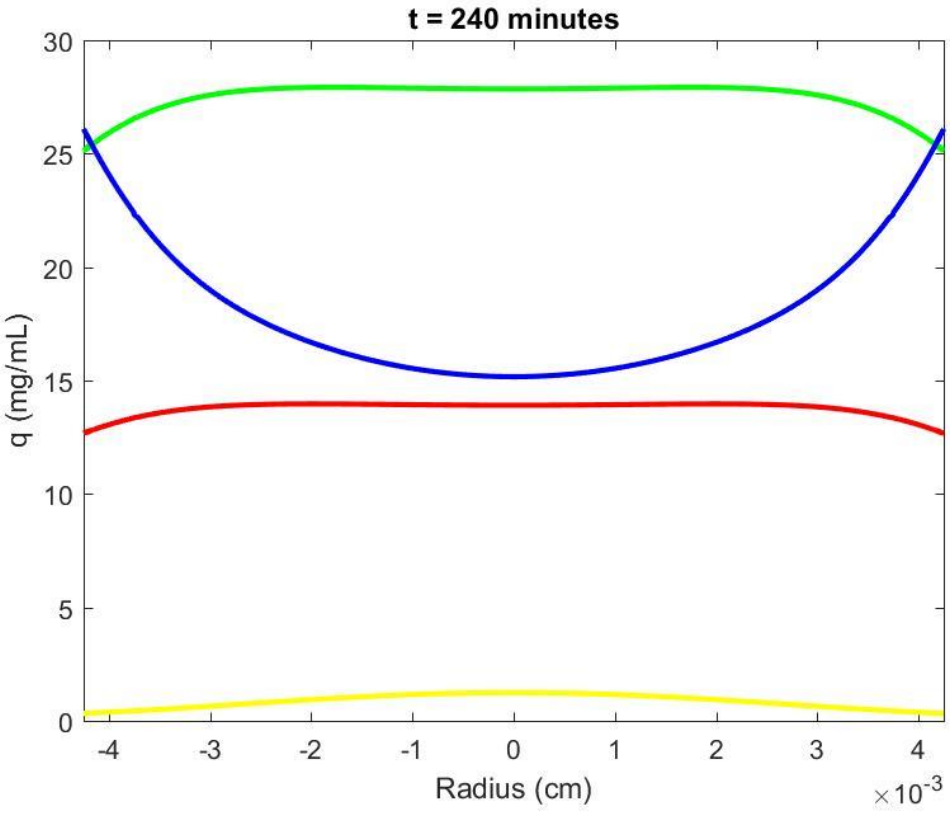
(B)

(C)


t = 30 minutes
q (mg/mL)
Radius (cm)
×10^-3

(D)


t = 60 minutes
q (mg/mL)
Radius (cm)
×10^-3



(E)

Figure 37: Kinetic case, q vs. radial position for four competitive components.
$c_1 = 0$ for all components. mAb component shown in green, weak binding hIgG component shown in yellow, moderate strength binding hIgG component shown in red, strong binding hIgG component shown in blue. Select times are (A) 1 minute (B) 15 minutes (C) 30 minutes (D) 60 minutes (E) 4 hours. q_{max} values and other key parameters can be found in Appendix II.

Appendix IV: Supplemental Data

Table 4: Elution data for mAb and hIgG on CaptivA PriMab chromatography from Weinberg et al., 2017

mAb		hIgG		mAb		hIgG	
Volume (mL)	UV280	Volume (mL)	UV280	Volume (mL)	pH	Volume (mL)	pH
-0.382000	-0.003369	0.000056	-0.382000	-0.695	6.921	-0.695	6.946
-0.262149	0.000448	-0.000158	-0.262160	-0.562	6.921	-0.562	6.946
-0.142298	-0.000018	-0.000724	-0.142320	-0.429	6.910	-0.429	6.935
-0.022447	0.000765	-0.001443	-0.022480	-0.295	6.898	-0.295	6.935
0.097403	0.000130	-0.001450	0.097360	-0.162	6.887	-0.162	6.935
0.217254	-0.000998	-0.001808	0.217200	-0.029	6.876	-0.029	6.935
0.337105	-0.000992	-0.001719	0.337040	0.104	6.853	0.104	6.946
0.456956	-0.000336	-0.002360	0.456879	0.238	6.819	0.238	6.946
0.576807	-0.000566	-0.001937	0.576719	0.371	6.785	0.371	6.946
0.696658	-0.000544	-0.002166	0.696559	0.504	6.740	0.504	6.946
0.816508	0.000136	-0.002379	0.816399	0.637	6.684	0.637	6.946
0.936359	-0.000275	-0.002316	0.936239	0.771	6.627	0.770	6.946
1.056210	0.000330	-0.002152	1.056079	0.904	6.571	0.904	6.946
1.176061	0.000402	-0.002120	1.175919	1.037	6.492	1.037	6.935
1.295912	0.000408	-0.002512	1.295759	1.170	6.424	1.170	6.935
1.415763	0.000387	-0.002232	1.415599	1.304	6.368	1.303	6.935
1.535613	0.000641	-0.003004	1.535439	1.437	6.311	1.437	6.935
1.655464	0.000986	-0.002337	1.655279	1.570	6.278	1.570	6.935
1.775315	0.001757	-0.002475	1.775119	1.703	6.255	1.703	6.946
1.895166	0.002640	-0.002591	1.894959	1.836	6.232	1.836	6.946
2.015017	0.002601	-0.003539	2.014798	1.970	6.232	1.969	6.946
2.134868	0.005186	-0.003468	2.134638	2.103	6.232	2.103	6.946
2.254718	0.006426	-0.003695	2.254478	2.236	6.266	2.236	6.946
2.374569	0.008241	-0.003485	2.374318	2.369	6.311	2.369	6.946
2.494420	0.010210	-0.003608	2.494158	2.503	6.368	2.502	6.946
2.614271	0.014262	-0.003590	2.613998	2.636	6.436	2.636	6.946
2.734122	0.017165	-0.004114	2.733838	2.769	6.492	2.769	6.946
2.853973	0.020827	-0.004087	2.853678	2.902	6.548	2.902	6.946
2.973823	0.023824	-0.004003	2.973518	3.036	6.605	3.035	6.946
3.093674	0.027378	-0.004111	3.093358	3.169	6.650	3.169	6.946
3.213525	0.031799	-0.004291	3.213198	3.302	6.695	3.302	6.946
3.333376	0.034400	-0.003686	3.333038	3.435	6.729	3.435	6.946
3.453227	0.036982	-0.004082	3.452878	3.569	6.763	3.568	6.946
3.573078	0.040766	-0.003999	3.572717	3.702	6.797	3.701	6.946

3.692928	0.043914	-0.004316	3.692557	3.835	6.819	3.835	6.946
3.812779	0.045078	-0.004059	3.812397	3.968	6.831	3.968	6.946
3.932630	0.047863	-0.004601	3.932237	4.102	6.853	4.101	6.946
4.052481	0.048447	-0.004332	4.052077	4.235	6.876	4.234	6.946
4.172332	0.048517	-0.004687	4.171917	4.368	6.887	4.368	6.946
4.292183	0.048713	-0.004642	4.291757	4.501	6.898	4.501	6.946
4.412033	0.048172	-0.004388	4.411597	4.634	6.898	4.634	6.946
4.531884	0.046853	-0.004471	4.531437	4.768	6.910	4.767	6.946
4.651735	0.045641	-0.004748	4.651277	4.901	6.921	4.900	6.946
4.771586	0.044576	-0.004579	4.771117	5.034	6.921	5.034	6.946
4.891437	0.042223	-0.004821	4.890957	5.167	6.921	5.167	6.946
5.011288	0.040563	-0.004689	5.010797	5.301	6.932	5.300	6.946
5.131138	0.037872	-0.004924	5.130636	5.434	6.932	5.433	6.946
5.250989	0.034977	-0.004794	5.250476	5.567	6.932	5.567	6.946
5.370840	0.033341	-0.005145	5.370316	5.700	6.932	5.700	6.946
5.490691	0.030356	-0.005264	5.490156	5.834	6.932	5.833	6.935
5.610542	0.027632	-0.004943	5.609996	5.967	6.932	5.966	6.935
5.730393	0.026150	-0.005256	5.729836	6.100	6.921	6.099	6.935
5.850244	0.023948	-0.004797	5.849676	6.233	6.921	6.233	6.924
5.970094	0.021254	-0.004741	5.969516	6.367	6.898	6.366	6.902
6.089945	0.018868	-0.005292	6.089356	6.500	6.864	6.499	6.891
6.209796	0.017843	-0.004835	6.209196	6.633	6.831	6.632	6.859
6.329647	0.016711	-0.004844	6.329036	6.766	6.785	6.766	6.826
6.449498	0.015381	-0.004603	6.448876	6.899	6.729	6.899	6.761
6.569349	0.012505	-0.004811	6.568715	7.033	6.639	7.032	6.696
6.689199	0.011837	-0.005090	6.688555	7.166	6.537	7.165	6.598
6.809050	0.011997	-0.005220	6.808395	7.299	6.413	7.298	6.479
6.928901	0.009995	-0.004476	6.928235	7.432	6.278	7.432	6.360
7.048752	0.009786	-0.004255	7.048075	7.566	6.165	7.565	6.241
7.168603	0.010243	-0.003957	7.167915	7.699	6.052	7.698	6.121
7.288454	0.009995	-0.004292	7.287755	7.832	5.950	7.831	6.024
7.408304	0.009323	-0.003521	7.407595	7.965	5.871	7.965	5.937
7.528155	0.009469	-0.003031	7.527435	8.099	5.804	8.098	5.872
7.648006	0.009193	-0.002724	7.647275	8.232	5.747	8.231	5.807
7.767857	0.009139	-0.002496	7.767115	8.365	5.691	8.364	5.764
7.887708	0.009671	-0.001732	7.886955	8.498	5.668	8.497	5.720
8.007559	0.009523	-0.001346	8.006795	8.632	5.634	8.631	5.688
8.127409	0.009345	-0.001262	8.126634	8.765	5.601	8.764	5.666
8.247260	0.010143	-0.000873	8.246474	8.898	5.589	8.897	5.633

8.367111	0.009913	-0.001185	8.366314	9.031	5.567	9.030	5.623
8.486962	0.010412	-0.000298	8.486154	9.164	5.544	9.164	5.601
8.606813	0.010606	0.000033	8.605994	9.298	5.544	9.297	5.590
8.726664	0.010394	0.000188	8.725834	9.431	5.544	9.430	5.579
8.846514	0.010627	0.000528	8.845674	9.564	5.533	9.563	5.568
8.966365	0.011289	0.000343	8.965514	9.697	5.533	9.696	5.568
9.086216	0.011141	0.000768	9.085354	9.831	5.533	9.830	5.568
9.206067	0.010963	0.000942	9.205194	9.964	5.533	9.963	5.568
9.325918	0.010896	0.000585	9.325034	10.097	5.533	10.096	5.568
9.445769	0.010624	0.001105	9.444874	10.230	5.522	10.229	5.568
9.565619	0.010318	0.000759	9.564714	10.364	5.522	10.363	5.568
9.685470	0.010639	0.001081	9.684553	10.497	5.522	10.496	5.557
9.805321	0.010760	0.001161	9.804393	10.630	5.522	10.629	5.568
9.925172	0.010902	0.000641	9.924233	10.763	5.533	10.762	5.568
10.045023	0.011806	0.001177	10.044073	10.897	5.533	10.896	5.557
10.164874	0.010923	0.000870	10.163913	11.030	5.533	11.029	5.557
10.284724	0.011041	0.000652	10.283753	11.163	5.533	11.162	5.557
10.404575	0.010618	0.001301	10.403593	11.296	5.533	11.295	5.557
10.524426	0.010370	0.001343	10.523433	11.430	5.533	11.428	5.557
10.644277	0.010684	0.001290	10.643273	11.563	5.533	11.562	5.568
10.764128	0.011053	0.001030	10.763113	11.696	5.533	11.695	5.568
10.883979	0.010470	0.000903	10.882953	11.829	5.533	11.828	5.568
11.003829	0.010736	0.000569	11.002793	11.962	5.533	11.961	5.568
11.123680	0.010164	0.000574	11.122633	12.096	5.533	12.095	5.568
11.243531	0.010179	0.000755	11.242472	12.229	5.533	12.228	5.568
11.363382	0.010031	0.000710	11.362312	12.362	5.533	12.361	5.568
11.483233	0.010312	0.000981	11.482152	12.495	5.533	12.494	5.568
11.603084	0.009862	0.000542	11.601992	12.629	5.533	12.627	5.568
11.722935	0.010578	0.000213	11.721832	12.762	5.544	12.761	5.568
11.842785	0.009898	0.000185	11.841672	12.895	5.544	12.894	5.568
11.962636	0.009956	0.000092	11.961512	13.028	5.544	13.027	5.568
12.082487	0.010119	0.000013	12.081352	13.162	5.544	13.160	5.568
12.202338	0.009402	0.000263	12.201192	13.295	5.544	13.294	5.568
12.322189	0.008876	-0.000077	12.321032	13.428	5.544	13.427	5.568
12.442040	0.009239	-0.000401	12.440872	13.561	5.544	13.560	5.568
12.561890	0.009018	-0.000205	12.560712	13.695	5.544	13.693	5.568
12.681741	0.009614	-0.000407	12.680552	13.828	5.544	13.826	5.568
12.801592	0.008386	-0.000726	12.800391	13.961	5.544	13.960	5.568
12.921443	0.008289	-0.000489	12.920231	14.094	5.544	14.093	5.568

13.041294	0.009012	-0.000572	13.040071	14.227	5.544	14.226	5.568
13.161145	0.008840	-0.000660	13.159911	14.361	5.544	14.359	5.568
13.280995	0.008613	-0.000948	13.279751	14.494	5.544	14.493	5.568
13.400846	0.009357	-0.000406	13.399591	14.627	5.555	14.626	5.568
13.520697	0.009381	-0.000816	13.519431	14.760	5.555	14.759	5.568
13.640548	0.007718	-0.000522	13.639271	14.894	5.555	14.892	5.568
13.760399	0.008867	-0.001100	13.759111	15.027	5.544	15.025	5.568
13.880250	0.008583	-0.001017	13.878951	15.160	5.555	15.159	5.568
14.000100	0.008344	-0.001116	13.998791	15.293	5.555	15.292	5.568
14.119951	0.008114	-0.000818	14.118631	15.427	5.544	15.425	5.568
14.239802	0.007636	-0.001368	14.238471	15.560	5.555	15.558	5.568
14.359653	0.009302	-0.001168	14.358310	15.693	5.555	15.692	5.568
14.479504	0.009689	-0.000939	14.478150	15.826	5.555	15.825	5.568
14.599355	0.007346	-0.000865	14.597990	15.960	5.555	15.958	5.568
14.719205	0.007687	-0.000958	14.717830	16.093	5.567	16.091	5.568
14.839056	0.007340	-0.001334	14.837670	16.226	5.567	16.224	5.568
14.958907	0.008828	-0.001030	14.957510	16.359	5.567	16.358	5.568
15.078758	0.008688	-0.001412	15.077350	16.492	5.567	16.491	5.568
15.198609	0.008265	-0.001222	15.197190	16.626	5.555	16.624	5.568
15.318460	0.007799	-0.001138	15.317030	16.759	5.555	16.757	5.568
15.438310	0.007615	-0.001374	15.436870	16.892	5.555	16.891	5.568
15.558161	0.007684	-0.001381	15.556710	17.025	5.555	17.024	5.568
15.678012	0.007524	-0.001033	15.676550	17.159	5.567	17.157	5.568
15.797863	0.007823	-0.001131	15.796390	17.292	5.567	17.290	5.568
15.917714	0.007917	-0.001011	15.916229	17.425	5.567	17.423	5.568
16.037565	0.008208	-0.001542	16.036069	17.558	5.567	17.557	5.568
16.157415	0.007621	-0.001693	16.155909	17.692	5.567	17.690	5.568
16.277266	0.007352	-0.001352	16.275749	17.825	5.567	17.823	5.568
16.397117	0.007031	-0.001254	16.395589	17.958	5.567	17.956	5.568
16.516968	0.007715	-0.001418	16.515429	18.091	5.567	18.090	5.568
16.636819	0.008214	-0.001805	16.635269	18.225	5.567	18.223	5.568
16.756670	0.007500	-0.000878	16.755109	18.358	5.567	18.356	5.568
16.876520	0.007261	-0.001829	16.874949	18.491	5.567	18.489	5.568
16.996371	0.007458	-0.001404	16.994789	18.624	5.567	18.623	5.568
17.116222	0.008770	-0.001599	17.114629	18.757	5.567	18.756	5.568
17.236073	0.007727	-0.001843	17.234469	18.891	5.567	18.889	5.568
17.355924	0.007358	-0.001399	17.354309	19.024	5.567	19.022	5.568
17.475775	0.007331	-0.001553	17.474148	19.157	5.567	19.155	5.568
17.595626	0.007312	-0.001475	17.593988	19.290	5.567	19.289	5.568

17.715476	0.007805	-0.001468	17.713828	19.424	5.567	19.422	5.568
17.835327	0.007149	-0.001323	17.833668	19.557	5.567	19.555	5.568
17.955178	0.007633	-0.001508	17.953508	19.690	5.567	19.688	5.568
18.075029	0.007013	-0.001653	18.073348	19.823	5.567	19.822	5.568
18.194880	0.007143	-0.001738	18.193188	19.957	5.567	19.955	5.568
18.314731	0.007566	-0.001653	18.313028	20.090	5.567	20.088	5.568
18.434581	0.007681	-0.001594	18.432868	20.223	5.567	20.221	5.568
18.554432	0.007506	-0.001827	18.552708	20.356	5.567	20.354	5.568
18.674283	0.007188	-0.002241	18.672548	20.490	5.567	20.488	5.568
18.794134	0.007334	-0.001791	18.792388	20.623	5.567	20.621	5.568
18.913985	0.007448	-0.001877	18.912228	20.756	5.578	20.754	5.568
19.033836	0.007808	-0.001818	19.032067	20.889	5.567	20.887	5.568
19.153686	0.007046	-0.001407	19.151907	21.023	5.567	21.021	5.568
19.273537	0.007802	-0.001696	19.271747	21.156	5.567	21.154	5.568
19.393388	0.006925	-0.001724	19.391587	21.289	5.567	21.287	5.568
19.513239	0.007343	-0.001849	19.511427	21.422	5.567	21.420	5.568
19.633090	0.007321	-0.001860	19.631267	21.555	5.567	21.553	5.568
19.752941	0.006711	-0.001561	19.751107	21.689	5.567	21.687	5.568
19.872791	0.006692	-0.001824	19.870947	21.822	5.567	21.820	5.568
19.992642	0.007258	-0.001738	19.990787	21.955	5.567	21.953	5.568
20.112493	0.007303	-0.001718	20.110627	22.088	5.567	22.086	5.568
20.232344	0.007243	-0.001976	20.230467	22.222	5.567	22.220	5.568
20.352195	0.007966	-0.002050	20.350307	22.355	5.567	22.353	5.568
20.472046	0.007343	-0.001812	20.470146	22.488	5.567	22.486	5.557
20.591896	0.007579	-0.001837	20.589986	22.621	5.567	22.619	5.557
20.711747	0.007681	-0.002039	20.709826	22.755	5.555	22.752	5.547
20.831598	0.007270	-0.001851	20.829666	22.888	5.544	22.886	5.547
20.951449	0.006753	-0.001497	20.949506	23.021	5.544	23.019	5.536
21.071300	0.007013	-0.001989	21.069346	23.154	5.544	23.152	5.536
21.191151	0.007210	-0.001998	21.189186	23.288	5.533	23.285	5.525
21.311001	0.006511	-0.001647	21.309026	23.421	5.533	23.419	5.525
21.430852	0.006484	-0.001705	21.428866	23.554	5.533	23.552	5.514
21.550703	0.006940	-0.002167	21.548706	23.687	5.522	23.685	5.514
21.670554	0.006974	-0.001352	21.668546	23.820	5.510	23.818	5.503
21.790405	0.007328	-0.001381	21.788386	23.954	5.510	23.951	5.503
21.910256	0.006859	-0.001595	21.908226	24.087	5.499	24.085	5.503
22.030106	0.007197	-0.001807	22.028065	24.220	5.499	24.218	5.492
22.149957	0.006635	-0.001700	22.147905	24.353	5.499	24.351	5.482
22.269808	0.006680	-0.001367	22.267745	24.487	5.476	24.484	5.471

22.389659	0.007070	-0.001465	22.387585	24.620	5.476	24.618	5.471
22.509510	0.007391	-0.001033	22.507425	24.753	5.465	24.751	5.471
22.629361	0.006832	-0.001318	22.627265	24.886	5.465	24.884	5.449
22.749211	0.007603	-0.001081	22.747105	25.020	5.454	25.017	5.449
22.869062	0.007282	-0.000923	22.866945	25.153	5.443	25.150	5.438
22.988913	0.006729	-0.000997	22.986785	25.286	5.443	25.284	5.438
23.108764	0.007461	-0.000768	23.106625	25.419	5.431	25.417	5.427
23.228615	0.007669	-0.000661	23.226465	25.553	5.431	25.550	5.416
23.348466	0.007007	-0.000654	23.346305	25.686	5.420	25.683	5.406
23.468317	0.007539	-0.000552	23.466145	25.819	5.409	25.817	5.406
23.588167	0.007325	-0.000539	23.585984	25.952	5.397	25.950	5.395
23.708018	0.006771	-0.000647	23.705824	26.085	5.397	26.083	5.384
23.827869	0.007049	-0.000042	23.825664	26.219	5.397	26.216	5.384
23.947720	0.007080	-0.000472	23.945504	26.352	5.386	26.350	5.373
24.067571	0.007518	-0.000246	24.065344	26.485	5.375	26.483	5.373
24.187422	0.006717	0.000176	24.185184	26.618	5.364	26.616	5.362
24.307272	0.007225	-0.000089	24.305024	26.752	5.352	26.749	5.351
24.427123	0.007216	0.000230	24.424864	26.885	5.341	26.882	5.341
24.546974	0.007639	0.000597	24.544704	27.018	5.341	27.016	5.330
24.666825	0.007095	0.000882	24.664544	27.151	5.341	27.149	5.319
24.786676	0.007893	0.001262	24.784384	27.285	5.330	27.282	5.319
24.906527	0.007820	0.001373	24.904224	27.418	5.330	27.415	5.308
25.026377	0.007672	0.001661	25.024064	27.551	5.307	27.549	5.308
25.146228	0.007585	0.001912	25.143903	27.684	5.307	27.682	5.308
25.266079	0.008150	0.002172	25.263743	27.818	5.296	27.815	5.286
25.385930	0.008561	0.002446	25.383583	27.951	5.285	27.948	5.276
25.505781	0.007524	0.002365	25.503423	28.084	5.273	28.081	5.276
25.625632	0.008537	0.002772	25.623263	28.217	5.273	28.215	5.265
25.745482	0.008927	0.003067	25.743103	28.351	5.262	28.348	5.254
25.865333	0.009426	0.003833	25.862943	28.484	5.251	28.481	5.254
25.985184	0.009381	0.003814	25.982783	28.617	5.239	28.614	5.243
26.105035	0.008976	0.003877	26.102623	28.750	5.239	28.748	5.232
26.224886	0.009030	0.004191	26.222463	28.883	5.239	28.881	5.221
26.344737	0.009714	0.004579	26.342303	29.017	5.228	29.014	5.221
26.464587	0.008843	0.004571	26.462143	29.150	5.217	29.147	5.210
26.584438	0.010309	0.005136	26.581983	29.283	5.206	29.280	5.200
26.704289	0.010240	0.005416	26.701822	29.416	5.194	29.414	5.189
26.824140	0.010194	0.005567	26.821662	29.550	5.194	29.547	5.189
26.943991	0.010010	0.005988	26.941502	29.683	5.183	29.680	5.178

27.063842	0.011135	0.006278	27.061342	29.816	5.172	29.813	5.167
27.183692	0.010536	0.006411	27.181182	29.949	5.172	29.947	5.156
27.303543	0.010766	0.006789	27.301022	30.083	5.160	30.080	5.156
27.423394	0.010917	0.006471	27.420862	30.216	5.160	30.213	5.145
27.543245	0.011328	0.007289	27.540702	30.349	5.138	30.346	5.135
27.663096	0.011589	0.006817	27.660542	30.482	5.138	30.479	5.135
27.782947	0.011897	0.007011	27.780382	30.616	5.127	30.613	5.124
27.902797	0.011449	0.007359	27.900222	30.749	5.115	30.746	5.113
28.022648	0.011558	0.006979	28.020062	30.882	5.115	30.879	5.113
28.142499	0.011970	0.007312	28.139902	31.015	5.115	31.012	5.091
28.262350	0.011456	0.007284	28.259741	31.148	5.104	31.146	5.091
28.382201	0.012069	0.007073	28.379581	31.282	5.104	31.279	5.091
28.502052	0.012015	0.007004	28.499421	31.415	5.081	31.412	5.080
28.621902	0.011846	0.006570	28.619261	31.548	5.081	31.545	5.069
28.741753	0.011685	0.006507	28.739101	31.681	5.070	31.678	5.059
28.861604	0.012009	0.006488	28.858941	31.815	5.059	31.812	5.059
28.981455	0.011664	0.006477	28.978781	31.948	5.048	31.945	5.048
29.101306	0.012429	0.005838	29.098621	32.081	5.048	32.078	5.037
29.221157	0.011419	0.005543	29.218461	32.214	5.036	32.211	5.037
29.341008	0.012299	0.005170	29.338301	32.348	5.036	32.345	5.026
29.460858	0.011870	0.005134	29.458141	32.481	5.025	32.478	5.015
29.580709	0.011498	0.004980	29.577981	32.614	5.014	32.611	5.004
29.700560	0.011685	0.004455	29.697821	32.747	5.014	32.744	5.004
29.820411	0.011371	0.004645	29.817660	32.881	5.003	32.877	4.994
29.940262	0.010987	0.003885	29.937500	33.014	4.991	33.011	4.994
30.060113	0.011592	0.003539	30.057340	33.147	4.980	33.144	4.972
30.179963	0.010854	0.003100	30.177180	33.280	4.980	33.277	4.972
30.299814	0.011510	0.003369	30.297020	33.413	4.969	33.410	4.961
30.419665	0.010464	0.002638	30.416860	33.547	4.969	33.544	4.950
30.539516	0.011165	0.002641	30.536700	33.680	4.957	33.677	4.939
30.659367	0.010820	0.002749	30.656540	33.813	4.946	33.810	4.939
30.779218	0.011117	0.002710	30.776380	33.946	4.935	33.943	4.929
30.899068	0.010975	0.002070	30.896220	34.080	4.935	34.077	4.929
31.018919	0.010627	0.001946	31.016060	34.213	4.924	34.210	4.918
31.138770	0.009965	0.002133	31.135900	34.346	4.924	34.343	4.907
31.258621	0.010282	0.001667	31.255740	34.479	4.912	34.476	4.907
31.378472	0.010624	0.001376	31.375579	34.613	4.901	34.609	4.896
31.498323	0.010808	0.001572	31.495419	34.746	4.901	34.743	4.885
31.618173	0.010630	0.001533	31.615259	34.879	4.890	34.876	4.874

31.738024	0.011313	0.001282	31.735099	35.012	4.878	35.009	4.863
31.857875	0.010854	0.000815	31.854939	35.146	4.867	35.142	4.863
31.977726	0.010787	0.000881	31.974779	35.279	4.867	35.276	4.853
32.097577	0.010297	0.001031	32.094619	35.412	4.856	35.409	4.842
32.217428	0.010563	0.001052	32.214459	35.545	4.856	35.542	4.831
32.337278	0.010494	0.001113	32.334299	35.679	4.845	35.675	4.831
32.457129	0.010950	0.001047	32.454139	35.812	4.833	35.808	4.820
32.576980	0.010866	0.000650	32.573979	35.945	4.833	35.942	4.820
32.696831	0.010125	0.001429	32.693819	36.078	4.822	36.075	4.809
32.816682	0.010760	0.000658	32.813659	36.211	4.811	36.208	4.798
32.936533	0.010663	0.000619	32.933498	36.345	4.799	36.341	4.798
33.056383	0.010545	0.000522	33.053338	36.478	4.799	36.475	4.788
33.176234	0.010736	0.000884	33.173178	36.611	4.799	36.608	4.777
33.296085	0.010282	0.000643	33.293018	36.744	4.777	36.741	4.766
33.415936	0.010941	0.000418	33.412858	36.878	4.777	36.874	4.766
33.535787	0.010364	0.001348	33.532698	37.011	4.777	37.007	4.755
33.655638	0.010306	0.000610	33.652538	37.144	4.766	37.141	4.744
33.775488	0.009943	0.000208	33.772378	37.277	4.754	37.274	4.744
33.895339	0.010451	0.000611	33.892218	37.411	4.743	37.407	4.733
34.015190	0.010938	0.000780	34.012058	37.544	4.743	37.540	4.733
34.135041	0.010197	0.000487	34.131898	37.677	4.732	37.674	4.722
34.254892	0.009865	0.000371	34.251738	37.810	4.732	37.807	4.712
34.374743	0.010400	0.000813	34.371577	37.944	4.720	37.940	4.701
34.494593	0.010473	-0.000241	34.491417	38.077	4.709	38.073	4.701
34.614444	0.011156	0.000291	34.611257	38.210	4.698	38.206	4.690
34.734295	0.010509	0.000896	34.731097	38.343	4.698	38.340	4.679
34.854146	0.010960	0.000578	34.850937	38.476	4.687	38.473	4.679
34.973997	0.010527	0.000442	34.970777	38.610	4.675	38.606	4.668
35.093848	0.011322	0.000531	35.090617	38.743	4.675	38.739	4.657
35.213699	0.011144	0.000480	35.210457	38.876	4.675	38.873	4.647
35.333549	0.010872	0.000287	35.330297	39.009	4.653	39.006	4.647
35.453400	0.010684	0.000581	35.450137	39.143	4.653	39.139	4.636
35.573251	0.011071	0.000672	35.569977	39.276	4.641	39.272	4.636
35.693102	0.010893	0.000459	35.689817	39.409	4.641	39.405	4.625
35.812953	0.011519	0.000854	35.809657	39.542	4.630	39.539	4.614
35.932804	0.010455	0.000549	35.929496	39.676	4.619	39.672	4.603
36.052654	0.011319	0.000459	36.049336	39.809	4.619	39.805	4.592
36.172505	0.010539	0.000912	36.169176	39.942	4.608	39.938	4.592
36.292356	0.011870	0.000868	36.289016	40.075	4.608	40.072	4.581

36.412207	0.010866	0.000317	36.408856	40.209	4.596	40.205	4.581
36.532058	0.011020	-0.000008	36.528696	40.342	4.585	40.338	4.571
36.651909	0.011764	0.000596	36.648536	40.475	4.574	40.471	4.560
36.771759	0.010687	0.000204	36.768376	40.608	4.574	40.604	4.549
36.891610	0.011691	0.000125	36.888216	40.741	4.574	40.738	4.549
37.011461	0.011474	0.000583	37.008056	40.875	4.562	40.871	4.538
37.131312	0.011123	0.000404	37.127896	41.008	4.551	41.004	4.527
37.251163	0.011153	0.000583	37.247736	41.141	4.540	41.137	4.527
37.371014	0.012094	0.000520	37.367576	41.274	4.540	41.271	4.516
37.490864	0.010923	0.000542	37.487415	41.408	4.540	41.404	4.516
37.610715	0.011930	0.000780	37.607255	41.541	4.517	41.537	4.506
37.730566	0.011794	0.000373	37.727095	41.674	4.517	41.670	4.495
37.850417	0.011150	0.000491	37.846935	41.807	4.506	41.804	4.484
37.970268	0.011165	0.000619	37.966775	41.941	4.506	41.937	4.484
38.090119	0.011852	0.000744	38.086615	42.074	4.483	42.070	4.473
38.209969	0.011407	0.000806	38.206455	42.207	4.483	42.203	4.462
38.329820	0.011350	0.000796	38.326295	42.340	4.472	42.336	4.462
38.449671	0.011462	0.001022	38.446135	42.474	4.472	42.470	4.451
38.569522	0.011933	0.000464	38.565975	42.607	4.461	42.603	4.451
38.689373	0.012100	0.000445	38.685815	42.740	4.461	42.736	4.430
38.809224	0.011707	0.000812	38.805655	42.873	4.450	42.869	4.430
38.929074	0.011743	0.000947	38.925495	43.006	4.450	43.003	4.419
39.048925	0.011906	0.001174	39.045334	43.140	4.438	43.136	4.419
39.168776	0.012196	0.000284	39.165174	43.273	4.438	43.269	4.419
39.288627	0.012205	0.000392	39.285014	43.406	4.427	43.402	4.397
39.408478	0.011410	0.000849	39.404854	43.539	4.416	43.535	4.397
39.528329	0.012202	0.000932	39.524694	43.673	4.404	43.669	4.386
39.648179	0.011513	0.001138	39.644534	43.806	4.404	43.802	4.386
39.768030	0.011486	0.001240	39.764374	43.939	4.393	43.935	4.375
39.887881	0.011338	0.000947	39.884214	44.072	4.382	44.068	4.365
40.007732	0.012478	0.000898	40.004054	44.206	4.382	44.202	4.365
40.127583	0.013270	0.001091	40.123894	44.339	4.382	44.335	4.354
40.247434	0.012596	0.000760	40.243734	44.472	4.371	44.468	4.343
40.367285	0.012523	0.001247	40.363574	44.605	4.359	44.601	4.343
40.487135	0.012078	0.001025	40.483414	44.739	4.348	44.734	4.332
40.606986	0.013536	0.001103	40.603253	44.872	4.348	44.868	4.321
40.726837	0.013482	0.001083	40.723093	45.005	4.337	45.001	4.321
40.846688	0.012363	0.000867	40.842933	45.138	4.325	45.134	4.310
40.966539	0.012202	0.001177	40.962773	45.272	4.325	45.267	4.300

41.086390	0.012626	0.001216	41.082613	45.405	4.314	45.401	4.300
41.206240	0.012296	0.001376	41.202453	45.538	4.314	45.534	4.289
41.326091	0.014298	0.000926	41.322293	45.671	4.303	45.667	4.278
41.445942	0.012813	0.001393	41.442133	45.804	4.303	45.800	4.267
41.565793	0.012714	0.001230	41.561973	45.938	4.292	45.933	4.267
41.685644	0.013058	0.001183	41.681813	46.071	4.280	46.067	4.256
41.805495	0.013947	0.001285	41.801653	46.204	4.280	46.200	4.256
41.925345	0.013781	0.001589	41.921493	46.337	4.269	46.333	4.245
42.045196	0.013987	0.001421	42.041333	46.471	4.258	46.466	4.245
42.165047	0.012668	0.001901	42.161172	46.604	4.246	46.600	4.234
42.284898	0.013669	0.001868	42.281012	46.737	4.246	46.733	4.224
42.404749	0.013560	0.001783	42.400852	46.870	4.235	46.866	4.213
42.524600	0.013969	0.001716	42.520692	47.004	4.235	46.999	4.213
42.644450	0.014425	0.001464	42.640532	47.137	4.224	47.132	4.202
42.764301	0.013808	0.001680	42.760372	47.270	4.224	47.266	4.191
42.884152	0.014522	0.001685	42.880212	47.403	4.213	47.399	4.191
43.004003	0.015242	0.001783	43.000052	47.537	4.201	47.532	4.180
43.123854	0.014954	0.002047	43.119892	47.670	4.190	47.665	4.169
43.243705	0.014622	0.001547	43.239732	47.803	4.179	47.799	4.169
43.363555	0.014897	0.001586	43.359572	47.936	4.179	47.932	4.159
43.483406	0.014794	0.001646	43.479412	48.069	4.179	48.065	4.148
43.603257	0.015490	0.001603	43.599252	48.203	4.167	48.198	4.137
43.723108	0.015674	0.002138	43.719091	48.336	4.156	48.331	4.137
43.842959	0.016040	0.002174	43.838931	48.469	4.156	48.465	4.126
43.962810	0.017501	0.002175	43.958771	48.602	4.145	48.598	4.115
44.082660	0.016340	0.002326	44.078611	48.736	4.145	48.731	4.115
44.202511	0.016303	0.001849	44.198451	48.869	4.134	48.864	4.104
44.322362	0.017619	0.002230	44.318291	49.002	4.122	48.998	4.104
44.442213	0.018094	0.002492	44.438131	49.135	4.122	49.131	4.094
44.562064	0.018813	0.001943	44.557971	49.269	4.111	49.264	4.083
44.681915	0.018314	0.002142	44.677811	49.402	4.100	49.397	4.072
44.801765	0.019324	0.002197	44.797651	49.535	4.088	49.531	4.072
44.921616	0.019609	0.002119	44.917491	49.668	4.077	49.664	4.061
45.041467	0.018877	0.002736	45.037331	49.802	4.077	49.797	4.050
45.161318	0.021278	0.002476	45.157171	49.935	4.066	49.930	4.039
45.281169	0.021172	0.002764	45.277010	50.068	4.066	50.063	4.039
45.401020	0.022246	0.002612	45.396850	50.201	4.055	50.197	4.028
45.520870	0.022606	0.002808	45.516690	50.334	4.043	50.330	4.018
45.640721	0.023440	0.002752	45.636530	50.468	4.043	50.463	4.007

45.760572	0.024798	0.002962	45.756370	50.601	4.032	50.596	4.007
45.880423	0.025500	0.003150	45.876210	50.734	4.021	50.730	3.996
46.000274	0.026643	0.002943	45.996050	50.867	4.021	50.863	3.985
46.120125	0.026504	0.003327	46.115890	51.001	4.010	50.996	3.985
46.239976	0.028902	0.003029	46.235730	51.134	3.998	51.129	3.985
46.359826	0.029461	0.003459	46.355570	51.267	3.987	51.262	3.974
46.479677	0.030042	0.003463	46.475410	51.400	3.987	51.396	3.963
46.599528	0.031754	0.004217	46.595250	51.534	3.976	51.529	3.953
46.719379	0.034203	0.004071	46.715089	51.667	3.976	51.662	3.942
46.839230	0.035416	0.003858	46.834929	51.800	3.964	51.795	3.942
46.959081	0.037001	0.004233	46.954769	51.933	3.953	51.929	3.920
47.078931	0.039260	0.004379	47.074609	52.067	3.953	52.062	3.920
47.198782	0.040587	0.004333	47.194449	52.200	3.942	52.195	3.920
47.318633	0.043675	0.005117	47.314289	52.333	3.942	52.328	3.909
47.438484	0.045913	0.004797	47.434129	52.466	3.931	52.461	3.898
47.558335	0.048078	0.005449	47.553969	52.600	3.919	52.595	3.887
47.678186	0.050682	0.005277	47.673809	52.733	3.908	52.728	3.877
47.798036	0.053186	0.005819	47.793649	52.866	3.897	52.861	3.877
47.917887	0.057066	0.005998	47.913489	52.999	3.897	52.994	3.866
48.037738	0.060414	0.006176	48.033329	53.132	3.885	53.128	3.866
48.157589	0.063770	0.006582	48.153169	53.266	3.885	53.261	3.855
48.277440	0.067345	0.006811	48.273008	53.399	3.874	53.394	3.855
48.397291	0.071337	0.007174	48.392848	53.532	3.863	53.527	3.833
48.517141	0.076166	0.007925	48.512688	53.665	3.852	53.660	3.822
48.636992	0.079989	0.008251	48.632528	53.799	3.840	53.794	3.822
48.756843	0.085073	0.008329	48.752368	53.932	3.840	53.927	3.812
48.876694	0.090171	0.009144	48.872208	54.065	3.829	54.060	3.801
48.996545	0.095963	0.009699	48.992048	54.198	3.818	54.193	3.790
49.116396	0.101176	0.010082	49.111888	54.332	3.818	54.327	3.790
49.236246	0.107007	0.010685	49.231728	54.465	3.806	54.460	3.779
49.356097	0.113140	0.011442	49.351568	54.598	3.795	54.593	3.768
49.475948	0.118901	0.011914	49.471408	54.731	3.784	54.726	3.757
49.595799	0.125034	0.012873	49.591248	54.865	3.784	54.859	3.757
49.715650	0.132298	0.013587	49.711088	54.998	3.773	54.993	3.747
49.835501	0.140082	0.014811	49.830927	55.131	3.773	55.126	3.747
49.955351	0.146554	0.016051	49.950767	55.264	3.750	55.259	3.736
50.075202	0.153307	0.017008	50.070607	55.397	3.750	55.392	3.725
50.195053	0.161000	0.018510	50.190447	55.531	3.739	55.526	3.714
50.314904	0.167931	0.019326	50.310287	55.664	3.727	55.659	3.703

50.434755	0.175365	0.021525	50.430127	55.797	3.716	55.792	3.703
50.554606	0.182514	0.023152	50.549967	55.930	3.716	55.925	3.692
50.674456	0.189560	0.025143	50.669807	56.064	3.705	56.058	3.681
50.794307	0.196888	0.027264	50.789647	56.197	3.694	56.192	3.671
50.914158	0.203985	0.030279	50.909487	56.330	3.694	56.325	3.671
51.034009	0.210617	0.033138	51.029327	56.463	3.682	56.458	3.660
51.153860	0.217609	0.036233	51.149167	56.597	3.682	56.591	3.649
51.273711	0.224042	0.040594	51.269007	56.730	3.660	56.725	3.638
51.393561	0.229806	0.044329	51.388846	56.863	3.648	56.858	3.627
51.513412	0.236395	0.049295	51.508686	56.996	3.648	56.991	3.627
51.633263	0.242958	0.055123	51.628526	57.130	3.637	57.124	3.616
51.753114	0.247941	0.061376	51.748366	57.263	3.626	57.258	3.606
51.872965	0.254522	0.068609	51.868206	57.396	3.615	57.391	3.595
51.992816	0.259696	0.076233	51.988046	57.529	3.603	57.524	3.595
52.112667	0.265297	0.085361	52.107886	57.662	3.592	57.657	3.573
52.232517	0.270166	0.095752	52.227726	57.796	3.592	57.790	3.573
52.352368	0.276523	0.107134	52.347566	57.929	3.581	57.924	3.562
52.472219	0.282583	0.118867	52.467406	58.062	3.569	58.057	3.551
52.592070	0.288592	0.132867	52.587246	58.195	3.569	58.190	3.540
52.711921	0.294907	0.148139	52.707086	58.329	3.558	58.323	3.540
52.831772	0.302724	0.165233	52.826926	58.462	3.547	58.457	3.530
52.951622	0.310399	0.183898	52.946765	58.595	3.547	58.590	3.508
53.071473	0.318574	0.204663	53.066605	58.728	3.524	58.723	3.508
53.191324	0.326941	0.227250	53.186445	58.862	3.524	58.856	3.508
53.311175	0.338037	0.250780	53.306285	58.995	3.513	58.989	3.497
53.431026	0.347875	0.277747	53.426125	59.128	3.513	59.123	3.486
53.550877	0.361263	0.306433	53.545965	59.261	3.490	59.256	3.475
53.670727	0.374315	0.338572	53.665805	59.395	3.490	59.389	3.465
53.790578	0.389541	0.372075	53.785645	59.528	3.479	59.522	3.465
53.910429	0.405730	0.407718	53.905485	59.661	3.468	59.656	3.443
54.030280	0.422853	0.445008	54.025325	59.794	3.457	59.789	3.443
54.150131	0.441312	0.485449	54.145165	59.928	3.457	59.922	3.432
54.269982	0.461532	0.527391	54.265005	60.061	3.445	60.055	3.421
54.389832	0.481920	0.569670	54.384845	60.194	3.445	60.188	3.410
54.509683	0.503268	0.613450	54.504684	60.327	3.434	60.322	3.410
54.629534	0.525057	0.659223	54.624524	60.460	3.434	60.455	3.399
54.749385	0.550067	0.702063	54.744364	60.594	3.423	60.588	3.389
54.869236	0.574723	0.744555	54.864204	60.727	3.400	60.721	3.378
54.989087	0.596666	0.785723	54.984044	60.860	3.400	60.855	3.378

55.108937	0.621349	0.825867	55.103884	60.993	3.389	60.988	3.367
55.228788	0.643927	0.863314	55.223724	61.127	3.389	61.121	3.356
55.348639	0.665686	0.896303	55.343564	61.260	3.378	61.254	3.356
55.468490	0.687046	0.925982	55.463404	61.393	3.366	61.387	3.345
55.588341	0.706428	0.952402	55.583244	61.526	3.355	61.521	3.334
55.708192	0.724978	0.971446	55.703084	61.660	3.344	61.654	3.324
55.828042	0.740449	0.985961	55.822924	61.793	3.344	61.787	3.313
55.947893	0.754182	0.997022	55.942764	61.926	3.321	61.920	3.302
56.067744	0.764274	1.000000	56.062603	62.059	3.321	62.054	3.302
56.187595	0.771928	0.996975	56.182443	62.193	3.310	62.187	3.291
56.307446	0.776978	0.992084	56.302283	62.326	3.299	62.320	3.280
56.427297	0.777495	0.977141	56.422123	62.459	3.299	62.453	3.280
56.547147	0.776673	0.958594	56.541963	62.592	3.287	62.586	3.269
56.666998	0.771979	0.934508	56.661803	62.725	3.276	62.720	3.259
56.786849	0.766932	0.905950	56.781643	62.859	3.265	62.853	3.248
56.906700	0.754503	0.873291	56.901483	62.992	3.253	62.986	3.237
57.026551	0.743089	0.839380	57.021323	63.125	3.253	63.119	3.226
57.146402	0.726950	0.801125	57.141163	63.258	3.242	63.253	3.226
57.266252	0.709310	0.760581	57.261003	63.392	3.231	63.386	3.215
57.386103	0.691555	0.719841	57.380843	63.525	3.231	63.519	3.215
57.505954	0.669215	0.675034	57.500683	63.658	3.220	63.652	3.204
57.625805	0.645319	0.631146	57.620522	63.791	3.208	63.785	3.193
57.745656	0.622725	0.586830	57.740362	63.925	3.197	63.919	3.183
57.865507	0.596249	0.543008	57.860202	64.058	3.197	64.052	3.183
57.985358	0.570298	0.501307	57.980042	64.191	3.186	64.185	3.161
58.105208	0.546096	0.460168	58.099882	64.324	3.186	64.318	3.161
58.225059	0.518960	0.419983	58.219722	64.458	3.163	64.452	3.150
58.344910	0.493742	0.382817	58.339562	64.591	3.163	64.585	3.150
58.464761	0.468284	0.345546	58.459402	64.724	3.152	64.718	3.139
58.584612	0.443211	0.312526	58.579242	64.857	3.141	64.851	3.128
58.704463	0.418728	0.281056	58.699082	64.990	3.129	64.985	3.118
58.824313	0.395079	0.251460	58.818922	65.124	3.129	65.118	3.118
58.944164	0.372464	0.224876	58.938762	65.257	3.118	65.251	3.107
59.064015	0.351117	0.200255	59.058602	65.390	3.118	65.384	3.096
59.183866	0.330489	0.178212	59.178441	65.523	3.107	65.517	3.085
59.303717	0.310115	0.157868	59.298281	65.657	3.095	65.651	3.085
59.423568	0.292687	0.139380	59.418121	65.790	3.084	65.784	3.074
59.543418	0.276214	0.122887	59.537961	65.923	3.084	65.917	3.063
59.663269	0.260171	0.108203	59.657801	66.056	3.073	66.050	3.052

59.783120	0.245800	0.095098	59.777641	66.190	3.062	66.184	3.052
59.902971	0.232760	0.083965	59.897481	66.323	3.062	66.317	3.042
60.022822	0.219777	0.073910	60.017321	66.456	3.050	66.450	3.031
60.142673	0.209895	0.064788	60.137161	66.589	3.039	66.583	3.020
60.262523	0.199367	0.056797	60.257001	66.723	3.028	66.716	3.020
60.382374	0.189575	0.049937	60.376841	66.856	3.028	66.850	3.009
60.502225	0.180772	0.043910	60.496681	66.989	3.016	66.983	2.998
60.622076	0.172921	0.039157	60.616520	67.122	3.005	67.116	2.998
60.741927	0.166655	0.034604	60.736360	67.256	2.994	67.249	2.987
60.861778	0.159957	0.030207	60.856200	67.389	2.994	67.383	2.987
60.981628	0.154577	0.027375	60.976040	67.522	2.983	67.516	2.977
61.101479	0.148928	0.024282	61.095880	67.655	2.983	67.649	2.966
61.221330	0.144911	0.022218	61.215720	67.788	2.971	67.782	2.955
61.341181	0.140067	0.019726	61.335560	67.922	2.960	67.915	2.944
61.461032	0.136607	0.017944	61.455400	68.055	2.949	68.049	2.944
61.580883	0.133943	0.016513	61.575240	68.188	2.949	68.182	2.933
61.700733	0.129286	0.014741	61.695080	68.321	2.949	68.315	2.933
61.820584	0.127689	0.014125	61.814920	68.455	2.926	68.448	2.922
61.940435	0.123649	0.013062	61.934760	68.588	2.926	68.582	2.912
62.060286	0.122115	0.011953	62.054600	68.721	2.926	68.715	2.901
62.180137	0.119953	0.011346	62.174439	68.854	2.915	68.848	2.901
62.299988	0.118015	0.010688	62.294279	68.988	2.892	68.981	2.890
62.419838	0.115995	0.010196	62.414119	69.121	2.892	69.114	2.890
62.539689	0.113878	0.010002	62.533959	69.254	2.892	69.248	2.868
62.659540	0.111900	0.009497	62.653799	69.387	2.881	69.381	2.868
62.779391	0.110445	0.009409	62.773639	69.521	2.870	69.514	2.857
62.899242	0.109166	0.008792	62.893479	69.654	2.859	69.647	2.846
63.019093	0.106998	0.008392	63.013319	69.787	2.859	69.781	2.846
63.138943	0.104687	0.007955	63.133159	69.920	2.847	69.914	2.836
63.258794	0.103363	0.008318	63.252999	70.053	2.847	70.047	2.836
63.378645	0.102138	0.007699	63.372839	70.187	2.836	70.180	2.825
63.498496	0.101143	0.007938	63.492679	70.320	2.825	70.313	2.814
63.618347	0.098270	0.007538	63.612519	70.453	2.813	70.447	2.814
63.738198	0.096377	0.007718	63.732358	70.586	2.802	70.580	2.803
63.858049	0.096455	0.007851	63.852198	70.720	2.802	70.713	2.792
63.977899	0.094003	0.007516	63.972038	70.853	2.791	70.846	2.781
64.097750	0.092216	0.007786	64.091878	70.986	2.791	70.980	2.771
64.217601	0.090177	0.007323	64.211718	71.119	2.791	71.113	2.771
64.337452	0.089352	0.007268	64.331558	71.253	2.768	71.246	2.760

64.457303	0.087078	0.007142	64.451398	71.386	2.768	71.379	2.749
64.577154	0.085305	0.006831	64.571238	71.519	2.757	71.512	2.749
64.697004	0.084053	0.007054	64.691078	71.652	2.746	71.646	2.738
64.816855	0.082227	0.006914	64.810918	71.786	2.734	71.779	2.738
64.936706	0.079838	0.006681	64.930758	71.919	2.734	71.912	2.716
65.056557	0.078193	0.006562	65.050598	72.052	2.723	72.045	2.716
65.176408	0.076469	0.006527	65.170438	72.185	2.712	72.179	2.716
65.296259	0.074458	0.006769	65.290277	72.318	2.701	72.312	2.705
65.416109	0.073197	0.006313	65.410117	72.452	2.701	72.445	2.695
65.535960	0.071110	0.007071	65.529957	72.585	2.701	72.578	2.684
65.655811	0.069250	0.005991	65.649797	72.718	2.689	72.712	2.684
65.775662	0.067003	0.006496	65.769637	72.851	2.678	72.845	2.673
65.895513	0.065791	0.006678	65.889477	72.985	2.667	72.978	2.673
66.015364	0.064315	0.006874	66.009317	73.118	2.667	73.111	2.662
66.135214	0.062645	0.005838	66.129157	73.251	2.655	73.244	2.651
66.255065	0.060347	0.006195	66.248997	73.384	2.644	73.378	2.640
66.374916	0.059010	0.006300	66.368837	73.518	2.633	73.511	2.640
66.494767	0.057783	0.006403	66.488677	73.651	2.633	73.644	2.630
66.614618	0.056065	0.006289	66.608517	73.784	2.633	73.777	2.619
66.734469	0.054692	0.006364	66.728357	73.917	2.622	73.911	2.608
66.854319	0.053246	0.006131	66.848196	74.051	2.622	74.044	2.608
66.974170	0.051048	0.005844	66.968036	74.184	2.610	74.177	2.608
67.094021	0.049161	0.006444	67.087876	74.317	2.599	74.310	2.586
67.213872	0.049354	0.006059	67.207716	74.450	2.599	74.443	2.586
67.333723	0.047745	0.005954	67.327556	74.583	2.588	74.577	2.586
67.453574	0.045284	0.006109	67.447396	74.717	2.588	74.710	2.586
67.573424	0.044543	0.006115	67.567236	74.850	2.576	74.843	2.575
67.693275	0.043330	0.006595	67.687076	74.983	2.565	74.976	2.565
67.813126	0.041125	0.006198	67.806916	75.116	2.565	75.110	2.565
67.932977	0.039997	0.006178	67.926756	75.250	2.554	75.243	2.554
68.052828	0.038380	0.006170	68.046596	75.383	2.554	75.376	2.554
68.172679	0.038150	0.005888	68.166436	75.516	2.554	75.509	2.554
68.292529	0.036616	0.005885	68.286276	75.649	2.554	75.642	2.554
68.412380	0.035927	0.006286	68.406115	75.783	2.554	75.776	2.543
68.532231	0.034312	0.006418	68.525955	75.916	2.543	75.909	2.543
68.652082	0.033462	0.005756	68.645795	76.049	2.543	76.042	2.543
68.771933	0.032740	0.006107	68.765635	76.182	2.543	76.175	2.543
68.891784	0.032401	0.006085	68.885475	76.316	2.531	76.309	2.543
69.011634	0.030744	0.006262	69.005315	76.449	2.531	76.442	2.543

69.131485	0.029994	0.005957	69.125155	76.582	2.531	76.575	2.543
69.251336	0.029655	0.005974	69.244995	76.715	2.531	76.708	2.532
69.371187	0.028506	0.006087	69.364835	76.849	2.531	76.841	2.532
69.491038	0.027438	0.006088	69.484675	76.982	2.531	76.975	2.532
69.610889	0.026969	0.005698	69.604515	77.115	2.531	77.108	2.532
69.730740	0.027190	0.005838	69.724355	77.248	2.531	77.241	2.532
69.850590	0.025739	0.006112	69.844195	77.381	2.531	77.374	2.532
69.970441	0.026074	0.006134	69.964034	77.515	2.531	77.508	2.521
70.090292	0.025182	0.005878	70.083874	77.648	2.531	77.641	2.521
70.210143	0.024078	0.006256	70.203714	77.781	2.531	77.774	2.532
70.329994	0.023906	0.006066	70.323554	77.914	2.531	77.907	2.532
70.449845	0.024444	0.006236	70.443394	78.048	2.531	78.040	2.521
70.569695	0.022947	0.006117	70.563234	78.181	2.531	78.174	2.532
70.689546	0.021980	0.006283	70.683074	78.314	2.531	78.307	2.532
70.809397	0.022669	0.006367	70.802914	78.447	2.531	78.440	2.532
70.929248	0.021344	0.006375	70.922754	78.581	2.531	78.573	2.521
71.049099	0.022624	0.006372	71.042594	78.714	2.531	78.707	2.532
71.168950	0.022052	0.006085	71.162434	78.847	2.531	78.840	2.532
71.288800	0.021538	0.006021	71.282274	78.980	2.531	78.973	2.532
71.408651	0.021130	0.006389	71.402114	79.114	2.531	79.106	2.521
71.528502	0.020709	0.006297	71.521953	79.247	2.531	79.239	2.521
71.648353	0.020700	0.006402	71.641793	79.380	2.531	79.373	2.532
71.768204	0.019346	0.006092	71.761633	79.513	2.531	79.506	2.532
71.888055	0.019929	0.006372	71.881473	79.646	2.531	79.639	2.532
72.007905	0.020734	0.006549	72.001313	79.780	2.531	79.772	2.521
72.127756	0.019566	0.006278	72.121153	79.913	2.531	79.906	2.521
72.247607	0.019975	0.006168	72.240993	80.046	2.531	80.039	2.532
72.367458	0.019926	0.006038	72.360833	80.179	2.531	80.172	2.532
72.487309	0.018844	0.006352	72.480673	80.313	2.531	80.305	2.521
72.607160	0.019539	0.006041	72.600513	80.446	2.531	80.439	2.521
72.727010	0.018641	0.006381	72.720353	80.579	2.531	80.572	2.521
72.846861	0.018674	0.006496	72.840193	80.712	2.531	80.705	2.532
72.966712	0.018943	0.006407	72.960033	80.846	2.531	80.838	2.521
73.086563	0.019339	0.006427	73.079872	80.979	2.531	80.971	2.521
73.206414	0.019473	0.006186	73.199712	81.112	2.531	81.105	2.521
73.326265	0.019267	0.005961	73.319552	81.245	2.520	81.238	2.521
73.446115	0.018804	0.006208	73.439392	81.379	2.520	81.371	2.521
73.565966	0.018958	0.006380	73.559232	81.512	2.531	81.504	2.521
73.685817	0.018744	0.006642	73.679072	81.645	2.531	81.638	2.521

73.805668	0.019267	0.006148	73.798912	81.778	2.531	81.771	2.521
73.925519	0.018369	0.006378	73.918752	81.911	2.520	81.904	2.521
74.045370	0.017906	0.006012	74.038592	82.045	2.520	82.037	2.521
74.165220	0.017960	0.006309	74.158432	82.178	2.531	82.170	2.521
74.285071	0.016817	0.005897	74.278272	82.311	2.531	82.304	2.521
74.404922	0.017652	0.005888	74.398112	82.444	2.531	82.437	2.521
74.524773	0.018205	0.005938	74.517951	82.578	2.531	82.570	2.521
74.644624	0.018045	0.006231	74.637791	82.711	2.531	82.703	2.521
74.764475	0.017452	0.005717	74.757631	82.844	2.531	82.837	2.521
74.884325	0.017870	0.005617	74.877471	82.977	2.531	82.970	2.521
75.004176	0.017749	0.005648	74.997311	83.111	2.531	83.103	2.521
75.124027	0.017232	0.005425	75.117151	83.244	2.531	83.236	2.521
75.243878	0.016693	0.005683	75.236991	83.377	2.531	83.369	2.521
75.363729	0.016787	0.005346	75.356831	83.510	2.531	83.503	2.521
75.483580	0.016935	0.005043	75.476671	83.644	2.531	83.636	2.532
75.603431	0.015834	0.004913	75.596511	83.777	2.543	83.769	2.532
75.723281	0.016905	0.004824	75.716351	83.910	2.565	83.902	2.554
75.843132	0.017135	0.005118	75.836191	84.043	2.576	84.036	2.565
75.962983	0.016288	0.004943	75.956031	84.177	2.599	84.169	2.597
76.082834	0.016285	0.004960	76.075870	84.310	2.633	84.302	2.619
76.202685	0.016152	0.004904	76.195710	84.443	2.678	84.435	2.651
76.322536	0.016748	0.004365	76.315550	84.576	2.723	84.568	2.705
76.442386	0.015614	0.004750	76.435390	84.709	2.780	84.702	2.749
76.562237	0.015517	0.004799	76.555230	84.843	2.825	84.835	2.803
76.682088	0.015314	0.004401	76.675070	84.976	2.870	84.968	2.846
76.801939	0.015723	0.004590	76.794910	85.109	2.926	85.101	2.901
76.921790	0.015611	0.004419	76.914750	85.242	2.971	85.235	2.944
77.041641	0.015831	0.004169	77.034590	85.376	3.005	85.368	2.987
77.161491	0.015311	0.004139	77.154430	85.509	3.050	85.501	3.009
77.281342	0.015520	0.004286	77.274270	85.642	3.073	85.634	3.031
77.401193	0.015472	0.004123	77.394110	85.775	3.095	85.767	3.063
77.521044	0.015544	0.004302	77.513950	85.909	3.129	85.901	3.085
77.640895	0.015499	0.004208	77.633789	86.042	3.152	86.034	3.107
77.760746	0.014731	0.004150	77.753629	86.175	3.163	86.167	3.128
77.880596	0.015435	0.003891	77.873469	86.308	3.186	86.300	3.139
78.000447	0.014431	0.003493	77.993309	86.442	3.197	86.434	3.150
78.120298	0.014226	0.003783	78.113149	86.575	3.208	86.567	3.172
78.240149	0.014477	0.003457	78.232989	86.708	3.220	86.700	3.183
78.360000	0.013929	0.003481	78.352829	86.841	3.242	86.833	3.193

78.479851	0.014220	0.003456	78.472669	86.974	3.253	86.966	3.215
78.599701	0.014985	0.003653	78.592509	87.108	3.253	87.100	3.215
78.719552	0.014827	0.003227	78.712349	87.241	3.265	87.233	3.226
78.839403	0.014604	0.003562	78.832189	87.374	3.276	87.366	3.237
78.959254	0.014936	0.003473	78.952029	87.507	3.287	87.499	3.248
79.079105	0.014217	0.003200	79.071869	87.641	3.299	87.633	3.248
79.198956	0.015118	0.003418	79.191708	87.774	3.299	87.766	3.259
79.318806	0.014649	0.002852	79.311548	87.907	3.310	87.899	3.259
79.438657	0.014697	0.003048	79.431388	88.040	3.310	88.032	3.280
79.558508	0.014513	0.002882	79.551228	88.174	3.321	88.166	3.280
79.678359	0.014785	0.002924	79.671068	88.307	3.321	88.299	3.280
79.798210	0.014313	0.002600	79.790908	88.440	3.332	88.432	3.291
79.918061	0.014728	0.002979	79.910748	88.573	3.332	88.565	3.291
80.037911	0.013533	0.002533	80.030588	88.707	3.344	88.698	3.291
80.157762	0.014483	0.003014	80.150428	88.840	3.344	88.832	3.291
80.277613	0.014694	0.002926	80.270268	88.973	3.344	88.965	3.302
80.397464	0.014728	0.002971	80.390108	89.106	3.355	89.098	3.302
80.517315	0.013932	0.002710	80.509948	89.239	3.355	89.231	3.313
80.637166	0.014180	0.002691	80.629788	89.373	3.366	89.365	3.324
80.757016	0.013990	0.002565	80.749627	89.506	3.366	89.498	3.324
80.876867	0.013820	0.002554	80.869467	89.639	3.366	89.631	3.324
80.996718	0.013941	0.002514	80.989307	89.772	3.366	89.764	3.324
81.116569	0.014238	0.002495	81.109147	89.906	3.378	89.897	3.334
81.236420	0.013932	0.002575	81.228987	90.039	3.389	90.031	3.345
81.356271	0.014017	0.002677	81.348827	90.172	3.423	90.164	3.367
81.476122	0.013448	0.002650	81.468667	90.305	3.468	90.297	3.410
81.595972	0.013034	0.001779	81.588507	90.439	3.547	90.430	3.475
81.715823	0.013473	0.002507	81.708347	90.572	3.671	90.564	3.573
81.835674	0.013591	0.002250	81.828187	90.705	3.953	90.697	3.801
81.955525	0.012889	0.002228	81.948027	90.838	4.506	90.830	4.300
82.075376	0.012487	0.002134	82.067867	90.972	5.127	90.963	4.863
82.195227	0.012481	0.001619	82.187707	91.105	5.567	91.096	5.308
82.315077	0.012786	0.001365	82.307546	91.238	5.883	91.230	5.644
82.434928	0.012378	0.001633	82.427386	91.371	6.097	91.363	5.883
82.554779	0.012723	0.001653	82.547226	91.505	6.210	91.496	6.024
82.674630	0.012036	0.001247	82.667066	91.638	6.266	91.629	6.089
82.794481	0.012426	0.001367	82.786906	91.771	6.311	91.763	6.132
82.914332	0.011967	0.000769	82.906746	91.904	6.345	91.896	6.176
83.034182	0.010814	0.001127	83.026586	92.037	6.390	92.029	6.208

83.154033	0.011147	0.000691	83.146426	92.171	6.424	92.162	6.241
83.273884	0.010929	0.000818	83.266266	92.304	6.458	92.295	6.273
83.393735	0.010645	0.000404	83.386106	92.437	6.492	92.429	6.306
83.513586	0.010633	0.000353	83.505946	92.570	6.515	92.562	6.327
83.633437	0.010418	0.000199	83.625786	92.704	6.537	92.695	6.360
83.753287	0.010279	-0.000190	83.745626	92.837	6.560	92.828	6.382
83.873138	0.010237	-0.000246	83.865465	92.970	6.582	92.962	6.392
83.992989	0.009904	-0.000409	83.985305	93.103	6.605	93.095	6.425
84.112840	0.009659	-0.000064	84.105145	93.237	6.627	93.228	6.447
84.232691	0.009587	-0.000382	84.224985	93.370	6.639	93.361	6.458
84.352542	0.009169	-0.000083	84.344825	93.503	6.673	93.494	6.490
84.472392	0.010334	-0.000072	84.464665	93.636	6.695	93.628	6.501
84.592243	0.010863	0.000381	84.584505	93.770	6.718	93.761	6.523
84.712094	0.011020	0.000588	84.704345	93.903	6.740	93.894	6.544
84.831945	0.011622	0.000821	84.824185	94.036	6.763	94.027	6.566
84.951796	0.013061	0.000931	84.944025	94.169	6.785	94.161	6.588
85.071647	0.013321	0.001567	85.063865	94.302	6.808	94.294	6.620
85.191497	0.015459	0.002460	85.183705	94.436	6.842	94.427	6.642
85.311348	0.015514	0.003158	85.303545	94.569	6.876	94.560	6.674
85.431199	0.016288	0.003642	85.423384	94.702	6.910	94.693	6.718
85.551050	0.017776	0.004252	85.543224	94.835	6.943	94.827	6.739
85.670901	0.018078	0.005256	85.663064	94.969	6.966	94.960	6.772
85.790752	0.019046	0.005101	85.782904	95.102	7.000	95.093	6.794
85.910602	0.019110	0.005145	85.902744	95.235	7.022	95.226	6.815
86.030453	0.019590	0.005164	86.022584	95.368	7.056	95.360	6.826
86.150304	0.019055	0.005385	86.142424	95.502	7.068	95.493	6.848
86.270155	0.018937	0.004910	86.262264	95.635	7.090	95.626	6.870
86.390006	0.017531	0.004255	86.382104	95.768	7.113	95.759	6.902
86.509857	0.017362	0.003934	86.501944	95.901	7.135	95.893	6.924
86.629707	0.017217	0.003904	86.621784	96.035	7.135	96.026	6.935
86.749558	0.016527	0.003437	86.741624	96.168	7.135	96.159	6.946
86.869409	0.015287	0.002982	86.861463	96.301	7.135	96.292	6.946
86.989260	0.014310	0.002678	86.981303	96.434	7.124	96.425	6.935
87.109111	0.014235	0.001986	87.101143	96.567	7.124	96.559	6.935
87.228962	0.012989	0.001821	87.220983	96.701	7.124	96.692	6.935
87.348813	0.013645	0.001656	87.340823	96.834	7.124	96.825	6.935
87.468663	0.012181	0.001006	87.460663	96.967	7.124	96.958	6.935
87.588514	0.012336	0.000860	87.580503	97.100	7.113	97.092	6.935
87.708365	0.011253	0.000321	87.700343	97.234	7.113	97.225	6.935

87.828216	0.010427	0.000462	87.820183	97.367	7.124	97.358	6.935
87.948067	0.011531	-0.000589	87.940023	97.500	7.124	97.491	6.935
88.067918	0.010539	-0.000182	88.059863	97.633	7.113	97.624	6.935
88.187768	0.010805	-0.000501	88.179703	97.767	7.113	97.758	6.924
88.307619	0.010569	-0.000945	88.299543	97.900	7.113	97.891	6.924
88.427470	0.009184	-0.000701	88.419382	98.033	7.113	98.024	6.924
88.547321	0.009783	-0.001197	88.539222	98.166	7.113	98.157	6.924
88.667172	0.008821	-0.001039	88.659062	98.300	7.113	98.291	6.924
88.787023	0.009326	-0.001244	88.778902	98.433	7.113	98.424	6.924
88.906873	0.008803	-0.001274	88.898742	98.566	7.113	98.557	6.924
89.026724	0.009166	-0.001832	89.018582	98.699	7.113	98.690	6.935
89.146575	0.008994	-0.001432	89.138422	98.832	7.113	98.823	6.935
89.266426	0.009193	-0.001138	89.258262	98.966	7.113	98.957	6.935
89.386277	0.009544	-0.001351	89.378102	99.099	7.113	99.090	6.935
89.506128	0.010246	-0.001613	89.497942	99.232	7.113	99.223	6.935
89.625978	0.007981	-0.001597	89.617782	99.365	7.113	99.356	6.924
89.745829	0.009360	-0.001638	89.737622	99.499	7.113	99.490	6.924
89.865680	0.009178	-0.001291	89.857462	99.632	7.113	99.623	6.924
89.985531	0.008643	-0.001631	89.977301	99.765	7.101	99.756	6.924
90.105382	0.009363	-0.001293	90.097141	99.898	7.101	99.889	6.924
90.225233	0.009021	-0.001327	90.216981	100.032	7.101	100.022	6.924
90.345083	0.008307	-0.000824	90.336821	100.165	7.101	100.156	6.924
90.464934	0.009426	-0.000660	90.456661	100.298	7.101	100.289	6.935
90.584785	0.009466	-0.000409	90.576501	100.431	7.101	100.422	6.924
90.704636	0.010494	0.000447	90.696341	100.565	7.101	100.555	6.924
90.824487	0.010663	0.000168	90.816181	100.698	7.101	100.689	6.935
90.944338	0.009568	0.000519	90.936021	100.831	7.101	100.822	6.924
91.064188	0.009901	0.001352	91.055861	100.964	7.101	100.955	6.924
91.184039	0.011201	0.001801	91.175701	101.098	7.101	101.088	6.935
91.303890	0.011389	0.002069	91.295541	101.231	7.101	101.221	6.924
91.423741	0.008725	0.001382	91.415381	101.364	7.101	101.355	6.924
91.543592	0.007397	0.000094	91.535220	101.497	7.101	101.488	6.924
91.663443	0.007267	-0.000959	91.655060	101.630	7.101	101.621	6.935
91.783293	0.006520	-0.000900	91.774900	101.764	7.101	101.754	6.924
91.903144	0.006272	-0.000691	91.894740	101.897	7.101	101.888	6.924
92.022995	0.005779	-0.000835	92.014580	102.030	7.090	102.021	6.935
92.142846	0.005011	-0.000956	92.134420	102.163	7.090	102.154	6.924
92.262697	0.005265	-0.001150	92.254260	102.297	7.090	102.287	6.924
92.382548	0.005195	-0.001819	92.374100	102.430	7.090	102.420	6.935

92.502398	0.005075	-0.001835	92.493940	102.563	7.090	102.554	6.935
92.622249	0.004848	-0.001774	92.613780	102.696	7.101	102.687	6.924
92.742100	0.005434	-0.001329	92.733620	102.830	7.101	102.820	6.924
92.861951	0.005117	-0.001262	92.853460	102.963	7.101	102.953	6.924
92.981802	0.004805	-0.001782	92.973300	103.096	7.101	103.087	6.935
93.101653	0.004494	-0.002040	93.093139	103.229	7.090	103.220	6.935
93.221504	0.004966	-0.002017	93.212979	103.363	7.101	103.353	6.924
93.341354	0.004869	-0.002056	93.332819	103.496	7.101	103.486	6.924
93.461205	0.004533	-0.001526	93.452659	103.629	7.090	103.620	6.924
93.581056	0.004718	-0.001743	93.572499	103.762	7.090	103.753	6.935
93.700907	0.004506	-0.001508	93.692339			103.886	6.935
93.820758	0.005785	-0.001580	93.812179				
93.940609	0.004606	-0.001425	93.932019				
94.060459	0.004518	-0.001147	94.051859				
94.180310	0.005362	-0.000561	94.171699				
94.300161	0.005280	-0.000616	94.291539				
94.420012	0.005404	-0.000205	94.411379				
94.539863	0.005583	-0.000163	94.531219				
94.659714	0.005791	0.000133	94.651058				
94.779564	0.005138	0.000386	94.770898				
94.899415	0.005180	0.000367	94.890738				
95.019266	0.005274	-0.000683	95.010578				
95.139117	0.004007	-0.000478	95.130418				
95.258968	0.004866	-0.000816	95.250258				
95.378819	0.004488	-0.000619	95.370098				
95.498669	0.005486	-0.001070	95.489938				
95.618520	0.004291	-0.000843	95.609778				
95.738371	0.004848	-0.001221	95.729618				
95.858222	0.005159	-0.001572	95.849458				
95.978073	0.004037	-0.002573	95.969298				
96.097924	0.003296	-0.003125	96.089138				
96.217774	0.002268	-0.003158	96.208977				
96.337625	0.002062	-0.003399	96.328817				
96.457476	0.002876	-0.003706	96.448657				
96.577327	0.002561	-0.004109	96.568497				
96.697178	0.003472	-0.003612	96.688337				
96.817029	0.002186	-0.003523	96.808177				
96.936879	0.002214	-0.004137	96.928017				
97.056730	0.001917	-0.003904	97.047857				

97.176581	0.001975	-0.004122	97.167697				
97.296432	0.001751	-0.003979	97.287537				
97.416283	0.001718	-0.004667	97.407377				
97.536134	0.001446	-0.004468	97.527217				
97.655984	0.001923	-0.003857	97.647057				
97.775835	0.001920	-0.004042	97.766896				
97.895686	0.001409	-0.003791	97.886736				
98.015537	0.000956	-0.003904	98.006576				
98.135388	0.002356	-0.004051	98.126416				
98.255239	0.001025	-0.003945	98.246256				
98.375089	0.001957	-0.004308	98.366096				
98.494940	0.000886	-0.003747	98.485936				
98.614791	0.001161	-0.004336	98.605776				
98.734642	0.001745	-0.004643	98.725616				
98.854493	0.001691	-0.004045	98.845456				
98.974344	0.001966	-0.004084	98.965296				
99.094195	0.002078	-0.004437	99.085136				
99.214045	0.001246	-0.004377	99.204976				
99.333896	0.001001	-0.004529	99.324815				
99.453747	0.001222	-0.004418	99.444655				
99.573598	0.000493	-0.004253	99.564495				
99.693449	0.001709	-0.004122	99.684335				
99.813300	0.001488	-0.004308	99.804175				
99.933150	0.001346	-0.004567	99.924015				
100.053001	0.000178	-0.004631	100.043855				
100.172852	0.001058	-0.004197	100.163695				
100.292703	0.000953	-0.004819	100.283535				
100.412554	0.001140	-0.004477	100.403375				
100.532405	0.001467	-0.004672	100.523215				
100.652255	0.001001	-0.004339	100.643055				
100.772106	0.001485	-0.004816	100.762894				
100.891957	0.001346	-0.004535	100.882734				
101.011808	0.000871	-0.004412	101.002574				
101.131659	0.001110	-0.004669	101.122414				
101.251510	0.001391	-0.004764	101.242254				
101.371360	0.000965	-0.004830	101.362094				
101.491211	0.000064	-0.004717	101.481934				
101.611062	0.000928	-0.004543	101.601774				
101.730913	0.001201	-0.004860	101.721614				

101.850764	0.001055	-0.004759	101.841454				
101.970615	0.000599	-0.005012	101.961294				
102.090465	0.000315	-0.004819	102.081134				
102.210316	0.001279	-0.004783	102.200974				
102.330167	0.000487	-0.004573	102.320813				
102.450018	0.001591	-0.004176	102.440653				
102.569869	0.001464	-0.004691	102.560493				
102.689720	0.000360	-0.004968	102.680333				
102.809570	0.001651	-0.004806	102.800173				
102.929421	0.000995	-0.004645	102.920013				
103.049272	0.001255	-0.004991	103.039853				
103.169123	0.000895	-0.004722	103.159693				
103.288974	0.000426	-0.004954	103.279533				
103.408825	0.000826	-0.005132	103.399373				
103.528675	0.000901	-0.005425	103.519213				
103.648526	0.000883	-0.004719	103.639053				
103.768377	0.000596	-0.005212	103.758893				
103.888228	0.000472	-0.004767	103.878732				
104.008079	0.000844	-0.004835	103.998572				

Appendix V: MATLAB Scripts

```
% READ_THIS_FIRST.m
%-------------------------------------------------------------------------
% This file explains all of the variables and nomenclature
%-------------------------------------------------------------------------
% created by Virginia Lane on 06/09/18
% last updated on 10/13/18
%
%-------------------------------------------------------------------------
% file names
%-------------------------------------------------------------------------
% eq_           these files are the equilibrium case
% kin_          these files are the kinetic case
%
% _ic_          initial conditions script
% _bc_          boundary conditions script
% _pde_         script containing the partial differential equations
% _solver_      script that solves the PDE with associated IC and BC
% _parameters_  contains system parameters
% _fit_         polynomial fitting script for concentration wrt radius
at final t
%
% _I            phase I
% _II           phase II
%
% _surfgraphpcolor    graph that uses pcolor to plot a contour of r vs
c
% _surfgraph          graph that uses surf to plot a contour of r vs c
vs t
% _qvrgraph           graph of q vs r
% _qcontourgraphs     graphs 9 q (solid phase) contour plots
% _qcontourgraph      graph of q (solid phase) contour plot at a given
t
% _cvrgraph           graph of c vs r
% _cvrfinal           graph of c vs r at final t (also shows the
polyfit)
% _contourgraphs      graphs 9 c (liquid phase) contour plots
% _contourgraph       graph of c (liquid phase) contour plot at a
given t
%
%
%-------------------------------------------------------------------------
% parameter names
%-------------------------------------------------------------------------
% r_            radius
% t_            time
% D_            diffusivity
% e_p_          porosity
% q_max_        maximum q value
% k_a_          affinity constant (k_a = K = k_f/k_r)
% k_d_          1/k_a
```

```
% k_r_          reverse reaction constant
% k_f_          forward reaction constant
% pH_           pH
% c_0_          initial solution concentration (outside the sphere)
% c_1_          initial concentration inside the sphere
%
% _1_           1st component, mAb
% _2_           2nd component, "moderate" IgG
% _3_           3rd component, "weak" IgG
% _4_           4th component, "strong" IgG
%
% _i            initial/inside
% _o            outside
% _s            steps
% _p            points
%
%------------------------------------------------------------------------
```

Diffusion in a Porous Sphere with Analytical Solution

```
%-------------------------------------------------------------------------
% parameters.m
%
% created by Virginia Lane 01/24/18
% edited by TP 02/07/18
% finalized 08/27/18
%-------------------------------------------------------------------------
clc                     % clear command window
clear                   % clear all variables

% parameters
%-------------------------------------------------------------------------
global m r t c_0 c_1 D R t_i t_f t_p t_s r_i r_o r_p r_s...
    mAb_map colororder colororder2

m   = 2;                % indicates spherical symmetry
c_0 = 2;                % initial concentration of solution in mg/mL
c_1 = 0;                % initial concentration of sphere in mg/mL
D   = 7.5e-8;           % diffusivity constant in (cm^2)/(s)
Dm  = 85e-4;            % diameter of sphere in cm
R   = Dm./2;            % radius of sphere in cm

t_i = 0;                % initial time point
t_f = 60;              % final time point
t_p = 1000;             % number of time points
t_s = t_f./(t_p-1);     % step size of time points
t   = linspace(t_i,t_f,t_p); % time point report-out in seconds

r_i = 0;                % inside radius
r_o = R;                % outside radius
r_p = 100;              % number of radius points
r_s = r_o./(r_p-1);     % step size of radius points
r = linspace(r_i,r_o,r_p);    % radial point report-out in cm

%colormap
%-------------------------------------------------------------------------
colorinitial = 0;             % specifies black initial color
colorfinal   = .85;
colorstep    = .01;
green        = (colorinitial:colorstep:colorfinal).';
red          = (zeros(1, colorfinal./colorstep+1)).';
blue         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map      = [red,green,blue];

%colororder
%-------------------------------------------------------------------------
colorinitial = 0;             % specifies black initial color
colorfinal   = .85;
colorstep    = (colorfinal-colorinitial)./(t_p-1);
green        = (colorinitial:colorstep:colorfinal).';
```

```
red          = (zeros(1, colorfinal./colorstep+1)).';
blue         = red;
colororder   = [red,green,blue];
colororder2  = prism;
```

```
%-------------------------------------------------------------------------
% ic.m
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
function u_0 = ic(r)
global c_1
u_0 = c_1;
```

```
%-------------------------------------------------------------------------
% bc.m
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = bc(rl,ul,rr,ur,t)
global c_0
pl  = 0;                % "left" bc for r = 0, represents no flux
ql  = 1;
pr  = ur-c_0;           % "right" bc for r = R, represents constant
qr  = 0;                %                       surface concentration
```

```
%-------------------------------------------------------------------------
% pde.m
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
function [c,f,s] = pde(r,t,u,DuDr)
global D
c = 1;
f = D*DuDr;
s = 0;                    % no source term (yet...)
```

```
%--------------------------------------------------------------------------
% solver.m
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%--------------------------------------------------------------------------
function solver
global m r t u
sol = pdepe(m,@pde,@ic,@bc,r,t);
u = sol(:,:);
```

```
%-------------------------------------------------------------------------
% solution.m
%
% analytical solution for diffusion into a sphere from Crank textbook
% concentration matrix (time points = rows, radius points = columns)
%
% created by Virginia Lane on 02/15/18
% finalized 08/27/18
%-------------------------------------------------------------------------
function solution
parameters
global c_0 c_1 D R u_sol t_i t_f t_s r_i r_o r_p r_s

% first time row at t = 0
%-------------------------------------------------------------------------
u_sol = c_1;
for n = 1:1:r_p-2
    u_sol = [u_sol c_1];
end
u_sol = [u_sol c_0];

% c matrix
%-------------------------------------------------------------------------
for t = t_i+t_s:t_s:t_f

    % function for c as r approaches 0
    %---------------------------------------------------------------------
    sum = 0;
    for n = 1:1:1000
        sum2 = (-1)^n.*exp(-D.*n^2*pi^2.*t./(R.^2));
        sum = sum2 + sum;
    end
    u_sol3 = (1+2*sum)*(c_0-c_1)+c_1;

    % function for c at larger times when r > 0 :
    %---------------------------------------------------------------------
    for r = r_i+r_s:r_s:r_o
        sum = 0;
        for n = 1:1:1000
            sum2 = (-1)^n./n.*sin(n*pi.*r./R).*...
                exp(-D.*n^2*pi^2.*t./(R.^2));
            sum = sum2 + sum;
        end
        u_sol2 = (1+2*R./(pi*r).*sum)*(c_0-c_1)+c_1;
    u_sol3 = [u_sol3 u_sol2];
    end
u_sol = [u_sol;u_sol3];
end
```

```
%-------------------------------------------------------------------------
% difference.m
%
% graphs a contour plot of the percent difference between the
analytical
% solution and MATLAB solver
% Run this script by itself
%
% created by Virginia Lane on 02/15/18
% finalized 08/27/18
%-------------------------------------------------------------------------
solution
parameters
solver

global t_i t_f r u t u_sol
figure(1);
for n = 0:1:3                  % always starts at time point 0
    subplot(1,4,n+1)

    n3   = (t_f-t_i)./4;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    k               = t(column2);
    theta           = 0:.1:33*pi/16;
    [th, rad]       = meshgrid(theta,r);
    X               = rad.*cos(th);
    Y               = rad.*sin(th);
    r_circle        = sqrt(X.^2 + Y.^2);
    [row, column]   = size(r_circle);
    u2              = abs((u(column2,:)-
u_sol(column2,:)))./u(column2,:);
    u2              = u2.';
    n2              = 0;
    U               = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(hot);
    axis('square');
    axis off
    title(['t = ' num2str(round(k)) ' sec']);
end
cb = colorbar('location','Manual','position',[.92 .4 .03 .25]);
title(cb,{'percent','error'});
```

```
%-------------------------------------------------------------------------
% surfgraph.m
%
% script gives a 3D surface plot of concentration vs radius vs time
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
parameters
solver

global r t u R c_0 mAb_map
figure(1);
surf(r,t,u);
colormap(mAb_map);
grid on
shading interp
xlabel('radius (cm)');
ylabel('time (seconds)');
zlabel('conc. (mg/mL)');
xlim([0 R])
zlim([0 c_0])
title('concentration vs. radius vs. time')
title(colorbar,'mAb conc.');
```

```
%-------------------------------------------------------------------------
% cvrgraph.m
%
% script that graphs a plot of time curves for concentration vs radius
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
parameters
solver

global  c_0 R t_i t_f r u colororder2
figure(1);
co = get(gca,'ColorOrder');
set(gca, 'ColorOrder', colororder2, 'NextPlot','replacechildren');
plot(r,u)
title('Concentration vs. Radius')
xlabel('radius (cm)')
ylabel('conc. (mg/mL)')
xlim([0 R])
ylim([0 c_0])
t2 = linspace(t_i,t_f,21);
t2 = string(t2);
labels = get(legend(t2),'string');
legend2 = legend(labels,'location','eastoutside');
title(legend2,'time (sec)')
```

```
%-------------------------------------------------------------------------
% surfgraphpcolor.m
%
% This 2D radius vs time plot shows concentration as a color contour
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
parameters
solver

global r t u c_0 mAb_map
figure(1);
pcolor(r,t,u)
colormap(mAb_map);
grid off
shading interp
xlabel('radius (cm)');
ylabel('time (seconds)');
title('time vs. radius')
caxis([0 c_0])
title(colorbar,{'mAb conc','mg/mL'});
```

```
%-------------------------------------------------------------------------
% contourgraph.m
%
% graphs a concentration contour plot for a spherical cross section
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
parameters
solver

global t_i t_f c_0 c_1 r u t mAb_map
prompt = ['what time point would you like between ' num2str(t_i)...
    ' and ' num2str(t_f) ' seconds? '];
time = input(prompt);

figure(1);
[row2 column2] = min(abs(t-time));
k = t(column2);

theta        = 0:.1:33*pi/16;
[th, rad]    = meshgrid(theta,r);
X            = rad.*cos(th);
Y            = rad.*sin(th);
r_circle     = sqrt(X.^2 + Y.^2);
[row column] = size(r_circle);
u            = u(column2,:);
u            = u.';
m = 0;
U = u;
for m = 1:1:column-1;
    U = [U,u];
    m = m + 1;
end

pcolor(X,Y,U)
shading interp;
colormap(mAb_map);
axis('square');
axis off
caxis([c_1 c_0])
k2 = round(k);
title([num2str(k2) ' seconds']);
title(colorbar,{'mg/mL'});
set(gca, 'fontsize',20)
```

```
%-------------------------------------------------------------------------
% contourgraphs.m
%
% graphs 9 concentration contour plots, cross section of bead
%
% created by Virginia Lane on 02/15/18
% finalized 08/27/18
%-------------------------------------------------------------------------
parameters
solver

global t_i t_f c_1 c_0 r u t mAb_map
figure(1);
for n = 0:1:3                  % always starts at time point 0
    subplot(1,4,n+1)

    n3   = (t_f-t_i)./4;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    k               = t(column2);
    theta           = 0:.1:33*pi/16;
    [th, rad]       = meshgrid(theta,r);
    X               = rad.*cos(th);
    Y               = rad.*sin(th);
    r_circle        = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2              = u(column2,:);
    u2              = u2.';
    n2              = 0;
    U               = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map);
    axis('square');
    axis off
    caxis([c_1 c_0])
    title(['t = ' num2str(round(k)) ' sec']);
end
%('Diffusion of mAbs into a Sphere')
cb = colorbar('location','Manual','position',[.92 .4 .03 .25]);
title(cb,{'mAb conc.','mg/mL'});
```

```
%-------------------------------------------------------------------------
% surfgraph_sol.m
%
% script graphs a 3D surface plot of concentration vs radius vs time
% script uses the analytical solution
%
% created by Virginia Lane on 02/14/18
% finalized 080/27/18
%-------------------------------------------------------------------------
solution
parameters

global c_0 R r t u_sol mAb_map
u = u_sol;

figure(2);
surf(r./R,t,u);
colormap(mAb_map);
grid on
shading interp
xlabel('r (cm)');
ylabel('time (seconds)');
zlabel('conc. (mg/mL)');
xlim([0 R])
zlim([0 c_0])
title('concentration vs. radius vs. time (analytical solution)')
title(colorbar,'mAb conc.');
```

```
%-------------------------------------------------------------------------
% surfgraphpcolor_sol.m
%
% This 2D radius vs time plot shows concentration as a color contour
% This file uses the analytical solution
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
solution
parameters

global r t u_sol mAb_map
u = u_sol;

figure(2);
pcolor(r,t,u)
colormap(mAb_map);
grid off
shading interp
xlabel('radius (cm)');
ylabel('Time (seconds)');
title('time vs. radius (analytical solution)')
caxis([0 2])
title(colorbar,{'mAb conc','mg/mL'});
```

```
%-------------------------------------------------------------------------
% cvrgraph_sol.m
%
% script graphs a plot of time curves for concentration vs radius
% script uses the analytical solution
%
% created by Virginia Lane on 02/14/18
% finalized 08/27/18
%-------------------------------------------------------------------------
solution
parameters

global  c_0 R t_i t_f r u_sol colororder2
u = u_sol;

figure(2);
co = get(gca,'ColorOrder');
set(gca, 'ColorOrder', colororder2, 'NextPlot','replacechildren');
plot(r,u)
title('Concentration vs. Radius (analytical solution)')
xlabel('radius (cm)')
ylabel('conc. (mg/mL)')
xlim([0 R])
ylim([0 c_0])
t2 = linspace(t_i,t_f,21);
t2 = string(t2);
labels = get(legend(t2),'string');
legend2 = legend(labels,'location','eastoutside');
title(legend2,'time (sec)')
```

```
%-------------------------------------------------------------------------
% contourgraphs_sol.m
%
% graphs 9 concentration contour plots for a spherical cross section
% script uses the analytical solution
%
% created by Virginia Lane on 02/15/18
% finalized 08/27/18
%-------------------------------------------------------------------------
solution
parameters

global t_i t_f c_1 c_0 r u_sol t mAb_map
u = u_sol;

figure(2);
for n = 0:1:3                  % always starts at time point 0
    subplot(1,4,n+1)

    n3    = (t_f-t_i)./4;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    k               = t(column2);
    theta           = 0:.1:33*pi/16;
    [th, rad]       = meshgrid(theta,r);
    X               = rad.*cos(th);
    Y               = rad.*sin(th);
    r_circle        = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2              = u(column2,:);
    u2              = u2.';
    n2              = 0;
    U               = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map);
    axis('square');
    axis off
    caxis([c_1 c_0])
    title(['t = ' num2str(round(k)) ' sec']);
end
%suptitle('diffusion of mAbs (analytical solution)')
cb = colorbar('location','Manual','position',[.92 .39 .03 .25]);
title(cb,{'mAb conc.','mg/mL'});
```

Adsorption and Diffusion in a Single Chromatographic Bead, Equilibrium Case

```
% ------------------------------------------------------------------
% parameters.m
%
% parameters for diffusion of mAbs into a sphere
%
% created by Virginia Lane 01/24/18
% finalized 09/06/18
% ------------------------------------------------------------------
clc                     % clear command window
clear                   % clear all variables
colors;

global m c_0 c_1 D R k_d k_a k_f k_r e_p q_max...
    c_0_2 c_1_2 D_2 R_2 k_d_2 k_a_2 k_f_2 k_r_2 e_p_2 q_max_2...
    r_i r_o t_i t_f t_p t_s r_p r_s...
    r t

% constant parameters
%-------------------------------------------------------------------
m     = 2;              % indicates spherical symmetry
c_0   = 2;              % initial concentration of solution in mg/mL
c_1   = 0;              % initial concentration of sphere in mg/mL
D     = 7.5e-8*60;      % diffusivity constant in (cm^2)/(min)
e_p   = .65;            % interstitial porosity
q_max = 61;             % mAb q_max, (mg mAb)/(mL) hydrated resin
Dm    = 85e-4;          % diameter of sphere in cm
R     = Dm./2;          % radius of sphere in cm

% reaction constants
%-------------------------------------------------------------------
k_a   = 61.806;             % k_a = K = k_f/k_r (association constant)
k_d   = 1./k_a;         % dissociaton constant
k_r   = .002;
k_f   = k_a*k_r;

% competitive/noncompetitive binding
%-------------------------------------------------------------------
c_0_2   = c_0;        % initial concentration of solution in mg/mL
c_1_2   = c_1;        % initial concentration of sphere in mg/mL
D_2     = D;          % diffusivity constant in (cm^2)/(s)
e_p_2   = e_p;        % interstitial porosity
q_max_2 = 84;         % mAb q_max, (mg mAb)/(mL) hydrated resin
R_2     = R;          % radius of sphere in cm
k_a_2   = k_a*.1;     % k_a = K = k_f/k_r (association constant)
k_d_2   = 1/k_a_2;    % dissociaton constant
k_r_2   = k_r;
k_f_2   = k_r_2*k_a_2;

% time parameters
%-------------------------------------------------------------------
```

```
t_i   = 0;                  % initial time point
t_f   = 30;                 % final time point in minutes
t_p   = 100;                % number of time points
t_s   = t_f./(t_p-1);       % step size of time points
t = linspace(t_i,t_f,t_p); % time point report-out in minutes

% radius parameters
%--------------------------------------------------------------------------
r_i   = 0;                  % inside radius
r_o   = R;                  % outside radius
r_p   = 100;                % number of radius points
r_s   = r_o./(r_p-1);       % step size of radius points
r = linspace(r_i,r_o,r_p); % radial point report-out in cm
```

```
%-------------------------------------------------------------------------
% colors.m
%
% gives color profile for graphs in this path
%
% created by Virginia Lane on 02/14/18
% finalized 09/11/18
%-------------------------------------------------------------------------

global mAb_map mAb_map_2 t_p colororder colororder2

% colormap (for IgG molecule)
%-------------------------------------------------------------------------
colorinitial = 0;       % specifies black initial color
colorfinal   = .85;
colorstep    = .01;
green        = (colorinitial:colorstep:colorfinal).';
red          = (zeros(1, colorfinal./colorstep+1)).';
blue         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map      = [red,green,blue];

% colormap 2 (for competitive/noncompetitive molecule)
%-------------------------------------------------------------------------
red_2          = (zeros(1, colorfinal./colorstep+1)).';
green_2        = (colorinitial:colorstep:colorfinal).';
blue_2         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_2      = [red_2,green_2,blue_2];

% colororder
%-------------------------------------------------------------------------
colorinitial = 0;        % specifies black initial color
colorfinal   = 1;
colorstep    = (colorfinal-colorinitial)./(t_p-1);
red          = (colorinitial:colorstep:colorfinal).';
green        = (zeros(1, round(colorfinal./colorstep+1))).';
blue         = green;
colororder   = [red,green,blue];

colororder2  = jet;
```

```
%-------------------------------------------------------------------------
% eq_ic.m
% equilibrium case
%
% pde initial conditions to accompany eq_solver.m
%
% created by Virginia Lane on 02/14/18
% finalized 09/11/18
%-------------------------------------------------------------------------
function u_0 = eq_ic(r)

global c_1

u_0 = c_1;
```

```
%-------------------------------------------------------------------------
% eq_bc.m
% equilibrium case
%
% pde boundary conditions to accompany eq_solver.m
%
% created by Virginia Lane on 02/14/18
% finalized 09/11/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = eq_bc(rl,ul,rr,ur,t)

global c_0

pl  = 0;                % "left" bc for r = 0 and represents no flux
ql  = 1;
pr  = ur-c_0;           % this "right" bc for r = R and...
qr  = 0;                % represents constant surface concentration
```

```
%-------------------------------------------------------------------------
% eq_pde.m
% equilibrium case
%
% pde function for diffusion with absorption to accompany eq_solver.m
%
% created by Virginia Lane on 02/20/18
% finalized 09/11/18
%-------------------------------------------------------------------------
function [c,f,s] = eq_pde(r,t,u,DuDr)

global D q_max k_a e_p

DqDu = k_a*q_max/((1+k_a*u).^2); % Derivative of the langmuir isotherm

c = (e_p+DqDu);
f = e_p*D*DuDr;
s = 0;                                          % no source term
(yet...)
```

```
%-------------------------------------------------------------------------
% eq_solver.m
% equilibrium case
%
% solver for eq_pde.m using eq_ic.m and eq_bc.m
%
% created by Virginia Lane on 02/14/18
% finalized 09/11/18
%-------------------------------------------------------------------------
function eq_solver

global m r t eq_u eq_q k_a q_max

sol = pdepe(m,@eq_pde,@eq_ic,@eq_bc,r,t);
eq_u = sol(:,:,1);

eq_q = k_a*q_max*eq_u/(1+k_a*q_max)
```

```
%-------------------------------------------------------------------------
% eq_contourgraphs.m
% equilibrium case
%
% graphs 6 concentration contour plots for a spherical cross section
% concentration of liquid inside pores
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/15/18
% finalized 09/11/18
%-------------------------------------------------------------------------

global t_i t_f c_1 c_0 r eq_u t mAb_map

if c_1 > c_0
    disp('this is the desorption case ')
end

figure(1);
for n = 0:1:5                   % always starts at time point 0
    subplot(1,6,n+1)

    n3   = (t_f-t_i)./6;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    minutes          = round(t(column2));
    theta          = 0:.1:33*pi/16;
    [th, rad]      = meshgrid(theta,r);
    X              = rad.*cos(th);
    Y              = rad.*sin(th);
    r_circle       = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2             = eq_u(column2,:);
    u2             = u2.';
    n2             = 0;
    U              = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map);
    axis('square');
    axis off
    title([num2str(minutes) ' min']);
    if c_1 > c_0
        caxis([c_0 c_1])
    else
        caxis([c_1 c_0])
    end
```

```
end
cb = colorbar('location','Manual','position',[.92 .45 .03 .18]);
title(cb,{'mAb','mg/mL'});
```

```
%-------------------------------------------------------------------------
% eq_qcontourgraphs.m
% equilibrium case
%
% graphs 6 concentration contour plots for a spherical cross section
% concentration of adsorbed protein
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/15/18
% finalized 09/11/18
%-------------------------------------------------------------------------

global t_i t_f r eq_u t mAb_map c_1 c_0 q_max

if c_1 > c_0
    disp('this is the desorption case ')
end

q = q_max*k_a*eq_u./(1+k_a*eq_u);

figure(1);
for n = 0:1:5                  % always starts at time point 0
    subplot(1,6,n+1)

    n3   = (t_f-t_i)./6;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    minutes         = round(t(column2));
    theta           = 0:.1:33*pi/16;
    [th, rad]       = meshgrid(theta,r);
    X               = rad.*cos(th);
    Y               = rad.*sin(th);
    r_circle        = sqrt(X.^2 + Y.^2);
    [row, column]   = size(r_circle);
    u2              = q(column2,:);
    u2              = u2.';
    n2              = 0;
    U               = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map);
    axis('square');
    axis off
    title([num2str(minutes) ' min']);
    %set(gca,'fontsize',15)
    if c_1 > c_0
       caxis([q_max 0])
```

```
    else

        caxis([0 q_max])
    end
end
cb = colorbar('location','Manual','position',[.92 .45 .03 .18]);
title(cb,{'q','mg/mL'});
```

```
%-------------------------------------------------------------------------
% eq_cvrgraph.m
% equilibrium case
%
% script that graphs a plot of time curves for q vs radius
%
% created by Virginia Lane on 02/23/18
% last updated on 03/28/18
%-------------------------------------------------------------------------

global  q_max k_a R t_i t_f r eq_u

q = q_max*k_a*eq_u./(1+k_a*eq_u);

figure(1);
plot(r,q/q_max)
xlabel('r (cm)')
ylabel('q/q_m_a_x (mg/mL)')
xlim([0 R])
t2 = round(linspace(t_i,t_f,7));
t2 = string(t2);
labels = get(legend(t2),'string');
legend2 = legend(labels,'location','eastoutside');
title(legend2,'time (min)')
```

```
%-------------------------------------------------------------------------
% eq_cvrgraph.m
% equilibrium case
%
% script that graphs a plot of time curves for q vs radius
%
% created by Virginia Lane on 02/23/18
% last updated on 03/28/18
%-------------------------------------------------------------------------

global  q_max k_a R t_i t_f r eq_u

q = q_max*k_a*eq_u./(1+k_a*eq_u);

figure(1);
plot(r,q/q_max)
xlabel('r (cm)')
ylabel('q/q_m_a_x (mg/mL)')
xlim([0 R])
t2 = round(linspace(t_i,t_f,7));
t2 = string(t2);
labels = get(legend(t2),'string');
legend2 = legend(labels,'location','eastoutside');
title(legend2,'time (min)')
```

Analytical Solution to Shrinking Core Model

```
%-------------------------------------------------------------------------
% scm.m
% kinetic case
%
% this is the analytical solution to the shrinking core model (SCM).
% run after eq_solver and kin_solver for the diffusion and adsorption
% single component case
%
% created by Virginia Lane on 02/28/18
% last updated on 04/07/18
%-------------------------------------------------------------------------

global  kin_q q_max t_i t_s r_i r_s t_p R eq_q c_0 D e_p

[row, column] = min(abs(kin_q(1,:)/q_max-.9));
data = [t_i, r_i+r_s*(column-1)];
for n = 2:1:t_p-1
    [row, column] = min(abs(kin_q(n,:)/q_max-.9));
    data = [data;t_i+t_s*(n-1), r_i+r_s*(column-1)];
    %first column is time, second is radius (at q=50%)
end

[row2, column2] = min(abs(eq_q(1,:)/c_0-.9));
data2 = [t_i, r_i+r_s*(column2-1)];
for n = 2:1:t_p-1
    [row2, column2] = min(abs(eq_q(n,:)/c_0-.9));
    data2 = [data2;t_i+t_s*(n-1), r_i+r_s*(column2-1)];
    %first column is time, second is radius (at q=50%)
end

tortuosity = 2.8
%D_0 = D*tortuosity/e_p
D_0 = 3.7e-7*60

figure(1)
plot(t, 6*D_0*c_0*t/q_max/R^2*(D/D_0),'linewidth',2)
ylim([0 1])
xlabel('time (minutes)')
ylabel('2(p_s)^3-3(p_s)^2+1')
hold on
plot(data(:,1),(2*((data(:,2))/R).^3-
3*((data(:,2))/R).^2+1),data2(:,1),...
    (2*((data2(:,2))/R).^3-3*((data2(:,2))/R).^2+1),'linewidth',2)
%yyaxis right+1
%ylabel('(R-r)/R')
xlim([0 30])
legend('Shrinking core model','kin. case, q/q_m_a_x = 90%',...
    'eq. case, q/q_m_a_x = 90%','location','Northwest')
hold off
```

Adsorption and Diffusion in a Single Chromatographic Bead, Kinetic Case

```
%-------------------------------------------------------------------------
% parameters.m
%
% this file gives parameters for diffusion of mAbs into a sphere
%
% created by Virginia Lane 01/24/18
% finalized 09/15/18
%-------------------------------------------------------------------------
clc                     % clear command window
clear                   % clear all variables
colors;

global m c_0 c_1 D R k_d k_a k_f k_r e_p q_max...
    c_0_2 c_1_2 D_2 R_2 k_d_2 k_a_2 k_f_2 k_r_2 e_p_2 q_max_2...
    r_i r_o t_i t_f t_p t_r t_s r_p r_s r t

% constant parameters
%-------------------------------------------------------------------------
-
m     = 2;              % indicates spherical symmetry
c_0   = 2;              % initial concentration of solution in mg/mL
c_1   = 0;              % initial concentration of sphere in mg/mL
D     = 7.5e-8*60;      % diffusivity constant in (cm^2)/(min)
e_p   = .65;            % interstitial porosity
q_max = 61;             % mAb q_max, mg/mL hydrated resin particles
Dm    = 85e-4;          % diameter of sphere in cm
R     = Dm./2;          % radius of sphere in cm

% reaction constants
%-------------------------------------------------------------------------
k_a   = 61.806;             % k_a = K = k_f/k_r (association constant)
k_d   = 1./k_a;         % dissociaton constant
k_r   = .002;
k_f   = k_a*k_r;

% competitive/noncompetitive binding
%-------------------------------------------------------------------------
c_0_2   = c_0;          % initial concentration of solution in mg/mL
c_1_2   = c_1;          % initial concentration of sphere in mg/mL
D_2     = D;            % diffusivity constant in (cm^2)/(s)
e_p_2   = e_p;          % interstitial porosity
q_max_2 = 84;           % mAb q_max, mg/mL hydrated resin particles
R_2     = R;            % radius of sphere in cm
k_a_2   = k_a*.1;       % k_a K = k_f/k_r (association constant)
k_d_2   = 1/k_a_2;      % dissociaton constant
k_r_2   = k_r;
k_f_2   = k_r_2*k_a_2;

% time parameters
%-------------------------------------------------------------------------
```

```
t_i   = 0;              % initial time point
t_f   = 30;             % final time point in minutes
t_p   = 121;             % number of time points
t_s   = t_f./(t_p-1);   % step size of time points
t = linspace(t_i,t_f,t_p); % time point report-out in minutes

% radius parameters
%--------------------------------------------------------------------------
r_i   = 0;              % inside radius
r_o   = R;              % outside radius
r_p   = 300;            % number of radius points
r_s   = r_o./(r_p-1);   % step size of radius points
r = linspace(r_i,r_o,r_p); % radial point report-out in cm
```

```
%-------------------------------------------------------------------------
% colors.m
%
% gives color profile for graphs in this path
%
% created by Virginia Lane on 02/14/18
% finalized 09/15/18
%-------------------------------------------------------------------------

global mAb_map mAb_map_2 t_p colororder colororder2

% colormap (mAb)
%-------------------------------------------------------------------------
colorinitial = 0;       % specifies black initial color
colorfinal   = .85;
colorstep    = .01;
green        = (colorinitial:colorstep:colorfinal).';
red          = (zeros(1, colorfinal./colorstep+1)).';
blue         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map      = [red,green,blue];

% colormap 2 (for competitive/noncompetitive molecule)
%-------------------------------------------------------------------------
red_2          = (zeros(1, colorfinal./colorstep+1)).';
green_2        = (colorinitial:colorstep:colorfinal).';
blue_2         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_2      = [red_2,green_2,blue_2];

% colororder
%-------------------------------------------------------------------------
colorinitial = 0;        % specifies black initial color
colorfinal   = 1;
colorstep    = (colorfinal-colorinitial)./(t_p-1);
red          = (colorinitial:colorstep:colorfinal).';
green        = (zeros(1, round(colorfinal./colorstep+1))).';
blue         = green;
colororder   = [red,green,blue];

colororder2  = jet;
```

```
%------------------------------------------------------------------------
% kin_ic.m
% kinetic case
%
% pde initial conditions to accompany kin_solver.m
%
% created by Virginia Lane on 02/14/18
% finalized 09/15/18
%------------------------------------------------------------------------
function u_0 = kin_ic(r)

global c_1 k_a q_max

u_0 = [c_1;k_a*q_max*c_1/(1+k_a*c_1)];
```

```
%-------------------------------------------------------------------------
% kin_bc.m
% kinetic case
%
% pde boundary conditions to accompany kin_solver.m
%
% created by Virginia Lane on 02/14/18
% finalized 09/15/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = kin_bc(rl,ul,rr,ur,t)

global c_0 k_a q_max

pl  = [0;0];          % "left" bc is for r = 0 and represents no flux
ql  = [1;1];
pr  = [ur(1)-c_0; ur(2)-k_a*q_max*c_0/(1+k_a*c_0)];  % "right" bc for
qr  = [0;0]; % r = R, represents constant surface concentration and q
```

```
%-------------------------------------------------------------------------
% kin_pde.m
% kinetic case
%
% pde function for diffusion with absorption to accompany kin_solver.m
%
% created by Virginia Lane on 02/20/18
% finalized 09/15/18
%-------------------------------------------------------------------------
function [c,f,s] = kin_pde(r,t,u,DuDr)

global D q_max k_f k_r e_p

c = [e_p; 1];
f = [e_p*D*DuDr(1);0];
s = [-k_f*u(1)*(q_max-u(2))+k_r*u(2); k_f*u(1).*(q_max-u(2))-
k_r*u(2)];
```

```
%-------------------------------------------------------------------------
% kin_solver.m
% kinetic case
%
% solver for kin_pde.m using kin_ic.m and kin_bc.m
%
% created by Virginia Lane on 02/14/18
% finalized 09/15/18
%-------------------------------------------------------------------------
function kin_solver

global m r t kin_u kin_q

sol = pdepe(m,@kin_pde,@kin_ic,@kin_bc,r,t);
kin_u = sol(:,:,1);
kin_q = sol(:,:,2);
```

```
%-------------------------------------------------------------------------
% kin_contourgraphs.m
% kinetic case
%
% graphs 6 concentration contour plots for a spherical cross section
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/15/18
% last updated on 04/07/18
%-------------------------------------------------------------------------

global r t t_i t_f c_1 c_0 kin_u mAb_map

if c_1 > c_0
    disp('this is the desorption case ')
end

figure(1);
for n = 0:1:5                  % always starts at time point 0
    subplot(1,6,n+1)

    n3   = (t_f-t_i)./6;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    minutes         = round(t(column2));
    theta          = 0:.1:33*pi/16;
    [th, rad]      = meshgrid(theta,r);
    X              = rad.*cos(th);
    Y              = rad.*sin(th);
    r_circle       = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2             = kin_u(column2,:);
    u2             = u2.';
    n2             = 0;
    U              = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map);
    axis('square');
    axis off
    title([num2str(minutes) ' min']);
    if c_1 > c_0
        caxis([c_0 c_1])
    else
        caxis([c_1 c_0])
    end
end
```

```
cb = colorbar('location','Manual','position',[.92 .45 .03 .18]);
title(cb,{'mAb conc.','mg/mL'});
```

```
%-------------------------------------------------------------------------
% kin_contourgraphs.m
% kinetic case
%
% graphs 9 concentration contour plots for a spherical cross section
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/15/18
% last updated on 03/28/18
%-------------------------------------------------------------------------

global r t t_i t_f kin_q mAb_map c_1 c_0 q_max

if c_1 > c_0
    disp('this is the desorption case ')
end

figure(1);
for n = 0:1:5                  % always starts at time point 0
    subplot(1,6,n+1)

    n3   = (t_f-t_i)./6;
    time = round(n3.*n);
    [row2, column2] = min(abs(t-time));
    minutes          = round(t(column2));
    theta          = 0:.1:33*pi/16;
    [th, rad]      = meshgrid(theta,r);
    X              = rad.*cos(th);
    Y              = rad.*sin(th);
    r_circle       = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2             = kin_q(column2,:);
    u2             = u2.';
    n2             = 0;
    U              = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map);
    axis('square');
    axis off
    caxis([0 q_max])
    title([num2str(minutes) ' min']);

end
cb = colorbar('location','Manual','position',[.92 .45 .03 .18]);
title(cb,{'q','mg/mL'});
```

```
%-------------------------------------------------------------------------
% kin_q50graph.m
% kinetic case
%
% script that graphs a plot of radius at 50% q value vs. time
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/28/18
% last updated on 04/07/18
%-------------------------------------------------------------------------

global  kin_q q_max t_i t_s r_i r_s t_p R kin_u c_0

[row, column] = min(abs(kin_q(1,:)/q_max-.5));
data = [t_i, r_i+r_s*(column-1)];
for n = 2:1:t_p-1
    [row, column] = min(abs(kin_q(n,:)/q_max-.5));
    data = [data;t_i+t_s*(n-1), r_i+r_s*(column-1)]; %first column is
time, second is radius (at q=50%)
end

[row2, column2] = min(abs(kin_u(1,:)/c_0-.5));
data2 = [t_i, r_i+r_s*(column2-1)];
for n = 2:1:t_p-1
    [row2, column2] = min(abs(kin_u(n,:)/c_0-.5));
    data2 = [data2;t_i+t_s*(n-1), r_i+r_s*(column2-1)]; %first column
is time, second is radius (at q=50%)
end


figure(1);
plot(data(:,1),data(:,2)/R,'-o')
grid on
title('radius vs. time (@ q/q_m_a_x = 50%)')
xlabel('time (minutes)')
xlim([0 30])
ylabel('r/R')
grid off

figure(2);
plot(data(:,1),data(:,2)/R,'-o',data2(:,1),data2(:,2)/R,'-o')
grid on
title('radius vs. time')
xlabel('time (minutes)')
xlim([0 40])
ylabel('r/R')
grid off
legend('q/q_m_a_x = 50%','c/c_0 = 50%')
```

```
%-------------------------------------------------------------------------
% kin_cvrgraph.m
% kinetic case
%
% script that graphs a plot of time curves for concentration vs radius
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/14/18
% last updated on 04/07/18
%-------------------------------------------------------------------------

global  c_0 R t_i t_f r kin_u

figure(2);
plot(r,kin_u)
xlabel('radius (cm)')
ylabel('concentration (mg/mL)')
t2 = round(linspace(t_i,t_f,7));
t2 = string(t2);
labels = get(legend(t2),'string');
legend2 = legend(labels,'location','eastoutside');
title(legend2,'time (min)')
xlim([0 R])
ylim([0 c_0])
```

```
%-------------------------------------------------------------------------
% kin_qvrgraph.m
% kinetic case
%
% script that graphs a plot of time curves for concentration vs radius
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 02/14/18
% last updated on 04/07/18
%-------------------------------------------------------------------------
-

global  q_max R t_i t_f r kin_q

figure(2);
plot(r,kin_q/q_max)
xlabel('radius (cm)')
ylabel('q/q_m_a_x (mg/mL)')
t2 = round(linspace(t_i,t_f,7));
t2 = string(t2);
labels = get(legend(t2),'string');
legend2 = legend(labels,'location','eastoutside');
title(legend2,'time (min)')
xlim([0 R])
ylim([0 1])
```

Gaussian Fitting of Elution Data

```
%--------------------------------------------------------------------
% elutiondata_1.m
%
% imports elution_data_1.xlsx file containing elution data for
% CaptivA PriMab media
% Data from Weinberg et al. 2017, Figure 5a
%
% Created by Virginia Lane 09/12/18
% Edited by TP
%--------------------------------------------------------------------
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: C:\Users\Virginia Lane\Desktop\elution_data_1.xlsx
%    Worksheet: CaptivA PriMAB Figure 2a
%
% To extend the code for use with different selected data or a
different
% spreadsheet, generate a function instead of a script.

% Auto-generated by MATLAB on 2018/09/12 09:49:08
%
% TP edits included 9/16/2018 - I see that you did sort out the
% mis-labelling of the mAb and hIgG data sets in the spreadsheet as
well
% as the reversal in order of the mAb volume and mAb UV signal columns
%
%% clear all variables
clear
clc

%% Import the data
[~, ~, raw] = xlsread('C:\Users\Virginia
Lane\Desktop\elution_data_1.xlsx','CaptivA PriMAB Figure
2a','A3:K874');
stringVectors = string(raw(:,[3,6,9]));
stringVectors(ismissing(stringVectors)) = '';
raw = raw(:,[1,2,4,5,7,8,10,11]);

%% Create output variable
data = reshape([raw{:}],size(raw));

%% Create table
elutiondata1 = table;

%% Allocate imported array to column variable names
elutiondata1.VolumemL = data(:,1);
elutiondata1.UV280 = data(:,2);
elutiondata1.VolumemL1 = data(:,3);
elutiondata1.UV1 = data(:,4);
```

```
elutiondata1.VolumemL2 = data(:,5);
elutiondata1.pH = data(:,6);
elutiondata1.VolumemL3 = data(:,7);
elutiondata1.pH1 = data(:,8);

%% rename variables to correct names
hIgG_mL_1  = elutiondata1.VolumemL;
mAb_UV = elutiondata1.VolumemL1;
mAb_mL_2  = elutiondata1.VolumemL2;
hIgG_mL_2 = elutiondata1.VolumemL3;
hIgG_UV    = elutiondata1.UV280;
mAb_mL_1   = elutiondata1.UV1;
mAb_pH    = elutiondata1.pH;
hIgG_pH   = elutiondata1.pH1;

%% Clear temporary variables
clearvars data raw stringVectors;

%% graph data
figure(1)
plot(mAb_mL_1, mAb_UV,'b')
ylim([0 1])
ylabel('Normalized UV 280')
xlim([40 80])
xlabel('Volume (mL)')
hold on
plot(hIgG_mL_1, hIgG_UV,'r')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(mAb_mL_2, mAb_pH,':b', hIgG_mL_2, hIgG_pH,':r')
hold off

%% mAb gaussian fit
figure(2)

f = fit(mAb_mL_1(300:700), mAb_UV(300:700), 'gauss1')
plot(f,mAb_mL_1(300:700), mAb_UV(300:700))
ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(mAb_mL_2(270:630), mAb_pH(270:630))
hold off
title('Gaussian fit of mAb elution data')
legend('elution data','Gaussian fit','pH')

c_mAb = coeffvalues(f);
```

```
area_mAb = c_mAb(1)*sqrt(pi)*c_mAb(3)
mean_pH_mAb = interp1q(mAb_mL_2(270:630),mAb_pH(270:630),c_mAb(2))
%
%% hIgG Gaussian fit
figure(3)

gauss3wconst = fittype('a1*exp(-((x-b1)/c1)^2)+a2*exp(-((x-
b2)/c2)^2)+a3*exp(-((x-b3)/c3)^2)+d',...
    'independent','x',...

'coefficients',{'a1','b1','c1','a2','b2','c2','a3','b3','c3','d'});
options = fitoptions(fittype(gauss3wconst));
options.StartPoint = [0.5 52 2 0.5 0.5 56.5 0.5 60 0.5,0.01];
options.Lower = [0 50 0.01 0 56 0.01 0 57.5 0.01 0];
options.Upper = [10 54 10 10 57 10 10 63 10 0.03];
f = fit(hIgG_mL_1(300:700), hIgG_UV(300:700),gauss3wconst,options)
plot(f,hIgG_mL_1(300:700), hIgG_UV(300:700))

c_hIgG = coeffvalues(f);
gaussfit1 = c_hIgG(1)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(2))/c_hIgG(3)).^2)+c_hIgG(10);
gaussfit2 = c_hIgG(4)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(5))/c_hIgG(6)).^2)+c_hIgG(10);
gaussfit3 = c_hIgG(7)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(8))/c_hIgG(9)).^2)+c_hIgG(10);
hold on
plot(hIgG_mL_1(270:630),gaussfit1,hIgG_mL_1(270:630),gaussfit2,hIgG_mL
_1(270:630),gaussfit3)

ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(hIgG_mL_2(270:630), hIgG_pH(270:630))

hold off
title('Components of hIgG polyclonal solution from Gaussian fitting')
legend('hIgG elution data','Gaussian
fit','hIgG_4','hIgG_1','hIgG_2','pH')

area_hIgG_1 = c_hIgG(1)*sqrt(pi)*c_hIgG(3)
mean_pH_hIgG_1 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(2))
area_hIgG_2 = c_hIgG(4)*sqrt(pi)*c_hIgG(6)
mean_pH_hIgG_2 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(5))
area_hIgG_3 = c_hIgG(7)*sqrt(pi)*c_hIgG(9)
mean_pH_hIgG_3 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(8))
```

```
figure(4);
gauss3wconst = fittype('a1*exp(-((x-b1)/c1)^2)+a2*exp(-((x-
b2)/c2)^2)+a3*exp(-((x-b3)/c3)^2)+d',...
    'independent','x',...

'coefficients',{'a1','b1','c1','a2','b2','c2','a3','b3','c3','d'});
options = fitoptions(fittype(gauss3wconst));
options.StartPoint = [0.5 52 2 0.5 0.5 56.5 0.5 60 0.5,0.01];
options.Lower = [0 50 0.01 0 56 0.01 0 57.5 0.01 0];
options.Upper = [10 54 10 10 57 10 10 63 10 0.03];
f = fit(hIgG_mL_1(300:700), hIgG_UV(300:700),gauss3wconst,options)
plot(f,hIgG_mL_1(300:700), hIgG_UV(300:700))

ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(hIgG_mL_2(270:630), hIgG_pH(270:630))

hold off
title('Gaussian fit of hIgG elution data')
legend('hIgG elution data','Gaussian fit','pH')
```

```
%--------------------------------------------------------------------
% elutiondata_2.m
%
% imports elution_data_1.xlsx file containing elution data for
% MabSelect media
% Data from Weinberg et al. 2017, Figure 5b
%
% Created by Virginia Lane 09/12/18
% Edited by TP
%--------------------------------------------------------------------
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: C:\Users\Virginia Lane\Desktop\elution_data_1.xlsx
%    Worksheet: CaptivA PriMAB Figure 2a
%
% To extend the code for use with different selected data or a
different
% spreadsheet, generate a function instead of a script.

% Auto-generated by MATLAB on 2018/09/12 09:49:08
%
% TP edits included 9/16/2018 - I see that you did sort out the
% mis-labelling of the mAb and hIgG data sets in the spreadsheet as
well
% as the reversal in order of the mAb volume and mAb UV signal columns
%
%% clear all variables
clear
clc

%% Import the data
[~, ~, raw] = xlsread('C:\Users\Virginia
Lane\Desktop\elution_data_1.xlsx','MabSelect Figure 2b','A3:K874');
stringVectors = string(raw(:,[3,6,9]));
stringVectors(ismissing(stringVectors)) = '';
raw = raw(:,[1,2,4,5,7,8,10,11]);

%% Create output variable
data = reshape([raw{:}],size(raw));

%% Create table
elutiondata1 = table;

%% Allocate imported array to column variable names
elutiondata1.VolumemL = data(:,1);
elutiondata1.UV280 = data(:,2);
elutiondata1.VolumemL1 = data(:,3);
elutiondata1.UV1 = data(:,4);
elutiondata1.VolumemL2 = data(:,5);
elutiondata1.pH = data(:,6);
elutiondata1.VolumemL3 = data(:,7);
```

```
elutiondata1.pH1 = data(:,8);

%% rename variables to correct names
hIgG_mL_1  = elutiondata1.VolumemL1;
mAb_UV = elutiondata1.UV280;
mAb_mL_2  = elutiondata1.VolumemL2;
hIgG_mL_2 = elutiondata1.VolumemL3
hIgG_UV    = elutiondata1.UV1;
mAb_mL_1   = elutiondata1.VolumemL;
mAb_pH    = elutiondata1.pH;
hIgG_pH   = elutiondata1.pH1

%% Clear temporary variables
clearvars data raw stringVectors;

%% graph data
figure(1)
plot(mAb_mL_1, mAb_UV,'b')
ylim([0 1])
ylabel('Normalized UV 280')
xlim([40 80])
xlabel('Volume (mL)')
hold on
plot(hIgG_mL_1, hIgG_UV,'r')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(mAb_mL_2, mAb_pH,':b', hIgG_mL_2, hIgG_pH,':r')
hold off

%% mAb gaussian fit
figure(2)

f = fit(mAb_mL_1(300:700), mAb_UV(300:700), 'gauss1')
plot(f,mAb_mL_1(300:700), mAb_UV(300:700))
ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(mAb_mL_2(270:630), mAb_pH(270:630))
hold off
title('Gaussian fit of mAb elution data')
legend('elution data','Gaussian fit','pH')

c_mAb = coeffvalues(f);
area_mAb = c_mAb(1)*sqrt(pi)*c_mAb(3)
mean_pH_mAb = interp1q(mAb_mL_2(270:630),mAb_pH(270:630),c_mAb(2))
%
```

```
%% hIgG Gaussian fit
figure(3)

gauss3wconst = fittype('a1*exp(-((x-b1)/c1)^2)+a2*exp(-((x-
b2)/c2)^2)+a3*exp(-((x-b3)/c3)^2)+d',...
    'independent','x',...

'coefficients',{'a1','b1','c1','a2','b2','c2','a3','b3','c3','d'});
options = fitoptions(fittype(gauss3wconst));
options.StartPoint = [0.5 52 2 0.5 0.5 56.5 0.5 60 0.5,0.01];
options.Lower = [0 50 0.01 0 56 0.01 0 57.5 0.01 0];
options.Upper = [10 54 10 10 57 10 10 63 10 0.03];
f = fit(hIgG_mL_1(300:700), hIgG_UV(300:700),gauss3wconst,options)
plot(f,hIgG_mL_1(300:700), hIgG_UV(300:700))

c_hIgG = coeffvalues(f);
gaussfit1 = c_hIgG(1)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(2))/c_hIgG(3)).^2)+c_hIgG(10);
gaussfit2 = c_hIgG(4)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(5))/c_hIgG(6)).^2)+c_hIgG(10);
gaussfit3 = c_hIgG(7)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(8))/c_hIgG(9)).^2)+c_hIgG(10);
hold on
plot(hIgG_mL_1(270:630),gaussfit1,hIgG_mL_1(270:630),gaussfit2,hIgG_mL
_1(270:630),gaussfit3)

ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(hIgG_mL_2(270:630), hIgG_pH(270:630))

hold off
title('Components of hIgG polyclonal solution from Gaussian fitting')
legend('hIgG elution data','Gaussian
fit','hIgG_4','hIgG_1','hIgG_2','pH')

area_hIgG_1 = c_hIgG(1)*sqrt(pi)*c_hIgG(3)
mean_pH_hIgG_1 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(2))
area_hIgG_2 = c_hIgG(4)*sqrt(pi)*c_hIgG(6)
mean_pH_hIgG_2 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(5))
area_hIgG_3 = c_hIgG(7)*sqrt(pi)*c_hIgG(9)
mean_pH_hIgG_3 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(8))

figure(4);
```

```
gauss3wconst = fittype('a1*exp(-((x-b1)/c1)^2)+a2*exp(-((x-
b2)/c2)^2)+a3*exp(-((x-b3)/c3)^2)+d',...
    'independent','x',...

'coefficients',{'a1','b1','c1','a2','b2','c2','a3','b3','c3','d'});
options = fitoptions(fittype(gauss3wconst));
options.StartPoint = [0.5 52 2 0.5 0.5 56.5 0.5 60 0.5,0.01];
options.Lower = [0 50 0.01 0 56 0.01 0 57.5 0.01 0];
options.Upper = [10 54 10 10 57 10 10 63 10 0.03];
f = fit(hIgG_mL_1(300:700), hIgG_UV(300:700),gauss3wconst,options)
plot(f,hIgG_mL_1(300:700), hIgG_UV(300:700))

ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(hIgG_mL_2(270:630), hIgG_pH(270:630))

hold off
title('Gaussian fit of hIgG elution data')
legend('hIgG elution data','Gaussian fit','pH')
```

```
%--------------------------------------------------------------------
% elutiondata_3.m
%
% imports elution_data_1.xlsx file containing elution data for
% MabSelect SuRe media
% Data from Weinberg et al. 2017, Figure 5c
%
% Created by Virginia Lane 09/12/18
% Edited by TP
%--------------------------------------------------------------------
%% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: C:\Users\Virginia Lane\Desktop\elution_data_1.xlsx
%    Worksheet: CaptivA PriMAB Figure 2a
%
% To extend the code for use with different selected data or a
different
% spreadsheet, generate a function instead of a script.

% Auto-generated by MATLAB on 2018/09/12 09:49:08
%
% TP edits included 9/16/2018 - I see that you did sort out the
% mis-labelling of the mAb and hIgG data sets in the spreadsheet as
well
% as the reversal in order of the mAb volume and mAb UV signal columns
%
%% clear all variables
clear
clc

%% Import the data
[~, ~, raw] = xlsread('C:\Users\Virginia
Lane\Desktop\elution_data_1.xlsx','MabSelect SuRe Figure
2c','A3:K874');
stringVectors = string(raw(:,[3,6,9]));
stringVectors(ismissing(stringVectors)) = '';
raw = raw(:,[1,2,4,5,7,8,10,11]);

%% Create output variable
data = reshape([raw{:}],size(raw));

%% Create table
elutiondata1 = table;

%% Allocate imported array to column variable names
elutiondata1.VolumemL = data(:,1);
elutiondata1.UV280 = data(:,2);
elutiondata1.VolumemL1 = data(:,3);
elutiondata1.UV1 = data(:,4);
elutiondata1.VolumemL2 = data(:,5);
elutiondata1.pH = data(:,6);
```

```
elutiondata1.VolumemL3 = data(:,7);
elutiondata1.pH1 = data(:,8);

%% rename variables to correct names
hIgG_mL_1  = elutiondata1.VolumemL1;
mAb_UV = elutiondata1.UV280;
mAb_mL_2  = elutiondata1.VolumemL2;
hIgG_mL_2 = elutiondata1.VolumemL3
hIgG_UV    = elutiondata1.UV1;
mAb_mL_1   = elutiondata1.VolumemL;
mAb_pH    = elutiondata1.pH;
hIgG_pH   = elutiondata1.pH1

%% Clear temporary variables
clearvars data raw stringVectors;

%% graph data
figure(1)
plot(mAb_mL_1, mAb_UV,'b')
ylim([0 1])
ylabel('Normalized UV 280')
xlim([40 80])
xlabel('Volume (mL)')
hold on
plot(hIgG_mL_1, hIgG_UV,'r')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(mAb_mL_2, mAb_pH,':b', hIgG_mL_2, hIgG_pH,':r')
hold off

%% mAb gaussian fit
figure(2)

f = fit(mAb_mL_1(300:700), mAb_UV(300:700), 'gauss1')
plot(f,mAb_mL_1(300:700), mAb_UV(300:700))
ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(mAb_mL_2(270:630), mAb_pH(270:630))
hold off
title('Gaussian fit of mAb elution data')
legend('elution data','Gaussian fit','pH')

c_mAb = coeffvalues(f);
area_mAb = c_mAb(1)*sqrt(pi)*c_mAb(3)
mean_pH_mAb = interp1q(mAb_mL_2(270:630),mAb_pH(270:630),c_mAb(2))
```

```
%% hIgG Gaussian fit
figure(3)

gauss3wconst = fittype('a1*exp(-((x-b1)/c1)^2)+a2*exp(-((x-
b2)/c2)^2)+a3*exp(-((x-b3)/c3)^2)+d',...
    'independent','x',...

'coefficients',{'a1','b1','c1','a2','b2','c2','a3','b3','c3','d'});
options = fitoptions(fittype(gauss3wconst));
options.StartPoint = [0.5 52 2 0.5 0.5 56.5 0.5 60 0.5,0.01];
options.Lower = [0 50 0.01 0 56 0.01 0 57.5 0.01 0];
options.Upper = [10 54 10 10 57 10 10 63 10 0.03];
f = fit(hIgG_mL_1(300:700), hIgG_UV(300:700),gauss3wconst,options)
plot(f,hIgG_mL_1(300:700), hIgG_UV(300:700))

c_hIgG = coeffvalues(f);
gaussfit1 = c_hIgG(1)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(2))/c_hIgG(3)).^2)+c_hIgG(10);
gaussfit2 = c_hIgG(4)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(5))/c_hIgG(6)).^2)+c_hIgG(10);
gaussfit3 = c_hIgG(7)*exp(-((hIgG_mL_1(270:630)-
c_hIgG(8))/c_hIgG(9)).^2)+c_hIgG(10);
hold on
plot(hIgG_mL_1(270:630),gaussfit1,hIgG_mL_1(270:630),gaussfit2,hIgG_mL
_1(270:630),gaussfit3)

ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(hIgG_mL_2(270:630), hIgG_pH(270:630))

hold off
title('Components of hIgG polyclonal solution from Gaussian fitting')
legend('hIgG elution data','Gaussian
fit','hIgG_4','hIgG_1','hIgG_2','pH')

area_hIgG_1 = c_hIgG(1)*sqrt(pi)*c_hIgG(3)
mean_pH_hIgG_1 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(2))
area_hIgG_2 = c_hIgG(4)*sqrt(pi)*c_hIgG(6)
mean_pH_hIgG_2 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(5))
area_hIgG_3 = c_hIgG(7)*sqrt(pi)*c_hIgG(9)
mean_pH_hIgG_3 =
interp1q(hIgG_mL_2(270:630),hIgG_pH(270:630),c_hIgG(8))

figure(4);
```

```
gauss3wconst = fittype('a1*exp(-((x-b1)/c1)^2)+a2*exp(-((x-
b2)/c2)^2)+a3*exp(-((x-b3)/c3)^2)+d',...
    'independent','x',...

'coefficients',{'a1','b1','c1','a2','b2','c2','a3','b3','c3','d'});
options = fitoptions(fittype(gauss3wconst));
options.StartPoint = [0.5 52 2 0.5 0.5 56.5 0.5 60 0.5,0.01];
options.Lower = [0 50 0.01 0 56 0.01 0 57.5 0.01 0];
options.Upper = [10 54 10 10 57 10 10 63 10 0.03];
f = fit(hIgG_mL_1(300:700), hIgG_UV(300:700),gauss3wconst,options)
plot(f,hIgG_mL_1(300:700), hIgG_UV(300:700))

ylim([0 1])
ylabel('Normalized UV 280')
xlabel('Volume (mL)')
hold on
yyaxis right
ylim([2 6])
ylabel('pH')
plot(hIgG_mL_2(270:630), hIgG_pH(270:630))

hold off
title('Gaussian fit of hIgG elution data')
legend('hIgG elution data','Gaussian fit','pH')
```

Competitive Binding Behavior in a Single Chromatographic Bead, Equilibrium Case

```
%-------------------------------------------------------------------------
% parameters.m
%
% these are the parameters that don't change based on time
% run this file first!!
%
% created by Virginia Lane 05/24/18
% last updated on 05/30/18
%-------------------------------------------------------------------------
clc                     % clear command window
clear                   % clear all variables
colors;

global m R r r_i r_i_2 r_o r_p r_p_2 r_s r_s_2... % radius
    D_1 k_d_1 k_a_1 k_f_1 k_r_1 e_p_1 q_max_1...  % mAb
    D_2 k_d_2 k_a_2 k_f_2 k_r_2 e_p_2 q_max_2...  % moderate IgG
    D_3 k_d_3 k_a_3 k_f_3 k_r_3 e_p_3 q_max_3...  % weak IgG
    D_4 k_d_4 k_a_4 k_f_4 k_r_4 e_p_4 q_max_4...  % strong IgG


% constant parameters
%-------------------------------------------------------------------------
m     = 2;              % indicates spherical symmetry
Dm    = 85e-4;          % diameter of sphere in cm
R     = Dm./2;          % radius of sphere in cm

% mAb constants
%-------------------------------------------------------------------------
D_1     = 7.5e-8*60;      % diffusivity constant in (cm^2)/(min)
e_p_1   = .65;            % accessible interstitial porosity
q_max_1 = 61;             % mAb q_max, mg/mL hydrated resin
k_a_1   = 61.806;         % k_a = K = k_f/k_r (association constant)
k_d_1   = 1./k_a_1;       % dissociaton constant
k_r_1   = .002;
k_f_1   = k_r_1*k_a_1;

% moderate IgG constants
%-------------------------------------------------------------------------
D_2     = 7.5e-8*60;
e_p_2   = .65;
q_max_2 = 50.1;
k_a_2   = 76.311;
k_d_2   = 1./k_a_2;
k_r_2   = .002;
k_f_2   = k_r_2*k_a_2;

% weak IgG constants
%-------------------------------------------------------------------------
D_3     = 7.5e-8*60;
e_p_3   = .65;
```

```
q_max_3 = 15.1;
k_a_3   = 22.947;
k_d_3   = 1./k_a_3;
k_r_3   = .002;
k_f_3   = k_r_3*k_a_3;

% strong IgG constants
%-------------------------------------------------------------------------
D_4     = 7.5e-8*60;
e_p_4   = .65;
q_max_4 = 84;
k_a_4   = 127.96;
k_d_4   = 1./k_a_4;
k_r_4   = .002;
k_f_4   = k_r_4*k_a_4;


% radius parameters
%-------------------------------------------------------------------------
r_i   = 0;              % inside radius
r_i_2 = abs(R-.0005);   % for generating more time points near edge
r_o   = R;              % outside radius
r_p   = 50;            % number of radius points
r_p_2 = 20;             % number of radius points near edge
r_s   = (r_i_2-r_i)/(r_p-1);   % step size of radius points
r_s_2 = (r_o-r_i_2)/(r_p_2-1); % step size of radius points near edge
r = [linspace(r_i,r_i_2,r_p) linspace(r_i_2+r_s_2,r_o,r_p_2)];
```

```
%-------------------------------------------------------------------------
% colors.m
%
% gives color profile for graphs in this path
%
% created by Virginia Lane on 05/13/18
% last updated on 10/12/18
%-------------------------------------------------------------------------

global mAb_map_1 mAb_map_2 mAb_map_3 mAb_map_4...
    color_1 color_2 color_3 color_4

% colormap 1 (green, for mAb)
%-------------------------------------------------------------------------
colorinitial = 0;       % specifies black initial color
colorfinal   = 1;
colorstep    = .01;
green        = (colorinitial:colorstep:colorfinal).';
red          = (zeros(1, colorfinal./colorstep+1)).';
blue         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_1    = [red,green,blue];
color_1      = [0 colorfinal 0];

% colormap 2 (red, for moderate IgG)
%-------------------------------------------------------------------------
green_2        = (zeros(1, colorfinal./colorstep+1)).';
red_2          = (colorinitial:colorstep:colorfinal).';
blue_2         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_2      = [red_2,green_2,blue_2];
color_2        = [colorfinal*2 0 0];

% colormap 3 (yellow, for weak IgG)
%-------------------------------------------------------------------------
red_3          = (colorinitial:colorstep:colorfinal).';
green_3        = (colorinitial:colorstep:colorfinal).';
blue_3         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_3      = [red_3,green_3,blue_3];
color_3        = [colorfinal colorfinal 0];

% colormap 4 (blue, for strong IgG)
%-------------------------------------------------------------------------
red_4          = (zeros(1, colorfinal./colorstep+1)).';
green_4        = (zeros(1, colorfinal./colorstep+1)).';
blue_4         = (colorinitial:colorstep:colorfinal).';
mAb_map_4      = [red_4,green_4,blue_4];
color_4        = [0 0 colorfinal];
```

```
%-------------------------------------------------------------------------
% eq_solver.m
% equilibrium, 1 component case
%
% solves phase I and phase II of the process
%
% created by Virginia Lane on 05/13/18
% last updated on 10/15/18
%-------------------------------------------------------------------------

parameters;           % runs the parameters not dependant on time
eq_parameters_I;      % runs the parameters for phase I of the process
eq_solver_I;          % solves phase I of the process
eq_parameters_II;     % runs the new parameters for phase II of the
process
eq_solver_II;         % solves phase II of the process
```

```
%-------------------------------------------------------------------------
% eq_parameters_I.m
%
% these are the parameters for phase I
%
% created by Virginia Lane 05/24/18
% last updated on 10/15/18
%-------------------------------------------------------------------------

global c_0_1_I c_1_1_I...            % mAb parameters
    c_0_2_I c_1_2_I...               % IgG parameters
    c_0_3_I c_1_3_I...               % weak IgG parameters
    c_0_4_I c_1_4_I...               % strong IgG parameters
    t t_I t_i t_f_I t_p_I t_s_I      % time parameters

% mAb parameters
%-------------------------------------------------------------------------
c_0_1_I   = 2;              % initial concentration of solution in
mg/mL
c_1_1_I   = 0;              % initial concentration of sphere in mg/mL

% moderate IgG parameters
%-------------------------------------------------------------------------
c_0_2_I   = 0;              % initial concentration of solution in
mg/mL
c_1_2_I   = 0;              % initial concentration of sphere in mg/mL

% weak IgG parameters
%-------------------------------------------------------------------------
c_0_3_I   = 0;              % initial concentration of solution in
mg/mL
c_1_3_I   = 0;              % initial concentration of sphere in mg/mL

% strong IgG parameters
%-------------------------------------------------------------------------
c_0_4_I   = 0;              % initial concentration of solution in
mg/mL
c_1_4_I   = 0;              % initial concentration of sphere in mg/mL


% time parameters
%-------------------------------------------------------------------------
t_i   = 0;                  % initial time point
t_f_I = 15;                 % final time point of phase I in minutes
t_p_I = 61;                 % number of time points in phase I
t_s_I = t_f_I./(t_p_I-1);       % step size of time points in phase I
t_I = linspace(t_i,t_f_I,t_p_I); % time point report-out in minutes
t   = t_I;
```

```
%-------------------------------------------------------------------------
% eq_ic_I.m
% equilibrium case
%
% pde initial conditions to accompany eq_solver_I.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function u_0 = eq_ic_I(r)

global c_1_1_I c_1_2_I c_1_3_I c_1_4_I

u_0 = [c_1_1_I; c_1_2_I; c_1_3_I; c_1_4_I];
```

```
%-------------------------------------------------------------------------
% eq_bc_I.m
% equilibrium case
%
% pde boundary conditions to accompany eq_solver_I.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = eq_bc_I(rl,ul,rr,ur,t_I)

global c_0_1_I c_0_2_I c_0_3_I c_0_4_I

pl  = [0;0;0;0];          % "left" bc is for r = 0
ql  = [1;1;1;1];          % and represents no flux
pr  = [ur(1)-c_0_1_I; ur(2)-c_0_2_I;...
    ur(3)-c_0_3_I; ur(4)-c_0_4_I]; % "right" bc for
qr  = [0;0;0;0]; % r = R
                % and represents constant surface concentration and q
```

```
%-------------------------------------------------------------------------
% eq_pde_I.m
% equilibrium case
%
% pde function to accompany eq_solver_I.m
%
% created by Virginia Lane on 02/20/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function [c,f,s] = eq_pde_I(r,t_I,u,DuDr)

global D_1 q_max_1 e_p_1 k_a_1...
    D_2 q_max_2 e_p_2 k_a_2...
    D_3 q_max_3 e_p_3 k_a_3...
    D_4 q_max_4 e_p_4 k_a_4

DqDu(1) = k_a_1*q_max_1*(1+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;
DqDu(2) = k_a_2*q_max_2*(1+k_a_1*u(1)+k_a_3*u(3)+k_a_4*u(4))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;
DqDu(3) = k_a_3*q_max_3*(1+k_a_2*u(2)+k_a_1*u(1)+k_a_4*u(4))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;
DqDu(4) = k_a_4*q_max_4*(1+k_a_2*u(2)+k_a_3*u(3)+k_a_1*u(1))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;

c = [e_p_1+DqDu(1); e_p_2+DqDu(2); e_p_3+DqDu(3); e_p_4+DqDu(4)];
f = [e_p_1*D_1*DuDr(1); e_p_2*D_2*DuDr(2);...
    e_p_3*D_3*DuDr(3); e_p_4*D_4*DuDr(4)];
s = [0;0;0;0];
```

```
%-------------------------------------------------------------------------
% eq_solver_I.m
% equilibrium case
%
% solver for eq_pde_I.m using eq_ic_I.m and eq_bc_I.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function eq_solver_I

global m r t_I...
    eq_u_1 eq_u_1_I eq_q_1 eq_q_1_I k_a_1 q_max_1...
    eq_u_2 eq_u_2_I eq_q_2 eq_q_2_I k_a_2 q_max_2...
    eq_u_3 eq_u_3_I eq_q_3 eq_q_3_I k_a_3 q_max_3...
    eq_u_4 eq_u_4_I eq_q_4 eq_q_4_I k_a_4 q_max_4

sol = pdepe(m,@eq_pde_I,@eq_ic_I,@eq_bc_I,r,t_I);
eq_u_1_I = sol(:,:,1);
eq_u_2_I = sol(:,:,2);
eq_u_3_I = sol(:,:,3);
eq_u_4_I = sol(:,:,4);
eq_q_1_I = k_a_1*q_max_1*eq_u_1_I./...
    (1+k_a_1*eq_u_1_I+k_a_2*eq_u_2_I+k_a_3*eq_u_3_I+k_a_4*eq_u_4_I);
eq_q_2_I = k_a_2*q_max_2*eq_u_2_I./...
    (1+k_a_1*eq_u_1_I+k_a_2*eq_u_2_I+k_a_3*eq_u_3_I+k_a_4*eq_u_4_I);
eq_q_3_I = k_a_3*q_max_3*eq_u_3_I./...
    (1+k_a_1*eq_u_1_I+k_a_2*eq_u_2_I+k_a_3*eq_u_3_I+k_a_4*eq_u_4_I);
eq_q_4_I = k_a_4*q_max_4*eq_u_4_I./...
    (1+k_a_1*eq_u_1_I+k_a_2*eq_u_2_I+k_a_3*eq_u_3_I+k_a_4*eq_u_4_I);

eq_u_1 = eq_u_1_I;
eq_u_2 = eq_u_2_I;
eq_u_3 = eq_u_3_I;
eq_u_4 = eq_u_4_I;
eq_q_1 = eq_q_1_I;
eq_q_2 = eq_q_2_I;
eq_q_3 = eq_q_3_I;
eq_q_4 = eq_q_4_I;
```

```
%-------------------------------------------------------------------------
% eq_parameters_II.m
%
% these are the parameters for phase I
%
% created by Virginia Lane 05/24/18
% last updated on 10/15/18
%-------------------------------------------------------------------------

global c_0_1_II...              % mAb parameters
    c_0_2_II...                 % IgG parameters
    c_0_3_II...                 % weak IgG parameters
    c_0_4_II...                 % strong IgG parameters
    t t_I t_II t_i_II t_f_I t_f_II t_p_II     % time parameters

% mAb parameters
%-------------------------------------------------------------------------
c_0_1_II   = 0;
% initial concentration of solution in mg/mL

% moderate IgG parameters
%-------------------------------------------------------------------------
c_0_2_II   = 2*.4972;
% phase II initial concentration of solution in mg/mL

% weak IgG parameters
%-------------------------------------------------------------------------
c_0_3_II   = 2*.1386;
% phase II initial concentration of solution in mg/mL

% strong IgG parameters
%-------------------------------------------------------------------------
c_0_4_II   = 2*.3642;
% phase II initial concentration of solution in mg/mL


% time parameters
%-------------------------------------------------------------------------
t_i_II   = t_f_I+.005;          % initial time point
t_f_II = 240;                   % final time point of phase I in minutes
t_p_II = 240;                   % number of time points in phase I
t_t_II = t_f_I+t_f_II;          % total process time
t_s_II = t_f_II./(t_p_II-1);    % step size of time points in phase II
t_II   = linspace(t_i_II,t_t_II,t_p_II); % time point report-out
t      = [t_I t_II];
```

```
%-------------------------------------------------------------------------
% eq_ic_II.m
% equilibrium case
%
% pde initial conditions to accompany eq_solver_II.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function u_0 = eq_ic_II(r2)

global r eq_u_1_I eq_u_2_I eq_u_3_I eq_u_4_I

u_0 = [interp1(r, eq_u_1_I(end,:), r2, 'spline');...
    interp1(r, eq_u_2_I(end,:), r2, 'spline');...
    interp1(r, eq_u_3_I(end,:), r2, 'spline');...
    interp1(r, eq_u_4_I(end,:), r2, 'spline')];
```

```
%-------------------------------------------------------------------------
% eq_bc_II.m
% equilibrium case
%
% pde boundary conditions to accompany eq_solver_II.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = eq_bc_II(rl,ul,rr,ur,t_II)

global c_0_1_II c_0_2_II c_0_3_II c_0_4_II

pl  = [0;0;0;0];         % "left" bc is for r = 0 and represents no
flux
ql  = [1;1;1;1];
pr  = [ur(1)-c_0_1_II; ur(2)-c_0_2_II;...
    ur(3)-c_0_3_II; ur(4)-c_0_4_II]; % "right" bc for
qr  = [0;0;0;0]; % r = R, represents constant surface concentration
and q
```

```
%-------------------------------------------------------------------------
% eq_pde_II.m
% equilibrium case
%
% pde function to accompany eq_solver_II.m
%
% created by Virginia Lane on 02/20/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function [c,f,s] = eq_pde_II(r,t_II,u,DuDr)

global D_1 q_max_1 e_p_1 k_a_1...
    D_2 q_max_2 e_p_2 k_a_2...
    D_3 q_max_3 e_p_3 k_a_3...
    D_4 q_max_4 e_p_4 k_a_4

DqDu(1) = k_a_1*q_max_1*(1+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;
DqDu(2) = k_a_2*q_max_2*(1+k_a_1*u(1)+k_a_3*u(3)+k_a_4*u(4))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;
DqDu(3) = k_a_3*q_max_3*(1+k_a_2*u(2)+k_a_1*u(1)+k_a_4*u(4))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;
DqDu(4) = k_a_4*q_max_4*(1+k_a_2*u(2)+k_a_3*u(3)+k_a_1*u(1))./...
    (1+k_a_1*u(1)+k_a_2*u(2)+k_a_3*u(3)+k_a_4*u(4)).^2;

c = [e_p_1+DqDu(1); e_p_2+DqDu(2); e_p_3+DqDu(3); e_p_4+DqDu(4)];
f = [e_p_1*D_1*DuDr(1); e_p_2*D_2*DuDr(2);...
    e_p_3*D_3*DuDr(3); e_p_4*D_4*DuDr(4)];
s = [0;0;0;0];
```

```
%-------------------------------------------------------------------------
% eq_solver_II.m
% equilibrium case
%
% solver for eq_pde_II.m using eq_ic_II.m and eq_bc_II.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/15/18
%-------------------------------------------------------------------------
function eq_solver_II

global m r t_II...
    eq_u_1 eq_u_1_I eq_u_1_II eq_q_1 eq_q_1_I eq_q_1_II k_a_1
q_max_1...
    eq_u_2 eq_u_2_I eq_u_2_II eq_q_2 eq_q_2_I eq_q_2_II k_a_2
q_max_2...
    eq_u_3 eq_u_3_I eq_u_3_II eq_q_3 eq_q_3_I eq_q_3_II k_a_3
q_max_3...
    eq_u_4 eq_u_4_I eq_u_4_II eq_q_4 eq_q_4_I eq_q_4_II k_a_4 q_max_4

sol = pdepe(m,@eq_pde_II,@eq_ic_II,@eq_bc_II,r,t_II);
eq_u_1_II = sol(:,:,1);
eq_u_2_II = sol(:,:,2);
eq_u_3_II = sol(:,:,3);
eq_u_4_II = sol(:,:,4);
eq_q_1_II = k_a_1*q_max_1*eq_u_1_II./...

(1+k_a_1*eq_u_1_II+k_a_2*eq_u_2_II+k_a_3*eq_u_3_II+k_a_4*eq_u_4_II);
eq_q_2_II = k_a_2*q_max_2*eq_u_2_II./...

(1+k_a_1*eq_u_1_II+k_a_2*eq_u_2_II+k_a_3*eq_u_3_II+k_a_4*eq_u_4_II);
eq_q_3_II = k_a_3*q_max_3*eq_u_3_II./...

(1+k_a_1*eq_u_1_II+k_a_2*eq_u_2_II+k_a_3*eq_u_3_II+k_a_4*eq_u_4_II);
eq_q_4_II = k_a_4*q_max_4*eq_u_4_II./...

(1+k_a_1*eq_u_1_II+k_a_2*eq_u_2_II+k_a_3*eq_u_3_II+k_a_4*eq_u_4_II);

eq_u_1 = [eq_u_1_I; eq_u_1_II];
eq_u_2 = [eq_u_2_I; eq_u_2_II];
eq_u_3 = [eq_u_3_I; eq_u_3_II];
eq_u_4 = [eq_u_4_I; eq_u_4_II];
eq_q_1 = [eq_q_1_I; eq_q_1_II];
eq_q_2 = [eq_q_2_I; eq_q_2_II];
eq_q_3 = [eq_q_3_I; eq_q_3_II];
eq_q_4 = [eq_q_4_I; eq_q_4_II];
```

```
%-------------------------------------------------------------------------
% kin_qcontourgraphs.m
% kinetic, competitive binding X 5 case
%
% graphs 9 concentration contour plots for a spherical cross section
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 05/14/18
% last updated on 05/24/18
%-------------------------------------------------------------------------

global t_II r mAb_map_1 mAb_map_2 mAb_map_3 mAb_map_4...
    eq_q_1_II eq_q_2_II eq_q_3_II eq_q_4_II

% mAb
%-------------------------------------------------------------------------
figure(1);
for n = 1:1:5                % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta         = 0:.1:33*pi/16;
    [th, rad]     = meshgrid(theta,r);
    X             = rad.*cos(th);
    Y             = rad.*sin(th);
    r_circle      = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2            = eq_q_1_II(column2,:);
    u2            = u2.';
    n2            = 0;
    U             = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_1);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(eq_q_1_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'mAb','q'});

% IgG
%-------------------------------------------------------------------------
figure(2);
```

```
for n = 1:1:5                  % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta         = 0:.1:33*pi/16;
    [th, rad]     = meshgrid(theta,r);
    X             = rad.*cos(th);
    Y             = rad.*sin(th);
    r_circle      = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2            = eq_q_2_II(column2,:);
    u2            = u2.';
    n2            = 0;
    U             = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_2);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(eq_q_2_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'IgG','q'});

% weak IgG
%-------------------------------------------------------------------------
figure(3);
for n = 1:1:5                  % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta         = 0:.1:33*pi/16;
    [th, rad]     = meshgrid(theta,r);
    X             = rad.*cos(th);
    Y             = rad.*sin(th);
    r_circle      = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2            = eq_q_3_II(column2,:);
    u2            = u2.';
    n2            = 0;
    U             = u2;
    for n2 = 1:1:column-1
```

```
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_3);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(eq_q_3_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'weak','IgG','q'});

% Strong IgG
%-------------------------------------------------------------------------
figure(4);
for n = 1:1:5                  % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta          = 0:.1:33*pi/16;
    [th, rad]      = meshgrid(theta,r);
    X              = rad.*cos(th);
    Y              = rad.*sin(th);
    r_circle       = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2             = eq_q_4_II(column2,:);
    u2             = u2.';
    n2             = 0;
    U              = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_4);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(eq_q_4_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'strong','IgG','q'});

% saving figures
%-------------------------------------------------------------------------
```

```
saveas(figure(1),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\mAb_qcontour.jpg']);
saveas(figure(2),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\IgG_qcontour.jpg']);
saveas(figure(3),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\weakIgG_qcontour.jpg']);
saveas(figure(4),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\strongIgG_qcontour.jpg']);
```

```
%-------------------------------------------------------------------------
% kin_cvrgraphs.m
% kinetic, competitive binding case
%
% plots profiles for concentration vs radius at select times
%
% created by Virginia Lane on 05/14/18
% last updated on 05/24/18
%-------------------------------------------------------------------------

global t_II r R...
    eq_u_1_II eq_u_2_II eq_u_3_II eq_u_4_II

for n = 1:1:5
    figure(n);
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row, column] = min(abs(t_II-time));
    u1            = eq_u_1_II(column,:);
    u2            = eq_u_2_II(column,:);
    u3            = eq_u_3_II(column,:);
    u4            = eq_u_4_II(column,:);
    plot(r, u1,'g', r, u2,'r', r, u3, 'y', r, u4, 'b',...
        -r, u1,'g', -r, u2,'r', -r, u3, 'y', -r, u4,
'b','linewidth',2)

    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-R R])
    title(['t = ' num2str(time-15),' minutes'])
    %legend('mAb', 'IgG', 'weak IgG', 'strong IgG',
'location','northwest')
end

% all lines on one graph
%-------------------------------------------------------------------------
figure(6)
n3   = [16,30,45,75,255];
for n = 1:1:5
    time = n3(n);
    [row, column] = min(abs(t_II-time));
    u1            = eq_u_1_II(column,:);
    u2            = eq_u_2_II(column,:);
    u3            = eq_u_3_II(column,:);
    u4            = eq_u_4_II(column,:);

    figure(7)
    plot(r, u1, 'g', -r, u1,'g')
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-R R])
    title('mAb')
```

```
    hold on

    figure(8)
    plot(r, u2, 'r', -r, u2,'r')
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-R R])
    title('IgG')
    hold on

    figure(9)
    plot(r, u3, 'y', -r, u3,'y')
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-R R])
    title('weak IgG')
    hold on

    figure(10)
    plot(r, u4, 'b', -r, u4,'b')
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-R R])
    title('strong IgG')
    hold on

    figure(6)
    plot(r, u1, 'g', r, u2, 'r', r, u3, 'y', r, u4, 'b',...
        -r, u1,'g', -r, u2,'r', -r, u3, 'y', -r, u4, 'b')
    hold on
end
hold off

xlabel('Radius (cm)')
ylabel('q (mg/mL)')
xlim([-R R])
title('q vs radius')
%legend('mAb', 'IgG', 'weak IgG', 'strong IgG',
'location','northwest')
set(gca,'fontsize',16)

% saving figures
%-------------------------------------------------------------------------
saveas(figure(1),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_1.jpg']);
saveas(figure(2),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_15.jpg']);
saveas(figure(3),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_30.jpg']);
saveas(figure(4),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_60.jpg']);
```

```
saveas(figure(5),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_240.jpg']);
saveas(figure(6),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_all.jpg']);
saveas(figure(7),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_mAb.jpg']);
saveas(figure(8),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_IgG.jpg']);
saveas(figure(9),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_weakIgG.jpg']);
saveas(figure(10),['C:\users\Virginia
Lane\Desktop\figures\eq_4comp\cvrgraph_strongIgG.jpg']);
```

Competitive Binding Behavior in a Single Chromatographic Bead, Kinetic Case

```
%-------------------------------------------------------------------------
% kin_solver.m
%
% kinetic, competitive binding X 4 case
% component 1: mAb
% component 2: moderate IgG
% component 3: weak IgG
% component 4: strong IgG
%
% this file solves phase I and phase II of the process
%
% created by Virginia Lane on 05/13/18
% last updated on 10/12/18
%-------------------------------------------------------------------------

parameters;          % runs the parameters not dependent on time
kin_parameters_I;    % runs the parameters for phase I
kin_solver_I;        % solves phase I of the process
kin_parameters_II;   % runs the new parameters for phase II
kin_solver_II;       % solves phase II of the process
```

```
%-------------------------------------------------------------------------
% parameters.m
%
% these are the parameters that don't change based on time
% run this file first!!
%
% created by Virginia Lane 05/24/18
% last updated on 05/30/18
%-------------------------------------------------------------------------
clc                     % clear command window
clear                   % clear all variables
colors;

global m R r r_i r_i_2 r_o r_p r_p_2 r_s r_s_2... % radius
    D_1 k_d_1 k_a_1 k_f_1 k_r_1 e_p_1 q_max_1...  % mAb
    D_2 k_d_2 k_a_2 k_f_2 k_r_2 e_p_2 q_max_2...  % moderate IgG
    D_3 k_d_3 k_a_3 k_f_3 k_r_3 e_p_3 q_max_3...  % weak IgG
    D_4 k_d_4 k_a_4 k_f_4 k_r_4 e_p_4 q_max_4...  % strong IgG


% constant parameters
%-------------------------------------------------------------------------
m     = 2;              % indicates spherical symmetry
Dm    = 85e-4;          % diameter of sphere in cm
R     = Dm./2;          % radius of sphere in cm

% mAb constants
%-------------------------------------------------------------------------
D_1     = 7.5e-8*60;      % diffusivity constant in (cm^2)/(min)
e_p_1   = .65;            % accessible interstitial porosity
q_max_1 = 61;             % mAb q_max, mg/mL hydrated resin
k_a_1   = 61.806;         % k_a = K = k_f/k_r (association constant)
k_d_1   = 1./k_a_1;       % dissociaton constant
k_r_1   = .002;
k_f_1   = k_r_1*k_a_1;

% moderate IgG constants
%-------------------------------------------------------------------------
D_2     = 7.5e-8*60;
e_p_2   = .65;
q_max_2 = 50.1;
k_a_2   = 76.311;
k_d_2   = 1./k_a_2;
k_r_2   = .002;
k_f_2   = k_r_2*k_a_2;

% weak IgG constants
%-------------------------------------------------------------------------
D_3     = 7.5e-8*60;
e_p_3   = .65;
q_max_3 = 15.1;
k_a_3   = 22.947;
```

```
k_d_3   = 1./k_a_3;
k_r_3   = .002;
k_f_3   = k_r_3*k_a_3;

% strong IgG constants
%-------------------------------------------------------------------------
D_4     = 7.5e-8*60;
e_p_4   = .65;
q_max_4 = 84;
k_a_4   = 127.96;
k_d_4   = 1./k_a_4;
k_r_4   = .002;
k_f_4   = k_r_4*k_a_4;


% radius parameters
%-------------------------------------------------------------------------
r_i   = 0;              % inside radius
r_i_2 = abs(R-.0005);   % for generating more time points near edge
r_o   = R;              % outside radius
r_p   = 500;            % number of radius points
r_p_2 = 200;             % number of radius points near edge
r_s   = (r_i_2-r_i)/(r_p-1);   % step size of radius points
r_s_2 = (r_o-r_i_2)/(r_p_2-1); % step size of radius points near edge
r = [linspace(r_i,r_i_2,r_p) linspace(r_i_2+r_s_2,r_o,r_p_2)];
```

```
%-------------------------------------------------------------------------
% colors.m
%
% gives color profile for graphs in this path
%
% created by Virginia Lane on 05/13/18
% last updated on 10/12/18
%-------------------------------------------------------------------------

global mAb_map_1 mAb_map_2 mAb_map_3 mAb_map_4...
    color_1 color_2 color_3 color_4

% colormap 1 (green, for mAb)
%-------------------------------------------------------------------------
colorinitial = 0;       % specifies black initial color
colorfinal   = 1;
colorstep    = .01;
green        = (colorinitial:colorstep:colorfinal).';
red          = (zeros(1, colorfinal./colorstep+1)).';
blue         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_1    = [red,green,blue];
color_1      = [0 colorfinal 0];

% colormap 2 (red, for moderate IgG)
%-------------------------------------------------------------------------
green_2        = (zeros(1, colorfinal./colorstep+1)).';
red_2          = (colorinitial:colorstep:colorfinal).';
blue_2         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_2      = [red_2,green_2,blue_2];
color_2        = [colorfinal*2 0 0];

% colormap 3 (yellow, for weak IgG)
%-------------------------------------------------------------------------
red_3          = (colorinitial:colorstep:colorfinal).';
green_3        = (colorinitial:colorstep:colorfinal).';
blue_3         = (zeros(1, colorfinal./colorstep+1)).';
mAb_map_3      = [red_3,green_3,blue_3];
color_3        = [colorfinal colorfinal 0];

% colormap 4 (blue, for strong IgG)
%-------------------------------------------------------------------------
red_4          = (zeros(1, colorfinal./colorstep+1)).';
green_4        = (zeros(1, colorfinal./colorstep+1)).';
blue_4         = (colorinitial:colorstep:colorfinal).';
mAb_map_4      = [red_4,green_4,blue_4];
color_4        = [0 0 colorfinal];
```

```
%-------------------------------------------------------------------------
% kin_parameters_I.m
%
% Parameters for phase I
%
% created by Virginia Lane 05/24/18
% last updated on 10/12/18
%-------------------------------------------------------------------------

global c_0_1_I c_1_1_I...            % mAb parameters
    c_0_2_I c_1_2_I...               % moderate IgG parameters
    c_0_3_I c_1_3_I...               % weak IgG parameters
    c_0_4_I c_1_4_I...               % strong IgG parameters
    t t_I t_i t_f_I t_p_I t_s_I      % time parameters

% mAb parameters
%-------------------------------------------------------------------------
c_0_1_I   = 2;          % initial concentration of solution in mg/mL
c_1_1_I   = 0;          % initial concentration of sphere in mg/mL

% moderate IgG parameters
%-------------------------------------------------------------------------
c_0_2_I   = 0;          % initial concentration of solution in mg/mL
c_1_2_I   = 0;          % initial concentration of sphere in mg/mL

% weak IgG parameters
%-------------------------------------------------------------------------
c_0_3_I   = 0;          % initial concentration of solution in mg/mL
c_1_3_I   = 0;          % initial concentration of sphere in mg/mL

% strong IgG parameters
%-------------------------------------------------------------------------
c_0_4_I   = 0;          % initial concentration of solution in mg/mL
c_1_4_I   = 0;          % initial concentration of sphere in mg/mL


% time parameters
%-------------------------------------------------------------------------
t_i   = 0;                  % initial time point
t_f_I = 15;                 % final time point of phase I in minutes
t_p_I = 61;                 % number of time points in phase I
t_s_I = t_f_I./(t_p_I-1);   % step size of time points in phase I
t_I = linspace(t_i,t_f_I,t_p_I); % time point report-out in minutes
t   = t_I;
```

```
%-------------------------------------------------------------------------
% kin_ic_I.m
% kinetic case
%
% pde initial conditions to accompany kin_solver_I.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function u_0 = kin_ic_I(r)

global c_1_1_I k_a_1 q_max_1...
    c_1_2_I k_a_2 q_max_2...
    c_1_3_I k_a_3 q_max_3...
    c_1_4_I k_a_4 q_max_4

u_0 = [c_1_1_I;...
    k_a_1*q_max_1*c_1_1_I/...
    (1+k_a_1*c_1_1_I+k_a_2*c_1_2_I+k_a_3*c_1_3_I+k_a_4*c_1_4_I);...
    c_1_2_I;...
    k_a_2*q_max_2*c_1_2_I/...
    (1+k_a_1*c_1_1_I+k_a_2*c_1_2_I+k_a_3*c_1_3_I+k_a_4*c_1_4_I);...
    c_1_3_I;...
    k_a_3*q_max_3*c_1_3_I/...
    (1+k_a_1*c_1_1_I+k_a_2*c_1_2_I+k_a_3*c_1_3_I+k_a_4*c_1_4_I);...
    c_1_4_I;...
    k_a_4*q_max_4*c_1_4_I/...
    (1+k_a_1*c_1_1_I+k_a_2*c_1_2_I+k_a_3*c_1_3_I+k_a_4*c_1_4_I)];
```

```
%-------------------------------------------------------------------------
% kin_bc_I.m
% kinetic case
%
% pde boundary conditions to accompany kin_solver_I.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = kin_bc_I(rl,ul,rr,ur,t_I)

global c_0_1_I k_a_1 q_max_1...
    c_0_2_I k_a_2 q_max_2...
    c_0_3_I k_a_3 q_max_3...
    c_0_4_I k_a_4 q_max_4

pl  = [0;0;0;0;0;0;0;0];          % "left" bc is for r = 0
ql  = [1;1;1;1;1;1;1;1];          % represents no flux
pr  = [ur(1)-c_0_1_I;...
    ur(2)-k_a_1*q_max_1*c_0_1_I/...
    (1+k_a_1*c_0_1_I+k_a_2*c_0_2_I+k_a_3*c_0_3_I+k_a_4*c_0_4_I);...
    ur(3)-c_0_2_I;...
    ur(4)-k_a_2*q_max_2*c_0_2_I/...
    (1+k_a_1*c_0_1_I+k_a_2*c_0_2_I+k_a_3*c_0_3_I+k_a_4*c_0_4_I);...
    ur(5)-c_0_3_I;...
    ur(6)-k_a_3*q_max_3*c_0_3_I/...
    (1+k_a_1*c_0_1_I+k_a_2*c_0_2_I+k_a_3*c_0_3_I+k_a_4*c_0_4_I);...
    ur(7)-c_0_4_I;...
    ur(8)-k_a_4*q_max_4*c_0_4_I/...
    (1+k_a_1*c_0_1_I+k_a_2*c_0_2_I+k_a_3*c_0_3_I+k_a_4*c_0_4_I)];
qr  = [0;0;0;0;0;0;0;0]; % "right" bc for r = R
                % represents constant surface concentration and q
```

```
%-------------------------------------------------------------------------
% kin_pde_I.m
% kinetic case
%
% pde function to accompany kin_solver_I.m
%
% created by Virginia Lane on 02/20/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function [c,f,s] = kin_pde_I(r,t_I,u,DuDr)

global D_1 q_max_1 k_f_1 k_r_1 e_p_1...
    D_2 q_max_2 k_f_2 k_r_2 e_p_2...
    D_3 q_max_3 k_f_3 k_r_3 e_p_3...
    D_4 q_max_4 k_f_4 k_r_4 e_p_4

c = [e_p_1;1;e_p_2;1;e_p_3;1;e_p_4;1];
f = [e_p_1*D_1*DuDr(1); 0; e_p_2*D_2*DuDr(3); 0;...
    e_p_3*D_3*DuDr(3); 0; e_p_4*D_4*DuDr(4); 0];
s = [-k_f_1*u(1).*(q_max_1-(u(2)+u(4)+u(6)+u(8)))+k_r_1*u(2);...
    k_f_1*u(1).*(q_max_1-(u(2)+u(4)+u(6)+u(8)))-k_r_1*u(2);...
    -k_f_2*u(3).*(q_max_2-(u(2)+u(4)+u(6)+u(8)))+k_r_2*u(4);...
    k_f_2*u(3).*(q_max_2-(u(2)+u(4)+u(6)+u(8)))-k_r_2*u(4);...
    -k_f_3*u(5).*(q_max_3-(u(2)+u(4)+u(6)+u(8)))+k_r_3*u(6);...
    k_f_3*u(5).*(q_max_3-(u(2)+u(4)+u(6)+u(8)))+k_r_3*u(6);...
    -k_f_4*u(7).*(q_max_4-(u(2)+u(4)+u(6)+u(8)))+k_r_1*u(8);...
    k_f_4*u(7).*(q_max_4-(u(2)+u(4)+u(6)+u(8)))+k_r_4*u(8)];
```

```
%-------------------------------------------------------------------------
% kin_solver_I.m
% kinetic case
%
% solver for kin_pde_I.m using kin_ic_I.m and kin_bc_I.m
%
% created by Virginia Lane on 02/14/18
% finalized 08/12/18
%-------------------------------------------------------------------------
function kin_solver_I

global m r t_I...
    kin_u_1 kin_q_1 kin_u_1_I kin_q_1_I...
    kin_u_2 kin_q_2 kin_u_2_I kin_q_2_I...
    kin_u_3 kin_q_3 kin_u_3_I kin_q_3_I...
    kin_u_4 kin_q_4 kin_u_4_I kin_q_4_I

sol = pdepe(m,@kin_pde_I,@kin_ic_I,@kin_bc_I,r,t_I);
kin_u_1_I = sol(:,:,1);
kin_q_1_I = sol(:,:,2);
kin_u_2_I = sol(:,:,3);
kin_q_2_I = sol(:,:,4);
kin_u_3_I = sol(:,:,5);
kin_q_3_I = sol(:,:,6);
kin_u_4_I = sol(:,:,7);
kin_q_4_I = sol(:,:,8);

kin_u_1 = kin_u_1_I;
kin_q_1 = kin_q_1_I;
kin_u_2 = kin_u_2_I;
kin_q_2 = kin_q_2_I;
kin_u_3 = kin_u_3_I;
kin_q_3 = kin_u_3_I;
kin_u_4 = kin_u_4_I;
kin_q_4 = kin_q_4_I;
```

```
%-------------------------------------------------------------------------
% kin_parameters_II.m
%
% kinetic case
% these are the parameters for phase II
%
% created by Virginia Lane 05/24/18
% last updated on 10/12/18
%-------------------------------------------------------------------------

global c_0_1_II c_0_2_II c_0_3_II c_0_4_II...
    t_II t_i_II t_f_I t_f_II t_t_II t_p_II t_I t


% mAb parameters
%-------------------------------------------------------------------------
c_0_1_II   = 0;
% phase II initial concentration of solution in mg/mL

% moderate IgG parameters
%-------------------------------------------------------------------------
c_0_2_II   = 2*.4972;
% phase II initial concentration of solution in mg/mL

% weak IgG parameters
%-------------------------------------------------------------------------
c_0_3_II   = 2*.1386;
% phase II initial concentration of solution in mg/mL

% strong IgG parameters
%-------------------------------------------------------------------------
c_0_4_II   = 2*.3642;
% phase II initial concentration of solution in mg/mL


% time parameters
%-------------------------------------------------------------------------
-
t_i_II = t_f_I+.001;             % initial time point of phase II
t_f_II = 240;                    % total time in phase II in minutes
t_t_II = t_f_I+t_f_II;           % total process time
t_p_II = 240;                    % number of time points in phase II
t_s_II = t_f_II./(t_p_II-1);     % step size of time points in phase
II
t_II   = linspace(t_i_II,t_t_II,t_p_II); % time point report-out
t      = [t_I t_II];
```

```
%-------------------------------------------------------------------------
% kin_ic_II.m
% kinetic case
%
% pde initial conditions to accompany kin_solver_II.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function u_0 = kin_ic_II(r2)

global  r kin_u_1_I kin_q_1_I...
    kin_u_2_I kin_q_2_I...
    kin_u_3_I kin_q_3_I...
    kin_u_4_I kin_q_4_I

u_0 = [interp1(r, kin_u_1_I(end,:), r2, 'spline');...
    interp1(r, kin_q_1_I(end,:), r2, 'spline');...
    interp1(r, kin_u_2_I(end,:), r2, 'spline');...
    interp1(r, kin_q_2_I(end,:), r2, 'spline');...
    interp1(r, kin_u_3_I(end,:), r2, 'spline');...
    interp1(r, kin_q_3_I(end,:), r2, 'spline');...
    interp1(r, kin_u_4_I(end,:), r2, 'spline');...
    interp1(r, kin_q_4_I(end,:), r2, 'spline')];
```

```
%-------------------------------------------------------------------------
% kin_bc_II.m
% kinetic case
%
% pde boundary conditions to accompany kin_solver_II.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function [pl,ql,pr,qr] = kin_bc_II(rl,ul,rr,ur,t_II)

global c_0_1_II k_a_1 q_max_1...
    c_0_2_II k_a_2 q_max_2...
    c_0_3_II k_a_3 q_max_3...
    c_0_4_II k_a_4 q_max_4

pl  = [0;0;0;0;0;0;0;0];          % "left" bc is for r = 0
ql  = [1;1;1;1;1;1;1;1];          %  and represents no flux
pr  = [ur(1)-c_0_1_II;...
    ur(2)-
k_a_1*q_max_1*c_0_1_II/(1+k_a_1*c_0_1_II+k_a_2*c_0_2_II+k_a_3*c_0_3_II
+k_a_4*c_0_4_II);...
    ur(3)-c_0_2_II;...
    ur(4)-
k_a_2*q_max_2*c_0_2_II/(1+k_a_1*c_0_1_II+k_a_2*c_0_2_II+k_a_3*c_0_3_II
+k_a_4*c_0_4_II);...
    ur(5)-c_0_3_II;...
    ur(6)-
k_a_3*q_max_3*c_0_3_II/(1+k_a_1*c_0_1_II+k_a_2*c_0_2_II+k_a_3*c_0_3_II
+k_a_4*c_0_4_II);...
    ur(7)-c_0_4_II;...
    ur(8)-
k_a_4*q_max_4*c_0_4_II/(1+k_a_1*c_0_1_II+k_a_2*c_0_2_II+k_a_3*c_0_3_II
+k_a_4*c_0_4_II)];
qr  = [0;0;0;0;0;0;0;0]; % "right" bc for r = R
               % represents constant surface concentration and q
```

```
%-------------------------------------------------------------------------
% kin_pde_II.m
% kinetic case
%
% pde function to accompany kin_solver_II.m
%
% created by Virginia Lane on 02/20/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function [c,f,s] = kin_pde_II(r,t_II,u,DuDr)

global D_1 q_max_1 k_f_1 k_r_1 e_p_1...
    D_2 q_max_2 k_f_2 k_r_2 e_p_2...
    D_3 q_max_3 k_f_3 k_r_3 e_p_3...
    D_4 q_max_4 k_f_4 k_r_4 e_p_4

c = [e_p_1;1;e_p_2;1;e_p_3;1;e_p_4;1];
f = [e_p_1*D_1*DuDr(1); 0; e_p_2*D_2*DuDr(3); 0;...
    e_p_3*D_3*DuDr(5); 0; e_p_4*D_4*DuDr(7); 0];
s = [-k_f_1*u(1)*q_max_1*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))+k_r_1*u(2);...
    k_f_1*u(1)*q_max_1*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))-k_r_1*u(2);...
    -k_f_2*u(3)*q_max_2*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))+k_r_2*u(4);...
    k_f_2*u(3)*q_max_2*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))-k_r_2*u(4);...
    -k_f_3*u(5)*q_max_3*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))+k_r_3*u(6);...
    k_f_3*u(5)*q_max_3*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))-k_r_3*u(6);...
    -k_f_4*u(7)*q_max_4*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))+k_r_4*u(8);...
    k_f_4*u(7)*q_max_4*(1-
(u(2)/q_max_1+u(4)/q_max_2+u(6)/q_max_3+u(8)/q_max_4))-k_r_4*u(8)];
```

```
%-------------------------------------------------------------------------
% kin_solver_II.m
% kinetic case
%
% solver for kin_pde_II.m using kin_ic_II.m and kin_bc_II.m
%
% created by Virginia Lane on 02/14/18
% finalized 10/12/18
%-------------------------------------------------------------------------
function kin_solver_II

global m r t_II...
    kin_u_1 kin_u_1_I kin_u_1_II kin_q_1 kin_q_1_I kin_q_1_II...
    kin_u_2 kin_u_2_I kin_u_2_II kin_q_2 kin_q_2_I kin_q_2_II...
    kin_u_3 kin_u_3_I kin_u_3_II kin_q_3 kin_q_3_I kin_q_3_II...
    kin_u_4 kin_u_4_I kin_u_4_II kin_q_4 kin_q_4_I kin_q_4_II

sol = pdepe(m,@kin_pde_II,@kin_ic_II,@kin_bc_II,r,t_II);
kin_u_1_II = sol(:,:,1);
kin_q_1_II = sol(:,:,2);
kin_u_2_II = sol(:,:,3);
kin_q_2_II = sol(:,:,4);
kin_u_3_II = sol(:,:,5);
kin_q_3_II = sol(:,:,6);
kin_u_4_II = sol(:,:,7);
kin_q_4_II = sol(:,:,8);

kin_u_1 = [kin_u_1_I; kin_u_1_II];
kin_q_1 = [kin_q_1_I; kin_q_1_II];
kin_u_2 = [kin_u_2_I; kin_u_2_II];
kin_q_2 = [kin_q_2_I; kin_q_2_II];
kin_u_3 = [kin_u_3_I; kin_u_3_II];
kin_q_3 = [kin_q_3_I; kin_q_3_II];
kin_u_4 = [kin_u_4_I; kin_u_4_II];
kin_q_4 = [kin_q_4_I; kin_q_4_II];
```

```
%-------------------------------------------------------------------------
% kin_qcontourgraphs.m
% kinetic, competitive binding X 5 case
%
% graphs 9 concentration contour plots for a spherical cross section
% script uses the matlab pdepe solver
%
% created by Virginia Lane on 05/14/18
% last updated on 05/24/18
%-------------------------------------------------------------------------

global t_II r mAb_map_1 mAb_map_2 mAb_map_3 mAb_map_4...
    kin_q_1_II kin_q_2_II kin_q_3_II kin_q_4_II

% mAb
%-------------------------------------------------------------------------
figure(1);
for n = 1:1:5                % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta           = 0:.1:33*pi/16;
    [th, rad]       = meshgrid(theta,r);
    X               = rad.*cos(th);
    Y               = rad.*sin(th);
    r_circle        = sqrt(X.^2 + Y.^2);
    [row, column]   = size(r_circle);
    u2              = kin_q_1_II(column2,:);
    u2              = u2.';
    n2              = 0;
    U               = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_1);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(kin_q_1_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'mAb','q'});

% IgG
%-------------------------------------------------------------------------
figure(2);
```

```
for n = 1:1:5                  % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta         = 0:.1:33*pi/16;
    [th, rad]     = meshgrid(theta,r);
    X             = rad.*cos(th);
    Y             = rad.*sin(th);
    r_circle      = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2            = kin_q_2_II(column2,:);
    u2            = u2.';
    n2            = 0;
    U             = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_2);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(kin_q_2_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'IgG','q'});

% weak IgG
%-------------------------------------------------------------------------
figure(3);
for n = 1:1:5                  % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta         = 0:.1:33*pi/16;
    [th, rad]     = meshgrid(theta,r);
    X             = rad.*cos(th);
    Y             = rad.*sin(th);
    r_circle      = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2            = kin_q_3_II(column2,:);
    u2            = u2.';
    n2            = 0;
    U             = u2;
    for n2 = 1:1:column-1
```

```
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_3);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(kin_q_3_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'weak','IgG','q'});

% Strong IgG
%-------------------------------------------------------------------------
figure(4);
for n = 1:1:5                  % always starts at time point 0
    subplot(1,5,n)
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row2, column2] = min(abs(t_II-time));
    minutes         = round(t_II(column2));
    theta          = 0:.1:33*pi/16;
    [th, rad]      = meshgrid(theta,r);
    X              = rad.*cos(th);
    Y              = rad.*sin(th);
    r_circle       = sqrt(X.^2 + Y.^2);
    [row, column] = size(r_circle);
    u2             = kin_q_4_II(column2,:);
    u2             = u2.';
    n2             = 0;
    U              = u2;
    for n2 = 1:1:column-1
        U  = [U,u2];
        n2 = n2 + 1;
    end

    pcolor(X,Y,U);
    shading interp;
    colormap(mAb_map_4);
    axis('square');
    axis off
    title([num2str(minutes-15) ' min']);
    caxis([0 max(kin_q_4_II(:))])
end
cb = colorbar('location','Manual','position',[.92 .43 .03 .18]);
title(cb,{'strong','IgG','q'});


% saving figures
```

```
%-------------------------------------------------------------------------
saveas(figure(1),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\mAb_qcontour.jpg']);
saveas(figure(2),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\IgG_qcontour.jpg']);
saveas(figure(3),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\weakIgG_qcontour.jpg']);
saveas(figure(4),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\strongIgG_qcontour.jpg']);
```

```
%-------------------------------------------------------------------------
% kin_qvrgraphs.m
% kinetic, competitive binding case
%
% created by Virginia Lane on 05/14/18
% last updated on 05/24/18
%-------------------------------------------------------------------------

global t_II r r_i R...
    kin_q_1_II kin_q_2_II kin_q_3_II kin_q_4_II

for n = 1:1:5
    figure(n);
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row, column] = min(abs(t_II-time));
    u1            = kin_q_1_II(column,:);
    u2            = kin_q_2_II(column,:);
    u3            = kin_q_3_II(column,:);
    u4            = kin_q_4_II(column,:);
    plot(r, u1,'g', r, u2,'r', r, u3, 'y', r, u4, 'b','linewidth',2)

    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    ylim([0 60]);
    xlim([r_i (R-.00005)])
    title(['t = ' num2str(time-15),' minutes'])
    legend('mAb', '"moderate" hIgG', '"weak" hIgG', '"strong" hIgG',
'location','northwest')
    set(gca,'fontsize',16)
end

% all lines on one graph
%-------------------------------------------------------------------------
figure(6)
n3   = [16,30,45,75,255];
for n = 1:1:5
    time = n3(n);
    [row, column] = min(abs(t_II-time));
    u1            = kin_q_1_II(column,:);
    u2            = kin_q_2_II(column,:);
    u3            = kin_q_3_II(column,:);
    u4            = kin_q_4_II(column,:);

    plot(r, u1, 'g', r, u2, 'r', r, u3, 'y', r, u4, 'b')
    hold on
end
hold off

xlabel('Radius (cm)')
ylabel('q (mg/mL)')
xlim([r_i R])
```

```
title('q vs radius')
set(0, 'defaultaxescolor','white')
legend('mAb', 'IgG', 'weak IgG', 'strong IgG', 'location','northwest')

% saving figures
%-------------------------------------------------------------------------
saveas(figure(1),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_1.jpg']);
saveas(figure(2),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_15.jpg']);
saveas(figure(3),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_30.jpg']);
saveas(figure(4),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_60.jpg']);
saveas(figure(5),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_240.jpg']);
saveas(figure(6),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_all.jpg']);
```

```
%-------------------------------------------------------------------------
% kin_qvrgraphs2.m
% kinetic, competitive binding case
%
% created by Virginia Lane on 05/14/18
% last updated on 05/24/18
%-------------------------------------------------------------------------

global t_II r R...
    kin_q_1_II kin_q_2_II kin_q_3_II kin_q_4_II

for n = 1:1:5
    figure(n);
    n3   = [16,30,45,75,255];
    time = n3(n);
    [row, column] = min(abs(t_II-time));
    u1            = kin_q_1_II(column,:);
    u2            = kin_q_2_II(column,:);
    u3            = kin_q_3_II(column,:);
    u4            = kin_q_4_II(column,:);
    plot(r, u1,'g', r, u2,'r', r, u3, 'y', r, u4, 'b',...
        -r, u1,'g', -r, u2,'r', -r, u3, 'y', -r, u4,
'b','linewidth',2)

    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-(R-.00005) (R-.00005)])
    ylim([0 60])
    title(['t = ' num2str(time-15),' minutes'])
    %legend('mAb', '"moderate" hIgG', '"weak" hIgG', '"strong" hIgG',
'location','northwest')
    set(gca,'fontsize',16)

end

% all lines on one graph
%-------------------------------------------------------------------------
figure(6)
n3   = [16,30,45,75,255];
for n = 1:1:5
    time = n3(n);
    [row, column] = min(abs(t_II-time));
    u1            = kin_q_1_II(column,:);
    u2            = kin_q_2_II(column,:);
    u3            = kin_q_3_II(column,:);
    u4            = kin_q_4_II(column,:);

    figure(7)
    plot(r, u1, 'g', -r, u1,'g','linewidth',2)
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-(R-.00005) (R-.00005)])
```

```
    ylim([0 60])
    title('mAb')
    hold on

    figure(8)
    plot(r, u2, 'r', -r, u2,'r','linewidth',2)
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-(R-.00005) (R-.00005)])
    ylim([0 60])
    title('IgG')
    hold on

    figure(9)
    plot(r, u3, 'y', -r, u3,'y','linewidth',2)
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-(R-.00005) (R-.00005)])
    ylim([0 60])
    title('weak IgG')
    hold on

    figure(10)
    plot(r, u4, 'b', -r, u4,'b','linewidth',2)
    xlabel('Radius (cm)')
    ylabel('q (mg/mL)')
    xlim([-(R-.00005) (R-.00005)])
    ylim([0 60])
    title('strong IgG')
    hold on

    figure(6)
    plot(r, u1, 'g', r, u2, 'r', r, u3, 'y', r, u4, 'b',...
        -r, u1,'g', -r, u2,'r', -r, u3, 'y', -r, u4,
'b','linewidth',2)
    hold on
end
hold off

xlabel('Radius (cm)')
ylabel('q (mg/mL)')
xlim([-(R-.00005) (R-.00005)])
ylim([0 60])
title('q vs radius')
legend('mAb', '"moderate" hIgG', '"weak" hIgG', '"strong" hIgG',
'location','northwest')
set(gca,'fontsize',16)


% saving figures
%-----------------------------------------------------------------------
```

```
saveas(figure(1),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_1.jpg']);
saveas(figure(2),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_15.jpg']);
saveas(figure(3),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_30.jpg']);
saveas(figure(4),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_60.jpg']);
saveas(figure(5),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_240.jpg']);
saveas(figure(6),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_all.jpg']);
saveas(figure(7),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_mAb.jpg']);
saveas(figure(8),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_IgG.jpg']);
saveas(figure(9),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_weakIgG.jpg']);
saveas(figure(10),['C:\users\Virginia
Lane\Desktop\figures\kin_4comp\qvrgraph_strongIgG.jpg']);
```

Quantitative Analysis of Experimental Data

```
%-------------------------------------------------------------------------
% image_5a.m
%
% script quantitatively analyzes experimental images to generate plots
% of color intensity vs. radial position
% CaptivA PriMab chromatography media
% Images taken from Weinberg et al. 2017
%
% created by Virginia Lane on 10/10/18
% last updated on 10/10/18
%-------------------------------------------------------------------------

% parameters
%-------------------------------------------------------------------------
Rmin = 95;
Rmax = 105;
R = .0043;
mAb_1_sensitivity = .93;
mAb_2_sensitivity = .93;
mAb_3_sensitivity = .95;
mAb_4_sensitivity = .96;
mAb_5_sensitivity = .96;
hIgG_1_sensitivity = .96;
hIgG_2_sensitivity = .93;
hIgG_3_sensitivity = .93;
hIgG_4_sensitivity = .93;
hIgG_5_sensitivity = .93;

% 5a_mAb_1
%-------------------------------------------------------------------------
mAb_1 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_mAb_1.jpg');
mAb_1_gray = rgb2gray(mAb_1);
[mAb_1_center, mAb_1_radius] = imfindcircles(mAb_1_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_1_sensitivity,'method','twostage');
x = round(mAb_1_center(1,1));
y = round(mAb_1_center(1,2));
[rows columns z] = size(mAb_1);

figure(1);
imshow(mAb_1)
viscircles(mAb_1_center, mAb_1_radius,'color','r');

mAb_1_vector = [];
mAb_1_gray_vector = [];
for n = 1:1:rows
    mAb_1_vector = [mAb_1_vector mAb_1(n,:,1)];
end
for n = 1:1:rows
```

```
    mAb_1_gray_vector = [mAb_1_gray_vector mAb_1_gray(n,:)];
end

mAb_1_dist = [];
mAb_1_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_1_dist = [mAb_1_dist; row];
    mAb_1_dist_vector = [mAb_1_dist_vector row];
end

mAb_1_intensity = [];
for n = 1:1:max(mAb_1_dist_vector)
    m = find(mAb_1_dist_vector==n);
    average = mean(mAb_1_gray_vector(m));
    mAb_1_intensity = [mAb_1_intensity average];
end

bins = 1:1:max(mAb_1_dist_vector);
mAb_1_cm = bins*R/mAb_1_radius;

% 5a_mAb_15
%--------------------------------------------------------------------------
mAb_15 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_mAb_15.jpg');
mAb_15_gray = rgb2gray(mAb_15);
[mAb_15_center, mAb_15_radius] = imfindcircles(mAb_15_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_2_sensitivity,'method','twostage');
x = round(mAb_15_center(1,1));
y = round(mAb_15_center(1,2));
[rows columns z] = size(mAb_15);

figure(2);
imshow(mAb_15)
viscircles(mAb_15_center, mAb_15_radius,'color','r');

mAb_15_vector = [];
mAb_15_gray_vector = [];
for n = 1:1:rows
    mAb_15_vector = [mAb_15_vector mAb_15(n,:,1)];
end
for n = 1:1:rows
    mAb_15_gray_vector = [mAb_15_gray_vector mAb_15_gray(n,:)];
end
```

```
mAb_15_dist = [];
mAb_15_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_15_dist = [mAb_15_dist; row];
    mAb_15_dist_vector = [mAb_15_dist_vector row];
end

mAb_15_intensity = [];
for n = 1:1:max(mAb_15_dist_vector)
    m = find(mAb_15_dist_vector==n);
    average = mean(mAb_15_gray_vector(m));
    mAb_15_intensity = [mAb_15_intensity average];
end

bins = 1:1:max(mAb_15_dist_vector);
mAb_15_cm = bins*R/mAb_15_radius;


% 5a_mAb_30
%--------------------------------------------------------------------------
mAb_30 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_mAb_30.jpg');
mAb_30_gray = rgb2gray(mAb_30);
[mAb_30_center, mAb_30_radius] = imfindcircles(mAb_30_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_3_sensitivity,'method','twostage');
x = round(mAb_30_center(1,1));
y = round(mAb_30_center(1,2));
[rows columns z] = size(mAb_30);

figure(3);
imshow(mAb_30)
viscircles(mAb_30_center, mAb_30_radius,'color','r');

mAb_30_vector = [];
mAb_30_gray_vector = [];
for n = 1:1:rows
    mAb_30_vector = [mAb_30_vector mAb_30(n,:,1)];
end
for n = 1:1:rows
    mAb_30_gray_vector = [mAb_30_gray_vector mAb_30_gray(n,:)];
end

mAb_30_dist = [];
mAb_30_dist_vector = [];
```

```
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_30_dist = [mAb_30_dist; row];
    mAb_30_dist_vector = [mAb_30_dist_vector row];
end

mAb_30_intensity = [];
for n = 1:1:max(mAb_30_dist_vector)
    m = find(mAb_30_dist_vector==n);
    average = mean(mAb_30_gray_vector(m));
    mAb_30_intensity = [mAb_30_intensity average];
end

bins = 1:1:max(mAb_30_dist_vector);
mAb_30_cm = bins*R/mAb_30_radius;

% 5a_mAb_60
%--------------------------------------------------------------------------
mAb_60 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_mAb_60.jpg');
mAb_60_gray = rgb2gray(mAb_60);
[mAb_60_center, mAb_60_radius] = imfindcircles(mAb_60_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_4_sensitivity,'method','twostage');
x = round(mAb_60_center(1,1));
y = round(mAb_60_center(1,2));
[rows columns z] = size(mAb_60);

figure(4);
imshow(mAb_60)
viscircles(mAb_60_center, mAb_60_radius,'color','r');

mAb_60_vector = [];
mAb_60_gray_vector = [];
for n = 1:1:rows
    mAb_60_vector = [mAb_60_vector mAb_60(n,:,1)];
end
for n = 1:1:rows
    mAb_60_gray_vector = [mAb_60_gray_vector mAb_60_gray(n,:)];
end

mAb_60_dist = [];
mAb_60_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
```

```
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_60_dist = [mAb_60_dist; row];
    mAb_60_dist_vector = [mAb_60_dist_vector row];
end

mAb_60_intensity = [];
for n = 1:1:max(mAb_60_dist_vector)
    m = find(mAb_60_dist_vector==n);
    average = mean(mAb_60_gray_vector(m));
    mAb_60_intensity = [mAb_60_intensity average];
end

bins = 1:1:max(mAb_60_dist_vector);
mAb_60_cm = bins*R/mAb_60_radius;

% 5a_mAb_240
%--------------------------------------------------------------------------
mAb_240 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_mAb_240.jpg');
mAb_240_gray = rgb2gray(mAb_240);
[mAb_240_center, mAb_240_radius] = imfindcircles(mAb_240_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_5_sensitivity,'method','twostage');
x = round(mAb_240_center(1,1));
y = round(mAb_240_center(1,2));
[rows columns z] = size(mAb_240);

figure(5);
imshow(mAb_240)
viscircles(mAb_240_center, mAb_240_radius,'color','r');

mAb_240_vector = [];
mAb_240_gray_vector = [];
for n = 1:1:rows
    mAb_240_vector = [mAb_240_vector mAb_240(n,:,1)];
end
for n = 1:1:rows
    mAb_240_gray_vector = [mAb_240_gray_vector mAb_240_gray(n,:)];
end

mAb_240_dist = [];
mAb_240_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
```

```
        row = [row round(D)];
    end
    mAb_240_dist = [mAb_240_dist; row];
    mAb_240_dist_vector = [mAb_240_dist_vector row];
end

mAb_240_intensity = [];
for n = 1:1:max(mAb_240_dist_vector)
    m = find(mAb_240_dist_vector==n);
    average = mean(mAb_240_gray_vector(m));
    mAb_240_intensity = [mAb_240_intensity average];
end

bins = 1:1:max(mAb_240_dist_vector);
mAb_240_cm = bins*R/mAb_240_radius;

% 5a_hIgG_1
%--------------------------------------------------------------------------
hIgG_1 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_hIgG_1.jpg');
hIgG_1_gray = rgb2gray(hIgG_1);
[hIgG_1_center, hIgG_1_radius] = imfindcircles(hIgG_1_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_1_sensitivity,'method','twostage');
x = round(hIgG_1_center(1,1));
y = round(hIgG_1_center(1,2));
[rows columns z] = size(hIgG_1);

figure(6);
imshow(hIgG_1)
viscircles(hIgG_1_center, hIgG_1_radius,'color','r');

hIgG_1_vector = [];
hIgG_1_gray_vector = [];
for n = 1:1:rows
    hIgG_1_vector = [hIgG_1_vector hIgG_1(n,:,1)];
end
for n = 1:1:rows
    hIgG_1_gray_vector = [hIgG_1_gray_vector hIgG_1_gray(n,:)];
end

hIgG_1_dist = [];
hIgG_1_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_1_dist = [hIgG_1_dist; row];
```

```
    hIgG_1_dist_vector = [hIgG_1_dist_vector row];
end

hIgG_1_intensity = [];
for n = 1:1:max(hIgG_1_dist_vector)
    m = find(hIgG_1_dist_vector==n);
    average = mean(hIgG_1_gray_vector(m));
    hIgG_1_intensity = [hIgG_1_intensity average];
end

bins = 1:1:max(hIgG_1_dist_vector);
hIgG_1_cm = bins*R/hIgG_1_radius;

% 5a_hIgG_15
%-------------------------------------------------------------------------
hIgG_15 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_hIgG_15.jpg');
hIgG_15_gray = rgb2gray(hIgG_15);
[hIgG_15_center, hIgG_15_radius] = imfindcircles(hIgG_15_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_2_sensitivity,'method','twostage');
x = round(hIgG_15_center(1,1));
y = round(hIgG_15_center(1,2));
[rows columns z] = size(hIgG_15);

figure(7);
imshow(hIgG_15)
viscircles(hIgG_15_center, hIgG_15_radius,'color','r');

hIgG_15_vector = [];
hIgG_15_gray_vector = [];
for n = 1:1:rows
    hIgG_15_vector = [hIgG_15_vector hIgG_15(n,:,1)];
end
for n = 1:1:rows
    hIgG_15_gray_vector = [hIgG_15_gray_vector hIgG_15_gray(n,:)];
end

hIgG_15_dist = [];
hIgG_15_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_15_dist = [hIgG_15_dist; row];
    hIgG_15_dist_vector = [hIgG_15_dist_vector row];
end
```

```
hIgG_15_intensity = [];
for n = 1:1:max(hIgG_15_dist_vector)
    m = find(hIgG_15_dist_vector==n);
    average = mean(hIgG_15_gray_vector(m));
    hIgG_15_intensity = [hIgG_15_intensity average];
end

bins = 1:1:max(hIgG_15_dist_vector);
hIgG_15_cm = bins*R/hIgG_15_radius;


% 5a_hIgG_30
%--------------------------------------------------------------------------
hIgG_30 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_hIgG_30.jpg');
hIgG_30_gray = rgb2gray(hIgG_30);
[hIgG_30_center, hIgG_30_radius] = imfindcircles(hIgG_30_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_3_sensitivity,'method','twostage');
x = round(hIgG_30_center(1,1));
y = round(hIgG_30_center(1,2));
[rows columns z] = size(hIgG_30);

figure(8);
imshow(hIgG_30)
viscircles(hIgG_30_center, hIgG_30_radius,'color','r');

hIgG_30_vector = [];
hIgG_30_gray_vector = [];
for n = 1:1:rows
    hIgG_30_vector = [hIgG_30_vector hIgG_30(n,:,1)];
end
for n = 1:1:rows
    hIgG_30_gray_vector = [hIgG_30_gray_vector hIgG_30_gray(n,:)];
end

hIgG_30_dist = [];
hIgG_30_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_30_dist = [hIgG_30_dist; row];
    hIgG_30_dist_vector = [hIgG_30_dist_vector row];
end

hIgG_30_intensity = [];
for n = 1:1:max(hIgG_30_dist_vector)
```

```
    m = find(hIgG_30_dist_vector==n);
    average = mean(hIgG_30_gray_vector(m));
    hIgG_30_intensity = [hIgG_30_intensity average];
end

bins = 1:1:max(hIgG_30_dist_vector);
hIgG_30_cm = bins*R/hIgG_30_radius;

% 5a_hIgG_60
%--------------------------------------------------------------------------
hIgG_60 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_hIgG_60.jpg');
hIgG_60_gray = rgb2gray(hIgG_60);
[hIgG_60_center, hIgG_60_radius] = imfindcircles(hIgG_60_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_4_sensitivity,'method','twostage');
x = round(hIgG_60_center(1,1));
y = round(hIgG_60_center(1,2));
[rows columns z] = size(hIgG_60);

figure(9);
imshow(hIgG_60)
viscircles(hIgG_60_center, hIgG_60_radius,'color','r');

hIgG_60_vector = [];
hIgG_60_gray_vector = [];
for n = 1:1:rows
    hIgG_60_vector = [hIgG_60_vector hIgG_60(n,:,1)];
end
for n = 1:1:rows
    hIgG_60_gray_vector = [hIgG_60_gray_vector hIgG_60_gray(n,:)];
end

hIgG_60_dist = [];
hIgG_60_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_60_dist = [hIgG_60_dist; row];
    hIgG_60_dist_vector = [hIgG_60_dist_vector row];
end

hIgG_60_intensity = [];
for n = 1:1:max(hIgG_60_dist_vector)
    m = find(hIgG_60_dist_vector==n);
    average = mean(hIgG_60_gray_vector(m));
    hIgG_60_intensity = [hIgG_60_intensity average];
```

```
end

bins = 1:1:max(hIgG_60_dist_vector);
hIgG_60_cm = bins*R/hIgG_60_radius;

% 5a_hIgG_240
%-------------------------------------------------------------------------
hIgG_240 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5a\5a_hIgG_240.jpg');
hIgG_240_gray = rgb2gray(hIgG_240);
[hIgG_240_center, hIgG_240_radius] = imfindcircles(hIgG_240_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_5_sensitivity,'method','twostage');
x = round(hIgG_240_center(1,1));
y = round(hIgG_240_center(1,2));
[rows columns z] = size(hIgG_240);

figure(10);
imshow(hIgG_240)
viscircles(hIgG_240_center, hIgG_240_radius,'color','r');

hIgG_240_vector = [];
hIgG_240_gray_vector = [];
for n = 1:1:rows
    hIgG_240_vector = [hIgG_240_vector hIgG_240(n,:,1)];
end
for n = 1:1:rows
    hIgG_240_gray_vector = [hIgG_240_gray_vector hIgG_240_gray(n,:)];
end

hIgG_240_dist = [];
hIgG_240_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_240_dist = [hIgG_240_dist; row];
    hIgG_240_dist_vector = [hIgG_240_dist_vector row];
end

hIgG_240_intensity = [];
for n = 1:1:max(hIgG_240_dist_vector)
    m = find(hIgG_240_dist_vector==n);
    average = mean(hIgG_240_gray_vector(m));
    hIgG_240_intensity = [hIgG_240_intensity average];
end

bins = 1:1:max(hIgG_240_dist_vector);
```

```
hIgG_240_cm = bins*R/hIgG_240_radius;


% plot data
%-------------------------------------------------------------------------
figure(11);
plot(mAb_1_cm,mAb_1_intensity,'g',hIgG_1_cm,hIgG_1_intensity,'r',...
    -mAb_1_cm,mAb_1_intensity,'g',-hIgG_1_cm,hIgG_1_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('1 minute')
legend('mAb','hIgG','location','northwest')

figure(12);
plot(mAb_15_cm,mAb_15_intensity,'g',hIgG_15_cm,hIgG_15_intensity,'r',.
..
    -mAb_15_cm,mAb_15_intensity,'g',-hIgG_15_cm,hIgG_15_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('15 minutes')
legend('mAb','hIgG','location','northwest')

figure(13);
plot(mAb_30_cm,mAb_30_intensity,'g',hIgG_30_cm,hIgG_30_intensity,'r',.
..
    -mAb_30_cm,mAb_30_intensity,'g',-hIgG_30_cm,hIgG_30_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('30 minutes')
legend('mAb','hIgG','location','northwest')

figure(14);
plot(mAb_60_cm,mAb_60_intensity,'g',hIgG_60_cm,hIgG_60_intensity,'r',.
..
    -mAb_60_cm,mAb_60_intensity,'g',-hIgG_60_cm,hIgG_60_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('60 minutes')
legend('mAb','hIgG','location','northwest')

figure(15);
plot(mAb_240_cm,mAb_240_intensity,'g',hIgG_240_cm,hIgG_240_intensity,'
r',...
    -mAb_240_cm,mAb_240_intensity,'g',-
hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
```

```
title('240 minutes')
legend('mAb','hIgG','location','northwest')

figure(16);
plot(mAb_1_cm,mAb_1_intensity,'g',hIgG_1_cm,hIgG_1_intensity,'r',...

mAb_15_cm,mAb_15_intensity,'g',hIgG_15_cm,hIgG_15_intensity,'r',...

mAb_30_cm,mAb_30_intensity,'g',hIgG_30_cm,hIgG_30_intensity,'r',...

mAb_60_cm,mAb_60_intensity,'g',hIgG_60_cm,hIgG_60_intensity,'r',...

mAb_240_cm,mAb_240_intensity,'g',hIgG_240_cm,hIgG_240_intensity,'r',..
.
    -mAb_1_cm,mAb_1_intensity,'g',-hIgG_1_cm,hIgG_1_intensity,'r',...
    -mAb_15_cm,mAb_15_intensity,'g',-
hIgG_15_cm,hIgG_15_intensity,'r',...
    -mAb_30_cm,mAb_30_intensity,'g',-
hIgG_30_cm,hIgG_30_intensity,'r',...
    -mAb_60_cm,mAb_60_intensity,'g',-
hIgG_60_cm,hIgG_60_intensity,'r',...
    -mAb_240_cm,mAb_240_intensity,'g',-
hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('mAb','hIgG','location','northwest')

figure(17);
plot(mAb_1_cm,mAb_1_intensity,'k',...
    mAb_15_cm,mAb_15_intensity,'r',...
    mAb_30_cm,mAb_30_intensity,'y',...
    mAb_60_cm,mAb_60_intensity,'g',...
    mAb_240_cm,mAb_240_intensity,'b',...
    -mAb_1_cm,mAb_1_intensity,'k',...
    -mAb_15_cm,mAb_15_intensity,'r',...
    -mAb_30_cm,mAb_30_intensity,'y',...
    -mAb_60_cm,mAb_60_intensity,'g',...
    -mAb_240_cm,mAb_240_intensity,'b','linewidth',1.5)
xlim([-.0045 .0045]);
xlabel('radius (cm)')
ylabel('average intensity')
title('mAb component: intensity vs. radius')
legend('1 min','15 min','30 min','60 min','240
min','location','southeast')

figure(18);
plot(hIgG_1_cm,hIgG_1_intensity,'k',...
    hIgG_15_cm,hIgG_15_intensity,'r',...
    hIgG_30_cm,hIgG_30_intensity,'y',...
    hIgG_60_cm,hIgG_60_intensity,'g',...
```

```
    hIgG_240_cm,hIgG_240_intensity,'b',...
    -hIgG_1_cm,hIgG_1_intensity,'k',...
    -hIgG_15_cm,hIgG_15_intensity,'r',...
    -hIgG_30_cm,hIgG_30_intensity,'y',...
    -hIgG_60_cm,hIgG_60_intensity,'g',...
    -hIgG_240_cm,hIgG_240_intensity,'b','linewidth',1.5)
xlim([-.0045 .0045]);
xlabel('radius (cm)')
ylabel('average intensity')
title('hIgG component: intensity vs. radius')
legend('1 min','15 min','30 min','60 min','240
min','location','southeast')

% save figures
%-----------------------------------------------------------------------
saveas(figure(11),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_1.jpg']);
saveas(figure(12),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_15.jpg']);
saveas(figure(13),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_30.jpg']);
saveas(figure(14),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_60.jpg']);
saveas(figure(15),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_240.jpg']);
saveas(figure(16),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_all.jpg']);
saveas(figure(17),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_mAb.jpg']);
saveas(figure(18),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5a\5a_hIgG.jpg']);
```

```
%-------------------------------------------------------------------------
% image_5b.m
%
% script quantitatively analyzes experimental images to generate plots
% of color intensity vs. radial position
% MabSelect chromatography media
% Images taken from Weinberg et al. 2017
%
% created by Virginia Lane on 10/10/18
% last updated on 10/10/18
%-------------------------------------------------------------------------

% parameters
%-------------------------------------------------------------------------
Rmin = 95;
Rmax = 105;
R = .0043;
mAb_1_sensitivity = .93;
mAb_2_sensitivity = .93;
mAb_3_sensitivity = .95;
mAb_4_sensitivity = .96;
mAb_5_sensitivity = .96;
hIgG_1_sensitivity = .98;
hIgG_2_sensitivity = .93;
hIgG_3_sensitivity = .93;
hIgG_4_sensitivity = .93;
hIgG_5_sensitivity = .93;

% 5b_mAb_1
%-------------------------------------------------------------------------
mAb_1 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_mAb_1.jpg');
mAb_1_gray = rgb2gray(mAb_1);
[mAb_1_center, mAb_1_radius] = imfindcircles(mAb_1_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_1_sensitivity,'method','twostage');
x = round(mAb_1_center(1,1));
y = round(mAb_1_center(1,2));
[rows columns z] = size(mAb_1);

figure(1);
imshow(mAb_1)
viscircles(mAb_1_center, mAb_1_radius,'color','r');

mAb_1_vector = [];
mAb_1_gray_vector = [];
for n = 1:1:rows
    mAb_1_vector = [mAb_1_vector mAb_1(n,:,1)];
end
for n = 1:1:rows
    mAb_1_gray_vector = [mAb_1_gray_vector mAb_1_gray(n,:)];
end
```

```
mAb_1_dist = [];
mAb_1_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_1_dist = [mAb_1_dist; row];
    mAb_1_dist_vector = [mAb_1_dist_vector row];
end

mAb_1_intensity = [];
for n = 1:1:max(mAb_1_dist_vector)
    m = find(mAb_1_dist_vector==n);
    average = mean(mAb_1_gray_vector(m));
    mAb_1_intensity = [mAb_1_intensity average];
end

bins = 1:1:max(mAb_1_dist_vector);
mAb_1_cm = bins*R/mAb_1_radius;

% 5b_mAb_15
%--------------------------------------------------------------------------
mAb_15 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_mAb_15.jpg');
mAb_15_gray = rgb2gray(mAb_15);
[mAb_15_center, mAb_15_radius] = imfindcircles(mAb_15_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_2_sensitivity,'method','twostage');
x = round(mAb_15_center(1,1));
y = round(mAb_15_center(1,2));
[rows columns z] = size(mAb_15);

figure(2);
imshow(mAb_15)
viscircles(mAb_15_center, mAb_15_radius,'color','r');

mAb_15_vector = [];
mAb_15_gray_vector = [];
for n = 1:1:rows
    mAb_15_vector = [mAb_15_vector mAb_15(n,:,1)];
end
for n = 1:1:rows
    mAb_15_gray_vector = [mAb_15_gray_vector mAb_15_gray(n,:)];
end

mAb_15_dist = [];
mAb_15_dist_vector = [];
```

```
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_15_dist = [mAb_15_dist; row];
    mAb_15_dist_vector = [mAb_15_dist_vector row];
end

mAb_15_intensity = [];
for n = 1:1:max(mAb_15_dist_vector)
    m = find(mAb_15_dist_vector==n);
    average = mean(mAb_15_gray_vector(m));
    mAb_15_intensity = [mAb_15_intensity average];
end

bins = 1:1:max(mAb_15_dist_vector);
mAb_15_cm = bins*R/mAb_15_radius;


% 5b_mAb_30
%--------------------------------------------------------------------------
mAb_30 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_mAb_30.jpg');
mAb_30_gray = rgb2gray(mAb_30);
[mAb_30_center, mAb_30_radius] = imfindcircles(mAb_30_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_3_sensitivity,'method','twostage');
x = round(mAb_30_center(1,1));
y = round(mAb_30_center(1,2));
[rows columns z] = size(mAb_30);

figure(3);
imshow(mAb_30)
viscircles(mAb_30_center, mAb_30_radius,'color','r');

mAb_30_vector = [];
mAb_30_gray_vector = [];
for n = 1:1:rows
    mAb_30_vector = [mAb_30_vector mAb_30(n,:,1)];
end
for n = 1:1:rows
    mAb_30_gray_vector = [mAb_30_gray_vector mAb_30_gray(n,:)];
end

mAb_30_dist = [];
mAb_30_dist_vector = [];
for n = 1:1:rows
    y2 = n;
```

```
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_30_dist = [mAb_30_dist; row];
    mAb_30_dist_vector = [mAb_30_dist_vector row];
end

mAb_30_intensity = [];
for n = 1:1:max(mAb_30_dist_vector)
    m = find(mAb_30_dist_vector==n);
    average = mean(mAb_30_gray_vector(m));
    mAb_30_intensity = [mAb_30_intensity average];
end

bins = 1:1:max(mAb_30_dist_vector);
mAb_30_cm = bins*R/mAb_30_radius;

% 5b_mAb_60
%--------------------------------------------------------------------------
mAb_60 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_mAb_60.jpg');
mAb_60_gray = rgb2gray(mAb_60);
[mAb_60_center, mAb_60_radius] = imfindcircles(mAb_60_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_4_sensitivity,'method','twostage');
x = round(mAb_60_center(1,1));
y = round(mAb_60_center(1,2));
[rows columns z] = size(mAb_60);

figure(4);
imshow(mAb_60)
viscircles(mAb_60_center, mAb_60_radius,'color','r');

mAb_60_vector = [];
mAb_60_gray_vector = [];
for n = 1:1:rows
    mAb_60_vector = [mAb_60_vector mAb_60(n,:,1)];
end
for n = 1:1:rows
    mAb_60_gray_vector = [mAb_60_gray_vector mAb_60_gray(n,:)];
end

mAb_60_dist = [];
mAb_60_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
```

```
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_60_dist = [mAb_60_dist; row];
    mAb_60_dist_vector = [mAb_60_dist_vector row];
end

mAb_60_intensity = [];
for n = 1:1:max(mAb_60_dist_vector)
    m = find(mAb_60_dist_vector==n);
    average = mean(mAb_60_gray_vector(m));
    mAb_60_intensity = [mAb_60_intensity average];
end

bins = 1:1:max(mAb_60_dist_vector);
mAb_60_cm = bins*R/mAb_60_radius;

% 5b_mAb_240
%-------------------------------------------------------------------------
mAb_240 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_mAb_240.jpg');
mAb_240_gray = rgb2gray(mAb_240);
[mAb_240_center, mAb_240_radius] = imfindcircles(mAb_240_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_5_sensitivity,'method','twostage');
x = round(mAb_240_center(1,1));
y = round(mAb_240_center(1,2));
[rows columns z] = size(mAb_240);

figure(5);
imshow(mAb_240)
viscircles(mAb_240_center, mAb_240_radius,'color','r');

mAb_240_vector = [];
mAb_240_gray_vector = [];
for n = 1:1:rows
    mAb_240_vector = [mAb_240_vector mAb_240(n,:,1)];
end
for n = 1:1:rows
    mAb_240_gray_vector = [mAb_240_gray_vector mAb_240_gray(n,:)];
end

mAb_240_dist = [];
mAb_240_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
```

```
    mAb_240_dist = [mAb_240_dist; row];
    mAb_240_dist_vector = [mAb_240_dist_vector row];
end

mAb_240_intensity = [];
for n = 1:1:max(mAb_240_dist_vector)
    m = find(mAb_240_dist_vector==n);
    average = mean(mAb_240_gray_vector(m));
    mAb_240_intensity = [mAb_240_intensity average];
end

bins = 1:1:max(mAb_240_dist_vector);
mAb_240_cm = bins*R/mAb_240_radius;

% 5b_hIgG_1
%--------------------------------------------------------------------------
hIgG_1 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_hIgG_1.jpg');
hIgG_1_gray = rgb2gray(hIgG_1);
[hIgG_1_center, hIgG_1_radius] = imfindcircles(hIgG_1_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_1_sensitivity,'method','twostage');
x = round(hIgG_1_center(1,1));
y = round(hIgG_1_center(1,2));
[rows columns z] = size(hIgG_1);

figure(6);
imshow(hIgG_1)
viscircles(hIgG_1_center, hIgG_1_radius,'color','r');

hIgG_1_vector = [];
hIgG_1_gray_vector = [];
for n = 1:1:rows
    hIgG_1_vector = [hIgG_1_vector hIgG_1(n,:,1)];
end
for n = 1:1:rows
    hIgG_1_gray_vector = [hIgG_1_gray_vector hIgG_1_gray(n,:)];
end

hIgG_1_dist = [];
hIgG_1_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_1_dist = [hIgG_1_dist; row];
    hIgG_1_dist_vector = [hIgG_1_dist_vector row];
end
```

```
hIgG_1_intensity = [];
for n = 1:1:max(hIgG_1_dist_vector)
    m = find(hIgG_1_dist_vector==n);
    average = mean(hIgG_1_gray_vector(m));
    hIgG_1_intensity = [hIgG_1_intensity average];
end

bins = 1:1:max(hIgG_1_dist_vector);
hIgG_1_cm = bins*R/hIgG_1_radius;

% 5b_hIgG_15
%-------------------------------------------------------------------------
hIgG_15 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_hIgG_15.jpg');
hIgG_15_gray = rgb2gray(hIgG_15);
[hIgG_15_center, hIgG_15_radius] = imfindcircles(hIgG_15_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_2_sensitivity,'method','twostage');
x = round(hIgG_15_center(1,1));
y = round(hIgG_15_center(1,2));
[rows columns z] = size(hIgG_15);

figure(7);
imshow(hIgG_15)
viscircles(hIgG_15_center, hIgG_15_radius,'color','r');

hIgG_15_vector = [];
hIgG_15_gray_vector = [];
for n = 1:1:rows
    hIgG_15_vector = [hIgG_15_vector hIgG_15(n,:,1)];
end
for n = 1:1:rows
    hIgG_15_gray_vector = [hIgG_15_gray_vector hIgG_15_gray(n,:)];
end

hIgG_15_dist = [];
hIgG_15_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_15_dist = [hIgG_15_dist; row];
    hIgG_15_dist_vector = [hIgG_15_dist_vector row];
end

hIgG_15_intensity = [];
for n = 1:1:max(hIgG_15_dist_vector)
```

```
    m = find(hIgG_15_dist_vector==n);
    average = mean(hIgG_15_gray_vector(m));
    hIgG_15_intensity = [hIgG_15_intensity average];
end

bins = 1:1:max(hIgG_15_dist_vector);
hIgG_15_cm = bins*R/hIgG_15_radius;


% 5b_hIgG_30
%-------------------------------------------------------------------------
hIgG_30 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_hIgG_30.jpg');
hIgG_30_gray = rgb2gray(hIgG_30);
[hIgG_30_center, hIgG_30_radius] = imfindcircles(hIgG_30_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_3_sensitivity,'method','twostage');
x = round(hIgG_30_center(1,1));
y = round(hIgG_30_center(1,2));
[rows columns z] = size(hIgG_30);

figure(8);
imshow(hIgG_30)
viscircles(hIgG_30_center, hIgG_30_radius,'color','r');

hIgG_30_vector = [];
hIgG_30_gray_vector = [];
for n = 1:1:rows
    hIgG_30_vector = [hIgG_30_vector hIgG_30(n,:,1)];
end
for n = 1:1:rows
    hIgG_30_gray_vector = [hIgG_30_gray_vector hIgG_30_gray(n,:)];
end

hIgG_30_dist = [];
hIgG_30_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_30_dist = [hIgG_30_dist; row];
    hIgG_30_dist_vector = [hIgG_30_dist_vector row];
end

hIgG_30_intensity = [];
for n = 1:1:max(hIgG_30_dist_vector)
    m = find(hIgG_30_dist_vector==n);
    average = mean(hIgG_30_gray_vector(m));
```

```
    hIgG_30_intensity = [hIgG_30_intensity average];
end

bins = 1:1:max(hIgG_30_dist_vector);
hIgG_30_cm = bins*R/hIgG_30_radius;

% 5b_hIgG_60
%-------------------------------------------------------------------------
hIgG_60 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_hIgG_60.jpg');
hIgG_60_gray = rgb2gray(hIgG_60);
[hIgG_60_center, hIgG_60_radius] = imfindcircles(hIgG_60_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_4_sensitivity,'method','twostage');
x = round(hIgG_60_center(1,1));
y = round(hIgG_60_center(1,2));
[rows columns z] = size(hIgG_60);

figure(9);
imshow(hIgG_60)
viscircles(hIgG_60_center, hIgG_60_radius,'color','r');

hIgG_60_vector = [];
hIgG_60_gray_vector = [];
for n = 1:1:rows
    hIgG_60_vector = [hIgG_60_vector hIgG_60(n,:,1)];
end
for n = 1:1:rows
    hIgG_60_gray_vector = [hIgG_60_gray_vector hIgG_60_gray(n,:)];
end

hIgG_60_dist = [];
hIgG_60_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_60_dist = [hIgG_60_dist; row];
    hIgG_60_dist_vector = [hIgG_60_dist_vector row];
end

hIgG_60_intensity = [];
for n = 1:1:max(hIgG_60_dist_vector)
    m = find(hIgG_60_dist_vector==n);
    average = mean(hIgG_60_gray_vector(m));
    hIgG_60_intensity = [hIgG_60_intensity average];
end
```

```
bins = 1:1:max(hIgG_60_dist_vector);
hIgG_60_cm = bins*R/hIgG_60_radius;

% 5b_hIgG_240
%-------------------------------------------------------------------------
hIgG_240 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5b\5b_hIgG_240.jpg');
hIgG_240_gray = rgb2gray(hIgG_240);
[hIgG_240_center, hIgG_240_radius] = imfindcircles(hIgG_240_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_5_sensitivity,'method','twostage');
x = round(hIgG_240_center(1,1));
y = round(hIgG_240_center(1,2));
[rows columns z] = size(hIgG_240);

figure(10);
imshow(hIgG_240)
viscircles(hIgG_240_center, hIgG_240_radius,'color','r');

hIgG_240_vector = [];
hIgG_240_gray_vector = [];
for n = 1:1:rows
    hIgG_240_vector = [hIgG_240_vector hIgG_240(n,:,1)];
end
for n = 1:1:rows
    hIgG_240_gray_vector = [hIgG_240_gray_vector hIgG_240_gray(n,:)];
end

hIgG_240_dist = [];
hIgG_240_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_240_dist = [hIgG_240_dist; row];
    hIgG_240_dist_vector = [hIgG_240_dist_vector row];
end

hIgG_240_intensity = [];
for n = 1:1:max(hIgG_240_dist_vector)
    m = find(hIgG_240_dist_vector==n);
    average = mean(hIgG_240_gray_vector(m));
    hIgG_240_intensity = [hIgG_240_intensity average];
end

bins = 1:1:max(hIgG_240_dist_vector);
hIgG_240_cm = bins*R/hIgG_240_radius;
```

```
% plot data
%-------------------------------------------------------------------------
figure(11);
plot(mAb_1_cm,mAb_1_intensity,'g',hIgG_1_cm,hIgG_1_intensity,'r',...
    -mAb_1_cm,mAb_1_intensity,'g',-hIgG_1_cm,hIgG_1_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('1 minute')
legend('mAb','hIgG','location','northwest')

figure(12);
plot(mAb_15_cm,mAb_15_intensity,'g',hIgG_15_cm,hIgG_15_intensity,'r',.
..
    -mAb_15_cm,mAb_15_intensity,'g',-hIgG_15_cm,hIgG_15_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('15 minutes')
legend('mAb','hIgG','location','northwest')

figure(13);
plot(mAb_30_cm,mAb_30_intensity,'g',hIgG_30_cm,hIgG_30_intensity,'r',.
..
    -mAb_30_cm,mAb_30_intensity,'g',-hIgG_30_cm,hIgG_30_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('30 minutes')
legend('mAb','hIgG','location','northwest')

figure(14);
plot(mAb_60_cm,mAb_60_intensity,'g',hIgG_60_cm,hIgG_60_intensity,'r',.
..
    -mAb_60_cm,mAb_60_intensity,'g',-hIgG_60_cm,hIgG_60_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('60 minutes')
legend('mAb','hIgG','location','northwest')

figure(15);
plot(mAb_240_cm,mAb_240_intensity,'g',hIgG_240_cm,hIgG_240_intensity,'
r',...
    -mAb_240_cm,mAb_240_intensity,'g',-
hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('240 minutes')
legend('mAb','hIgG','location','northwest')
```

```
figure(16);
plot(mAb_1_cm,mAb_1_intensity,'g',hIgG_1_cm,hIgG_1_intensity,'r',...

mAb_15_cm,mAb_15_intensity,'g',hIgG_15_cm,hIgG_15_intensity,'r',...

mAb_30_cm,mAb_30_intensity,'g',hIgG_30_cm,hIgG_30_intensity,'r',...

mAb_60_cm,mAb_60_intensity,'g',hIgG_60_cm,hIgG_60_intensity,'r',...

mAb_240_cm,mAb_240_intensity,'g',hIgG_240_cm,hIgG_240_intensity,'r',..
.
    -mAb_1_cm,mAb_1_intensity,'g',-hIgG_1_cm,hIgG_1_intensity,'r',...
    -mAb_15_cm,mAb_15_intensity,'g',-
hIgG_15_cm,hIgG_15_intensity,'r',...
    -mAb_30_cm,mAb_30_intensity,'g',-
hIgG_30_cm,hIgG_30_intensity,'r',...
    -mAb_60_cm,mAb_60_intensity,'g',-
hIgG_60_cm,hIgG_60_intensity,'r',...
    -mAb_240_cm,mAb_240_intensity,'g',-
hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('mAb','hIgG','location','northwest')

figure(17);
plot(mAb_1_cm,mAb_1_intensity,'g',...
    mAb_15_cm,mAb_15_intensity,'g',...
    mAb_30_cm,mAb_30_intensity,'g',...
    mAb_60_cm,mAb_60_intensity,'g',...
    mAb_240_cm,mAb_240_intensity,'g',...
    -mAb_1_cm,mAb_1_intensity,'g',...
    -mAb_15_cm,mAb_15_intensity,'g',...
    -mAb_30_cm,mAb_30_intensity,'g',...
    -mAb_60_cm,mAb_60_intensity,'g',...
    -mAb_240_cm,mAb_240_intensity,'g')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('mAb','location','northwest')

figure(18);
plot(hIgG_1_cm,hIgG_1_intensity,'r',...
    hIgG_15_cm,hIgG_15_intensity,'r',...
    hIgG_30_cm,hIgG_30_intensity,'r',...
    hIgG_60_cm,hIgG_60_intensity,'r',...
    hIgG_240_cm,hIgG_240_intensity,'r',...
    -hIgG_1_cm,hIgG_1_intensity,'r',...
    -hIgG_15_cm,hIgG_15_intensity,'r',...
```

```
    -hIgG_30_cm,hIgG_30_intensity,'r',...
    -hIgG_60_cm,hIgG_60_intensity,'r',...
    -hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('hIgG','location','northwest')


% save figures
%-------------------------------------------------------------------------
saveas(figure(11),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_1.jpg']);
saveas(figure(12),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_15.jpg']);
saveas(figure(13),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_30.jpg']);
saveas(figure(14),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_60.jpg']);
saveas(figure(15),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_240.jpg']);
saveas(figure(16),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_all.jpg']);
saveas(figure(17),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_mAb.jpg']);
saveas(figure(18),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5b\5b_hIgG.jpg']);
```

```
%-------------------------------------------------------------------------
% image_5c.m
%
% script quantitatively analyzes experimental images to generate plots
% of color intensity vs. radial position
% MabSelect SuRe chromatography media
% Images taken from Weinberg et al. 2017
%
% created by Virginia Lane on 10/10/18
% last updated on 10/10/18
%-------------------------------------------------------------------------

% parameters
%-------------------------------------------------------------------------
Rmin = 95;
Rmax = 110;
R = .0043;
mAb_1_sensitivity = .95;
mAb_2_sensitivity = .95;
mAb_3_sensitivity = .96;
mAb_4_sensitivity = .96;
mAb_5_sensitivity = .96;
hIgG_1_sensitivity = .96;
hIgG_2_sensitivity = .95;
hIgG_3_sensitivity = .93;
hIgG_4_sensitivity = .95;
hIgG_5_sensitivity = .93;

% 5c_mAb_1
%-------------------------------------------------------------------------
mAb_1 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_mAb_1.jpg');
mAb_1_gray = rgb2gray(mAb_1);
[mAb_1_center, mAb_1_radius] = imfindcircles(mAb_1_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_1_sensitivity,'method','twostage');
x = round(mAb_1_center(1,1));
y = round(mAb_1_center(1,2));
[rows columns z] = size(mAb_1);

figure(1);
imshow(mAb_1)
viscircles(mAb_1_center, mAb_1_radius,'color','r');

mAb_1_vector = [];
mAb_1_gray_vector = [];
for n = 1:1:rows
    mAb_1_vector = [mAb_1_vector mAb_1(n,:,1)];
end
for n = 1:1:rows
    mAb_1_gray_vector = [mAb_1_gray_vector mAb_1_gray(n,:)];
end
```

```
mAb_1_dist = [];
mAb_1_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_1_dist = [mAb_1_dist; row];
    mAb_1_dist_vector = [mAb_1_dist_vector row];
end

mAb_1_intensity = [];
for n = 1:1:max(mAb_1_dist_vector)
    m = find(mAb_1_dist_vector==n);
    average = mean(mAb_1_gray_vector(m));
    mAb_1_intensity = [mAb_1_intensity average];
end

bins = 1:1:max(mAb_1_dist_vector);
mAb_1_cm = bins*R/mAb_1_radius;

% 5c_mAb_15
%--------------------------------------------------------------------------
mAb_15 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_mAb_15.jpg');
mAb_15_gray = rgb2gray(mAb_15);
[mAb_15_center, mAb_15_radius] = imfindcircles(mAb_15_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_2_sensitivity,'method','twostage');
x = round(mAb_15_center(1,1));
y = round(mAb_15_center(1,2));
[rows columns z] = size(mAb_15);

figure(2);
imshow(mAb_15)
viscircles(mAb_15_center, mAb_15_radius,'color','r');

mAb_15_vector = [];
mAb_15_gray_vector = [];
for n = 1:1:rows
    mAb_15_vector = [mAb_15_vector mAb_15(n,:,1)];
end
for n = 1:1:rows
    mAb_15_gray_vector = [mAb_15_gray_vector mAb_15_gray(n,:)];
end

mAb_15_dist = [];
mAb_15_dist_vector = [];
```

```
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_15_dist = [mAb_15_dist; row];
    mAb_15_dist_vector = [mAb_15_dist_vector row];
end

mAb_15_intensity = [];
for n = 1:1:max(mAb_15_dist_vector)
    m = find(mAb_15_dist_vector==n);
    average = mean(mAb_15_gray_vector(m));
    mAb_15_intensity = [mAb_15_intensity average];
end

bins = 1:1:max(mAb_15_dist_vector);
mAb_15_cm = bins*R/mAb_15_radius;


% 5c_mAb_30
%-------------------------------------------------------------------------
mAb_30 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_mAb_30.jpg');
mAb_30_gray = rgb2gray(mAb_30);
[mAb_30_center, mAb_30_radius] = imfindcircles(mAb_30_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_3_sensitivity,'method','twostage');
x = round(mAb_30_center(1,1));
y = round(mAb_30_center(1,2));
[rows columns z] = size(mAb_30);

figure(3);
imshow(mAb_30)
viscircles(mAb_30_center, mAb_30_radius,'color','r');

mAb_30_vector = [];
mAb_30_gray_vector = [];
for n = 1:1:rows
    mAb_30_vector = [mAb_30_vector mAb_30(n,:,1)];
end
for n = 1:1:rows
    mAb_30_gray_vector = [mAb_30_gray_vector mAb_30_gray(n,:)];
end

mAb_30_dist = [];
mAb_30_dist_vector = [];
for n = 1:1:rows
    y2 = n;
```

```
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_30_dist = [mAb_30_dist; row];
    mAb_30_dist_vector = [mAb_30_dist_vector row];
end

mAb_30_intensity = [];
for n = 1:1:max(mAb_30_dist_vector)
    m = find(mAb_30_dist_vector==n);
    average = mean(mAb_30_gray_vector(m));
    mAb_30_intensity = [mAb_30_intensity average];
end

bins = 1:1:max(mAb_30_dist_vector);
mAb_30_cm = bins*R/mAb_30_radius;

% 5c_mAb_60
%--------------------------------------------------------------------------
mAb_60 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_mAb_60.jpg');
mAb_60_gray = rgb2gray(mAb_60);
[mAb_60_center, mAb_60_radius] = imfindcircles(mAb_60_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_4_sensitivity,'method','twostage');
x = round(mAb_60_center(1,1));
y = round(mAb_60_center(1,2));
[rows columns z] = size(mAb_60);

figure(4);
imshow(mAb_60)
viscircles(mAb_60_center, mAb_60_radius,'color','r');

mAb_60_vector = [];
mAb_60_gray_vector = [];
for n = 1:1:rows
    mAb_60_vector = [mAb_60_vector mAb_60(n,:,1)];
end
for n = 1:1:rows
    mAb_60_gray_vector = [mAb_60_gray_vector mAb_60_gray(n,:)];
end

mAb_60_dist = [];
mAb_60_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
```

```
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    mAb_60_dist = [mAb_60_dist; row];
    mAb_60_dist_vector = [mAb_60_dist_vector row];
end

mAb_60_intensity = [];
for n = 1:1:max(mAb_60_dist_vector)
    m = find(mAb_60_dist_vector==n);
    average = mean(mAb_60_gray_vector(m));
    mAb_60_intensity = [mAb_60_intensity average];
end

bins = 1:1:max(mAb_60_dist_vector);
mAb_60_cm = bins*R/mAb_60_radius;

% 5c_mAb_240
%--------------------------------------------------------------------------
mAb_240 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_mAb_240.jpg');
mAb_240_gray = rgb2gray(mAb_240);
[mAb_240_center, mAb_240_radius] = imfindcircles(mAb_240_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    mAb_5_sensitivity,'method','twostage');
x = round(mAb_240_center(1,1));
y = round(mAb_240_center(1,2));
[rows columns z] = size(mAb_240);

figure(5);
imshow(mAb_240)
viscircles(mAb_240_center, mAb_240_radius,'color','r');

mAb_240_vector = [];
mAb_240_gray_vector = [];
for n = 1:1:rows
    mAb_240_vector = [mAb_240_vector mAb_240(n,:,1)];
end
for n = 1:1:rows
    mAb_240_gray_vector = [mAb_240_gray_vector mAb_240_gray(n,:)];
end

mAb_240_dist = [];
mAb_240_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
```

```
    mAb_240_dist = [mAb_240_dist; row];
    mAb_240_dist_vector = [mAb_240_dist_vector row];
end

mAb_240_intensity = [];
for n = 1:1:max(mAb_240_dist_vector)
    m = find(mAb_240_dist_vector==n);
    average = mean(mAb_240_gray_vector(m));
    mAb_240_intensity = [mAb_240_intensity average];
end

bins = 1:1:max(mAb_240_dist_vector);
mAb_240_cm = bins*R/mAb_240_radius;

% 5c_hIgG_1
%--------------------------------------------------------------------------
hIgG_1 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_hIgG_1.jpg');
hIgG_1_gray = rgb2gray(hIgG_1);
[hIgG_1_center, hIgG_1_radius] = imfindcircles(hIgG_1_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_1_sensitivity,'method','twostage');
x = round(hIgG_1_center(1,1));
y = round(hIgG_1_center(1,2));
[rows columns z] = size(hIgG_1);

figure(6);
imshow(hIgG_1)
viscircles(hIgG_1_center, hIgG_1_radius,'color','r');

hIgG_1_vector = [];
hIgG_1_gray_vector = [];
for n = 1:1:rows
    hIgG_1_vector = [hIgG_1_vector hIgG_1(n,:,1)];
end
for n = 1:1:rows
    hIgG_1_gray_vector = [hIgG_1_gray_vector hIgG_1_gray(n,:)];
end

hIgG_1_dist = [];
hIgG_1_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_1_dist = [hIgG_1_dist; row];
    hIgG_1_dist_vector = [hIgG_1_dist_vector row];
end
```

```
hIgG_1_intensity = [];
for n = 1:1:max(hIgG_1_dist_vector)
    m = find(hIgG_1_dist_vector==n);
    average = mean(hIgG_1_gray_vector(m));
    hIgG_1_intensity = [hIgG_1_intensity average];
end

bins = 1:1:max(hIgG_1_dist_vector);
hIgG_1_cm = bins*R/hIgG_1_radius;

% 5c_hIgG_15
%--------------------------------------------------------------------------
hIgG_15 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_hIgG_15.jpg');
hIgG_15_gray = rgb2gray(hIgG_15);
[hIgG_15_center, hIgG_15_radius] = imfindcircles(hIgG_15_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_2_sensitivity,'method','twostage');
x = round(hIgG_15_center(1,1));
y = round(hIgG_15_center(1,2));
[rows columns z] = size(hIgG_15);

figure(7);
imshow(hIgG_15)
viscircles(hIgG_15_center, hIgG_15_radius,'color','r');

hIgG_15_vector = [];
hIgG_15_gray_vector = [];
for n = 1:1:rows
    hIgG_15_vector = [hIgG_15_vector hIgG_15(n,:,1)];
end
for n = 1:1:rows
    hIgG_15_gray_vector = [hIgG_15_gray_vector hIgG_15_gray(n,:)];
end

hIgG_15_dist = [];
hIgG_15_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_15_dist = [hIgG_15_dist; row];
    hIgG_15_dist_vector = [hIgG_15_dist_vector row];
end

hIgG_15_intensity = [];
for n = 1:1:max(hIgG_15_dist_vector)
```

```
    m = find(hIgG_15_dist_vector==n);
    average = mean(hIgG_15_gray_vector(m));
    hIgG_15_intensity = [hIgG_15_intensity average];
end

bins = 1:1:max(hIgG_15_dist_vector);
hIgG_15_cm = bins*R/hIgG_15_radius;


% 5c_hIgG_30
%-------------------------------------------------------------------------
hIgG_30 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_hIgG_30.jpg');
hIgG_30_gray = rgb2gray(hIgG_30);
[hIgG_30_center, hIgG_30_radius] = imfindcircles(hIgG_30_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_3_sensitivity,'method','twostage');
x = round(hIgG_30_center(1,1));
y = round(hIgG_30_center(1,2));
[rows columns z] = size(hIgG_30);

figure(8);
imshow(hIgG_30)
viscircles(hIgG_30_center, hIgG_30_radius,'color','r');

hIgG_30_vector = [];
hIgG_30_gray_vector = [];
for n = 1:1:rows
    hIgG_30_vector = [hIgG_30_vector hIgG_30(n,:,1)];
end
for n = 1:1:rows
    hIgG_30_gray_vector = [hIgG_30_gray_vector hIgG_30_gray(n,:)];
end

hIgG_30_dist = [];
hIgG_30_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_30_dist = [hIgG_30_dist; row];
    hIgG_30_dist_vector = [hIgG_30_dist_vector row];
end

hIgG_30_intensity = [];
for n = 1:1:max(hIgG_30_dist_vector)
    m = find(hIgG_30_dist_vector==n);
    average = mean(hIgG_30_gray_vector(m));
```

```
    hIgG_30_intensity = [hIgG_30_intensity average];
end

bins = 1:1:max(hIgG_30_dist_vector);
hIgG_30_cm = bins*R/hIgG_30_radius;

% 5c_hIgG_60
%-------------------------------------------------------------------------
hIgG_60 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_hIgG_60.jpg');
hIgG_60_gray = rgb2gray(hIgG_60);
[hIgG_60_center, hIgG_60_radius] = imfindcircles(hIgG_60_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_4_sensitivity,'method','twostage');
x = round(hIgG_60_center(1,1));
y = round(hIgG_60_center(1,2));
[rows columns z] = size(hIgG_60);

figure(9);
imshow(hIgG_60)
viscircles(hIgG_60_center, hIgG_60_radius,'color','r');

hIgG_60_vector = [];
hIgG_60_gray_vector = [];
for n = 1:1:rows
    hIgG_60_vector = [hIgG_60_vector hIgG_60(n,:,1)];
end
for n = 1:1:rows
    hIgG_60_gray_vector = [hIgG_60_gray_vector hIgG_60_gray(n,:)];
end

hIgG_60_dist = [];
hIgG_60_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_60_dist = [hIgG_60_dist; row];
    hIgG_60_dist_vector = [hIgG_60_dist_vector row];
end

hIgG_60_intensity = [];
for n = 1:1:max(hIgG_60_dist_vector)
    m = find(hIgG_60_dist_vector==n);
    average = mean(hIgG_60_gray_vector(m));
    hIgG_60_intensity = [hIgG_60_intensity average];
end
```

```
bins = 1:1:max(hIgG_60_dist_vector);
hIgG_60_cm = bins*R/hIgG_60_radius;

% 5c_hIgG_240
%-------------------------------------------------------------------------
hIgG_240 = imread('C:\users\Virginia
Lane\Desktop\figures\experimental\5c\5c_hIgG_240.jpg');
hIgG_240_gray = rgb2gray(hIgG_240);
[hIgG_240_center, hIgG_240_radius] = imfindcircles(hIgG_240_gray,...
    [Rmin Rmax], 'ObjectPolarity','bright','sensitivity',...
    hIgG_5_sensitivity,'method','twostage');
x = round(hIgG_240_center(1,1));
y = round(hIgG_240_center(1,2));
[rows columns z] = size(hIgG_240);

figure(10);
imshow(hIgG_240)
viscircles(hIgG_240_center, hIgG_240_radius,'color','r');

hIgG_240_vector = [];
hIgG_240_gray_vector = [];
for n = 1:1:rows
    hIgG_240_vector = [hIgG_240_vector hIgG_240(n,:,1)];
end
for n = 1:1:rows
    hIgG_240_gray_vector = [hIgG_240_gray_vector hIgG_240_gray(n,:)];
end

hIgG_240_dist = [];
hIgG_240_dist_vector = [];
for n = 1:1:rows
    y2 = n;
    row = [];
    for m = 1:1:columns
        x2 = m;
        D = ((x2-x)^2+(y2-y)^2)^.5;
        row = [row round(D)];
    end
    hIgG_240_dist = [hIgG_240_dist; row];
    hIgG_240_dist_vector = [hIgG_240_dist_vector row];
end

hIgG_240_intensity = [];
for n = 1:1:max(hIgG_240_dist_vector)
    m = find(hIgG_240_dist_vector==n);
    average = mean(hIgG_240_gray_vector(m));
    hIgG_240_intensity = [hIgG_240_intensity average];
end

bins = 1:1:max(hIgG_240_dist_vector);
hIgG_240_cm = bins*R/hIgG_240_radius;
```

```
% plot data
%-------------------------------------------------------------------------
figure(11);
plot(mAb_1_cm,mAb_1_intensity,'g',hIgG_1_cm,hIgG_1_intensity,'r',...
    -mAb_1_cm,mAb_1_intensity,'g',-hIgG_1_cm,hIgG_1_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('1 minute')
legend('mAb','hIgG','location','northwest')

figure(12);
plot(mAb_15_cm,mAb_15_intensity,'g',hIgG_15_cm,hIgG_15_intensity,'r',.
..
    -mAb_15_cm,mAb_15_intensity,'g',-hIgG_15_cm,hIgG_15_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('15 minutes')
legend('mAb','hIgG','location','northwest')

figure(13);
plot(mAb_30_cm,mAb_30_intensity,'g',hIgG_30_cm,hIgG_30_intensity,'r',.
..
    -mAb_30_cm,mAb_30_intensity,'g',-hIgG_30_cm,hIgG_30_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('30 minutes')
legend('mAb','hIgG','location','northwest')

figure(14);
plot(mAb_60_cm,mAb_60_intensity,'g',hIgG_60_cm,hIgG_60_intensity,'r',.
..
    -mAb_60_cm,mAb_60_intensity,'g',-hIgG_60_cm,hIgG_60_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('60 minutes')
legend('mAb','hIgG','location','northwest')

figure(15);
plot(mAb_240_cm,mAb_240_intensity,'g',hIgG_240_cm,hIgG_240_intensity,'
r',...
    -mAb_240_cm,mAb_240_intensity,'g',-
hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('240 minutes')
legend('mAb','hIgG','location','northwest')
```

```
figure(16);
plot(mAb_1_cm,mAb_1_intensity,'g',hIgG_1_cm,hIgG_1_intensity,'r',...

mAb_15_cm,mAb_15_intensity,'g',hIgG_15_cm,hIgG_15_intensity,'r',...

mAb_30_cm,mAb_30_intensity,'g',hIgG_30_cm,hIgG_30_intensity,'r',...

mAb_60_cm,mAb_60_intensity,'g',hIgG_60_cm,hIgG_60_intensity,'r',...

mAb_240_cm,mAb_240_intensity,'g',hIgG_240_cm,hIgG_240_intensity,'r',..
.
    -mAb_1_cm,mAb_1_intensity,'g',-hIgG_1_cm,hIgG_1_intensity,'r',...
    -mAb_15_cm,mAb_15_intensity,'g',-
hIgG_15_cm,hIgG_15_intensity,'r',...
    -mAb_30_cm,mAb_30_intensity,'g',-
hIgG_30_cm,hIgG_30_intensity,'r',...
    -mAb_60_cm,mAb_60_intensity,'g',-
hIgG_60_cm,hIgG_60_intensity,'r',...
    -mAb_240_cm,mAb_240_intensity,'g',-
hIgG_240_cm,hIgG_240_intensity,'r','LineWidth',2)
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('mAb','hIgG','location','northwest')
set(gca,'fontsize',16)

figure(17);
plot(mAb_1_cm,mAb_1_intensity,'g',...
    mAb_15_cm,mAb_15_intensity,'g',...
    mAb_30_cm,mAb_30_intensity,'g',...
    mAb_60_cm,mAb_60_intensity,'g',...
    mAb_240_cm,mAb_240_intensity,'g',...
    -mAb_1_cm,mAb_1_intensity,'g',...
    -mAb_15_cm,mAb_15_intensity,'g',...
    -mAb_30_cm,mAb_30_intensity,'g',...
    -mAb_60_cm,mAb_60_intensity,'g',...
    -mAb_240_cm,mAb_240_intensity,'g')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('mAb','location','northwest')

figure(18);
plot(hIgG_1_cm,hIgG_1_intensity,'r',...
    hIgG_15_cm,hIgG_15_intensity,'r',...
    hIgG_30_cm,hIgG_30_intensity,'r',...
    hIgG_60_cm,hIgG_60_intensity,'r',...
    hIgG_240_cm,hIgG_240_intensity,'r',...
    -hIgG_1_cm,hIgG_1_intensity,'r',...
```

```
    -hIgG_15_cm,hIgG_15_intensity,'r',...
    -hIgG_30_cm,hIgG_30_intensity,'r',...
    -hIgG_60_cm,hIgG_60_intensity,'r',...
    -hIgG_240_cm,hIgG_240_intensity,'r')
xlim([-.0043 .0043]);
xlabel('radius (cm)')
ylabel('average intensity')
title('intensity vs. radius')
legend('hIgG','location','northwest')


% save figures
%-------------------------------------------------------------------------
saveas(figure(11),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_1.jpg']);
saveas(figure(12),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_15.jpg']);
saveas(figure(13),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_30.jpg']);
saveas(figure(14),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_60.jpg']);
saveas(figure(15),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_240.jpg']);
saveas(figure(16),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_all.jpg']);
saveas(figure(17),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_mAb.jpg']);
saveas(figure(18),['C:\users\Virginia
Lane\Desktop\figures\Experimental\5c\5c_hIgG.jpg']);
```