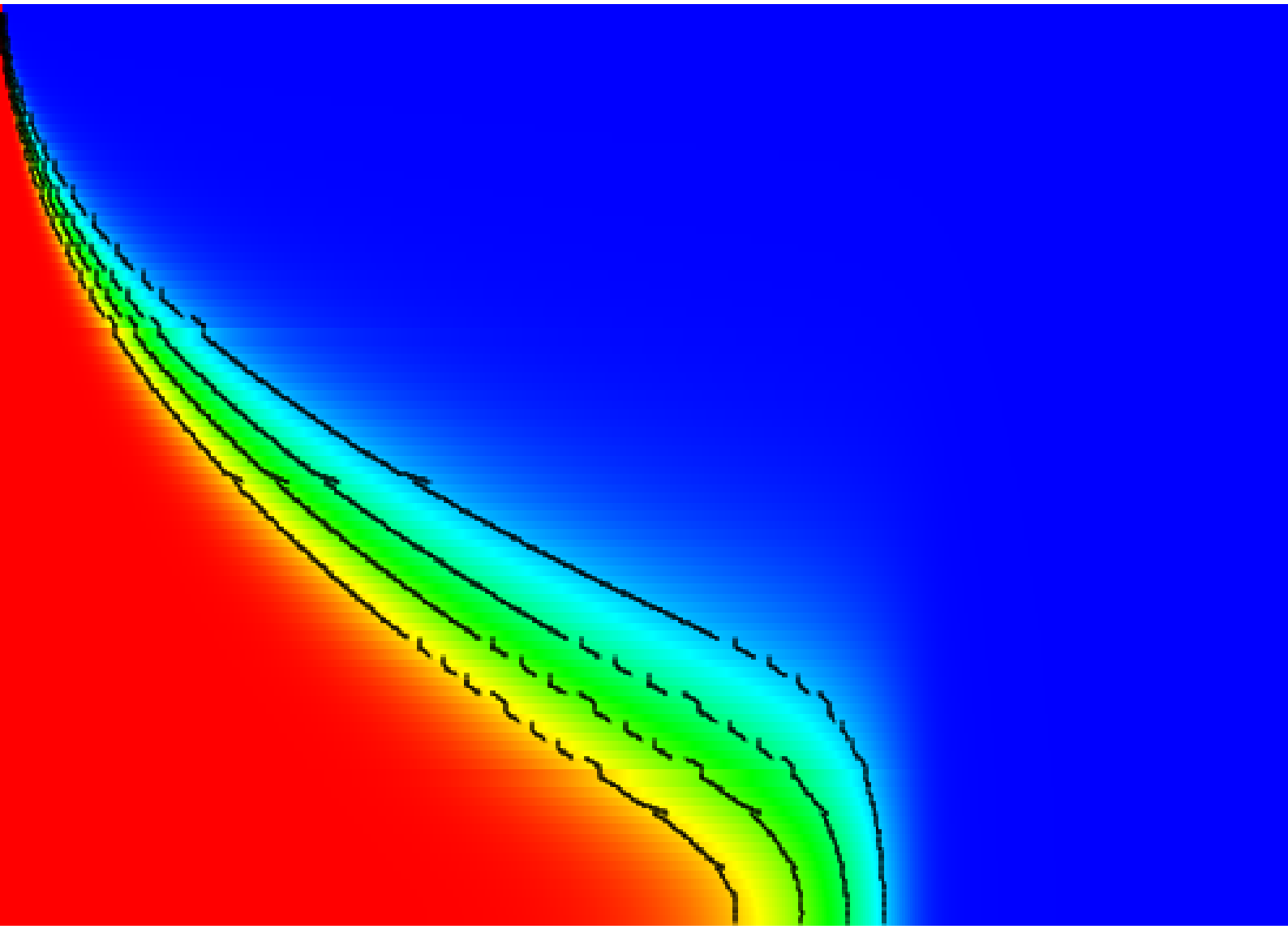
Masterthesis

Hydrogels: pH-sensitive swelling and use as a sensor

Pieter Biewenga



Supervisors:

S. Van Berkum
B.H. Erné

December 2012 - June 2014

Van 't Hoff laboratory for physical and colloid chemistry
Utrecht University

Abstract

In this work pH-sensitive magnetic hydrogels are investigated for their potential as chemical sensors in aqueous media. For this purpose thin layers of pH-sensitive hydrogels were made that incorporate magnetic particles. It was found that the magnetic field caused by these magnetic hydrogels changes as a function of pH. These magnetic fields are compared with theory.

The swelling process in pH-sensitive hydrogels is discussed in more detail based on numerical models. These models determine that the swelling and shrinking of a pH-sensitive hydrogel are two different processes with their own dynamics. These differing dynamics cause the shrinking process to occur at a much higher speed than the swelling process.

Table of Contents

Abstract 3
1. Introduction 5
2. Theory 6
2.1 Hydrogels 6
2.1.1 General 6
2.1.2 The acrylic acid/HEA gel 6
2.2 Magnetic signal 8
2.2.1 Relaxation 8
2.2.2 Magnetic field at the sensor 9
2.2.3 Effects of swelling on the magnetic field 11
3. Experimental: Methods 13
3.1 Materials 13
3.2 Bulk Swelling 13
3.3 Synthesis of thin magnetic hydrogels 13
3.4 Gauss probe measurements 14
4. Experimental: Results 16
4.1 Bulk swelling 16
4.2 Thickness dependent measurements 18
4.3 pH dependence of relaxation 19
5. Calculations: Methods 21
5.1 Diffusion model 21
5.1.1 Calculation method 21
5.1.2 Starting configuration 22
5.1.3 Diffusion, theory 22
5.1.4 Diffusion, calculation 24
5.1.5 Acid-base equilibrium 26
5.1.6 Measuring gel size 29
5.2 Buffered model 30
5.2.1 Calculation method 30
5.2.2 Hydrogen ion diffusion: theory 31
5.2.3 Hydrogen ion diffusion: calculation 33
5.2.4 Donnan theory 35
5.2.e Gel size 36
6. Calculations: Results 39
6.1 Diffusion model 39
6.1.1 Initial configuration 39
6.1.2 Size 39
6.2 Buffered model 40
6.2.1 Initial configuration 40
6.2.2 Size 41
Conclusions 45
Bibliography 46
Appendix A: Diffusion model computer code 47
Appendix B: Buffered model computer code 57

1. Introduction

A hydrogel is a polymer network swollen with water. The amount of water inside the gel depends on the properties of the hydrogel. Currently hydrogels are being investigated for a number of different applications including drug delivery[1] and treatment of injuries.[2]

One specific kind of gel is a stimulus-sensitive hydrogel. These kinds of gels respond to external stimuli by changing their chemical or physical properties. Stimuli can include pH, temperature, concentrations of ions, electrical field [3], composition of the solvent, and light. Hydrogels responsive to specific biomolecules can also be designed. [4]

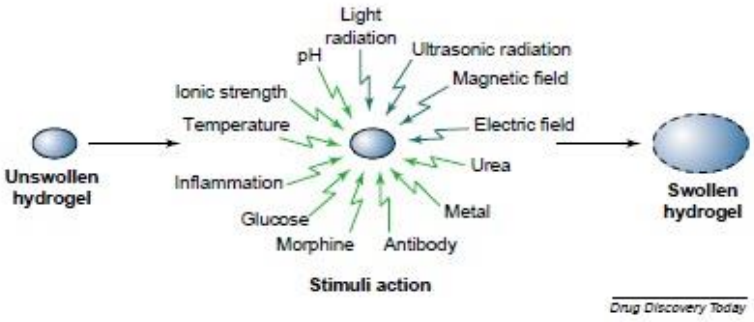


Figure 1: *Schematic illustration of a number of different stimuli changing the size of a hydrogel. Couresy of P. Gupta et al.[7]*

This property of stimulus-sensitive hydrogels can be used for sensing applications. [5] One example of a hydrogel being considered for sensing applications is a pH sensitive hydrogel.[6] Such a hydrogel often works by incorporating weakly acidic or basic groups into the polymer structure.

One of the problems in using hydrogels for sensing applications is the signal transducer. A signal transducer is what translates the non-electrical changes in property into a signal that can be interpreted by a sensor. One proposal for a signal transducer for a hydrogel sensor is to measure the magnetic response of superparamagnetic particles incorporated into the hydrogel. [8] This proposal uses alternating magnetic fields which are relatively difficult to use. In this work I will focus on a variant to this proposal that uses static magnetic fields rather than alternating magnetic fields.

A static magnetic field could come from ferromagnetic particles that have been bound to the hydrogel's polymer network. If these particles are bound to this network they cannot rotate, meaning that any loss of magnetization will occur much more slowly. If the change in the size of the hydrogel will lead to a difference in the strength of the magnetic field this could be measured and used to measure the size of the hydrogel.

In this work I will look into various aspects of pH sensitive hydrogels. First I will look into the relationship between the physical size of a hydrogel incorporating magnetized ferromagnetic particles and the strength of its magnetic field. This will be treated both from theory and experiment.

The other main goal of this work is to provide a theoretical model to describe the swelling process in a pH sensitive hydrogel. Especially the time-dependent description of the swelling process will be considered. To achieve this, numerical calculations of the time-dependent swelling of a pH sensitive hydrogel were done and will be treated in this work.

2. Theory

In the first part of this chapter the general mechanism of the swelling and shrinking of hydrogels is discussed as well as the model system used in this research. In the second part the theory behind magnetic hydrogels is discussed, this includes both the way in which these magnetic hydrogels are made and the field that can be expected for a given hydrogel.

2.1 Hydrogels

2.1.1 General

A hydrogel is a polymer network swollen with water. The amount of water inside this gel depends on the properties of this hydrogel. In the case of a pH-sensitive hydrogel the water content depends on the pH. A hydrogel responsive to pH can be made by including (co)monomers with weak acidic or weak basic sidegroups. These sidegroups have a charge that will depend on their pH. As these sidegroups become charged there must be additional counterions present to make the gel electrically neutral. These counterions however cause an osmotic pressure difference between the gel and the solution. As a result of this osmotic pressure the gel will take up solvent and swell until the elastic forces in the hydrogel are in equilibrium with the osmotic force. [9]

This pH-dependent swelling behavior is also dependent on one other very important parameter: The ionic strength of the solution. This can be seen in the most extreme case where, apart from water only OH^- and H^+ are present in the solution. In this case if the hydrogel contains weakly acidic sidegroups these sidegroups will release H^+, but the pH will rapidly drop causing hardly any swelling. This is because there must be other ions present for which H^+ can be exchanged. This means that a gel will hardly swell if very few ions are present in the solution. When a lot of ions are present in the solution (i.e. the solution has a high ionic strength) swelling is also problematic as the solution outside of the gel already has a very high osmotic pressure and the osmotic pressure inside the gel does not differ significantly from that outside the gel. [5] Therefore the pH-sensitive sensor is also sensitive to ionic strength and the ionic strength needs to be controlled during experiments.

2.1.2 The acrylic acid/HEA gel

The hydrogel used for this research has acrylic acid monomer groups to act as the weak acid group. This is combined with a Hydroxyethyl acrylate (HEA) comonomer, HEA does not act as an acid or base group at low pH and is added to keep the effect of swelling within limits. Furthermore Diethylene glycol diacrylate (DEGDA) is added as a crosslinker monomer.

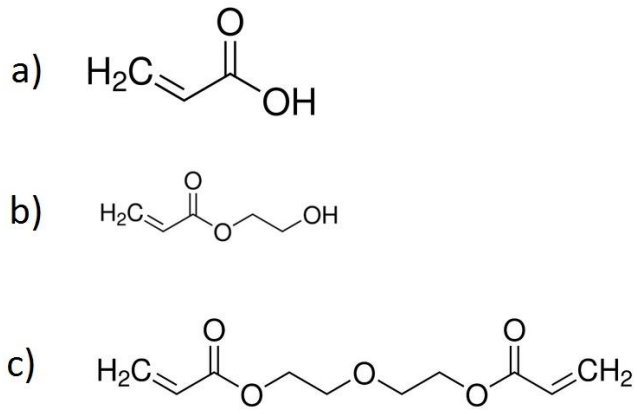
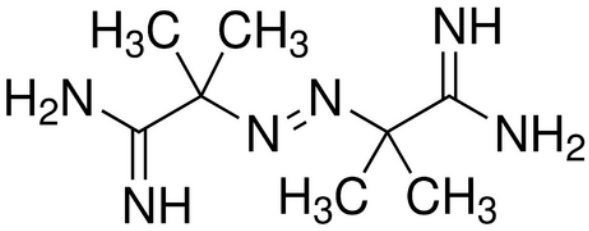


***Figure 2**: monomers used: **a)** acrylic acid **b)** hydroxyethyl acrylate **c)** diethylene glycol diacrylate*

The monomers are polymerized through radical polymerization. In this process the carbon-carbon double bond is attacked by a free radical causing the free radical to bond to the monomer. This newly formed unit can then act as a radical itself and continue the polymerization reaction. The reaction is stopped when two radicals react with each other. To create radicals 2,2'-azobis(2-methylpropionamidine)dihydrochloride (V-50) is used, the molecular structure is given in figure 2.



V-50 has a C-N=N-C bond which is unstable at elevated temperatures. When V-50 is heated it will break up into molecular nitrogen and two radical groups. This allows the use of V-50 as an initiator simply by heating it.

Gels for magnetic study were prepared attached to a glass substrate. To achieve this the glass substrate was coated with 3-(Trimethoxysilyl)propyl methacrylate (TPM) . TPM (figure 3) is an ester of methacrylic acid which is very similar to acrylic acid, it also has a trimethoxysilyl group that can react with regular glass under acidic conditions. This coats the glass with methacrylate groups that can react in the same way as the polymer monomers. This covalently binds the gel to the glass substrate allowing better study of the gel's properties as its position is more accurately determined because of its binding to the glass substrate. This is especially important for thin layers of gel, it is preferred when the gel is relatively thin as the entire swelling and shrinking process is a diffusion-driven process. This means that the swelling rate is proportional to the square of the thickness of the gel.

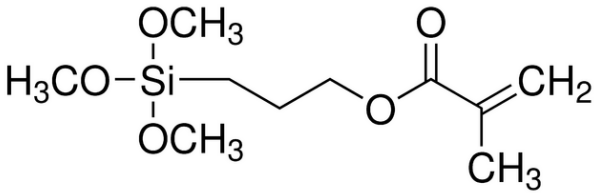


***Figure 3:** 3-(Trimethoxysilyl)propyl methacrylate (TPM)*

2.2 Magnetic signal

To understand how the signal from a magnetic gel relates to its size it is good to look into the theory behind the magnetic signal. Ferromagnetic bulk materials consist of many different magnetic domains which all have different orientations. This means that bulk amounts of these materials do not have a net magnetic moment. However colloids with a size smaller than the magnetic domain size of the material have only a single magnetic domain. These single domain particles therefore have a permanent magnetic moment and their behavior in a solvent can be described as superparamagnetic. This means that a dispersion of these single domain particles does not have a fixed magnetic moment because the particles move randomly due thermal motion, but that the particles readily align in an external magnetic field. The overall dispersion then also quickly loses this magnetic moment once the external magnetic field is removed through relaxation.

2.2.1 Relaxation

Relaxation of a collection of single domain magnetic particles can occur through two different mechanisms, namely Néel relaxation and Brownian relaxation. Néel relaxation happens when the magnetic axis relaxes within a particle while Brownian relaxation happens when the particle as a whole rotates. The variation of the magnetization of an magnetic system can generally be described through[10]:

$$M(t) = M(t_0)e^{-t/\tau_{\text{eff}}} \quad 1.$$

Where M(t) is the magnetization at time t and τ_{eff} is an effective relaxation time. This relaxation time depends both on a characteristic Néel frequency and a characteristic Brownian frequency:

$$\frac{1}{\tau_{\text{eff}}} = \frac{1}{\tau_N} + \frac{1}{\tau_B} \quad 2.$$

In practice the larger of the two frequencies usually dominates. The lowest energy magnetization of a single domain magnetic particle lays on a certain easy axis, magnetizing the domain in a different direction means that the particle will try to reorient its magnetization into the direction of this easy axis. This is the driving force behind Néel relaxation where the relaxation frequency can be described using[8]:

$$\frac{1}{\tau_N} = f_0 \exp\left(-\frac{KV_p}{K_B T}\right) \quad 3.$$

Where KV_p is an energy barrier between the current state and the relaxed state. f_0 is the attempt rate which is often assumed to be in the order of 10^{-9} but is actually not very well known, this means that the above equation primarily gives a qualitative description of the Néel relaxation process. The

constant K is the anisotropy constant of the particle, this anisotropy consists of bulk, surface and shape anisotropy, there can be multiple of these anisotropy constants if there are multiple routes for relaxation, but usually only the dominant constant is taken into consideration. V_p denotes the volume of the magnetic particle. When looking at the Néel relaxation it becomes apparent that the relaxation frequency strongly depends on the particle size, namely as an exponential of the diameter to the power three, so Néel relaxation dominates for smaller particles. Brownian relaxation is given by:

$$\frac{1}{\tau_B} = \frac{K_B T}{3\eta \mathrm{V_H}} \qquad 4.$$

Here η is the solvent viscosity and $\mathrm{V_H}$ the hydrodynamic volume of the magnetic particle. This may be different from the magnetic volume given by V_p earlier. For the magnetic hydrogels it is assumed that the magnetic particles are locked in place by their bonds to the polymer network, this would mean that only Néel relaxation should be present in the system. [11]

2.2.2 Magnetic field at the sensor

To describe the field of a hydrogel containing magnetic particles one can consider the entire gel as one magnet. This magnet can be described using magnetic 'charges', similar to electric charges. These 'charges' are spread out over the poles of the magnet.

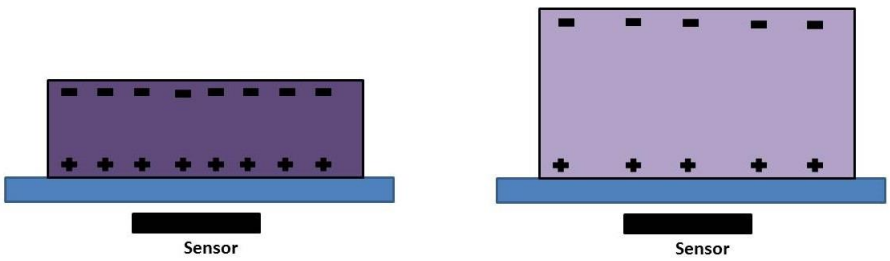


Figure 4: *Schematic representation of the strength of a magnet using magnetic 'charges'. The left figure is of a compact magnetic hydrogel with a relatively high concentration of magnetic particles. The figure on the right is of a less compact magnetic hydrogel, such as is the case for a swollen hydrogel.*

In figure 4 we can see that there are two effects on the magnetic field as a function of the hydrogel size. Firstly the negative charges are further away from the sensor reducing their effect on the overall magnetic field, secondly the total number of charges decreases due to dilution. The magnet will overall still have the same magnetic moment as that is defined in this model as the number of charge pairs multiplied by their separation. This means that the exact geometry is what effects the final observed magnetic field.

Now if we take a sensor that detects fields in the vertical direction in a gel that is also magnetized in this direction, we can see if we can make a mathematical description of the field strength at the sensor. For the sake of calculation it is assumed that the gels are cylindrical and are relatively thin in

the vertical direction. Making gels thin in at least one direction is needed to allow the gel to swell or shrink sufficiently fast, as this process occurs through diffusion which is very slow at longer distances. Making this direction the vertical direction allows us to use thin layers of a gel on a layer of glass. The situation would look like that in figure 5:

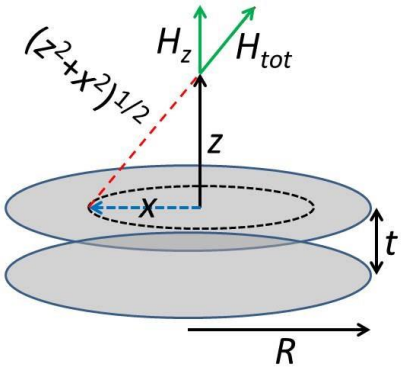


Figure 5: *The various parameters for calculating the horizontal magnetic field H_z. The magnetic gel itself is a cylinder with radius R and thickness t, the sensor is at a distance z directly over the center of the magnetic gel. The disks represent both of the poles of the magnetic gel.*

To calculate the field the magnetic field caused by a single magnetic charge at a given point we can simply use the magnetic analog of Coulomb's law:

$$H_{tot}{}^{*} = \frac{q_m}{4\pi r^2} \qquad 5.$$

Here q_m is the magnetic 'charge' of the point an r the distance between the charge and the point where the field is observed. Now instead of loose charges we have two planes with a certain charge density. To describe the charges on a circular disk as a function of the distance from the center of this disk it is possible to use the following equation:

$$q_m = H_{in} 2\pi x dx \qquad 6.$$

Here H_{in} is the charge density of the disk and x is the distance from the centre of the disk. Using equation 6 into equation 5 yields:

$$H_{tot}{}^{*} = \frac{H_{in} 2\pi x dx}{4\pi r^2} = \frac{H_{in} 2\pi x dx}{4\pi (x^2 + z^2)} \qquad 7.$$

Where the radius r can be calculated from the distance from the center of the disk x and the distance of the sensor from the disk z through Pythagoras' theorem. However as we are only measuring the vertical magnetic field and not the total magnetic field we need to know how much of this field is vertical. If the ratio of the field in the z direction with respect to the total field equals the ratio of the distance in the z direction with respect to the total distance from the magnetic charge, we can write:

$$H_z^* = H_{tot}\frac{z}{\sqrt{x^2+z^2}} = \frac{H_{in}2\pi x dx}{4\pi(x^2+z^2)}\frac{z}{\sqrt{x^2+z^2}} = \frac{H_{in}xz}{2(x^2+z^2)^{3/2}}dx \qquad 8.$$

To get the vertical magnetic field from an entire disk this equation must be integrated for the entire disk:

$$H_z = \int_0^R \frac{H_{in}xz}{2(x^2+z^2)^{3/2}}dx = -\frac{H_{in}z}{2}\left[\frac{1}{\sqrt{x^2+z^2}}\right]_{x=0}^{x=R} = \frac{H_{in}}{2}\left(1-\frac{1}{\sqrt{R^2+z^2}}\right) \qquad 9.$$

To get the magnetic field from the entire magnet another disk laying at z+t must be subtracted, this is the same equation as above but with z replaced by z+t. This yields the equation used to calculate the final magnetic field:

$$\begin{aligned} H_z &= \frac{H_{in}}{2}\left(1-\frac{1}{\sqrt{R^2+z^2}}\right) - \frac{H_{in}}{2}\left(1-\frac{1}{\sqrt{R^2+(z+t)^2}}\right) \\ &= \frac{H_{in}}{2}\left(\frac{1}{\sqrt{R^2+(z+t)^2}} - \frac{1}{\sqrt{R^2+z^2}}\right) \end{aligned} \qquad 10.$$

2.2.3 Effects of swelling on the magnetic field

It can be seen in equation 10 that the magnetic field is dependent on quite a few parameters which together determine the overall magnetic field. Two of these parameters change when a gel shrinks or swells, namely the charge density H_{in} and the separation of the magnetic poles t. The charge density scales with the volume of the gel as it depends on the concentration of magnetic particles inside the gel, the magnetic separation scales only with the thickness of the gel. Since the gels are made to be fixed on a substrate the gels can only swell in the vertical direction. This means that the separation of the magnetic poles and the charge density scale the same way. It is possible to define a swelling factor S that is definied as the thickness in the swollen state divided by the thickness in the shrunk state. Then the vertical magnetic field for the swollen state is given by:

$$H_z = \frac{H_{in}}{2S}\left(\frac{1}{\sqrt{R^2+(z+St)^2}} - \frac{1}{\sqrt{R^2+z^2}}\right) \qquad 11.$$

This equation combined with equation 10 helps to find the best parameters for the size of the gel and the distance of the sensor to the gel for a given gel thickness and swelling factor. If we take some typical values of gel parameters we can see what the ideal distance from the sensor should be:

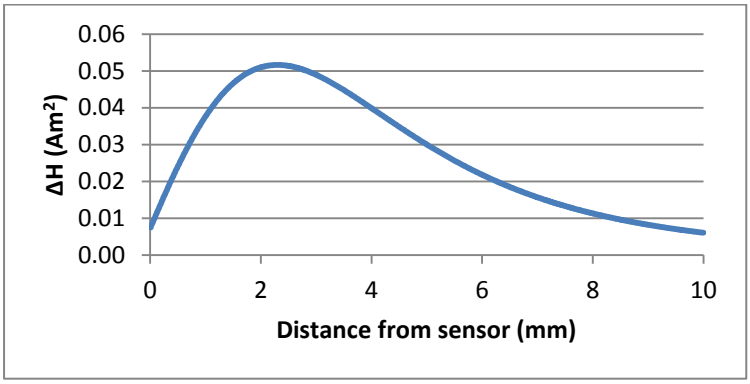


Figure 6: *Expected net signal as a function of the distance to the sensor. The net signal is the signal of the swollen gel minus the signal of the shrunk gel. Parameters used: R = 2mm, t = 0.2 mm, H_{in} = 10 Am^2.*

Overall the magnetic signal decreases as the gel swells, this means that the dilution of magnetic particles is a more important effect than the increase in distance between both magnetic poles.

3. Experimental: Methods

3.1 Materials

2,2'-azobis(2-methylpropionamidine)dihydrochloride (V-50, 98%, Acros Organics), acrylic acid (AA, 99% anhydrous, Aldrich), hydroxylethyl acrylate (HEA, 97%, Acros Organics), diethyleneglycol diacrylate (DEGDA, 75%, Aldrich), acetic acid (glacial 100%, Merck), sodium chloride (NaCl, p.a., Merck), sodium hydroxide (p.a. ≥99%, Merck), phosphoric acid (p.a. 85% in water, Acros Organics) and 3-(trimethoxysilyl) propyl methacrylate (TPM, 98%, Aldrich) were used as received. Deionized water from a Millipore purification system (MILLIPORE Synergy 185) was used for all experiments. A suspension of cobalt ferrite contained positively charged cobalt-ferrite particles and was obtained from Susanne van Berkum, prepared according to Tourinho et al. [12]

3.2 Bulk Swelling

To find out how the amount of swelling of different hydrogels depend on their composition gels of different compostition were made to swell up to their equilibrium size. These gels were made from varying precursor mixes that were made using varying molar ratios of hydroxyethyl acrylate (HEA) and acrylic acid. To all of the precursor mixes 1% by volume of diethylene glycol diacrylate (DEGDA) was added to function as a crosslinker. From these precursor mixes a gelmix was made by adding the precursor mix to milliQ water and adding a initiator mix. The final composition of these gel mixes were (by volume) 10% of precursor mix, 80% of water and 10% of initiator stock. The initiator stock was a mix of 3.019 g/mL solution of 2,2'-azobis(2-methylpropionamidine)dihydrochloride (V-50) in water.

Gels were made by taking 0.5 mL of gel mix and heating this for 30 minutes at 70°C in a closed Eppendorf cup. The gels were then weighed and transferred into about 45 mL of a pH 2 phosphate buffer and left there for 6 weeks before they were weighed again. Subsequently the gels spend 2 months in a pH 5 acetate buffer before being weighed again. Finally the gels were left to dry at 80°C overnight before being weighed the final time.

All buffers where made by titrating an acid (acetic acid in the case of the pH 5 buffer, phosphoric acid in the case of the pH 2 buffer) with a concentrated sodium hydroxide solution after adding an amount of sodium chloride sufficient to get the final ionic strength at 100mM. All buffers had a concentration of 50mM of the buffering species.

3.3 Synthesis of thin magnetic hydrogels

To synthesize a hydrogel containing magnetic particles the following procedure was followed: First a mixture of a given ratio of acrylic acid and Hydroxyethyl acrylate (HEA) was made, unless stated otherwise the molar ratio was 1 acrylic acid monomer for every 10 HEA monomers. To this mixture

1% (volume) of Diethylene glycol diacrylate (DEGDA) was added as a crosslinker. This mixture was then mixed with water, a suspension of cobalt ferrite and a initiator solution forming the gel mixture.

The initiator solution was a solution of 2,2'-azobis(2-methylpropionamidine)dihydrochloride (V-50) in water, the overall concentration of V-50 in the gel mixture was 1.25 mg/mL. The suspension of cobalt ferrite contained positively charged cobalt-ferrite particles and was obtained from Susanne van Berkum. The overall concentration of cobalt-ferrite in the gel mixture was 10 mg/mL. The amount of monomers in the gel mixture was 10% by volume.

The gel mixture was then placed in an Eppendorf cup in an oven at 70°C for 7 minutes to prevent bubble formation. Bubbles can form because nitrogen gas forms when V-50 is heated, heating (and partially polymerizing) the mixture before casting the final gel helps to reduce the number of bubbles in the final gel. The final gel was then cast the same day the gel mixture was made as it was found that gel mixtures left standing start to demix over time.

The gels were cast on top of a slide that was coated with TPM to make sure the gel remained attached to the glass. The gel's dimensions were controlled by placing a mask made of scotch tape on top of the TPM coated glass, the size of the hole in this mask controlled the lateral size of the gel and the number of scotch tape layers controlled the thickness of the gel. A regular glass-slide was placed on top of the mask to seal of the compartment in which the gel was cast. It was found that a mask of 4 layers of tape yields a gel of around 200 micrometers thickness. See figure 7 for a representation of how such a gel is cast.

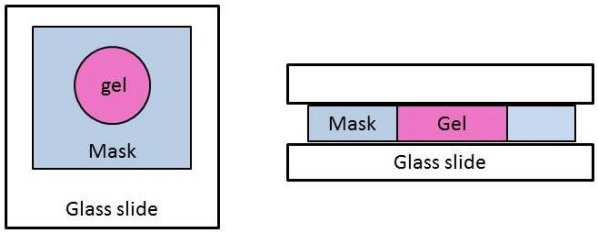


***Figure 7**: Schematic representation of how a gel is cast*

Glass was coated with TPM by first etching the glass in a solution of KOH for at least 30 minutes before rinsing the glass and putting it in a solution of 3-(Trimethoxysilyl)propyl methacrylate (TPM) for an hour and drying it overnight. The solution of TPM was made by adding acetic acid to water to obtain a pH between 3 and 4 and dissolving 0.4% by volume of TPM in this acidified water under stirring for at least two hours.

The gel was then finally made by casting the gel and placing it at 70°C for 30 minutes.

3.4 Gauss probe measurements

To measure the magnetic field from a magnetic gel a Gauss probe was used. This Gauss probe was equipped with a Lakeshore HMMT-6J Hall effect sensor. It was found however that the background signal of the probe was not constant over time, this was attributed to the temperature of the probe. It was tried to place the probe inside a temperature controlled environment, but it was found that the time needed for the magnetic signal to stabilize was too long to be useful when doing swelling experiments. Because of this all experiments with the Gauss probe where conducted at room temperature. To compensate for the shifting background signal over time each measurement was conducted by measuring the background signal first. Then, placing the sample near the probe,

measure the sample again before measuring the background signal again. This is repeated a number of times for each measurement to obtain the final data.

4. Experimental: Results

4.1 Bulk swelling

A number of gels were made to find out how the swelling of different hydrogels depend on their composition. To do this hydrogels of different ratios of acrylic acid and HEA were created and put into a pH 2 buffer and subsequently a pH 5 buffer so that the size of the gels could be determined. The molar ratios of acrylic acid and HEA were as follows:

Sample name:	Acrylic acid : HEA ratio
Z1	0:10
Z2	0.1:10
Z3	0.5:10
Z4	1:10
Z5	1.5:10
Z6	2:10
Z7	3:10

The amount of water the polymer network takes up from its surroundings as a function of pH can be calculated from the data (figure 8a). We are seeing an expected trend for the water uptake at pH 5, namely that the more acrylic acid a gel contains the more water is taken up by the gel. A rather unexpected outcome of this experiment is that the more acrylic acid is present in the gel the less water is taken up at pH 2. The effect is rather large, with a polymer consisting of only HEA taking up about 4 times as much water as a gel containing 3 parts acrylic acid per 10 parts HEA. This means that gels with high concentrations of acrylic acid have a larger difference in size between their size at low pH and at high pH. This is in part because of the gel being larger at pH 5 with a higher concentration of acrylic acid and partly because the gel is smaller at pH 2 with a higher concentration of acrylic acid. It is possible to express this difference between the state at pH 5 and the state at pH 2 as the swelling ratio, this is plotted in figure 8b.

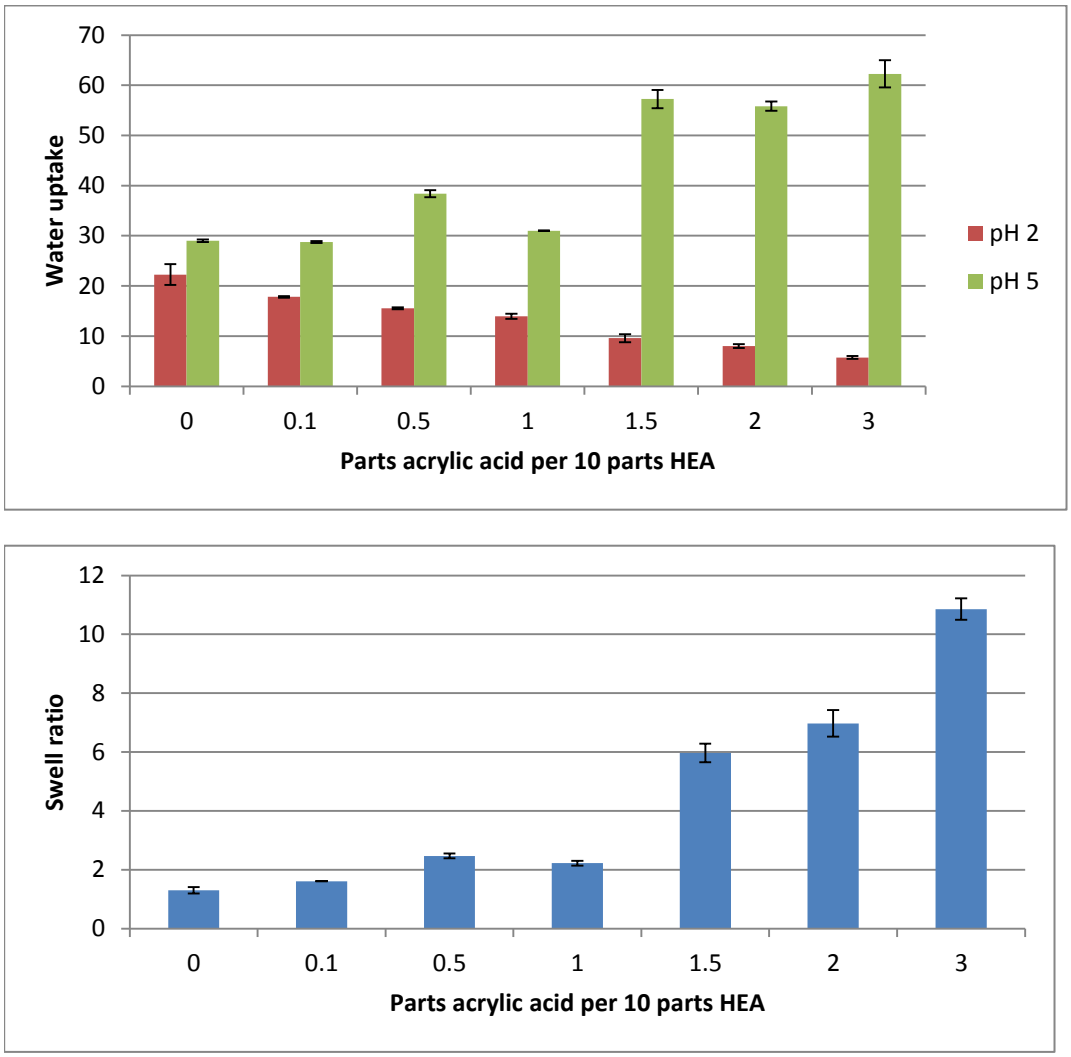


***Figure 8**: **(a)** Uptake of water at pH 2 and pH 5 at various concentrations of acrylic acid, average of three measurement. The water uptake is defined here as the weight of the gel at the given pH divided by the weight of the gel after drying. **(b)** The swell ratio of the same gels as in (a). The swell ratio is defined as the weight of the gel at pH 5 divided by the weight at pH 2.*

It is remarkable that a gel also swells between pH 2 and pH 5 when no acrylic acid is present. One possible explaination for this could be that the Young's modulus of the material changes with pH, for example S. K. De and coworkers found that hydrogels made from acrylic acid and hydroxyl-ethyl-methacrylate acid comonomers have a decreased Young's modulus at higher pH. [13] If we model the swelling of a gel by the balancing of some swelling force and the force from the stretching of the polymer network, then a change in the Young's modulus can cause swelling and deswelling even in the absence of acid groups.

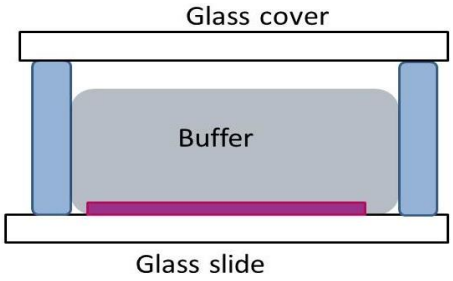


***Figure 9**: Schemetic representation of how magnetic gels are stored inside a buffer after they are made, the gel is in purple here.*

4.2 Thickness dependent measurements

Two magnetic hydrogels were synthesized to measure the size dependent magnetic signal of magnetic hydrogels. These were made using a mask that was approximately 0.5x0.5 cm and had 10 layers of scotch tape. The gels were immediately put in a pH 5.0 acetic acid buffer by putting a glass ring on top of the coated slide and filling it up with the buffer. (See figure 7) Whenever the gels needed to be stored they were stored in this pH 5 buffer in this manner. One of the gels was kept in pH 5 all the time to act as a control while the other one was alternately placed in a pH 2 buffer and a pH 5 buffer to see how the signal changes between the swollen en shrunk states. Both gels were magnetized by holding them next to a Neodymium magnet for 4 minutes.

The swelling and shrinking sample (gel 27) was placed inside a pH 2 buffer for between 4 and 7 hours to fully become shrunk, after this the gel was placed inside a pH 5 buffer again overnight to get back to its swollen state. The magnetic field was measured after each cycle, the results are plotted in figure 10:

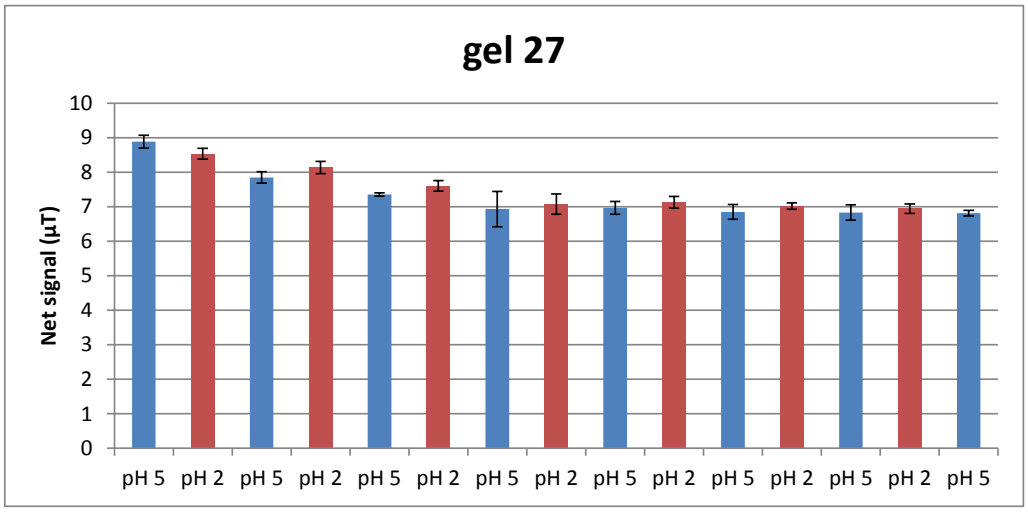


***Figure10**: Net magnetic signal (magnetic signal minus the background signal) at pH5 and pH2 after a number of cycles.*

From this data it becomes apparent that the red bars (denoting pH 2) are generally higher than the blue bars (denoting pH5), this means that the theory that the magnetic signal is higher when the gel is smaller seems to be holding up. However the size of the effect is very small. It is also apparent that the bars become smaller the further to the right these bars are. The further to the right the bars are the later in time they were measured showing that there is a non-negligible magnetic relaxation still present.

To see if the decreasing signal at pH 5 is due to Neél relaxation or also due to the effect of repeated swelling and shrinking it is possible to compare the gel that was swollen en shrunk multiple times (gel 27) with the gel that was kept at pH 5 all the time (gel 28). When we compare the signal of gel 27 and gel 28 over time it can be seen that the signal decreases more for gel 27 than for gel 28:

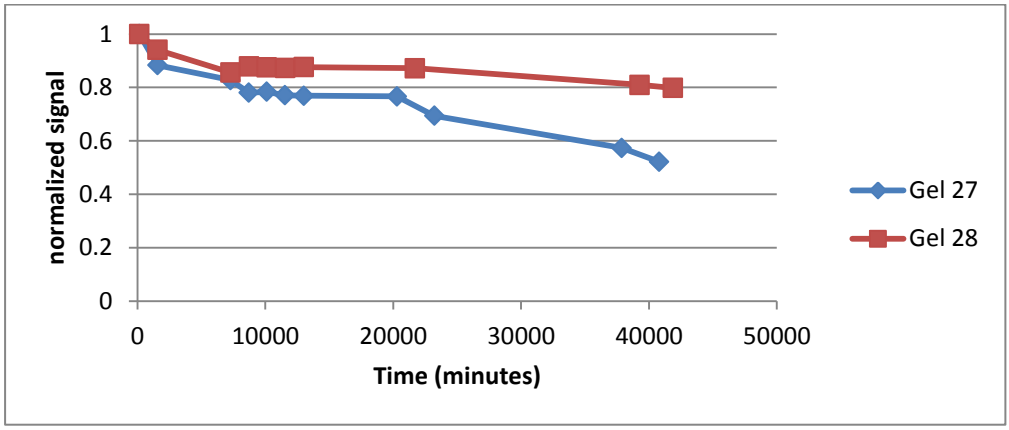


***Figure11**: Normalized signal at pH5 for gels 27 and 28. The signal of both is compared by normalizing both datasets by assuming that the first point equals 1*

One possible explanation that gel 27's signal decreases faster than the signal of gel 28 is that cobalt ferrite particles slowly dissolve at low pH, this was confirmed by leaving one gel at pH 2 for multiple weeks after which the gel became transparent indicating that all the cobalt ferrite had been dissolved. This result indicates that the system is not entirely reversible when the gel is shrunk with low pH buffer systems.

4.3 pH dependence of relaxation

It is interesting to look more closely into the statement that relaxation proceeds faster at pH 2 than at pH 5. To study this two gels were made using a mask that was approximately 0.5x0.5 cm and had 10 layers of scotch tape. Both gels were stored in a pH 5 buffer. One of the gels was then transferred to a pH 2 buffer and left there for 24 hours. Both gels were then magnetized by holding them next to a neodymium magnet for 5 minutes. Both gels had their signal measured multiple times to obtain a time dependent signal:

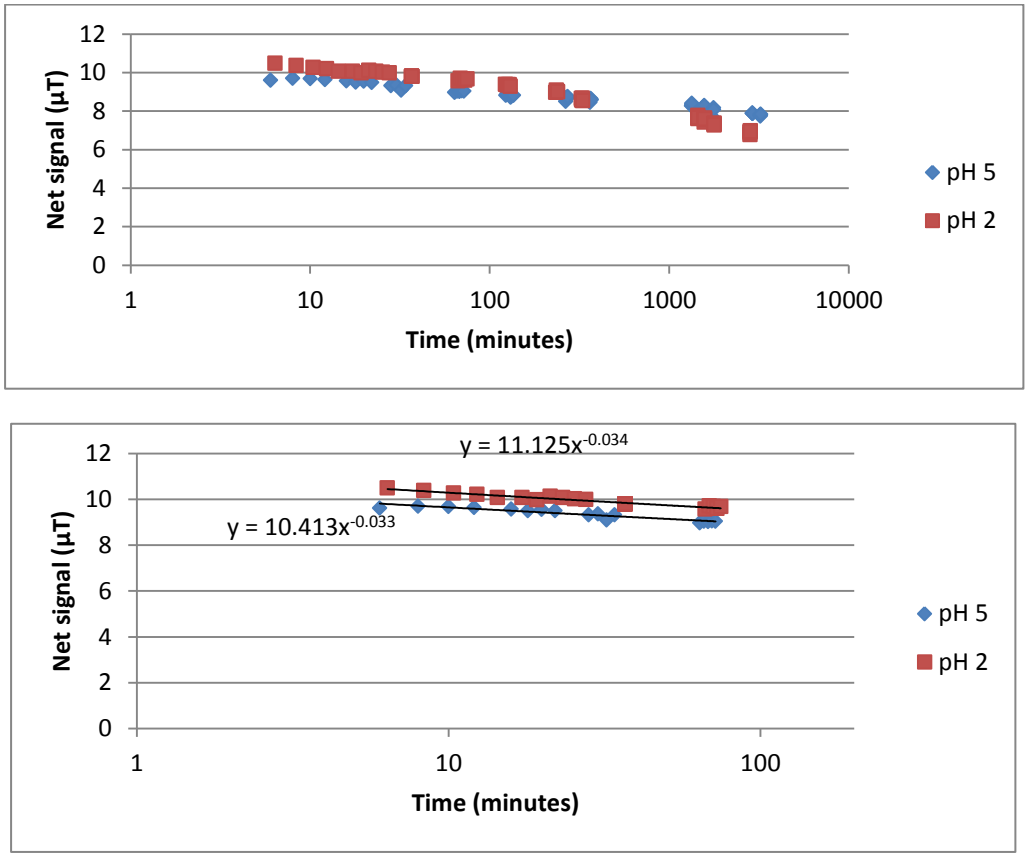


***Figure12**: Above: Magnetic signal at the sensor for two magnetic hydrogels stored at two different pH values Below: These same gels, but only the points from the first 100 minutes fitted with a powerlaw fit.*

From this data it is apparent that the signal decays approximately the same for both gels in the first approximately 100 minutes. Both gels show decay through a powerlaw. When the datapoints up to 100 minutes are fitted the exponent of the powerlaw is almost the same for both fits meaning that relaxation occurs at the same rate for gels kept under pH 5 conditions and pH 2 conditions. After 100 seconds the signal of the gel stored at pH 2 starts decreasing more rapidly yielding additional proof that the dissolving of cobalt ferrite particles lowers the magnetic signal over time.

5. Calculations: Methods

To understand more about the dynamics of the swelling of hydrogels, a number of calculations were done. Two similar, but different numerical models were implemented to try and understand the dynamics of swelling and shrinking. Both of these models will be described here. For both models, a finite element method was used to calculate differential equations. This method means essentially that something is divided into discrete steps small enough to make the following assumption:

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x} \qquad 12.$$

where y is the variable of interest which is a function of x.

5.1 Diffusion model

To measure the swelling and shrinking time of a hydrogel experimentally, a thin layer of hydrogel (around 200 µm thick) was covalently bound to a glass slide. It was found that the lateral swelling was significantly lower than the swelling in the direction perpendicular to the glass slide. This, along with the fact that the layer was quite thin, means that it was assumed that the problem could be treated as a one dimensional one if a point away from the edges is considered.

The diffusion model is based on calculating the diffusion of ions throughout the gel. At the same time it is assumed that the entire gel is at chemical equilibrium at all time. The osmotic pressure is calculated at various points in time as a measure for the size of the gel.

5.1.1 Calculation method

To calculate the diffusion of ions, the gel is divided into a number of small elements called bins. Each of these bins was typically about a micrometer in size. At a given point in time the values for the flow of ions between bins is calculated from the concentrations of ions inside these bins. This value for the flow is then used to get the concentrations one time-step later, essentially using the following approximation for ion k:

$$[k]_{new} = [k]_{old} + \frac{\partial [k]}{\partial t}\Delta t \qquad 13.$$

The newly found concentrations are then adjusted to the amounts at chemical equilibrium. The concentrations found in this way are then used to calculate the flow at this new point in time. This process is repeated many times to progress time forward from the starting value to see how various parameters change over time. The overall process looks like this:

a- Diffuse species and record new amounts in every bin due to diffusion
b- React [H^+] with gel acid groups and record [H^+] in every bin at chemical equilibrium

c- Progress time by Δt
d- Repeat

The gel environment (the buffer solution outside the gel) is modeled like an additional gel volume bin having a constant concentration of species that borders on the gel on one side. The actual bin size is assumed to be fixed; this allows working with amounts of atoms rather than concentrations which is easier to compute. This does obviously introduce an error due to the fact that in real life the gel changes size over time.

5.1.2 Starting configuration

At the start all gel bins are filled with H^+ ions using the relationship $nH^+ = 10^{-pH} \cdot N_A \cdot V$ where V is the volume of the finite size bin. The pH inside the gel itself and inside the buffer solution outside of the gel can be specified separately for this purpose. All bins have a set number of gel acid groups given by $ngel = [gel] * N_A * V$ where [gel] is estimated to be about 0.08 M. The number of charged acid groups in one bin is tracked and is initially given by: $ngel^- = ngel\left(\frac{K_a}{K_a+[H^+]}\right)$ Furthermore the concentrations of major counterions are also tracked. For the sake of simplicity, only two of these counterions are modeled: The positively charged Na^+ and negatively charged Cl^-. The initial amounts are determined using the ionic strength of the solution: $nCl^- = ionicstrength * N_A * V$ and $nNa^+ = nCl^- - nH^+$

5.1.3 Diffusion, theory

To calculate diffusion through the gel, it is assumed that diffusion works the same inside the gel as it would in water. The presence of a polymer matrix that occupies part of the available volume of the gel and the presence of charges on this matrix are neglected for calculating the diffusion. The gel is also assumed to be of constant size, so the distance between bins is kept constant. Diffusion can be described using Fick's first law:

$$J(x,t) = -\frac{Dc}{KT}\frac{\partial \mu}{\partial x} \qquad 14.$$

where J is the net flux, c is the concentration at the diffusing plane, D the diffusion coefficient and μ the chemical potential. For ideal mixtures the chemical potential is given by:

$$\mu = \mu_0 + KTln(c) \qquad 15.$$

Combining these equations means that $\frac{d\mu}{dx} = KT\frac{1}{c}\frac{dc}{dx}$, which means that the flux for ideal mixtures is given by:

$$J(x,t) = -D\frac{dc}{dx} \qquad 16.$$

However in practice electrical fields can be present due to varying diffusion rates (e.g. H^+ diffuses much faster than Cl^-) and fixed charge on the gel itself. These local electrical fields will influence the movement of ions as they are charged. As such we cannot simply assume that diffusion will work according to this equation. This can also be seen as in equilibrium. The net diffusion into the gel must be zero despite the fact that there is a concentration difference between the outside and inside of the gel as the gel has additional counter ions to make the gel charge-neutral. This non-ideality should not be ignored as it is the very thing that makes the gel swell.

For ions in an electric field the chemical potential can be described by the electrochemical potential which is:

$$\eta_k = \mu + z_k e\Phi \tag{17.}$$

where μ is the chemical potential of the ideal mixture, z_k is the valency of ion k, e is the elementary charge, and Φ is the electrostatic potential. If this electrochemical potential is used in equation 14 we get for ion k:

$$J_k(x,t) = -\frac{D_k c_k}{KT}\frac{\partial \eta_k}{\partial x} = -\frac{D_k c_k}{KT}\frac{\partial \mu_k}{\partial x} - D_k z_k \frac{e c_k}{KT}\frac{d\Phi}{dx} \tag{18.}$$

$$J_k(x,t) = -D_k \frac{\partial c_k}{\partial x} - D_k z_k \frac{e c_k}{KT}\frac{d\Phi}{dx} \tag{19.}$$

This equation is often written down in terms of the ionic mobility which is, very confusingly, also denoted by μ:

$$\mu_k = D_k \frac{e}{KT} = D_k \frac{F}{KT} \tag{20.}$$

The electrical current due to diffusion of a charged particle can simply be described by:

$$I_k = q_k J_k = z_k e\, J_k \tag{21.}$$

Now if we assume that there is no (or very little) electrical current due to diffusing ions we get for a system of N different ions j that:

$$\sum_{j=1}^{N} z_j e\, J_j = 0 \tag{22.}$$

This can be rewritten to yield:

$$\sum_{j=1}^{N} -z_j e\, D_j \left(\frac{\partial c_j}{\partial x} + z_j \frac{e c_j}{KT}\frac{d\Phi}{dx} \right) = 0 \tag{23.}$$

$$\sum_{j=1}^{N} -z_j e\, D_j \left(\frac{\partial c_j}{\partial x} + z_j \frac{ec_j}{KT} \frac{d\Phi}{dx} \right) = 0 \quad (24.)$$

$$\sum_{j=1}^{N} -z_j e\, D_j \frac{\partial c_j}{\partial x} = \frac{d\Phi}{dx} \sum_{j=1}^{N} e\, D_j {z_j}^2 \frac{ec_j}{KT} \quad (25.)$$

$$\frac{d\Phi}{dx} = -\frac{\sum_{j=1}^{N} z_j\, D_j \frac{\partial c_j}{\partial x}}{\sum_{j=1}^{N} D_j {z_j}^2 \frac{ec_j}{KT}} \quad (26.)$$

This last equation expresses the local electrical potential field as a function of concentrations and diffusion coefficients. This equation may then be inserted into the equation 19 to obtain the final diffusion equation assumed in this model:

$$J_k(x,t) = -D_k \frac{\partial c_k}{\partial x} + D_k c_k z_k \frac{\sum_{j=1}^{N} z_j\, D_j \frac{\partial c_j}{\partial x}}{\sum_{j=1}^{N} {z_j}^2 D_j c_j} \quad (27.)$$

5.1.4 Diffusion, calculation

To calculate the flow of ions, the method of using finite bins described broadly before is used. For this the gel is divided in finite volume elements of a fixed volume of 1 μm^3. The thickness of each element can be calculated by dividing the total gel thickness over the number of bins. From this the area through which all ions will diffuse can be easily be calculated, this is given by the constant 'sim_area'.

In a small time interval Δt the net number of ions diffusing through a plane of area 'sim_area' can be approximated by:

$$\Delta N_k \approx J_k(x,t) \cdot N_A \cdot simarea \cdot \Delta t \quad (28.)$$

We can also approximate differentials in space:

$$\frac{dc_k}{dx} \approx \frac{\Delta c_k}{\Delta x} \quad (29.)$$

Using this approximation it is possible to express the gradient between bin i and bin i+1 as:

$$\frac{\Delta c_k}{\Delta x} = \frac{c_k(x_i) - c_k(x_{i+1})}{x_i - x_{i+1}} \quad (30.)$$

In the program (appendix A) positions of the elements are not stored, only their thickness, this means that Δx must be given by $\frac{r_i+r_{i+1}}{2}$ where r_i is the thickness of element i. So we get:

$$\frac{\Delta c_k}{\Delta x} = 2\frac{c_k(i) - c_k(i+1)}{r_i + r_{i+1}} \quad 31.$$

Meanwhile it is assumed that the concentration at the plane between two bins equals the average concentrations of the two bins, so again for bins i and i+1:

$$c_k \approx \frac{c_k(x_i) + c_k(x_{i+1})}{2} \quad 32.$$

However concentrations are not recorded in the program, only amounts. But these can easily be converted into concentrations by:

$$c_k(i) = \frac{N_k(i)}{N_A \cdot V_i} = \frac{N_k(i)}{N_A \cdot simarea \cdot r_i} \quad 33.$$

Using equations 31, 32 and 33 to approximate equation 19 the following equation is obtained:

$$J_k(x,t) = \frac{-D_k\, 2}{N_A \cdot simarea}\frac{\frac{N_k(i)}{r_i} - \frac{N_k(i+1)}{r_{i+1}}}{r_i + r_{i+1}} + \frac{D_k z_k\left(\frac{N_k(i)}{r_i} + \frac{N_k(i+1)}{r_{i+1}}\right)}{2N_A \cdot simarea}\left(\frac{-e}{KT}\frac{d\Phi}{dx}\right) \quad 34.$$

This equation can be simplified by assuming that the volume is constant so that $r_i = r_{i+1} = r$. And, remembering our assumption $\Delta N_k \approx J_k(x,t) \cdot N_A \cdot simarea \cdot \Delta t$ (equation 28) we can get:

$$\frac{\Delta N_k}{\Delta t} = \frac{-D_k}{r^2}\big(N_k(i) - N_k(i+1)\big) + \frac{D_k z_k}{2r}\big(N_k(i) + N_k(i+1)\big)\left(\frac{-e}{KT}\frac{d\Phi}{dx}\right) \quad 35.$$

And, using equation 26 a value for $\frac{d\Phi}{dx}$ can also be approximated:

$$\frac{-e}{KT}\frac{d\Phi}{dx} \approx \frac{\sum_{j=1}^{N} z_j\, D_j \frac{\partial c_j}{\partial x}}{\sum_{j=1}^{N} {z_j}^2 D_j c_j} = \frac{2}{r}\frac{\sum_{j=1}^{N} z_j\, D_j\big[N_j(i) - N_j(i+1)\big]}{\sum_{j=1}^{N} {z_j}^2 D_j\big[N_j(i) + N_j(i+1)\big]} \quad 36.$$

Combining these last two equations allows to find a value for ΔN_k. Which is the net flow of ions of species k through the plane between bins i and i+1 in the time interval Δt:

$$\Delta N_k(i+1 \to i) = Dif_k\big(N_k(i+1) - N_k(i)\big) + Dif_k z_k\big(N_k(i) + N_k(i+1)\big)\frac{\sum_{j=1}^{N} z_j\, Dif_j\big[N_j(i) - N_j(i+1)\big]}{\sum_{j=1}^{N} {z_j}^2 Dif_j\big[N_j(i) + N_j(i+1)\big]} \quad 37.$$

Here a new effective diffusion coefficient call Dif_k is used which combines a number of constants into a single constants, it is defined as:

$$Dif_k = \frac{D_k}{r^2}\Delta t \quad 38.$$

Equation 37 is the main equation used to calculate diffusion between bins. For each bin the amount of ions diffusing into or out of this bin is calculated by looking both at the flow of ions from bin i+1 to bin i and the flow of ions from bin i to bin i-1. The net change in concentration is then given by the flow from bin i+1 to bin i minus the flow from bin i to bin i-1:

$$\Delta N_k = \Delta N_k(i+1 \to i) - \Delta N_k(i \to i-1) \quad 39.$$

5.1.5 Acid-base equilibrium

The hydrogel contains acid groups bound to the polymer matrix. These acid groups do not diffuse through the gel but do affect diffusion of H^+ ions through acid-base reactions. So it is important to also calculate the acid-base equilibrium to understand the effect of the presence of these groups to the diffusion behavior of H^+ through their buffering effect. The model does not take into account the buffering effect due to the self-ionisation of water, nor does it consider the diffusion of OH^-. Both of these make that the model is only valid when $[H^+] >> [OH^-]$. So only the reaction of H^+ with the acrylic acid subunits of the polymer chain is taken into account. Acrylic acid makes up about 0.557 vol% of the reaction mixture, which with a density close to that of water and a molecular weight of 72 g/mol the estimated concentration of acrylic acid subunits is about 0.08 mol/L. The reaction of H^+ to acrylic acid is a simple acid-base reaction:

$$AA^- + H^+ \to HAA \quad 40.$$

Here AA^- denotes acrylic acid in its deprotonated form and HAA denotes acrylic acid in its protonated form. The reaction equilibrium between these forms can be described by:

$$K_a = [H^+]\frac{[AA^-]}{[HAA]} \quad 41.$$

For the model it is assumed that reaching this equilibrium happens nearly instantaneously. So the rate of reaction is assumed to be much faster than the rate of diffusion. To use the above equation, an estimate of the value of the equilibrium constant K_a is needed. Acrylic acid has a pKa of 4.35, and

this is expected to be higher when it is inside a gel network due to loss of pi-conjugation stabilization of the conjugated base upon polymerization. I estimate the pKa of polymerized acrylic acid to be around 5.0 based on a predicted increase in pKa upon oligomerization ofmethacrylic acid. [14] Using these assumptions an equation for finding the number of protonated and deprotonated acrylic acid subunits can be found. In the following the acrylic acid subunits will simply be called gel units; all other components of the gel are neglected.

Since the total number of gel units won't change as long as the volume of the simulation element is unchanged the number of charged gel units can be expressed as:

$$[gel^-] + [Hgel] = [gel] \quad 42.$$

$$[Hgel] = [gel] - [gel^-] \quad 43.$$

Combing equations 41 and 43 yields:

$$[gel] - [gel^-] = \frac{[H^+][gel^-]}{K_a} \quad 44.$$

$$[gel] = [gel^-]\left(\frac{[H^+]}{K_a} + 1\right) \quad 45.$$

$$[gel^-] = [gel]\left(\frac{K_a}{K_a + [H^+]}\right) \quad 46.$$

Equation 46 is the basic equation used to calculate the degree of ionization of the gel at the initial pH.

However this equation cannot be used to calculate the change in $[gel^-]$ directly, as this equation is only valid at constant $[H^+]$. When the concentration of H^+ ions changes, due to diffusion the gel^- concentration will change in response, which in turn will change the H^+ concentration due to an acid-base reaction:

$$Hgel \rightarrow gel^- + H^+ \quad 47.$$

So in effect equation 46 becomes an equation with two unknowns. To solve it an additional equation is required. To solve it anyway it is possible to look at the change in concentration in more detail. The change in concentration between two points in time is given by:

$$[gel^-]_{new} = [gel^-]_{old} + \Delta[gel^-] \quad 48.$$

$$[H^+]_{new} = [H^+]_{old} + \Delta[H^+] = [H^+]_{old} + \Delta[H^+]_{diffusion} + \Delta[H^+]_{reaction} \quad 49.$$

This holds for the gel units and the H^+ ions respectively. In equation 49 the change in H^+ ions is further defined to consist of both a change due to diffusion and a change due to chemical interactions with the gel units. Now from simple reaction stoichiometry it is known that:

$$\Delta[H^+]_{reaction} = \Delta[gel^-] \tag{50.}$$

These relationships can be used in equation 46 to obtain:

$$[gel^-]_{new} = \frac{[gel]K_a}{K_a + [H^+]_{new}} = \frac{[gel]K_a}{K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} + \Delta[gel^-]} \tag{51.}$$

$$\begin{aligned}\frac{[gel]K_a}{[gel^-]_{new}} &= K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} + \Delta[gel^-] \\ &= K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} + [gel^-]_{new} - [gel^-]_{old}\end{aligned} \tag{52.}$$

$$\frac{[gel]K_a}{[gel^-]_{new}} - [gel^-]_{new} = K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} - [gel^-]_{old} \tag{53.}$$

$$\begin{aligned}[gel^-]_{new}^2 + \left(K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} - [gel^-]_{old}\right)[gel^-]_{new} \\ - [gel]K_a = 0\end{aligned} \tag{54.}$$

This last equation is a quadratic equation that can be solved using the quadratic formula to yield:

$$\begin{aligned}&[gel^-]_{new} \\ &= \frac{-\left(K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} - [gel^-]_{old}\right)}{2} \\ &\pm \frac{\sqrt{\left(K_a + [H^+]_{old} + \Delta[H^+]_{diffusion} - [gel^-]_{old}\right)^2 + 4[gel]K_a}}{2}\end{aligned} \tag{55.}$$

This equation has two solutions, but since negative concentrations are unphysical only the + variety is a correct solution. Furthermore we can rewrite this equation knowing that $[H^+]_{old} + \Delta[H^+]_{diffusion}$ is simply the H^+ concentration before the reaction step is applied:

$$\begin{aligned}[gel^-]_{new} = &-\left(\frac{K_a + [H^+] - [gel^-]_{old}}{2}\right) \\ &+ \sqrt{\left(\frac{K_a + [H^+] - [gel^-]_{old}}{2}\right)^2 + [gel]K_a}\end{aligned} \tag{56.}$$

Since amounts are stored rather than concentrations it is preferable if this equation can be written in terms of amounts rather than concentrations. To do this, the value of K_a must be changed using:

$$NK_a = K_a * \frac{N_A}{V} \quad (57.)$$

This leads to the final equation to calculate amounts of negatively charged gel in the software program:

$$ngel^-_{new} = \sqrt{\left(\frac{NK_a + nH^+ - ngel^-_{old}}{2}\right)^2 + ngel * NK_a} - \left(\frac{NK_a + nH^+ - ngel^-_{old}}{2}\right) \quad (58.)$$

5.1.6 Measuring gel size

One of the most important output parameters to be obtained from these calculations is the osmotic pressure. Since theory states that the swelling of hydrogels with acidic groups is primarily due to an increase in osmotic pressure inside the gel due to an excess of ions present to compensate for the negative polymer charges, it is important to get a measure for osmotic pressure from these calculations. To calculate the osmotic pressure the following equation can be used:

$$P_{osmotic} = RT \sum c_k \quad (59.)$$

Here c_k denotes the concentration of species k. The summation is performed over all concentrations assumed to play a role in the osmotic pressure; the gel units are not included in this calculation as they are assumed immobile. For practical purposes only H^+, Na^+ and Cl^- are considered here. The sodium and chloride ions are included as the main salt ions present in the solution. As the salt concentration in the gel is assumed to have a large impact on the amount of swelling [5] it is very important to take salt concentration into consideration in this model. It also affects the diffusion rate for H^+ through equation 26. To get the overall osmotic pressure difference between outside and inside the gel the osmotic pressure outside the gel must be subtracted from the osmotic pressure inside the gel. This quantity will simply be called the osmotic pressure here:

$$P_{osmotic} = RT([H^+]_{in} + [Na^+]_{in} + [Cl^-]_{in} - [H^+]_{out} - [Na^+]_{out} - [Cl^-]_{out}) \quad (60.)$$

$$P_{osmotic} = \frac{V}{N_A} RT(nH^+_{in} + nNa^+_{in} + nCl^-_{in} - nH^+_{out} - nNa^+_{out} - nCl^-_{out}) \quad (61.)$$

5.2 Buffered model

An alternative method for doing calculations on gel size was also used. This method is based on a paper by S.K. De and N.R. Aluru. [15] This model also includes buffering species besides the acrylic acid monomer groups, something that was noticeably missing from my earlier calculations. This model also allows the gel to swell and shrink, thereby changing the overall distance between separate bins. As with the diffusion model the gel is assumed to be at acid-base equilibrium all the time, it also assumes that the entire process can be described as a one-dimensional process with the same rationale as the earlier model.

5.2.1 Calculation method

As with the diffusion model, the gel is divided into a number of small elements called bins. However, unlike in the diffusion model, the bins do not have a fixed size. It is assumed that the gel size is caused exclusively by the osmotic pressure difference between the outside and the inside of the gel. So the gel would collapse and fully phase-separate into a polymer and water if no charges were present on the gel's polymer matrix.

In every bin the concentration of H^+ ions is tracked over time. These amounts of ions change over time as a consequence of the diffusion of H^+ through the gel. At a given point in time the value for the change in H^+ concentration with respect to time is calculated. This value for the rate of change in concentration is then used to get the concentration of H^+ one time-step later. The concentration at this new point in time is then used to find the equilibrium size of each bin at that point in time. From this size and the new H^+ concentration the rate of change in H^+ concentration at this new point in time is calculated. This process is repeated many times to track the development of gel size over time. This entire algorithm assumes that the change in volume of the gel does not cause any significant flow causing changes in concentrations. It also assumes that the gel size instantaneously adapts to the new H^+ concentration profile.

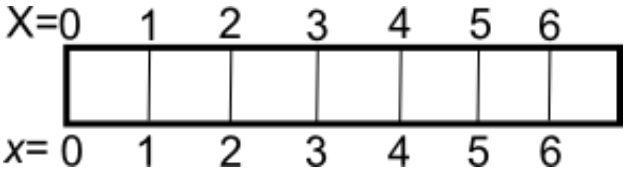
To track the gel size as a function of time, a parameter called the hydration value is calculated. The hydration is defined to be the volume of liquid V_l divided by the volume of polymer V_s in the gel; in this way it provides a useful swelling factor where at a hydration of zero the polymer and liquid are fully separated. The larger the value for the hydration, the larger the gel is. The hydration will be denoted by a capital H, defined as:

$$H = \frac{V_l}{V_s} \qquad 62.$$

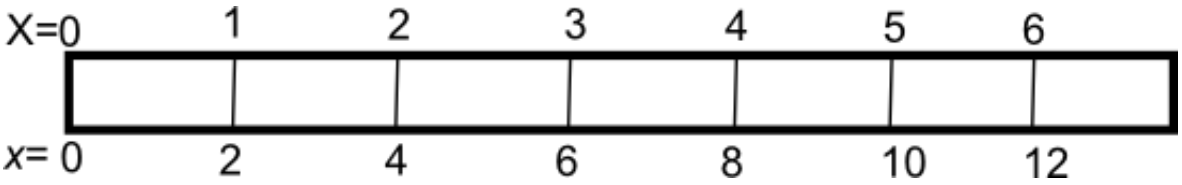
The actual bins used in the calculation have a variable size to take into account the fact that the gel is changing in size. This requires some thinking about the position of a given bin in space. If the hydration would be the same for all bins, the position of a bin can be given by:

$$x = (1 + H)X \qquad 63.$$

Here x is the Cartesian position on the x-axis for that specific bin and X is the relative position of the bin. This value X has the same units as the position x and equals the value of x in the case of a fully dry gel consisting of only polymer. So in case of a gel that consists of only the polymer, that is V_l is 0, the positions would be:



When the gel consists of an equal amount of polymer and solvent and the hydration level is still the same everywhere the situation would be:



In practice the hydration values do differ between bins, however this does not immediately cause problems as long as only differences in position are considered within a region where the hydration can be considered constant.

5.2.2 Hydrogen ion diffusion: theory

The principal time-dependence for the change in size of a hydrogel is the diffusion of H^+ ions. H^+ ions can diffuse in two different ways, that is diffusion of plain H^+ ions and diffusion of buffer that carries H^+. In this model the overall concentration of buffer molecules is considered to be constant; however the amount of buffer molecules that have H^+ bound to them is not constant. This approximation allows calculating the diffusion of H^+ through the buffer route without calculating the diffusion of the buffering molecules themselves. For simple diffusion of a substance k we can use Fick's first law of diffusion to yield the flux of a species k:

$$J_k = \phi\left[-\overline{D_k}\frac{\partial c_k}{\partial x}\right] \qquad 64.$$

This is essentially the same equation as was used for the diffusion model described earlier (equation 16) but with a few differences. Instead of a simple diffusion coefficient D_k an effective diffusion coefficient $\overline{D_k}$ is used. This effective diffusion coefficient is related to the regular diffusion coefficient for species k through:

$$\overline{D_k} = D_k\left(\frac{H}{2+H}\right)^2 \qquad 65.$$

This takes into account the fact that the polymer network acts as a sort of membrane causing the rate of diffusion to slow down. The above equation introduces a sort of tortuosity factor to the diffusion coefficient to account for the more complicated path taken by diffusing molecules inside a porous network. Additionally this effective diffusion coefficient is multiplied by a so called 'delay factor'. This is an arbitrary fitting parameter to make the swelling and shrinking rates found from the numerical calculations more in line with those found experimentally. This additional delay factor would include effects that make the effective diffusion through a gel slower than just the tortuosity accounts for.

Equation 64 also contains an additional factor ϕ which is the gel porosity. This factor accounts for the fact that diffusion only occurs through the solvent, not through the polymer matrix. This gel porosity simply equals the volume of liquid over the volume of solid:

$$\phi = \frac{V_l}{V_l + V_s} = \frac{H}{1 + H} \tag{66.}$$

The total amount of hydrogen must be preserved; this includes loose H^+ ions and H^+ ions bound to either the polymer acid groups or the buffer molecules. This means that if in a certain bin the number of hydrogen ions (bound plus unbound) changes over time this must be because hydrogen ions flow into or out of this bin. This simply means that the change of hydrogen over time in a certain bin must equal the net flow of hydrogen into the gel:

$$\frac{\partial}{\partial t}\left(Hc_H + Hc_{Hgel} + Hc_{HB}\right) = -\frac{\partial(J_H + J_{HB})}{\partial X} \tag{67.}$$

Here c_H is the concentration of free H^+, c_{Hgel} the concentration of H^+ bound to the gel acid groups and c_{HB} the concentration H^+ bound to buffer molecules. The above equation has a derivative of the flux of H^+ with respect to position as the net flow into and out of the bin is actually given by the difference of what flows in and what flows out. At a set point in space this must equal the space-derivative of the flow. As the flow between bins, not between set points in space, is what is considered, the relative position X is used rather than the real position x.

We can express the concentrations of H^+ bound to the gel polymer and to the buffer in terms of the concentration of loose H^+ by considering the equilibrium constant, giving these two equations:

$$c_{HB} = c_H \frac{c_T}{K_B + c_H} \tag{68.}$$

$$c_{Hgel} = \frac{c_{gel}^s}{H} \frac{c_H}{K + c_H} \tag{69.}$$

Here K_B denotes the dissociation constant of the buffer and K the dissociation constant of the gel acid groups. c_{gel}^s is the concentration of acid groups in dry polymer, so the concentration of acid groups is given by $\frac{c_{gel}^s}{H}$. Similarly c_T is the total concentration of buffer, both with and without H^+ bound to it, which is assumed to be constant throughout space. It is possible to use equation 57 to relate the flux of hydrogen ions carried by the buffer to the flux of free hydrogen ions. This can be used to find an expression for the flow of hydrogen ions carried by the buffer:

$$J_{HB} = \frac{\overline{\overline{D_{HB}}}}{\overline{D_H}} \frac{c_T}{K_B + c_H} J_H \qquad 70.$$

Using equations 68, 69, and 70 in equation 67 yields:

$$\frac{\partial}{\partial t}\left[Hc_H + \frac{c_{gel}^s c_H}{K + c_H} + \frac{Hc_T c_H}{K_B + c_H}\right] = \frac{\partial}{\partial X}\left[\phi\left(1 + \frac{\overline{\overline{D_{HB}}}}{\overline{D_H}} \frac{c_T}{K_B + c_H}\right)\left(\overline{D_H}\frac{\partial c_H}{\partial x}\right)\right] \qquad 71.$$

This is the hydrogen continuity equation, which is to be solved for $\frac{\partial c_H}{\partial t}$ to yield a profile of the concentration of hydrogen ions in time and space which can be used to model the time-dependence of swelling.

5.2.3 Hydrogen ion diffusion: calculation

To solve equation 71 for $\frac{\partial c_H}{\partial t}$ it is first necessary to rewrite the left hand side of that equation. If it is assumed that during a diffusion time-interval the hydration remains constant it is possible to use the product rule to obtain the following equation:

$$\begin{aligned}\frac{\partial}{\partial t}\left[Hc_H + \frac{c_{gel}^s c_H}{K + c_H} + \frac{Hc_T c_H}{K_B + c_H}\right] &= H\frac{\partial c_H}{\partial t} + \frac{c_{gel}^s}{K + c_H}\frac{\partial c_H}{\partial t} + c_{gel}^s c_H \frac{\partial}{\partial t}\frac{1}{K + c_H} + \frac{Hc_T}{K_B + c_H}\frac{\partial c_H}{\partial t} \\ &+ Hc_T c_H \frac{\partial}{\partial t}\frac{1}{K + c_H}\end{aligned} \qquad 72.$$

Using the quotient rule we know that $\frac{\partial}{\partial t}\frac{1}{K+c_H} = -\frac{1}{(K+c_H)^2}\frac{\partial c_H}{\partial t}$ so that the above equation becomes:

$$\left[H + \frac{c_{gel}^s}{K + c_H} - \frac{c_{gel}^s c_H}{(K + c_H)^2} + \frac{Hc_T}{K_B + c_H} - \frac{Hc_T c_H}{(K_B + c_H)^2}\right]\frac{\partial c_H}{\partial t} \qquad 73.$$

The entire part between the square brackets is calculated for each bin using the known concentrations and the hydration value inside each bin. This value is known in the software as 'Tcon'

$$Tcon = H + \frac{c_{gel}^{s}}{K + c_H} - \frac{c_{gel}^{s} c_H}{(K + c_H)^2} + \frac{H c_T}{K_B + c_H} - \frac{H c_T c_H}{(K_B + c_H)^2} \quad 74.$$

In the diffusion model described earlier diffusion of ions was calculated using the flow of ions through a plane between two bins. Here the net flow of ions into bin i was the flow of ions from bin i+1 to bin i minus the flow from bin i to bin i-1. For calculating the right hand side of equation 60 a similar method is used. Because the right hand side of equation 71 essentially contains a second derivative with respect to place at least three points are needed to get a good value for the second derivative. To get a value for the second derivative that includes as little as possible information from outside the bin itself the two points for finding the value for $\frac{\partial c_H}{\partial x}$ in bin i are points i+½ and i-½. The values for the concentrations at these points are assumed to equal the average value of two bins. So the value for the concentrations and hydration at point i+½ are the average of those at point i and point i+1. This allows approximating $\frac{\partial c_H}{\partial x}$ as:

$$\frac{\partial c_H}{\partial x} \approx \frac{\Delta c_H}{\Delta x} \quad 75.$$

$$\frac{\Delta c_H}{\Delta x}\left[i + \frac{1}{2}\right] = \frac{c_H[i+1] - c_H[i]}{2 + H[i] + H[i+1]} \frac{2}{\Delta X} \quad 76.$$

$$\frac{\Delta c_H}{\Delta x}\left[i - \frac{1}{2}\right] = \frac{c_H[i] - c_H[i-1]}{2 + H[i] + H[i-1]} \frac{2}{\Delta X} \quad 77.$$

Here the following relation is used (derived directly from equation 63):

$$\Delta x = (1 + H)\Delta X \quad 78.$$

This allows us to approximate the right hand side of equation 71 using the finite bins as:

$$\frac{\Delta}{\Delta X}\left[\phi\left(1+\frac{\overline{\overline{D_{HB}}}}{\overline{D_H}}\frac{c_T}{K_B+c_H}\right)\left(\overline{D_H}\frac{\Delta c_H}{\Delta x}\right)\right](i) \qquad 79.$$

$$=\left(\frac{\phi[i]+\phi[i+1]}{2}\right)\frac{\left(1+\frac{\overline{D_{HB}}}{\overline{D_H}}\frac{c_T}{K_B+\left(\frac{c_H[i]+c_H[i+1]}{2}\right)}\right)}{\Delta X}\left(\overline{D_H}\frac{c_H[i+1]-c_H[i]}{2+H[i]+H[i+1]}\frac{2}{\Delta X}\right)$$

$$-\left(\frac{\phi[i]+\phi[i-1]}{2}\right)\frac{\left(1+\frac{\overline{D_{HB}}}{\overline{D_H}}\frac{c_T}{K_B+\left(\frac{c_H[i]+c_H[i-1]}{2}\right)}\right)}{\Delta X}\left(\overline{D_H}\frac{c_H[i]-c_H[i-1]}{2+H[i]+H[i-1]}\frac{2}{\Delta X}\right)$$

Calculating this value for each bin and dividing it by Tcon (equation 74) allows us to calculate a value for $\frac{\Delta c_H}{\Delta t}$. This value for the flow can then be used to calculate the concentration of H^+ at a time Δt later. At this later time the new values for the hydration are then calculated.

5.2.4 Donnan theory

As has been stated before, it is assumed that the size of the gel is dependent only on the osmotic pressure difference between the outside of the gel and the gel's interior. This difference in osmotic pressure is caused by the gel's acid groups being partially deprotonated causing a net charge on the polymer matrix of the gel. This charge must be compensated; this can happen by having a higher concentration of positive counterions or by a deficit of negatively charged co-ions present in the gel. Given a limited ionic strength, the effect of excess counterions will be stronger. To calculate the exact effect, Donnan theory can be used. This relates the concentration of positive and negative ions inside the gel to the concentrations outside of the gel by:

$$\left(\frac{c_{k+}}{c_{k+}^0}\right)^{\frac{1}{|z_{k+}|}}=\left(\frac{c_{k-}^0}{c_{k-}}\right)^{\frac{1}{|z_{k-}|}} \qquad 80.$$

The concentrations c_{k+}^0 and c_{k-}^0 are the concentrations of positive and negative ions outside of the gel; the concentrations of ions inside the gel are given by c_{k+} and c_{k-}. For easy calculation it is assumed that all positive ions are Na^+ and all negative charges come from Cl^-. With only monovalent ions being included both absolute z-values are simply 1. This simplifies the above equation to:

$$\frac{c_{k+}}{c_{k+}^0}=\frac{c_{k-}^0}{c_{k-}} \qquad 81.$$

Knowing that the total charge should be zero it is possible to write for the concentration inside the gel $c_{k+}=c_{k-}+c_f$. Here c_f is the concentration of charges from the polymer backbone which can be calculated by:

$$c_f = \frac{K c_{gel}^s}{H(K + c_H)} \quad 82.$$

Combining the relation $c_{k+} = c_{k-} + c_f$ with the Donnan equilibrium yields:

$$c_{k+} = \frac{c_{k+}^0 c_{k-}^0}{c_{k+} - c_f} \quad 83.$$

$$c_{k+}{}^2 - c_{k+} c_f - c_{k+}^0 c_{k-}^0 = 0 \quad 84.$$

This can be solved for the positive ion concentration to yield:

$$c_{k+} = \frac{\sqrt{c_f{}^2 + 4 c_{k+}^0 c_{k-}^0} + c_f}{2} \quad 85.$$

From this the negative ion concentration can be calculated as:

$$c_{k-} = \frac{c_{k+}^0 c_{k-}^0}{c_{k+}} \quad 86.$$

5.2.e Gel size

If the concentrations of ions inside and outside of the gel are calculated from equations 85 and 86 the osmotic pressure can simply be calculated as:

$$P_{osmotic} = RT\left(c_{k+} - c_{k+}^0 + c_{k-} - c_{k-}^0\right) \quad 87.$$

To calculate how fast the gel subsequently swells, the displacement as a function of stress can be described as:

$$\rho \frac{\partial^2 u}{\partial t^2} = \nabla \cdot \sigma \quad 88.$$

Here ρ is the effective density of the gel, u is the vector of the displacements, and σ is the stress tensor. In the simple one-dimensional system used for these calculations, the stress-tensor does not have shear stresses and only a single component for the normal stress.

$$\sigma = B\frac{du_x}{dx} - P_{osmotic} \quad (89.)$$

Here B is an effective spring constant that relates displacement to stress. When the Poisson's ratio of the hydrogel is assumed to be 0 this effective spring constant equals the Young's modulus. This can be placed inside equation 88 to yield:

$$\rho\frac{\partial^2 u}{\partial t^2} = B\frac{\partial^2 u_x}{\partial x^2} - \frac{\partial P_{osmotic}}{\partial x} \quad (90.)$$

If the inertial terms are not dominant then $\rho\frac{\partial^2 u}{\partial t^2} = 0$ so that

$$B\frac{\partial^2 u_x}{\partial x^2} = \frac{\partial P_{osmotic}}{\partial x} \quad (91.)$$

The strain is assumed to be directly linked to the first derivative of the displacement vector. If the displacement that the gel experiences would be constant throughout the gel, the gel would simply move. If there is a difference in displacements throughout the gel this indicates that the gel is changing size. So the strain is assumed to be:

$$\epsilon_x = \frac{du_x}{dx} = H \quad (92.)$$

This description of the strain essentially requires that the displacement vector is measured from the dry gel state, so the displacement that a point experiences is the displacement from position x = X. To calculate the hydration H for the entire gel, we must integrate over this gel to get:

$$\int_{x=0}^{x=L} B\frac{\partial^2 u_x}{\partial x^2}dx = B\int_{x=0}^{x=L}\frac{\partial \epsilon_x}{\partial x}dx = \int_{x=0}^{x=L}\frac{\partial P_{osmotic}}{\partial x}dx \quad (93.)$$

$$P_{osmotic} = B\epsilon_x = B \cdot H \quad (94.)$$

Where L is the total thickness of the gel and the osmotic pressure is defined to be the difference between the osmotic pressure at point x=L and the pressure at point x=0. Equation 94 is used to calculate the size of each bin element.

However it is possible that once a new size is calculated, the outcome for the Donnan potential changes. This is because the concentration of polymer acid groups is dependent on the overall size of the gel. To remedy this situation, equations 85 and 86 are recalculated using a hydration value that is the average of the newly calculated hydration value and the previously calculated hydration value. Subsequently a new hydration value is calculated using equation 94. If this hydration value does not

differ significantly from the previously calculated value, the value is kept; otherwise, the process is repeated as long as it takes to find the proper value for the hydration.

6. Calculations: Results

6.1 Diffusion model

6.1.1 Initial configuration

Based on the theoretical model given before a number of calculations where done. The first step is to determine the starting configurations. Initially the starting configurations are prepared as described in the section **Starting configuration** in the calculations method. However these starting configurations are only approximate. To get a good configuration at a given pH it is possible to apply the diffusion model to a gel at the pH of interest. This allows for the calculation of all the various concentrations that the model will converge to at a given pH.

Calculating the initial configuration can be done using very few bins, strongly reducing calculations times. This is because the diffusion model will, at equilibrium, have the same concentrations everywhere inside the gel. Initial configurations where created at pH 2 and pH 5, then the outside pH was changed leading to swelling or shrinking calculations.

	Swelling	Shrinking
Gel pKa	5	5
pH outside gel	5	2
pH inside gel	1.99983	4.92141
Ionic strength	0.1 M	0.1 M
Gel size	200 µm	200 µm
Timestep	0.00001 s	0.00001 s
Total calculation time	9600 s	1000 s

Table 1: *Constants used in the diffusion model calculations*

Using a timestep of only 0.00001 s means that a maximum of 2.3% of hydrogen ions will move into or out of a bin per time step, other ions will diffuse even slower. Such a low percentage was chosen to make sure that the assumption that $\frac{d[H^+]}{dx} \approx \frac{\Delta[H^+]}{\Delta x}$ remains correct. From the above table it also becomes apparent that the pH inside the gel is slightly lower than outside the gel, this is due to the acidity of the gel's acid groups.

6.1.2 Size

In the diffusion model the osmotic pressure is recorded as an indicator for size. The calculations according to the setup described above were performed using a homebrew computer program. This yielded results as indicated in figure 13.

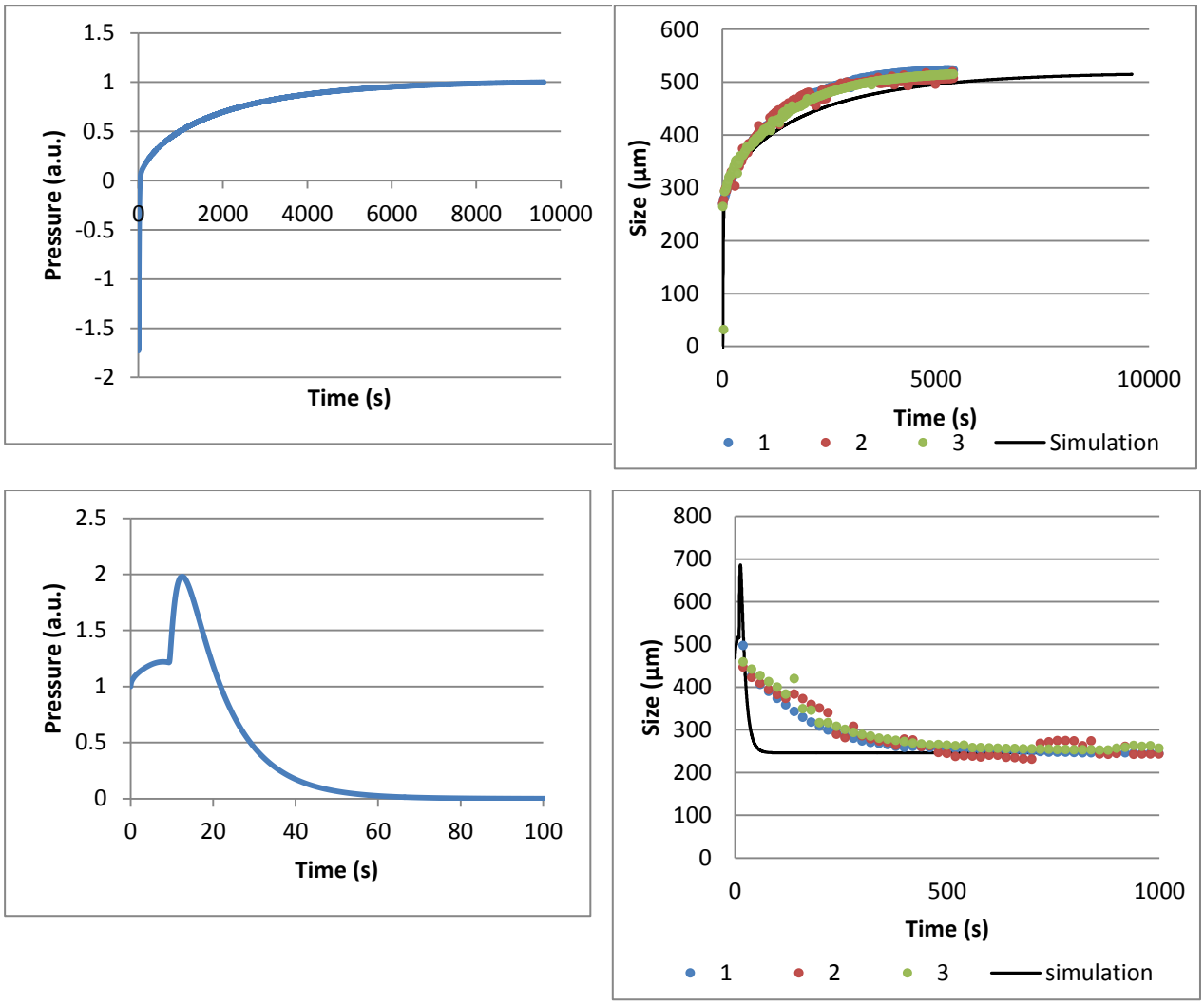


Figure 13: Top row left*: The osmotic pressure as a function of time for a gel swelling from pH 2 to pH 5 according to the diffusion model.* ***Top row right****: The data of the osmotic pressure as a function of time for a swelling hydrogel fitted to actual measurements of gel thickness over time.* ***Bottom row left****: Osmotic pressure as a function of time for a gel shrinking from pH 5 to pH 2 according to the diffusion model.* ***Bottom row right****: Data from the osmotic pressure fitted with actual measurements of gel thickness over time. Fits were performed by fitting the initial thickness and the final thickness of the gel to the pressure curve. The data from swelling and shrinking gels is courtesy of Suzanne Verkleij[16].*

When the data above is analyzed it can be seen that the diffusion model overestimates the difference between the swelling and shrinking curves.

6.2 Buffered model

6.2.1 Initial configuration

For the buffered model, the initial configuration is more easily obtained as the concentration of H^+ inside the gel at equilibrium simply equals the concentration outside the gel. As the calculation loop is such that the size is calculated at each point in time based only on the concentration of H+ (and

various constants) any somewhat reasonable initial configuration where the pH is correct will quickly converge to the correct size. Nonetheless the same approach as for the diffusion model was taken and the calculation was simply applied with the pH identical on the inside and outside to obtain the starting values.

Constant	**Value**
D_H	$9.311 \cdot 10^{-9}$ dm^2 s^{-1}
D_{HB}	$1.089 \cdot 10^{-9}$ dm^2 s^{-1}
c_{gel}^{s}	0.146 M
c_T	0.1 M
K	$10^{-4.35}$
K_B	$10^{-4.76}$
Ionic strength	0.1 M
H_0	10
B	0.01073
timestep	10^{-5} s
Delay factor	70

Table 2: *Constants used in the diffusion model calculations*

6.2.2 Size

The values for H_0 and B are chosen such that the thickness of the gel is roughly equal to the experimental values at pH 2 and 5. H_0 represents the equilibrium hydration value of a gel without any polymer acid groups. B represents the force counteracting the osmotic pressure caused by the Donnan potential. From early calculations it was found that the swelling and shrinking processes are much faster than found experimentally. It is possible that this is due to diffusion being slower inside the gel than in solution. The model includes this by reducing the diffusion coefficient through a tortuosity factor. However this may not incorporate all of the effects that cause diffusion to slow down. An arbitrary delay factor of 70 was included into the diffusion coefficient, this makes the experimental and theoretical timescales comparable. This factor of 70 was chosen as it corresponds well with the shrinking curve. When the program is then run the outcome for the theoretical gel size over time can be compared to experiment. The outcome can be seen in figure 14.

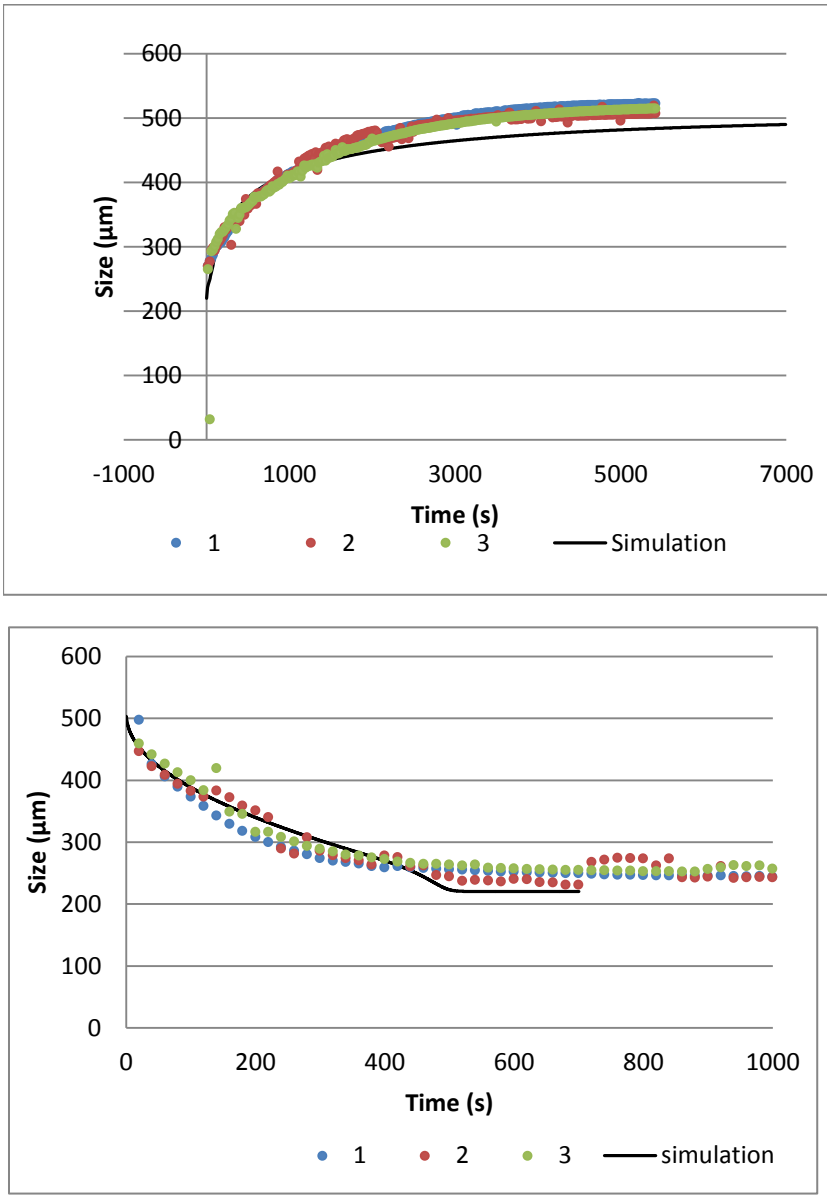


Figure 14: *Gel sizes found from the buffered model compared to experimental data.*

Both graphs show the size as a function of time for a gel. The simulation line shows the thickness of the gel according to the buffered model, the other points represent experimental data. The experimental data from swelling and shrinking gels is courtesy of Suzanne Verkleij[16]. ***Top:*** *Data for a gel swelling from pH 2 to pH 5.* ***Bottom****: Data for a gel shrinking from pH 5 to pH 2.*

As can be seen from figure 14 the difference between the rate of swelling and shrinking is close to the difference found in experiment. This provides a major improvement over the diffusion model that grossly overestimated the difference. This agreement could have been even closer if not for the strange acceleration of shrinking between 400 and 500 seconds after the start. This acceleration is because when even the region that is the farthest away from the outside buffer solution starts shrinking the absolute amount of buffer is decreasing. This decrease in the amount of buffer makes diffusion go faster accelerating the process. This decrease in buffer is only partially due to a lower concentration, it is also due to the gel getting thinner and thus containing less buffer in total, as mass

is not conserved in the buffered model. This means that the acceleration of the shrinking process between 400 and 500 seconds is probably somewhat of an artifact of the fact that in the buffered model the concentration of buffer is conserved, not the total amount.

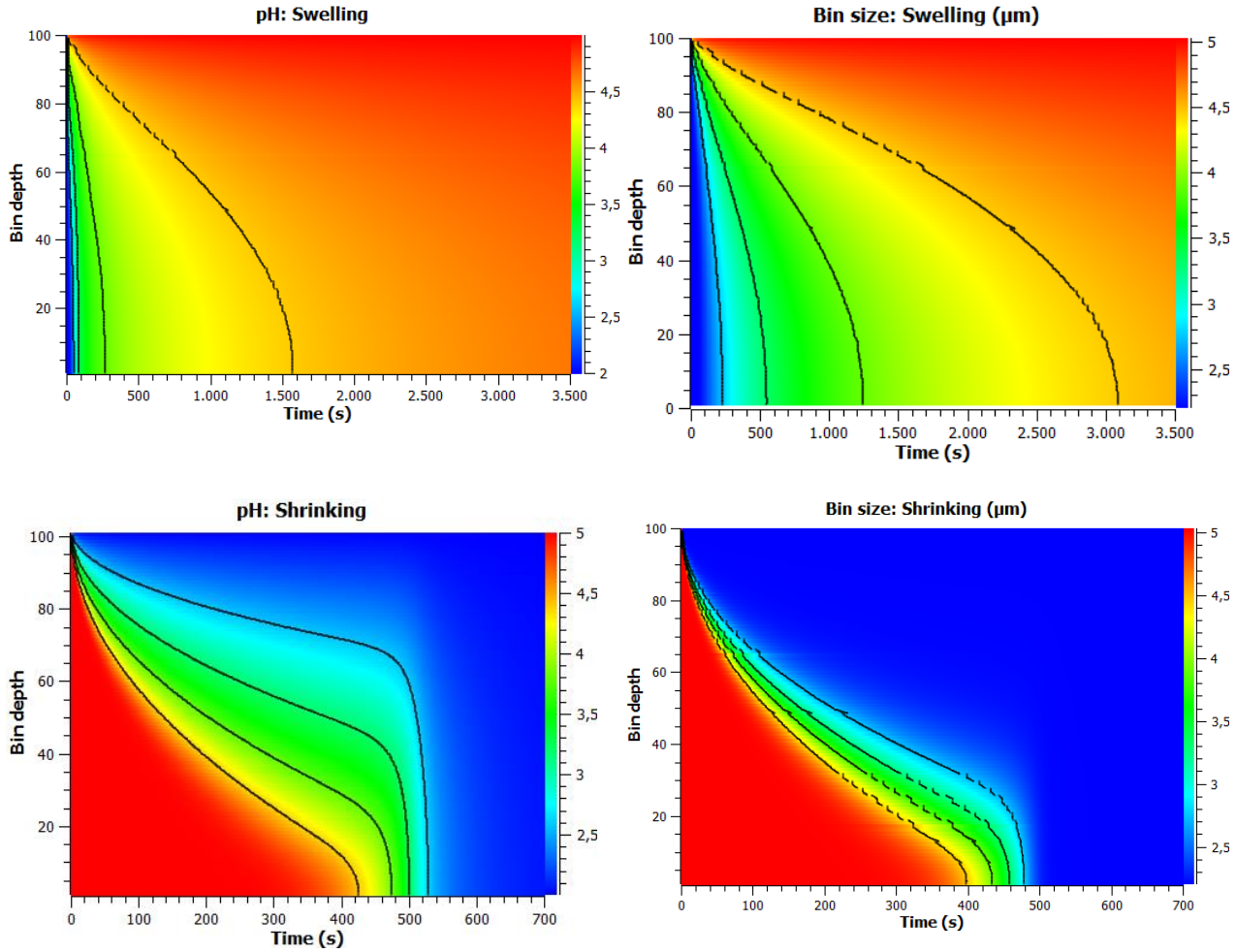


Figure 15: *A comparison of the swelling process (top row) and the shrinking process (bottom row). In all graphs a variable is plotted against both time and position using color coding. The position is given by the bin depth; this refers to how far away a point is from the substrate in a fixed frame of reference, the higher the bin depth the closer a point is to the outside buffer solution. The figures on the left show how the pH develops inside the gel over time in the case of a swelling gel or a shrinking gel. The figures on the right show how the local size of the gel changes over time, the size that is given here is the size of a single simulation unit, the overall size of the gel is calculated as the sum of all these elemental sizes.*

Figure 15 shows more of why swelling is so different from shrinking. Throughout the swelling period the pH is relatively uniform throughout the gel. The gradients appear much more strongly in the shrinking curve. As the speed at which diffusion occurs depends on the gradient this largely explains why swelling is so much slower. This is made even more important because the diffusion rate linearly depends on the concentration of H^+ rather than pH. This means that the effect of having only slight gradients during the swelling process is even more extreme than suggested by the pH versus time and place plots in figure 15. Figure 15 also shows the clear acceleration in shrinking between 400 and 500 seconds as was discussed earlier.

Conclusions

In this work the swelling process of a pH-sensitive hydrogel was investigated. Experiments were done to find whether or not this swelling process could be used for pH sensing using ferromagnetic particles.

It was found that the magnetic field resulting from magnetized cobaltferrite particles inside a layer of hydrogel gel based on an acrylic acid and hydroxyethyl acrylate copolymer, did indeed change depending on the pH of the surrounding medium. This change in magnetic field between the different states was very small. Additionally the total strength of the magnetic field was also found to decrease over time. Both Neél relaxation inside the cobalt ferrite particles and the dissolution of cobalt ferrite at low pH contributed to this. Improvements to the net signal of the sensor could be increased by improving the geometry of the set-up. However the low total magnetic field caused by the magnetic hydrogel was also very low, significantly below the earth magnetic field. These low signals will be challenging for a magnetic hydrogel based pH sensor.

The swelling process of a pH-sensitive hydrogel was investigated through numerical calculations. It was found that a large difference between the rate of swelling and the rate of shrinking could be explained through through two important notions. Firstly the size of the hydrogel does not respond linearly to the concentration of H^+, but is much more sensitive to changes in $[H^+]$ at very low concentrations. At the same time the rate of diffusion of H^+ does depend linearly on concentrations. Secondly the buffering effect of the polymer backbone at relatively low concentrations of H^+ slows down diffusion, especially in the swelling process as smaller gradients are present in the case of swelling.

It was also found that the presence of other buffers, besides the polymer backbone, helps to reduce the difference in swelling and shrinking rates. The current model to describe the swelling and shrinking process still cannot reproduce the actual swelling rates found in experiment; there still is an arbitrary delay factor. Further investigation is needed to find out exactly which physical processes cause these reduced rates of swelling and shrinking.

Bibliography

[1] T.R. Hoare, D.S. Kohane, Polymer, 2008, 49, 1993-2007

[2] A.J. Quarfoot et al. U.S. Patent No. 4,909,244. 20 Mar. 1990.

[3] T. Tanaka et al. Science, 1982, vol 218, 4571, 467-469

[4] T. Miyata et al. Advanced Drug Delivery Reviews, 2002, 54, 79-98

[5] H.J. Van der Linden et al. Analyst, 2003, 128, 325-331

[6] A. Richter et al. Sensors 2008, 8, 561-581

[7] P. Gupta et al. DDT 2002, 7, 569-579

[8] M.P.B. Van Bruggen, J.B.A. Van Zon. Sensors and Actuators A, 2010, 158, 240-248

[9] L. Brannon-Peppas, N.A. Peppas. Chemical Engineering Science, 1991, 46, 715-722

[10] R. Kötitz et al. Journal of Magnetism and Magnetic Materials, 1995, 149, 42-46

[11] S. Van Berkum et al. Int. J. Mol. Sci. 2013, 14, 10162-10177

[12] F. Tourinho et al. Mater. Sci. 1990, 25, 3249-3254

[13] S.K. De et al. Journal of Microelectromechanical systems 2002, 11, 544-555

[14] Dong, H., Du, H., Qian, X. J. Phys. Chem. A 2008, 112, 12687-12694

[15] S.K. De, N.R. Aluru, Mechanics of Materials 36 (2004) 395-410

[16] S.P. Verkleij," Synthesis and swelling behavior of magnetic pH-responsive hydrogels", Masterthesis, November 2013

Appendix A: Diffusion model computer code

Below is the the source code for the computer software used to make the calculations in chapter 6.1. The version used is version 8.01. The software is written in the C programming language.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define bins 100
#define version "swell8.01"

//Some parameters that can be tuned
//The values here are the defaults
//The values the user wants are set later.
double pKa = 5.0;
double gelpH = 5.0;
double pH = 5.0;
double ionic_strength = 0.1; //Molar
double gelsize = 100.0;     //micrometers
double timestep = 1e-5;   //The finite time step length, in seconds
double simtime = 100.0;       //Time to be simulated in seconds
int writetime = 10000;       //amount of steps between data collection events

//Some constants:
double NA = 6.0221429E23;       //Avogadro constant (mol-1)
double element_volume = 1e-15;  //1 cubic micrometer, expressed in L
double Nwater = 55.50843506;    //Mol/L
double Ngel = 0.08;             //Mol/L (definition will change below)
double element_thickness, element_area, NKa, pressure_const; //Some other globals that will be set later
double NH, NCl, NNa, NAc, NHAc;       //Amounts outside of the gel.

//Diffusion coefficients in dm2/s
//All of these will be changed in other, convenient forms later
double difH = 93.11e-8;
double difNa = 13.34e-8;
double difCl = 20.3e-8;
double difAc = 10.89e-8;

//Also important, the arrays with data:
double gel_NH[bins];
double gel_Ngelmin[bins];
double gel_NCl[bins];
double gel_NNa[bins];
double gel_NAc[bins];
double gel_NHAc[bins];
double gel_Ncharge[bins];
double gel_pressure[bins];

int set_parameters()
{
   int i;
   //First, see if the user wants to read from file...
   FILE* input = fopen("input.txt","r");
   if (input)
```

```
    {
        //Aha! File found! We'll try to use that as input then...
        double temp;
        fscanf(input,"Bins: %d\n",&i);
        if (i != bins)
        {
            printf("Error in reading input!");
        }
        //Okay, now for the actual values:
        fscanf(input,"%lg ",&temp);
        for (i=0;i<bins;i++)
            fscanf(input,", %lg ",&gel_NH[i]);
        fscanf(input,"\n%lg ",&temp);
        for (i=0;i<bins;i++)
            fscanf(input,", %lg ",&gel_Ngelmin[i]);
        fscanf(input,"\n%lg ",&temp);
        for (i=0;i<bins;i++)
            fscanf(input,", %lg ",&gel_NCl[i]);
        fscanf(input,"\n%lg ",&temp);
        for (i=0;i<bins;i++)
            fscanf(input,", %lg ",&gel_NNa[i]);
        fscanf(input,"\n%lg ",&NHAc);
        for (i=0;i<bins;i++)
            fscanf(input,", %lg ",&gel_NHAc[i]);
        fscanf(input,"\n%lg ",&NAc);
        for (i=0;i<bins;i++)
            fscanf(input,", %lg ",&gel_NAc[i]);
        fscanf(input,"\n\npH: %lg",&pH);
        fscanf(input,"\nsize: %lg",&gelsize);
        fscanf(input,"\nTimestep: %lg",&timestep);
        fscanf(input,"\nsimtime: %lg",&simtime);
        fscanf(input,"\nwritetime: %d",&writetime);
        fscanf(input,"\nIS: %lg",&ionic_strength);

        //The essential parameters here are the ionic strength
        //and the pH, as we'll fill the outside values based on those.
        //Also the values of NAc and NHAc will be used as presented
        //Values collected in temp will simply be discarded.
        fclose(input);
    }
    else
    {
        //file named 'input'
        //Asking the user for parameters then...
        printf("Please enter the parameters (Use 0 for default):\n");
        double newvalue[7] = {0.0,0.0,0.0,0.0,0.0,0.0,0.0};

        printf("gel pH, pH, ionic strength, Gel size, Simulation time, Timestep, # steps between datapoints\n");
        scanf("%lg, %lg, %lg, %lg, %lg,
%lg",&newvalue[0],&newvalue[1],&newvalue[2],&newvalue[3],&newvalue[4],&newvalue[5]);
        if (newvalue[0] > 0)
            gelpH = (double)newvalue[0];
        if (newvalue[1] > 0)
            pH = (double)newvalue[1];
        if (newvalue[2] > 0)
```

```
        ionic_strength = (double)newvalue[2];
    if (newvalue[3] > 0)
        gelsize = (double)newvalue[3];
    if (newvalue[4] > 0)
        simtime = (double)newvalue[4];
    if (newvalue[5] > 0)
        timestep = (double)newvalue[5];
    else
        timestep *= bins / gelsize;
    if (newvalue[6] > 0)
        writetime = (int)newvalue[6];

    NAc = 0.0;
    NHAc = 0.0;
    //Fill the simulation arrays with initial values:
    double m = pow(10.0,-gelpH);
    double n = NA * element_volume * m;
    for (i=0; i < bins; i++)
        gel_NH[i] = n;
    n = Ngel * NA * element_volume * pow(10.0,-pKa) / (pow(10.0,-pKa) + m);
    for (i=0; i < bins; i++)
        gel_Ngelmin[i] = n;
    m = ionic_strength * NA * element_volume - NAc;
    for (i=0; i < bins; i++)
        gel_NCl[i] = m;
    m = ionic_strength * NA * element_volume - gel_NH[0] + n;
    for (i=0; i < bins; i++)
        gel_NNa[i] = m;
    for (i=0; i < bins; i++)
    {
        gel_NAc[i] = NAc;
        gel_NHAc[i] = NHAc;
    }


}

//Here we set the necessary parameters
//Here we change the constants to their required values:
element_thickness = gelsize / bins;                    //Thickness of each volume element
element_area = element_volume / (element_thickness * 1.0e-5);//Area of each volume element
Ngel *= NA * element_volume;                           //Number of gel units per volume element
NKa = pow(10,-pKa) * NA * element_volume;              //Acid constant times NA times the volume of
the volume element
NH = pow(10,-pH) * NA * element_volume;                //Number of H+ atoms in volume elment
outside gel.
NCl = ionic_strength * NA * element_volume - NAc;
NNa = NCl - NH + NAc;
pressure_const = 8.314462175*298 * (element_area * 0.01)  / (NA * element_volume); //0.01 to make
dm2 into m2


//Here the diffusion coefficients are changed
//For details, see supporting info
difH *= timestep / pow(element_thickness * 1.0e-5,2);
```

```
    difNa *= timestep / pow(element_thickness * 1.0e-5,2);
    difCl *= timestep / pow(element_thickness * 1.0e-5,2);
    difAc *= timestep / pow(element_thickness * 1.0e-5,2);



    //Okay, everything is set here. Time to print everything on screen so the user knows the final set of
parameters:
    printf("Final parameters:\n");
    printf("pKa: %lg \n",pKa);
    printf("pH outside gel: %lg \n",pH);
    printf("pH inside gel: %lg \n",-log10(gel_NH[bins/2]/(NA * element_volume)));
    printf("bins: %d \n",bins);
    printf("gelsize: %lg \n",gelsize);
    printf("timestep: %lg \n",timestep);
    printf("time to be simulated: %lg \n",simtime);
    printf("amount of steps between data collection: %d \n",writetime);

    //And print all the derived stuff as well:
    printf("\nSimulation volume: %lg L\n",element_volume);
    printf("Element thickness: %lg micrometre\n",element_thickness);
    printf("Simulation area: %lg dm2\n",element_area);
    printf("Hydrogen diffusion coefficient: %lg per %lg s\n",difH,timestep);
    printf("Chlorine diffusion coefficient: %lg per %lg s\n",difCl,timestep);
    printf("Sodium diffusion coefficient: %lg per %lg s\n",difNa,timestep);
    printf("Acetate diffusion coefficient: %lg per %lg s\n",difAc,timestep);

    //Here we do exactly the same but now we write everything to a file:
    FILE* expdata = fopen("parameters.txt","w");
    fprintf(expdata, "Sofware version: %s\n",version);
    fprintf(expdata, "Final parameters:\n");
    fprintf(expdata, "pKa: %lg \n",pKa);
    fprintf(expdata, "pH outside gel: %lg \n",pH);
    fprintf(expdata, "pH inside gel: %lg \n",-log10(gel_NH[bins/2]/(NA * element_volume)));
    fprintf(expdata, "bins: %d \n",bins);
    fprintf(expdata, "gelsize: %lg \n",gelsize);
    fprintf(expdata, "timestep: %lg \n",timestep);
    fprintf(expdata, "time to be simulated: %lg \n",simtime);
    fprintf(expdata, "amount of steps between data collection: %d \n",writetime);
    fprintf(expdata, "\nSimulation volume: %lg L\n",element_volume);
    fprintf(expdata, "Element thickness: %lg micrometre\n",element_thickness);
    fprintf(expdata, "Simulation area: %lg dm2\n",element_area);
    fprintf(expdata, "Hydrogen diffusion coefficient: %lg per %lg s\n",difH,timestep);
    fprintf(expdata, "Chlorine diffusion coefficient: %lg per %lg s\n",difCl,timestep);
    fprintf(expdata, "Sodium diffusion coefficient: %lg per %lg s\n",difNa,timestep);
    fprintf(expdata, "Acetate diffusion coefficient: %lg per %lg s\n",difAc,timestep);
    fclose(expdata);
    return 0;
}
int write_data(FILE* dumpfile, double (*appdata)[bins], double* time)
{
    int i;
    fprintf(dumpfile,"%.20lg ",*time);
    for (i=0; i < bins; i++)
        fprintf(dumpfile,", %.20lg ",(*appdata)[i]);
```

```
    fprintf(dumpfile,"\n");
    return 0;
}
int pressure()
{
    int i;
    for (i=0;i<bins;i++)
    {
        gel_pressure[i] = gel_NH[i]+gel_NCl[i]+gel_NNa[i]+gel_NAc[i]+gel_NHAc[i] -NH -NCl -NNa -
NAc -NHAc;
        gel_pressure[i] *= pressure_const;
    }
    return 0;
}
int charge()
{
    int i;
    for (i=0;i<bins-1;i++)
    {
        gel_Ncharge[i] = difH * (gel_NH[i]-gel_NH[i+1]) + difNa * (gel_NNa[i]-gel_NNa[i+1]) - difCl *
(gel_NCl[i]-gel_NCl[i+1]) - difAc * (gel_NAc[i]-gel_NAc[i+1]);
        gel_Ncharge[i] /= difH * (gel_NH[i]+gel_NH[i+1]) + difNa * (gel_NNa[i]+gel_NNa[i+1]) + difCl *
(gel_NCl[i]+gel_NCl[i+1]) + difAc * (gel_NAc[i]+gel_NAc[i+1]);
    }
    //Here we need the factor for the field between the final bin and the outside
    //This is slightly different from the rest of the gel.
    gel_Ncharge[bins-1] = difH * (gel_NH[bins-1]-NH) + difNa * (gel_NNa[bins-1]-NNa) - difCl *
(gel_NCl[bins-1]-NCl) - difAc * (gel_NAc[bins-1]-NAc);
    gel_Ncharge[bins-1] /= difH * (gel_NH[bins-1]+NH) + difNa * (gel_NNa[bins-1]+NNa) + difCl *
(gel_NCl[bins-1]+NCl) + difAc * (gel_NAc[bins-1]+NAc);
    return 0;
}

int diffuse(double (*amounts)[bins], double outside_conc, double diffusion_const, int z)
{
    int i;
    double tempvalue[bins];
    for (i=0; i < bins - 1; i++)
    {
        tempvalue[i] = diffusion_const * (((*amounts)[i+1] - (*amounts)[i]) + z *
((*amounts)[i+1]+(*amounts)[i]) * gel_Ncharge[i]);
    }
    tempvalue[bins-1] = diffusion_const * ((outside_conc - (*amounts)[bins-1]) + z * (outside_conc +
(*amounts)[i]) * gel_Ncharge[i]);
    /*if (outside_conc == NH)
        printf("\nH in: %lg %lg",tempvalue[bins-1],tempvalue[bins-1]/(diffusion_const * ((*amounts)[i+1] -
(*amounts)[i])));
    if (outside_conc == NNa)
        printf("\nNa in: %lg %lg",tempvalue[bins-1],tempvalue[bins-1]/(diffusion_const * ((*amounts)[i+1] -
(*amounts)[i])));
    if (outside_conc == NCl)
        printf("\nCl in: %lg %lg",tempvalue[bins-1],tempvalue[bins-1]/(diffusion_const * ((*amounts)[i+1] -
(*amounts)[i])));*/
    //Set the new values and we're done!
    for (i=0; i < bins -1; i++)
```

```
    {
        (*amounts)[i] += tempvalue[i];
        (*amounts)[i+1] -= tempvalue[i];
    }
    (*amounts)[bins-1] += tempvalue[bins-1];
    return 0;
}
int ionisation()
{
    //Here is our wonderful ionisation equation, see the additional information for info about this equation.
    //Here it is split in two to reduce the amount of calculating steps (Trying to reduce computing time)
    double ionconst, newgelmin, outconst = Ngel*NKa;
    int i;
    for (i=0; i < bins; i++)
    {
        ionconst = (NKa + gel_NH[i] - gel_Ngelmin[i]) / 2;
        newgelmin = sqrt(pow(ionconst,2.0) + outconst) - ionconst;
        gel_NH[i] += newgelmin - gel_Ngelmin[i];
        gel_Ngelmin[i] = newgelmin;
    }
    return 0;
}
int simulate()
{
    //Some variables used throughout:
    FILE* NHout;
    FILE* gelminout;
    FILE* NClout;
    FILE* NNaout;
    FILE* NHAcout;
    FILE* NAcout;
    FILE* pressureout;
    //Some timing variables:
    int cycles = 0;
    double time = 0.0;

    //Here we open the files for writing data:
    NHout = fopen("NH.txt", "w");
    gelminout = fopen("gelmin.txt", "w");
    NClout = fopen("NCl.txt", "w");
    NNaout = fopen("NNa.txt", "w");
    NAcout = fopen("NAc.txt", "w");
    NHAcout = fopen("NHAc.txt", "w");
    pressureout = fopen("pressure.txt", "w");

    //And write out the starting state:
    pressure();
    write_data(NHout,&gel_NH,&time);
    write_data(gelminout,&gel_Ngelmin,&time);
    write_data(NClout,&gel_NCl,&time);
    write_data(NNaout,&gel_NNa,&time);
    write_data(NAcout,&gel_NAc,&time);
    write_data(NHAcout,&gel_NHAc,&time);
    write_data(pressureout,&gel_pressure,&time);
```

```
    //And here's the actual simulation:
    printf("Calculating...");
    for (;time < simtime; time += timestep)
    {
        //Diffusion:
        charge();
        diffuse(&gel_NH,NH,difH,1);
        diffuse(&gel_NNa,NNa,difNa,1);
        diffuse(&gel_NCl,NCl,difCl,-1);
        diffuse(&gel_NAc,NAc,difAc,-1);
        diffuse(&gel_NHAc,NHAc,difAc,0);

        //ionisation:
        ionisation();

        cycles += 1;
        if (cycles % writetime == 0)
        {
            //Time to write to the files:
            write_data(NHout,&gel_NH,&time);
            write_data(gelminout,&gel_Ngelmin,&time);
            write_data(NClout,&gel_NCl,&time);
            write_data(NNaout,&gel_NNa,&time);
            write_data(NAcout,&gel_NAc,&time);
            write_data(NHAcout,&gel_NHAc,&time);
            //Also, calculate pressure
            //and write this to file:
            pressure();
            write_data(pressureout,&gel_pressure,&time);
        }
    }
    fclose(NHout);
    fclose(gelminout);
    fclose(NClout);
    fclose(NNaout);
    fclose(NAcout);
    fclose(NHAcout);
    fclose(pressureout);
    return(1);
}
int putout(double computingtime)
{
    double outvalue = 0.0;
    double avgionisation = 0.0;
    int i = 0, l = 0;
    int j = (int)(simtime / (timestep * (double)writetime)) + 2;
    FILE* input;
    FILE* output;
    FILE* sizeout;

    //Now the final values for all concentrations
    //Just so we can read them out again if we need to:
    output = fopen("out_data.txt","w");
    fprintf(output,"Bins: %d\n",bins);
```

```
write_data(output,&gel_NH,&NH);
write_data(output,&gel_Ngelmin,&Ngel);
write_data(output,&gel_NCl,&NCl);
write_data(output,&gel_NNa,&NNa);
write_data(output,&gel_NHAc,&NHAc);
write_data(output,&gel_NAc,&NAc);
fprintf(output,"\npH: %lg",pH);
fprintf(output,"\nsize: %lg",gelsize);
fprintf(output,"\nTimestep: %lg",timestep);
fprintf(output,"\nsimtime: %lg",simtime);
fprintf(output,"\nwritetime: %d",writetime);
fprintf(output,"\nIS: %lg",ionic_strength);
fclose(output);

//Equilibrium output:
/*output = fopen("Equidata.txt","w");
fprintf(output,"%.20lg ",NH);
for (i=0; i < 100; i++)
   fprintf(output,", %.20lg ",gel_NH[1]);
   fprintf(output,"\n%.20lg ",Ngel);
for (i=0; i < 100; i++)
   fprintf(output,", %.20lg ",gel_Ngelmin[1]);
   fprintf(output,"\n%.20lg ",NCl);
for (i=0; i < 100; i++)
   fprintf(output,", %.20lg ",gel_NCl[1]);
   fprintf(output,"\n%.20lg ",NNa);
for (i=0; i < 100; i++)
   fprintf(output,", %.20lg ",gel_NNa[1]);
   fprintf(output,"\n%.20lg ",NHAc);
for (i=0; i < 100; i++)
   fprintf(output,", %.20lg ",gel_NHAc[1]);
   fprintf(output,"\n%.20lg ",NAc);
for (i=0; i < 100; i++)
   fprintf(output,", %.20lg ",gel_NAc[1]);
fclose(output);*/

input = fopen("NH.txt","r");
output = fopen("pH.txt","w");
for(l=0;l<j;l++)
{
   outvalue = 0.0;
   fscanf(input,"%lg ",&outvalue);
   fprintf(output,"%.20lg ",outvalue);
   for (i=0;i<bins;i++)
   {
      fscanf(input,", %lg ",&outvalue);
      fprintf(output,", %.20lg ",-log10(outvalue/(NA * element_volume)));
   }
   fscanf(input,"\n");
   fprintf(output,"\n");
}
fclose(output);
fclose(input);

input = fopen("gelmin.txt","r");
```

```
output = fopen("ionisation.txt","w");
sizeout = fopen("size.txt","w");
for(l=0;l<j;l++)
{
   outvalue = 0.0;
   fscanf(input,"%lg ",&outvalue);
   fprintf(output,"%.20lg ",outvalue);
   fprintf(sizeout,"%.20lg ",outvalue);
   avgionisation = 0.0;
   for (i=0;i<bins;i++)
   {
      fscanf(input,", %lg ",&outvalue);
      outvalue /= Ngel;
      fprintf(output,", %.20lg ",outvalue);
      avgionisation += outvalue;
   }
   avgionisation /= bins;
   fprintf(sizeout,", %.20lg ",avgionisation);
   fscanf(input,"\n");
   fprintf(output,"\n");
}
fclose(sizeout);
fclose(output);
fclose(input);



//Put out the charge as well:
double totcharge[5] = {0.0,0.0,0.0,0.0,0.0};
FILE* input_NH = fopen("NH.txt","r");
FILE* input_NNa = fopen("NNa.txt","r");
FILE* input_NCl = fopen("NCl.txt","r");
FILE* input_NAc = fopen("NAc.txt","r");
FILE* input_Ngelmin = fopen("gelmin.txt","r");
output = fopen("totcharge.txt","w");
for(l=0;l<j;l++)
{
   fscanf(input_NH,"%lg ",&outvalue);
   fscanf(input_NNa,"%lg ",&outvalue);
   fscanf(input_NCl,"%lg ",&outvalue);
   fscanf(input_NAc,"%lg ",&outvalue);
   fscanf(input_Ngelmin,"%lg ",&outvalue);
   fprintf(output,"%.20lg ",outvalue);
   for (i=0;i<bins;i++)
   {
      fscanf(input_NH,", %lg",&totcharge[0]);
      fscanf(input_NNa,", %lg",&totcharge[1]);
      fscanf(input_NCl,", %lg",&totcharge[2]);
      fscanf(input_NAc,", %lg",&totcharge[3]);
      fscanf(input_Ngelmin,", %lg",&totcharge[4]);
      fprintf(output,", %.20lg",totcharge[0]+totcharge[1]-totcharge[2]+totcharge[3]-totcharge[4]);
   }
   fscanf(input_NH,"\n");
   fscanf(input_NNa,"\n");
```

```
        fscanf(input_NCl,"\n");
        fscanf(input_NAc,"\n");
        fscanf(input_Ngelmin,"\n");
        fprintf(output,"\n");
    }
    fclose(input_NH);
    fclose(input_NNa);
    fclose(input_NCl);
    fclose(input_NAc);
    fclose(input_Ngelmin);


    //Append the parameter section so we can look up how long the actual calculation took.
    output = fopen("parameters.txt","a");
    fprintf(output,"Computing time: %lg\n",computingtime);
    fclose(output);

    return 0;
}
int main()
{
    printf("Welcome to SSSSS!\n\n\n");
    set_parameters();
    double computingtime = 0.0;
    clock_t starttime, endtime;
    starttime = clock();
    simulate();
    printf("\n\nCalculation done! Now calculating final output parameters\n");
    endtime = clock();
    computingtime = (double)(endtime-starttime)/CLOCKS_PER_SEC;
    putout(computingtime);
    printf("Done!\n\n");
    return 0;
}
```

Appendix B: Buffered model computer code

Below is the the source code for the computer software used to make the calculations in chapter 6.2. The version used is version 6.0. The software is written in the C programming language.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define version "altswell6.0"
#define bins 100
//#include <time.h>

//Thermal energy:
const double KBT = 2494.33863; //J/mol  Equals 300 K * 8.3144621 J/K*mol

//standard units to be used: dm for length, mole for amounts, mole/litre for concentrations
//Here are some general variables that will be used:
double springconst;     //Equal to E/((1+u)(1-2u))
double cT;              //Total buffer concentration, this is assumed constant everywhere (Molar)
double cMon;            //Monomer concentration in the dry polymer (Molar)
double KB;              //Acid-base equilibrium constant for the buffer
double Kgel;            //Acid-base equilibrium constant for the acidic groups in the polymer
double difHzero;        //Diffusion constant of H+ in water ( dm^2 s-1 )
double difBzero;        //Diffusion constant of the buffer in water ( dm^2 s-1 )
double cOut;            //Outside concentration of H+ (Molar)
double cNaOut;          //Outside concentration of Na+ (Molar)
double cClOut;          //Outside concentration of Cl- (Molar)
double zerothickness;   //Thickness of the pure polymer (dm)
double timestep;        //Time interval between calculation steps (s)
double simtime;         //Total time to be simulated (s)

double thicknull;       //Base hydration value
double delay;           //Delay factor for diffusion

//Probably not needed in this version:
//double difH;          //Diffusion constant of H+ in the gel ( dm^2 s-1 )
//double difB;          //Diffusion constant of the buffer in the gel ( dm^2 s-1 )
//double hydration;     //Hydration value, equals volume of the liquid over the volume of the polymer


//And the all important arrays:
double gelsize[bins];       //Whoho! location dependent hydration value, equals volume of the liquid over the
volume of the polymer
double flow[bins];      //Value for d(cH)/d(t) (Molar s-1)
double cH[bins];        //Concentration of H+ at every bin (Molar)
double Tcon[bins];      //Array holding some intermediate values used for calculating the flow, Called X-term in
the documentation
double Xterm[bins];     //Array holding some intermediate values used for calculating the flow, Called t-term in
the documentation
double grad[bins];      //Value for d(cH)/d(x) (Molar dm-1)
double charge[bins];    //Concentration of negatively charged groups in the polymer (Molar)
double cNa[bins];       //Concentration of Na+ inside the gel (Molar)
double cCl[bins];       //Concentration of Cl- inside the gel (Molar)

double difH[bins];      //Diffusion coefficient for H+ ions
```

```
double difB[bins];      //Diffusion coefficient for buffer molecules
double phi[bins];       //Value for phi, equals H/(1+H); Kept for ease of use...
double pressure[bins];  //Osmotic pressure inside the gel (Units: N dm^-2 ??? Check!!!)

int get_hydration()
{
   /*Calculates hydration from osmotic pressure (calculated in the Donnan function)*/
   int i;
   for(i=0; i < bins; i++)
   {
      gelsize[i] = thicknull + pressure[i] / springconst;
      //phi[i] = gelsize[i] / (1.0 + gelsize[i]);
   }
   return 1;
}
int get_diffusivity()
{
   /*Here the values for difH and difB are calculated.
   Note that the value for difH[i] or difB[i] actually
   refers to the value at point [i+0.5]
   This is calculated by taking the average hydration value*/
   int i;
   double diffa, diffb;
   for (i=0; i < bins - 1; i++)
   {
      diffa = (gelsize[i+1]+gelsize[i])/2.0;
      diffb = pow(diffa/(2.0+diffa),2.0);
      difH[i] = difHzero * diffb;
      difB[i] = difBzero * diffb;
   }
   //For the final bin the hydration outside is taken to be equal to the hydration in the final bin.
   diffb = pow(gelsize[bins-1]/(2.0+gelsize[bins-1]),2.0);
   difH[bins-1] = difHzero * diffb;
   difB[bins-1] = difBzero * diffb;
   return 1;
}
int get_grad()
{
   /*This function gets the value for dcH/dx. Note that the value for grad[i]
   actually refers to the value at point grad[i+0.5]
   For calculating the size of each individual bin, the following equation is used:
   binsize = (1+H)*zerothickness / bins*/
   double zeroconst = 2.0 * bins / zerothickness;
   int i;
   grad[0] = zeroconst * (cH[1] - cH[0]) / (2.0 + gelsize[1] + gelsize[0]);
   for (i=1; i < bins-1; i++)
   {
      grad[i] = zeroconst * (cH[i+1] - cH[i])/(2.0 + gelsize[i+1] + gelsize[i]);
   }
   //The outside binsize is assumed to be the same as the bin bordering it (The bin bordering the solvent)
   grad[bins-1] = zeroconst * (cOut - cH[bins-1])/(2.0 + gelsize[bins-1] + gelsize[bins-1]);
   return 1;
}
int solvexterm()
{
```

```
int i;
double dX = zerothickness / bins;
double avgconcplus, avgconcmin; //The average concentrations at point i+0.5 and i-0.5 respectively
double Havg;
//First calculate phi:
for (i=0; i < bins-1; i++)
{
   Havg = gelsize[i+1] + gelsize[i];
   phi[i] = Havg / (2.0 + Havg);
}
phi[bins-1] = gelsize[bins-1] / (1.0 + gelsize[bins-1]);


//For the first point it is assumed that there's another point before point 0
//that has exactly the same properties as point 0.
//This means that dcH/dx is zero at point -0.5, greatly simplifying equations

avgconcplus = (cH[0]+cH[1])/2.0;
Xterm[0] = phi[0]*(1.0 + difB[0]*cT/(difH[0]*(KB + avgconcplus)))*difH[0]*grad[0];
Xterm[0] /= dX;
avgconcmin = (cH[1]+cH[0])/2.0;

for(i=1; i < bins-1; i++)
{
   avgconcplus = (cH[i]+cH[i+1])/2.0;
   Xterm[i] = phi[i]*(1.0 + difB[i]*cT/(difH[i]*(KB + avgconcplus)))*difH[i]*grad[i];
   Xterm[i] -= phi[i-1]*(1.0 + difB[i-1]*cT/(difH[i-1]*(KB + avgconcmin)))*difH[i-1]*grad[i-1];
   Xterm[i] /= dX;
   avgconcmin = avgconcplus;
}
avgconcplus = (cH[bins-1]+cOut)/2.0;
Xterm[bins-1] = phi[bins-1]*(1.0 + difB[bins-1]*cT/(difH[bins-1]*(KB + avgconcplus)))*difH[bins-
1]*grad[bins-1];
Xterm[bins-1] -= phi[bins-2]*(1.0 + difB[bins-2]*cT/(difH[bins-2]*(KB + avgconcmin)))*difH[bins-
2]*grad[bins-2];
Xterm[bins-1] /= dX;

return 1;
/*Some old rubbish:


dGdX = 2.0*((cH[1] - cH[0])/(2.0 + gelsize[1] + gelsize[0])) / dX;
Xterm[0] = phi[0]*dGdX * (difH[0] + difB[0]*cT/(KB + cH[0]));
//dGdX = (cH[1] - cH[0]) / dX;
//Xterm[0] -= dGdX * phi[0] * difB[0]*cT*grad[0] / pow(KB + cH[0],2.0);
for (i=1; i < bins-1; i++)
{
   dGdX = 2.0*((cH[i+1] - cH[i])/(2.0 + gelsize[i+1] + gelsize[i]) - (cH[i] - cH[i-1])/(2.0 + gelsize[i] +
gelsize[i-1])) / dX;
   Xterm[i] = phi[i]*dGdX * (difH[i] + difB[i]*cT/(KB + cH[i]));
   //dGdX = ((cH[i] - cH[i-1])+(cH[i+1] - cH[i])) / (2.0*dX);
   //Xterm[i] -= dGdX * phi[i] * difB[i]*cT*grad[i] / pow(KB + cH[i],2.0);
}
dGdX = 2.0*((cOut - cH[bins-1])/(2.0 + gelsize[bins-1] + gelsize[bins-1]) - (cH[bins-1] - cH[bins-
2])/(2.0 + gelsize[bins-1] + gelsize[bins-2])) / dX;
```

```
    //The hydration value for the outside world is simply assumed to be equal to the hydration in the last bin.
    Xterm[bins-1] = phi[bins-1]*dGdX * (difH[bins-1] + phi[bins-1] * difß[bins-1]*grad[bins-1]/(KB +
cH[bins-1]));
    //dGdX = ((cH[bins-1] - cH[bins-2])+(cOut - cH[bins-1])) / (2.0*dX);
    //Xterm[bins-1] -= dGdX * phi[bins-1] * difß[bins-1]*cT*grad[bins-1] / pow(KB + cH[bins-1],2.0);
    */
}

double solveeleven()
{
    /*Function to calculate the value for dcH/dt*/
    int i;          //Tracker variable for loops
    double temp;    //Holds temporary values

    //Let's fill up Tcon, which are a lot of constants bundled for ease of use:
    for (i=0; i < bins; i++)
    {
        temp = Kgel + cH[i];
        Tcon[i] = gelsize[i] + cMon/temp - cMon*cH[i]/(temp*temp) + gelsize[i]*cT / (KB + cH[i]) -
gelsize[i]*cT*cH[i]/((KB + cH[i])*(KB + cH[i]));
    }
    //Find the value for deltaX here:
    get_diffusivity();  //Get the correct diffusion coefficients
    get_grad();  //Get dH/dx
    solvexterm();       //Get the Xcon array
    //And solve the flow:
    for(i=0; i <bins; i++)
    {
        flow[i] = Xterm[i]/Tcon[i];
        cH[i] += flow[i]*timestep;
    }
    return 0.0;
}
int solve_Donnan()
{
    /*Function to find values for the Na+ and Cl- concentrations given a certain H+ concentration*/
    int i;
    for(i=0; i<bins; i++)
    {
        //First find the charge concentration on the polymer:
        charge[i] = cMon*Kgel/ (gelsize[i]*(Kgel + cH[i]));
        //From this, deduce the Na+ concentration (see notes):
        cNa[i] = (sqrt(charge[i]*charge[i] + 4*cNaOut*cClOut) - charge[i]) / 2;
        //Cl- concentration:
        cCl[i] = cClOut*cNaOut/cNa[i];
        //For added bonus the osmotic pressure can be calculated:
        pressure[i] = KBT * ((cNa[i]-cNaOut)+(cCl[i]-cClOut));
    }
    return 1;
}
int set_parameters()
{
    /*This function sets all of the parameters for the calculation*/
    //First some default values
    cT = 0.1;               //Molar
```

```
cMon = 0.146;          //Molar in dry gel. About 10 %vol, 1.051 g/mL 72.06 g/mol
//cMon = 1.8;            //Molar in dry gel, value as used in De et al. 2002
KB = pow(10.0,-4.76);   //pKa of acetic acid is 4.76
//KB = 6.2*pow(10.0,-5.0);
Kgel = pow(10.0,-4.35); //pKa of loose acrylic acid monomers ~4.35
//Kgel = pow(10.0,-2.0);
difHzero = 93.11E-8;    //dm^2 s^-1
difBzero = 10.89E-8;    //acetate diffusion constant
//difBzero = 8.79E-8
zerothickness = 20E-5;  //20 um in dm.
cOut = 1E-5;
cNaOut = 0.1;
cClOut = 0.1;
timestep = 1e-5;
simtime = 1;

springconst = 0.01073;
thicknull = 10;
delay = 70;
difHzero /= delay;
difBzero /= delay;
//And fill in the arrays:
int i;
for(i=0; i < bins; i++)
{
   gelsize[i] = 10;       //This is the starting hydration value;  Vl/Vs
   phi[i] = gelsize[i] / (1.0 + gelsize[i]);
   cH[i] = cOut;
   cNa[i] = cNaOut;
   cCl[i] = cClOut;
}
//Read any parameters from a file if needed:
//These values simply replace the default ones
FILE* input = fopen("input.txt","r");
if (input)
{
   double dummy; //For variables I don't want to keep...
   printf("Reading input file");
   fscanf(input, "Buffer: %lg \n",&cT);
   fscanf(input, "Monomer: %lg \n",&cMon);
   fscanf(input, "zerothickness: %lg \n",&zerothickness);
   fscanf(input, "Outside H: %lg \n",&cOut);
   fscanf(input, "Outside Na: %lg \n",&cNaOut);
   fscanf(input, "Outside Cl: %lg \n",&cClOut);
   fscanf(input, "Simulation time: %lg \n",&simtime);
   fscanf(input, "Time step: %lg \n",&timestep);
   fscanf(input, "Spring constant: %lg \n",&springconst);
   fscanf(input, "Overall size: %lg \n",&dummy);


   for(i=0; i<bins; i++)
      fscanf(input, "%lg, ",&cH[i]);
   fscanf(input,"\n");
   for(i=0; i<bins; i++)
      fscanf(input, "%lg, ",&cNa[i]);
```

```
        fscanf(input,"\n");
        for(i=0; i<bins; i++)
            fscanf(input, "%lg, ",&cCl[i]);
        fscanf(input,"\n");
        for(i=0; i<bins; i++)
            fscanf(input, "%lg, ",&gelsize[i]);
        fscanf(input,"\n");
        for(i=0; i<bins; i++)
            fscanf(input, "%lg, ",&dummy);
        fscanf(input,"\n");
        fclose(input);
        printf("\n");
    }
    return 1;
}
int putoutfinalstats(int a)
{
    /*This function prints essential parameters to an output file.
    The format is such that it can be read as input by the program.*/
    int i;
    double gelthick = 0.0;
    for (i=0; i<bins; i++)
        gelthick += (1.0+gelsize[i])*zerothickness/bins;

    FILE* expdata;
    if (a == 1)
        expdata = fopen("initial.txt","w");
    else
        expdata = fopen("putout.txt","w");

    fprintf(expdata, "Buffer: %lg \n",cT);
    fprintf(expdata, "Monomer: %lg \n",cMon);
    fprintf(expdata, "zerothickness: %lg \n",zerothickness);
    fprintf(expdata, "Outside H: %lg \n",cOut);
    fprintf(expdata, "Outside Na: %lg \n",cNaOut);
    fprintf(expdata, "Outside Cl: %lg \n",cClOut);
    fprintf(expdata, "Simulation time: %lg \n",simtime);
    fprintf(expdata, "Time step: %lg \n",timestep);
    fprintf(expdata, "Spring constant: %lg \n",springconst);
    fprintf(expdata, "Overall size: %lg \n",gelthick);
    for(i=0; i<bins; i++)
        fprintf(expdata, "%lg, ",cH[i]);
    fprintf(expdata,"\n");
    for(i=0; i<bins; i++)
        fprintf(expdata, "%lg, ",cNa[i]);
    fprintf(expdata,"\n");
    for(i=0; i<bins; i++)
        fprintf(expdata, "%lg, ",cCl[i]);
    fprintf(expdata,"\n");
    for(i=0; i<bins; i++)
        fprintf(expdata, "%lg, ",gelsize[i]);
    fprintf(expdata,"\n");
    for(i=0; i<bins; i++)
        fprintf(expdata, "%lg, ",(1.0+gelsize[i])*zerothickness/bins);
    fprintf(expdata,"\n");
```

```
    fclose(expdata);
    return 1;
}
int putoutvars(FILE* output, double* time, double (*arrdata)[bins], double* outsideconc)
{
    /*Here various data coming from the simulate function is parsed to a file*/
    int i;
    fprintf(output,"%lg, ",*time);
    for(i=0;i<bins;i++)
        fprintf(output,"%lg, ",(*arrdata)[i]);
    fprintf(output,"%lg\n",*outsideconc);
    return(1);
}
int simulate()
{
    /*This is the location for the main calculation loop*/

    double dummy = 0.0;             //Dummy variable, used to easily pass 0.0 to the file writing function
    double curtime = 0.0;           //Time elapsed
    double oldH[bins];              //The previous value for "hydration", used to tell if the value for hydration has
equilibriated
    double deltaH = 0.001;          //Set the difference in hydration over two goes that results in a pass
    int outputfrequency = 10000;    //Data will be written to an output file once every 'outputfrequency' steps
    int i = outputfrequency;        //Tracker variable for outputting data, set to equal outputfrequency so the
initial situation will be recorded too
    int tries;                      //Tracker variable to break the hydration loop if the result does not converge
    double outsize[bins];           //The size of the gel, used for file output
    double outtotsize;              //The total size of the gel, used for file output
    int j;                          //Minor loop tracker variable
    int dHsize;                     //Determines the number of bins where the new hydration value is more than
deltaH from the old hydration value.

    //These are the output files to monitor various variables
    FILE* outH = fopen("outH.txt","w");
    FILE* outNa = fopen("outNa.txt","w");
    FILE* outCl = fopen("outCl.txt","w");
    FILE* outP = fopen("outP.txt","w");
    FILE* outcharge = fopen("outcharge.txt","w");
    FILE* outflow = fopen("outflow.txt","w");
    FILE* outhydra = fopen("outhydration.txt","w");
    FILE* outgelsize = fopen("outsize.txt","w");

    fprintf(outH,"Test, Test\n"); //This is a REALLY strange line, but Windows notepad sometimes doesn't
recognize the correct encoding for "OutH.txt" This solves this problem somehow

    //Here's the main loop!
    for (; curtime < simtime; curtime += timestep)
    {
        solveeleven();      //Solves the transient Nernst-Planck Equation for dH/dt

        for (j=0; j < bins; j++)
        oldH[j] = gelsize[j];
        solve_Donnan();     //Solves the Donnan equilibrium
        get_hydration();    //Solves the hydration for a given Donnan outcome
```

```
    dHsize = 1;         //Make sure the next for loop will do at least one loop

    for(tries = 0 ; dHsize > 0 && tries < 100 ; tries++)
    {
       /*This loop is designed to find the correct hydration
       This is because the Donnan equilibrium is changed by the hydration
       and the hydration is chaged by the Donnan equilibrium.
       This process simply assumes a new equilibrium halfway between the
       two previous hydrations to make it quickly go towards the correct hydration.*/
       for (j=0; j < bins; j++)
       {
          gelsize[j] = (gelsize[j] + oldH[j]) / 2;
          oldH[j] = gelsize[j];
       }
       solve_Donnan();
       get_hydration();
       //Let's check if there are still bins for which the tolerances are too large:
       dHsize = 0;
       for (j=0; j < bins; j++)
       {
          if (fabs(gelsize[j] - oldH[j]) > deltaH)
             dHsize++;
       }

    }
    if (tries >= 98)
       printf("Error, hydration not converging");

    i += 1;
    if (i >= outputfrequency)
    {
       //Here variables are writen to a file:
       putoutvars(outH,&curtime,&cH,&cOut);
       putoutvars(outNa,&curtime,&cNa,&cNaOut);
       putoutvars(outCl,&curtime,&cCl,&cClOut);
       putoutvars(outP,&curtime,&pressure,&dummy);
       putoutvars(outcharge,&curtime,&charge,&dummy);
       putoutvars(outflow,&curtime,&flow,&dummy);
       putoutvars(outhydra,&curtime,&gelsize,&dummy);
       outtotsize = 0.0;
       for(i=0; i < bins; i++)
       {
          outsize[i] = (1.0+gelsize[i]) * zerothickness / bins;
          outtotsize += outsize[i];
       }
       putoutvars(outgelsize,&curtime,&gelsize,&outtotsize);
       i = 0;
    }

}
//After the simulation is done, close the output files:
fclose(outH);
fclose(outNa);
fclose(outCl);
fclose(outP);
```

```
    fclose(outcharge);
    fclose(outflow);
    fclose(outhydra);
    fclose(outgelsize);
    return 1;
}
int main()
{
    //Set (or get) the initial values for different parameters
    set_parameters();
    putoutfinalstats(1);    //Puts out the initial stats in "intial.txt"
    printf("Simulating...\nPlease wait...\n");
    simulate();
    printf("Done!\n");
    putoutfinalstats(0);    //Puts out final stats in "putout.txt"
    return 0;
}
```