

Simulation of confined water in nanoporous silicates with varying surface curvatures

by

Mathilde Nygaard Kamperud

THESIS
for the degree of
MASTER OF SCIENCE



Faculty of Mathematics and Natural Sciences
Department of Physics
University of Oslo
June 2014

Abstract

In this master thesis we perform a systematic investigation of how pore geometry affects the behavior of confined water. The water is confined in nanoporous silica in an amorphous glass state. Experimental results report that water confined in nanoporous silica has different properties than water in bulk, and since the behavior of water in silica is important for many geological, biological and physical processes we wish to investigate this further. We suspect that a curved surface frustrates the packing or configuration of the water molecules inside the pore, and that the degree of frustration is dependent on the curvature of the pore surface.

The studies are based on large-scale molecular dynamics simulations with an interatomic potential that allows chemical reactions between atoms in SiO₂ and H₂O.

We perform simulations of single pore systems, where the pores in the silica have simple geometric shapes with constant Gaussian curvature on the entire pore surface. We simulated six systems with spherical pores of different curvatures, with radii between 1.3 nm and 4.3 nm. In addition we simulated one system with a cylindrical pore and one with plane pore, for comparison.

Both structural and diffusive properties are measured on the water inside the pores. We find that the pore could be divided into two regions of different behaviors, the surface region and the confined region. In the confined region, the properties measured are virtually constant, while in the surface region the properties vary with the distance to the pore surface. These regions are separated at the distance d_{surf} , which for the structural properties is $d_{\text{surf}} \sim 7\text{-}8 \text{ \AA}$, while for the diffusive properties it is $d_{\text{surf}} \sim 9\text{-}10 \text{ \AA}$.

The structural properties showed no dependence on the curvature of the pores, which is in disagreement with what we suspected. The diffusive properties showed dependence on the radius of the pore, which is most likely due to varying fraction of water in the surface region compared to the water in the confined region.

Acknowledgements

I was first introduced to Molecular Dynamics simulations through a project in the course FYS4460, held by my supervisor Anders Malthe-Sørenssen, and found it very interesting. I especially enjoyed writing my own MD simulation code, which could be used to make movies showing the trajectories of a large number of interacting atoms. I contacted Anders Malthe-Sørenssen, and he provided me with an interesting topic for a master thesis where we would use MD simulations to investigate water confined in silica. I would like to thank him for his time and guidance during the past year. Our weekly talks have been essential during this master project.

I would also like to thank my co-supervisor, Camilla Kirkemo Alm, who has been a great help, especially during the last hectic weeks of the project. Thank you for reading my thesis, and giving very helpful feedback.

Master student and my collaborator, Filip Sund, was given a master project very similar to mine, and naturally we became collaborators. I have appreciated the time we have spent writing code together, and thank you for teaching me technical skills and helping me whenever I had a computer related problem.

I want to thank the people at the Computational Physics department, and an extra thanks to the three people I share office with: Emilie Fjørner, Jonas van den Brink and Øydis Larsen. I have enjoyed sharing space and time with you the past year.

My gratitude goes to my family, who have endured all my science related questions throughout my life. I love hanging out with you. A big thank you to my parents, Kari Anne and Nils Edvard, who have supported me throughout my entire education. I could not have done this without you.

Finally, I want to thank Andreas Våvång Solbrå. You are very important to me. Thank you for being there for me, encouraging me and for answering all my questions. You are awesome!

*Mathilde Nygaard Kamperud
Blindern, June 2014*

Contents

1	Introduction	9
1.1	Porous materials and confined fluids	9
1.2	Choice of nanoporous system	10
1.2.1	Hydrophilic silica and water	10
1.2.2	Pore shapes studied in this thesis	11
1.3	Choice of numerical method	11
1.4	Goals	12
1.5	Structure of the thesis	13
2	Characterization of pore surfaces	15
2.1	Differential geometry	15
2.1.1	Curvature of curves	15
2.1.2	Gaussian curvature	19
2.2	Pore shapes studied in this thesis	19
3	Introduction to molecular dynamics	21
3.1	Potentials	22
3.2	Integrator - Verlet algorithm	25
3.3	Periodic boundary conditions	27
3.4	Thermostats	28
3.4.1	Andersen Thermostat	29
3.4.2	Berendsen Thermostat	29
3.4.3	Nosé-Hoover Thermostat	31
3.5	Thermalization	32
3.6	Defining neighbors	33
3.6.1	Voronoi diagrams	33
3.7	Physical quantities	35
3.7.1	Temperature	35
3.7.2	Density	36
3.7.3	Diffusion	37
3.7.4	The radial distribution function	38

4 Generating a porous media	45
4.1 Creating amorphous glass	45
4.2 Methods for creating pores	46
4.3 Passivating (hydroxylate) the porous media	48
4.3.1 Developing a method for passivation	48
4.3.2 Before and after passivation	52
4.3.3 The steepest descent method	53
4.4 Injecting water	55
5 Systems	57
5.1 Description of the different systems	57
5.1.1 Spherical pores	57
5.1.2 Different pores with the same volume	60
5.1.3 Bulk systems	62
5.2 System evolution	62
5.2.1 Pore evolution	62
5.2.2 Nanobubbles	64
6 Measurements	67
6.1 Density	67
6.1.1 Water density as function of distance to the pore wall	68
6.2 Tetrahedral order parameter - TOP	71
6.2.1 TOP in the pores	72
6.3 Diffusion	76
6.3.1 Model for the diffusion constant	82
6.4 Cage correlation function	88
6.4.1 Defining the cage correlation function	88
6.4.2 Bulk water	92
6.4.3 Spherical pores	101
6.4.4 Same volume pores - three pore shapes	112
6.5 Life span	113
7 Discussion and conclusions	119
7.1 Revisiting the goals	119
7.2 Summary of the results and discussion	120
7.3 Future work	123
8 Appendix	125
8.1 Outline of the matlab lsqcurvefit	125
8.2 Differential geometry	127
8.2.1 Proof: Curvature of regular curves	127
8.2.2 Proof: Curvature of plane curves	129
8.3 Figures	129

1

Introduction

1.1 Porous materials and confined fluids

A porous material is a material, often a solid, which has regions where the solid is absent, also known as *pores*. The parts of the materials that are not pore space are called the matrix. There are many variants of porous materials in the world, such as rocks, sponges, cork or concrete to mention a few, with a wide spectrum of pore sizes from the centimeter length scale down to the nanometer length scale. The latter type of materials are called nanoporous materials. The applications of porous materials are many, including industrial applications and medical applications. Figure 1.1 shows two examples of porous materials, a porous ceramic material and a Swiss cheese-like polymer. The Swiss cheese-like polymer picture is from the University of Washington where scientists are researching the use of such polymers as porous structures where cardiac cells can grow and build heart tissue.

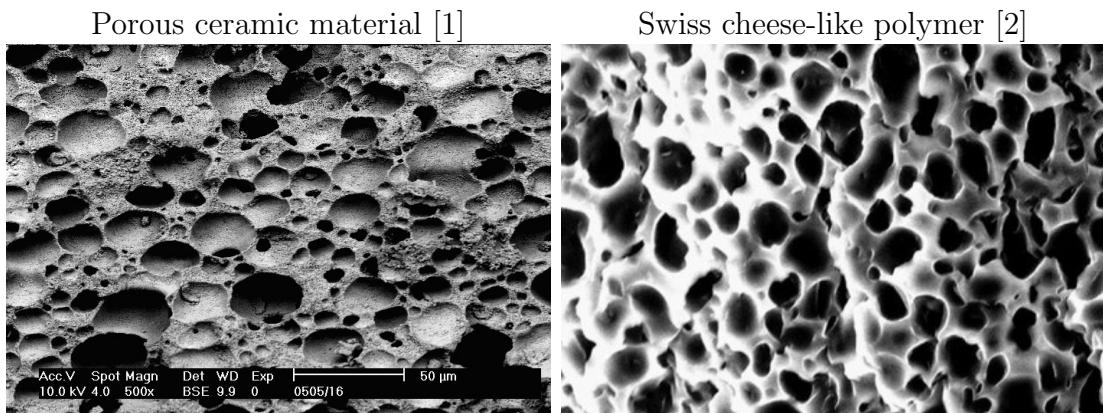


Figure 1.1: Examples of porous materials. These pictures are taken from web articles [1] and [2].

In this master thesis we are going to focus on pores in the nanometer length scale, and our focus will actually not be on the pores themselves, but on fluid that is confined in the pores. The behavior of the fluid is of course dependent on the material of the pore walls, and we will study the system of the matrix and the fluid in the pore space as a whole, but our main interest is to investigate the behavior of the fluid. Increased understanding of the physics of fluids in nanometer sized cavities may increase our understanding of

geological, physical and biomedical processes, and help us discover and improve industrial applications of nanoporous materials. Examples of applications of nanoporous materials used today are zeolite for removal of ammonia from drinking waters, and as a catalyst in the petroleum industry [3].

1.2 Choice of nanoporous system

1.2.1 Hydrophilic silica and water

The system we choose to investigate consists of water (H_2O) confined in nanoporous silica (SiO_2). Silica contributes to approximately 60% of the mass in the crust of the earth [4], where it present itselfs most commonly as quartz (see figure 1.2). Nanoporous silica has many industrial applications, for example filtration, catalysis, phase separation, medical delivery systems and CO_2 capture and storage, and the reason for this is that the pore sizes can be tuned and the pore surfaces functionalized. A form of nanoporous silica we utilize in our everyday life is silica gel in beaded form wrapped in small paper bags labelled with the characteristic warning “DO NOT EAT” (figure 1.3). We find these little bags inside newly bought purses etc., where they are placed to minimize the humidity in the product. The silica gel is used for this task because the silica is hydrophilic and has the ability to adsorb water molecules, and the fact that the silica gel is porous means it has large surface area where the water can be adsorbed.



Figure 1.2: Quartz, a crystalline version of silica, at the Senckenberg Natural History Museum. This picture is taken from [5].



Figure 1.3: Silica gel beads and packets. These pictures are taken from [6] and [7].

Experimental results report that water confined in nanoporous silica has different properties than bulk water [8][9][10], and since the behavior of water in silica is important for many geological, biological and physical processes we want to investigate this further. More particularly, we want to investigate how the curvature of the pore wall affects the water inside the pore. We suspect that a curved surface will affect the structure and dynamics of the water differently than a plane surface will. The curved surfaces set up a boundary conditions that would frustrate the packing or configuration of the water, forcing the water molecules to organize themselves differently than the water molecules would in bulk water. We expect the degree of frustration to be dependent on the curvature of the pore surface.

1.2.2 Pore shapes studied in this thesis

When we are going to investigate how the surface curvature affects the water, the easiest approach is to work with pores that have simple geometrical shapes where it is possible to find an analytical expression for the curvature at every point on the surface. This means that we *create* the pores the way we want them before we start investigating, instead of investigating a complex system where we measure the curvature of the pore walls, even though it is the complex system we expect to find in nature.

We choose to look at geometrical shapes that has constant curvature at all points on the surface, and these are the sphere, the open cylinder and the plane. If we had chosen a pore shape such as for instance an ellipsoid, we would have had an analytical expression for the curvature, but it would vary along the surface. To investigate the behavior of the water as a function of the curvature, we had to somehow find what curvature is affecting each molecule, meaning which part of the surface the molecule is affected by the most. The fact that the molecules move around inside the pore makes this task very challenging. Therefore, we let each system have just one curvature, and create multiple systems with different curvatures. This means that each system has a single pore with a simple geometric shape.

In nature we find complex pore structures, usually not simple geometrically shaped pores. There are several ways to model a porous material with a complex pore space. One popular way is to look at the matrix as randomly placed spheres where the pore space is the space between the spheres. Another way is to view the pore space as a network of spherical pores in a solid connected through passages shaped as cylinders or cracks with plane walls, and this is the model we use in this thesis. When we investigate our systems of single pores of spheres, cylinders and planes, we think of it as we investigate the component parts of this model.

Recap: we look exclusively at single pores shaped either as a sphere, cylinder or cracks with plane walls.

1.3 Choice of numerical method

Our investigation of nanoporous silica and confined water is through large-scale computer simulations at the atomic level. We simulate the atoms trajectories in phase space, based on three-body interatomic potentials, using a numerical technique called *Molecular Dynamics simulations* (MD simulations). The interatomic potential is developed

by Vashishta et al.[11] (currently at the University of Southern California, USC), and it covers interactions between the atoms in SiO_2 and H_2O in addition to internal interactions between the atoms in bulk SiO_2 and bulk H_2O , which allows for chemical reactions between the water and the silica on the pore surface.

To perform the calculations needed to simulate the physical trajectories of the atoms, we use a highly optimized molecular dynamics simulator (Fortran), developed by researchers at USC, which is run on UiO's own supercomputer ABEL [12].

1.4 Goals

We have established a set of goals for this master project, which will be revisited in chapter 7, Discussion and conclusions. The goals are:

1. Simulate systems of single pores of different geometric shapes such as spheres, cylinders and plane pore walls. The reason we want to do this is that it is possible to model a porous media as a network of spherical pores connected with cylindrical or “flat” pores, and in this thesis we would like to investigate the behavior of fluids inside the individual compartments building up such a network.
2. Improve already existing methods for preparing a silica/water MD-system. This includes
 - developing C++ code for making pores in silica
 - further developing the method for passivating the pore walls, and implementing the method in C++ code
 - develop a better method than the method used today for injecting water molecules into the system, and implementing the method in C++ code
3. Measure both diffusive and structural properties of water confined in pores and on bulk water. See if/how the measured results change with the curvature of the pores, and compare them with measured results for bulk water.

The structural properties:

- Density
- Tetrahedral order parameter

The diffusive properties:

- Diffusion constant (from the mean squared displacement function)
- Cage correlation function

1.5 Structure of the thesis

This document is divided into seven chapters.

Chapter 1, 2 and 3 introduces the overall background for this thesis, ch. 2 the characterization of the pore surfaces, ch. 3 the molecular dynamics simulation method.

Chapter 4 and 5 have detailed descriptions of the systems we are working with, where chapter 4 is about how the systems are generated and prepared and chapter 5 is about the specific details of each system.

Chapter 6 has descriptions of the measurements performed on the SiO₂ and H₂O systems, together with the results of the measurements.

The last chapter, chapter 7, is the discussion and conclusions part of the document. Here we discuss the connections between the various results found in chapter 6, and revisit the goals listed in chapter 1.

2

Characterization of pore surfaces

In this thesis we characterize the pore surface by its curvature, and in this chapter we look at how the curvature is defined from a differential geometry perspective. We start with the curvature of curves in space and in the plane, and end up at the curvature of surfaces. This part of the chapter is rather theoretical, so examples have been included. The last part of this chapter is about the curvatures of the surfaces of the pore shapes we have chosen to study.

2.1 Differential geometry

2.1.1 Curvature of curves

Curvature should say something about the extent of which a curve bends and curls at a given point. Naturally, by this vague definition, a straight line should have curvature $\kappa = 0$ for all points. For a circle, the larger a circle is, the slower it curves so the curvature should decrease as the radius of the circle increases. We have an intuitive understanding when it comes to deciding which curves should have high curvature and which should have low, but we need a precise mathematical definition that provide us with the ability to specify the curvature with a value, a number.

All definitions in this chapter is from *Carmo's* book called *Differential Geometry of Curves and Surfaces*[13].

Parametrized curves

Before we are ready to give a definition of the curvature, we first have to become familiar with some key concepts, such as parametrized differentiable curves, and regular curves which, by the way, is the only type of curves we define the curvature for.

All types of curves of interest to us are parametrized differentiable curves. A curve is differentiable if it is real and at all points have derivatives of all orders. The curve is defined in cartesian coordinates as

$$\alpha(t) = [x(t), y(t), z(t)],$$

where t is an arbitrary parametrization variable.

Example 2.1.1. We can parametrize the parabola as $\alpha(t) = [x(t), y(t), z(t)] = [t, t^2, 0]$. The curve α is differentiable because all functions $x(t), y(t), z(t)$ have derivatives of all orders.

Curvature of regular curves

According to *Carmo's Differential Geometry of Curves and Surfaces*[13] it is crucial for the study of the differential geometry of a curve that the tangent line exists at every point. The tangent line to $\alpha(t)$ is the straight line that goes through $\alpha(t)$ in the direction of $\vec{T} = \alpha'(t)$. If the derivative of $\alpha(t)$ is zero, we do not have a direction for our tangent line. Points where $\vec{T} = \alpha'(t) = 0$ are called singular points.

Definition 2.1.2. A parametrized differentiable curve $\alpha : I \rightarrow R^2$ is said to be regular if $\alpha'(t) \neq 0$ for all $t \in I$

We will only study regular curves. For simplifications sake we will restrict ourselves to curves parametrized by arc length. Arc length at t from some t_0 is defined by

$$s(t) = \int_{t_0}^t |\alpha'(t)| dt, \quad (2.1)$$

$$\Rightarrow \frac{ds}{dt} = |\alpha'(t)|. \quad (2.2)$$

For curves parametrized by arc length we have $\frac{ds}{dt} = 1$ which means the velocity vector $\vec{T} = \alpha'(t)$ is of unit length. We are now ready to define curvature.

Definition 2.1.3. Let $\alpha : I \rightarrow R^3$ be a curve parametrized by arc length $s \in I$. The number $\kappa(s) = |\alpha''(s)|$ is called the curvature of α at s .

Does this make sense? Let's check. Imagine that we have found the tangent vector for a regular curve α at s ; $\alpha'(s)$, and then moves an infinitesimal distance ds and find the tangent vector $\alpha'(s + ds)$. Since α is regular and parametrized by arc length we know that the tangent vectors are of unit length. In figure 2.1 there is a drawing of a regular curve with the tangent vectors at s and $s + ds$ drawn together. The drawing is just for illustration, and the angle between the tangent vectors is exaggerated. Since ds is an infinitesimal distance, we can assume that the angle θ between the tangent vectors is quite small and we use the approximation $\theta \approx \sin(\theta) \approx \frac{|\alpha''(s)|}{|\alpha'(s)|} = |\alpha''(s)|$. The angle θ is a measure of how fast the curve shies away from the tangent vector. If the curve was a straight line we would get $\theta = 0$ and hence also the curvature, which fits with our intuitive understanding of curvature.

We have defined curvature for curves parametrized by arc length. It would be more convenient to have a definition of curvature for arbitrary parametrization parameter. Lucky for us, all regular curves of arbitrary parametrization can be reparametrized by arc length (page 21-22 in Carmo's book [13]). This means we can use definition 2.1.3 to find the curvature of regular curves. In fact, the curvature of a regular parametrized curve $\alpha(t)$ is given by

$$\kappa(t) = \frac{|\alpha' \times \alpha''|}{|\alpha'|^3}, \quad (2.3)$$

where $\alpha' = \frac{d\alpha}{dt}$ and $\alpha'' = \frac{d^2\alpha}{dt^2}$. The proof of that can be found in appendix 8.2.1

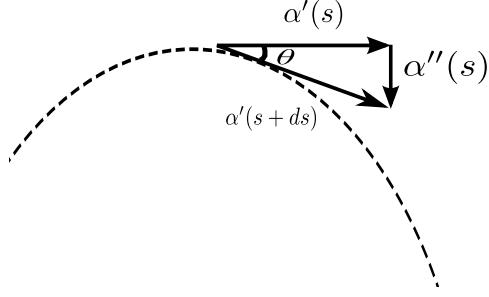


Figure 2.1: Drawing of a curve with its tangent vector and acceleration vector. The angle θ is exaggerated.

Curvature of plane curves

For the special case of plane regular curves there exists a formula for the curvature that uses the cartesian components of the curve. Let the curve be parametrized by t

$$\alpha(t) = [x(t), y(t)], \quad (2.4)$$

then the curvature is given by

$$\kappa(t) = \left| \frac{x'y'' - x''y'}{[(x')^2 + (y')^2]^{3/2}} \right|. \quad (2.5)$$

This is a convenient formula, and the proof can be found in appendix 8.2.2 where we insert eq. 2.4 into eq. 2.3.

We look at some examples, where the first example is the curvature of the circle which is relevant to our systems because it is needed in the process of calculating the curvature of the spherical and cylindrical surface.

Example 2.1.4. The curvature of a circle with radius R .

A circle with radius R is defined as all the points that fulfills the equation $x^2 + y^2 = R^2$. One possible parametrization of the circle is $\alpha(t) = [t, \sqrt{R^2 - t^2}]$. Let's find the derivatives

$$\begin{aligned} x &= t; \quad y = \sqrt{R^2 - t^2} \\ x' &= 1; \quad x'' = 0; \\ y' &= \frac{-t}{\sqrt{R^2 - t^2}} \\ y'' &= \frac{-R^2}{(R^2 - t^2)^{3/2}} \end{aligned} \quad (2.6)$$

Now we insert the derivatives into 2.5 to find the curvature

$$\kappa(t) = \left| \frac{\frac{-R^2}{(R^2 - t^2)^{3/2}}}{\left[1 + \frac{t^2}{(R^2 - t^2)}\right]^{3/2}} \right| = \left| \frac{-R^2}{(R^2 - t^2 + t^2)^{3/2}} \right| = \frac{1}{R}.$$

The curvature of a circle is independent of t and inverse proportional to the radius, which fits with what we discussed at the beginning of this chapter. When the radius increases, the curvature decreases. It is nice to know that the precise definition of curvature fits with our intuitive understanding.

Example 2.1.5. Curvature of the parabola.

One parametrization of the parabola is $\alpha(t) = (t, t^2)$.

$$\begin{aligned}x &= t; \quad y = t^2 \\x' &= 1; \quad y' = 2t \\x'' &= 0; \quad y'' = 2\end{aligned}$$

We insert this into 2.5 and find the curvature

$$\kappa(t) = \frac{2}{(1 + 3t^2)^{3/2}}.$$

We see that the curvature of the parabola has a maximum for $t = 0$, $\kappa(t = 0) = 2$, which is where the function $f(x) = x^2$ has a minimum.

The quantity inside the absolute value signs of eq. 2.5 is called the signed curvature. The fact that the curvature can have a sign is special for plane curves. Whether the signed curvature is negative or positive is dependent on the choice of parametrization. For instance, in example 2.1.4, the signed curvature of the circle was $k_-(t) = -\frac{1}{R}$, but we could have chosen the parametrization $\alpha(t) = [-t, \sqrt{R^2 - t^2}]$, and the signed curvature would be $k_+(t) = \frac{1}{R}$. The signed curvature of a plane curve is

$$k(t) = \frac{x'y'' - x''y'}{[(x')^2 + (y')^2]^{3/2}}. \quad (2.7)$$

Example 2.1.6. Signed curvature of the ellipse.

One parametrization of the ellipse is $\alpha(t) = (a \sin(t), b \cos(t))$. We find the derivatives

$$\begin{aligned}x &= a \sin(t); \quad y = b \cos(t) \\x' &= a \cos(t); \quad y' = -b \sin(t) \\x'' &= -a \sin(t); \quad y'' = -b \cos(t)\end{aligned}$$

Insert this into 2.7 to find the signed curvature

$$k(t) = \frac{-ab \cos(t)^2 - ab \sin(t)^2}{[a^2 \cos(t)^2 + b^2 \sin(t)^2]^{3/2}} = \frac{-ab}{[a^2 \cos(t)^2 + b^2 \sin(t)^2]^{3/2}}$$

Let's find the curvature at the vertices $(a, 0), (0, b)$. For vertex $(a, 0)$ we have $x = a \sin(t) = a$ and $y = b \cos(t) = 0$. This means that $\sin(t) = 1$ and $\cos(t) = 0$. The curvature at vertex a is

$$k(t_a) = \frac{-ab}{b^3} = \frac{-a}{b^2}.$$

With the same logic we find the curvature at the other vertex

$$k(t_b) = \frac{-b}{a^2}.$$

We notice that for $a = b \equiv R$, which is the special case of an ellipse that is a circle, we get the signed curvature $k(t_a) = k(t_b) = \frac{-1}{R}$, which is consistent with the results in example 2.1.4.

2.1.2 Gaussian curvature

Let us see how we can extend the definition of curvature to describe a surface. Imagine standing at the bottom of a u-shaped valley. If you walk directly towards one of the “walls” of the valley, your experience would be that your path is very curved, and the curvature of the path, which actually is a curve in space (plane curve if you walk straight up the hill), would have a high value. If you, on the other hand, walk along the bottom of the valley, the curvature of your path would be zero. So, the curvature you experience is determined by the direction of your path. This gives us infinite many curvatures for that point on the surface, but we want to characterize the curvature of that point with only one value. One option is to use the Gaussian curvature.

The definition of Gaussian curvature at a point, \vec{r} , on a surface is the product of the principal curvatures of the point,

$$\kappa(\vec{r}) = k_1(\vec{r})k_2(\vec{r}) \quad (2.8)$$

where k_1 and k_2 are the principal curvatures. So, what are the principal curvatures, and how can we find them?

Let $\vec{n}(\vec{r})$ be the normal vector of the surface at the point \vec{r} , and imagine a plane parallel to \vec{n} ‘cutting’ the surface through the point \vec{r} . The intersection between the plane and the surface is a plane curve. We can calculate the signed curvature of this plane curve by the formula in eq. 2.7, and we call it the normal curvature k_n . We do, of course, have infinite number of such planes, because we can rotate the plane and still have it be parallel with the normal vector, and thus we have infinite number of normal curvatures at \vec{r} . The principal curvatures are the maximum and minimum value of the set of normal vectors at \vec{r} ,

$$\begin{aligned} k_1 &= \max\{k_n\}, \\ k_2 &= \min\{k_n\}. \end{aligned}$$

The sign of the Gaussian curvature tells us a lot about the surface: [14]

- $\kappa > 0$ means that the principal curvatures have the same sign at this point on the surface. These points are called elliptic points, and the surface will look like an arch or dome at such points.
- $\kappa < 0$ means that the principal curvatures have different signs. These points are called hyperbolic points, and the surface will have a typical saddle shape.
- $\kappa = 0$ means that at least one of the principal curvatures is zero. The point is either parabolic or planar. This means that in at least one direction from the point \vec{r} the surface does not bend or curve.

In the next section we calculate the Gaussian curvature of the surfaces of the pore shapes we have chosen for this thesis.

2.2 Pore shapes studied in this thesis

In this thesis we have chosen to investigate spherical, cylindrical and plane pores. The Gaussian curvature of a plane pore surface is very easy to calculate, as all curves intersecting with a plane normal to the surface will be straight lines with zero curvature.

Both principal curvatures are zero which makes the Gaussian curvature zero. The next two examples finds the curvature of the two remaining pore shapes, the sphere and the cylinder.

Example 2.2.1. Gaussian curvature of spherical surfaces

Let's find the Gaussian curvature at a point \vec{r} at the surface of a sphere with radius R . The normal vector \vec{n} points away from the center of the sphere, and no matter how we rotate the plane parallel to \vec{n} , the intersection between the plane and the surface will be a circle with radius R . We have already found an expression for the curvature of a circle of radius R in example 2.1.4, and it was $k_n = \frac{1}{R}$. Since the set of normal curvatures just contain one element, it is unproblematic to find the maximum and minimum of the set. The principal curvatures are

$$k_1 = k_2 = \frac{1}{R},$$

and the Gaussian curvature at any point on the surface of the sphere is

$$\kappa = \frac{1}{R^2}.$$

Example 2.2.2. Gaussian curvature of cylindrical surfaces

We look at a point on the surface of the cylinder with radius R . The normal vector \vec{n} will point away from the line in the center of the cylinder parallel to the length of the cylinder. We start with the plane normal to the direction of the cylinder, so that the intersection between the plane and the surface is the circle of radius R . The normal curvature for this plane is as usual $k_n = \frac{1}{R}$. We start to rotate the plane, and observe that the intersection evolves into an ellipse that flattens out as we rotate (the normal curvature decreases). Eventually we end up with the plane parallel to the direction of the cylinder, and the intersection is just a straight line and the normal curvature is $k_n = 0$. If we continue to rotate the plane the intersection is again the ellipse and the the normal curvature increases as we rotate from this point. This all means that the largest normal curvature is

$$k_1 = \max\{k_n\} = \frac{1}{R}$$

and the smallest is

$$k_2 = \min\{k_n\} = 0,$$

and the Gaussian curvature

$$\kappa = k_1 k_2 = 0.$$

Since the cylinder is a symmetric creature, the curvature is the same at every point on the surface, if we exclude the end points.

Apparently, the cylinder has the same Gaussian curvature as a plane surface. Surfaces with zero Gaussian curvature are called developable surfaces and they have the property that they can be folded from a plane sheet without being stretched or compressed[15]. The cone is another example of developable surfaces, in addition to the cylinder. The sphere is not developable, which means that the surface of a sphere can not be flattened out on a plane sheet. This is why the planar map of the world looks a little funny compared to a globe, because we have tried flattening out a surface with non-zero Gaussian curvature, which leads to some parts being stretched and others compressed.

3

Introduction to molecular dynamics

The field of Molecular Dynamics (MD) simulations is about numerically solving many-body problems for particles that follow the laws of classical mechanics. MD simulations are used to model and study complex systems of interest at the atomic level. The technique is applied in multiple fields, such as medicine, biology and material science, and the reason for this is that performing a molecular dynamics simulation is very similar to performing the physical experiment. Ideally, we would have liked to see how the system or sample would evolve in the physical world, at the atomic level, but that is rarely possible. Instead, we do the next best thing, we model the system and perform the experiment on the computer. The various research fields might have very different problems they want solved using molecular dynamics simulations. An engineer could wish to investigate how much pressure it is possible to put on a material before the material is damaged, or she could want to see how a material evolves if it already has a flaw, for instance a crack. A biologist might want to investigate mutations in proteins or other biological macromolecules [16]. The common denominator is that they all want to model a system of multiple particles/atoms where the particles interact and affect each others trajectories.

We assume that the particles behave classically, so the position, velocity and acceleration a particle in the system, say particle k , is determined by Newton's equation of motion (Newton's second law),

$$m_k \ddot{\vec{r}}_k(t) = \vec{F}_k(\vec{r}_k(t)), \quad (3.1)$$

where m_k is the mass of the particle, $\ddot{\vec{r}}_k$ is the acceleration, which is the double derivative of the position of the particle and $\vec{F}_k(\vec{r}_k(t))$ is the sum of the forces acting on particle k . The magnitude and direction of the force acting on the particle is determined by what rule we have set for the particles to follow when they interact, and by the configuration of the rest of the particles in the system with respect to particle k . The rule that determines the interactions between particles has to be decided by us before we start the simulation, and is tightly connected to the type of system we want to simulate. For instance, if we wanted to simulate a gravitational system, the rule could be that each particle pair attract each other with the force $F = G \frac{m_1 m_2}{r^2}$.

Often, instead of having a rule about the forces between particles directly, we let the rule be a potential function $U(\vec{r}_k, \vec{r}_{config})$. When we have a potential function we can find

the force on particle k by calculating the gradient at \vec{r}_k

$$\vec{F}(\vec{r}_k, \vec{r}_{config}) = -\nabla U(\vec{r}_k, \vec{r}_{config}), \quad (3.2)$$

where \vec{r}_k is the position of the particle of interest and \vec{r}_{config} is the configuration of the rest of the system.

In short, to perform MD simulations basically means calculating the positions and velocities of all the particles in the system at the next time step, based on the positions and velocities the particles have at the current time step. To be able to do this, we first have to calculate the force acting on each particle in the system. This is done by differentiating the potential field set up by the particles. When we have the forces acting on all the particles, we find the state at the next time step by solving Newton's second law (eq. 3.1) numerically for all the particles, with the help of an integrator algorithm. Before we can do all that, we need to select a potential function that fit to our problem and our particles.

3.1 Potentials

There are many types of potential functions that describe interactions in different types of systems. One example of a potential function is the square-well potential used by B. J. Alder and T. E. Wainwright when they first introduced the technique of Molecular Dynamics simulations in 1959 [17]. In Alder and Wainwright's experiments, the square-well potential was used to describe systems of hard discs and spheres interacting through perfect collision. It is a two-body potential, where each pair of discs in the system contributes with

$$V(r) = \begin{cases} \infty & \text{if } r < \sigma_1, \\ V_0 & \text{if } \sigma_1 < r < \sigma_2 \\ 0 & \text{if } r > \sigma_2 \end{cases} \quad (3.3)$$

where r is the distance between the centers of two discs. V_0 , σ_1 and σ_2 are constants.

The total potential disc k feels is the sum

$$U(\vec{r}_k, \vec{r}_{config}) = \sum_{i \neq k} V(|\vec{r}_i - \vec{r}_k|) \quad \text{where } \vec{r}_i \in \vec{r}_{config}. \quad (3.4)$$

Equation 3.4 applies to all two-body potentials.

Another example is the very common potential used by A. Rahman in 1964 when he studied the motion of atoms in liquid argon[18]. He used the Lennard-Jones potential which describes van der Waals interactions between atoms. The Lennard-Jones potential is given as

$$V_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (3.5)$$

where r is the distance between the two particles in consideration. The r^{-12} term approximates the steric repulsion due to the Pauli principle which ensures that the particles do not come to close to each other. This term dominates for small distances, and gives the

steep part of the potential, which can be seen in figure 3.1. The r^{-6} term is the one that gives the attractive long-range van der Waals forces between the particles.

The Lennard-Jones potential is used to describe gases, and is especially precise when it comes to modeling the noble gases because they are monatomic gases where most interactions come from the van der Waal forces.

When calculating the force on an arbitrary particle (say k) in a Lennard-Jones system of N particles, we need to find the $N - 1$ distances from particle k to all the other particles, which makes the force calculations the most time consuming part of the simulations.

In figure 3.1, the shape of the Lennard-Jones potential is visualized. We see here that for particle-pairs with distances larger than $r_{flat} = 2.5-3\sigma$ the potential is pretty much horizontal, which means that the force contribution is approximately zero. To reduce computation time it is common to only calculate the forces between particles that are closer together than a cut-off radius, r_c , which could for instance be chosen as 3σ .

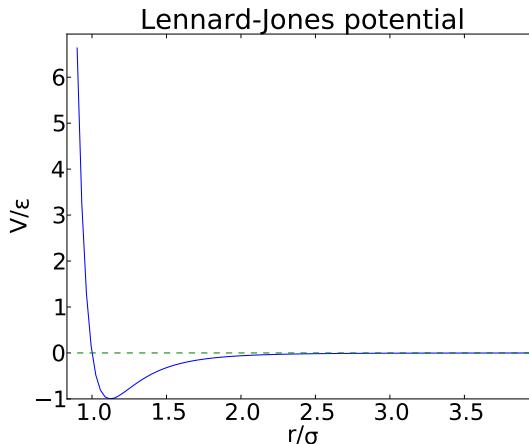


Figure 3.1: The shape of the common Lennard-Jones potential. The left-most part of the potential is where the term r^{-12} is most dominant, and this gives us steric repulsion between particles. The $-r^{-6}$ term gives us the attractive long-range forces, that hold the system together.

In the later years more potentials have been developed in order to describe chemical reactions between different types of atoms. These potentials are much more complex than the Lennard-Jones two body potential. Many of these potentials have included multiple-body terms, which means that the potential field in the entire system is not only determined by the distances between pairs of atoms (Lennard-Jones), but by the configuration of multiple atoms. For instance a three body term can include the angles between threes of particles in the system. One example is the ReaxFF force field which takes into account continuous bond breaking and formation [19], which is a requirement in describing chemical reactions.

The silica-water potential

In this project we will use an interatomic potential that covers chemical reactions between H_2O and SiO_2 , developed by Vashishta et al. The details described in this section are also described here [20], which is a supplementary document to an article about collapsing nanobubbles in water on a silica surface, where they use the same potential. The

potential consists of a two-body term and a three-body term. The two-body term covers steric repulsion, Coulomb interaction, charge-dipole interaction and van der Waals interaction, while the three-body term covers covalent bond bending and stretching. The total potential energy of the system can be written as sums over these two potential types

$$E_{tot} = \sum_{i < j} V_{ij}^{[2]}(r_{ij}) + \sum_{i < j < k} V_{ijk}^{[3]}(\vec{r}_{ij}, \vec{r}_{ik}) \quad (3.6)$$

where $V_{ij}^{[2]}(r_{ij})$ is the two-body term set up by particle i and j with distance $r_{ij} = |\vec{r}_{ij}|$ where \vec{r}_{ij} is the vector pointing from particle i to j . The two body sum in eq. 3.6 sums over all particles within the sphere with radius equal to the cut-off range r_c . $V_{ijk}^{[3]}(\vec{r}_{ij}, \vec{r}_{jk})$ is the three-body term which input is three particles i , j and k , or more precisely the vector pointing from particle i to j , and the vector pointing from particle i to k .

Definition 3.1.1. The two-body term

$$V_{ij}^{[2]}(r) = \frac{H_{ij}}{r^{n_{ij}}} + \frac{Z_i Z_j}{r} \exp(-r/r_{1s}) - \frac{D_{ij}}{2r^4} \exp(-r/r_{4s}) - \frac{\omega_{ij}}{r^6} \quad (3.7)$$

Steric repulsion

- H_{ij} is the steric repulsion strength
- n_{ij} is the exponent of the steric repulsion (In the Lennard-Jones potential it has the value 12)

Coulomb potential with screening

- Z_i is the effective charge of particle i in units of electronic charge
- r_{1s} is the screening length for the Coulomb potential.

Charge-Dipole

- D_{ij} is the charge-dipole strength
- r_{4s} is the screening length for the charge-dipole interactions

van der Waals attractions

- ω_{ij} is the strength of the van der Waals attractions

Definition 3.1.2. The three-body term

$$V_{ijk}^{[3]}(\vec{r}_{ij}, \vec{r}_{ik}) = B_{ijk} \exp\left(\frac{\xi}{r_{ij} - r_0} + \frac{\xi}{r_{ik} - r_0}\right) \frac{(\cos \theta_{ijk} - \cos \theta_0)^2}{1 + C_{ijk}(\cos \theta_{ijk} - \cos \theta_0)^2} \quad (r_{ij}, r_{ik} \leq r_0) \quad (3.8)$$

r_{ij} is the length of the vector from particle i to j , $r_{ij} = |\vec{r}_{ij}|$.

Angle: θ_{ijk} is the angle between vector \vec{r}_{ij} and \vec{r}_{ik}

Strength: B_{ijk} is the strength of the three-body interaction.

Constants: ξ and C_{ijk} are constants.

The potential parameters for the H₂O and SiO₂ system was chosen by first determining the parameters for pure SiO₂ and pure H₂O, and then combining these two sets to find the set of parameters used to describe the reactions at the water/silica border. The details concerning the combining process of the two sets of parameters is described in [21]

The parameters for pure silica were chosen to fit experimental values of bulk modulus, equilibrium volume and the melting temperature of silica. For pure water the parameters were fit to match the force constants of symmetric bond stretching and bending of a single water molecule.

3.2 Integrator - Verlet algorithm

Algorithms used for solving eq. 3.1 are called integrators. The reason is that solving this equation is the same as integrating this expression

$$\ddot{\vec{r}}_k(t) = \frac{-\nabla U(\vec{r}_k, \vec{r}_{config})}{m_k} \quad (3.9)$$

twice to retrieve $\vec{r}_k(t)$. Here we have substituted the force term of eq. 3.1 with the gradient of the potential, because it is more common to have an expression for the potential than to have one for the force directly.

A popular integrator in the field of Molecular Dynamics is the Verlet algorithm. The Verlet algorithm has been discovered a few times in history, including by Verlet who used it for computer simulations of Molecular Dynamics in the 1960's[22]. The Verlet algorithm is so popular because it is simple, and at the same time gives little long-term energy drift and is time reversible [23] which is a trait of Newton's equation of motion we want to conserve in our calculations.

Derivation of the Verlet algorithm

The algorithm can be found by the sum of two Taylor expansions. For convenience, we use the short hand notation $\vec{r}_{n+k} = \vec{r}(t + k\Delta t)$, $\vec{v}_n = \dot{\vec{r}}_n = \frac{\partial \vec{r}}{\partial t}|_n$ and $\vec{a}_n = \ddot{\vec{r}}_n$. The two Taylor expansions are

$$\begin{aligned} \vec{r}_{n+1} &= \vec{r}_n + \vec{v}_n \Delta t + \frac{1}{2} \vec{a}_n \Delta t^2 + \frac{1}{6} \dot{\vec{a}}_n \Delta t^3 + \mathcal{O}(\Delta t^4), \\ \vec{r}_{n-1} &= \vec{r}_n - \vec{v}_n \Delta t + \frac{1}{2} \vec{a}_n \Delta t^2 - \frac{1}{6} \dot{\vec{a}}_n \Delta t^3 + \mathcal{O}(\Delta t^4). \end{aligned}$$

The sum of these Taylor expansions give us

$$\begin{aligned} \vec{r}_{n+1} + \vec{r}_{n-1} &= 2\vec{r}_n + \vec{a}_n \Delta t^2 + \mathcal{O}(\Delta t^4), \\ \vec{r}_{n+1} &= 2\vec{r}_n - \vec{r}_{n-1} + \vec{a}_n \Delta t^2 + \mathcal{O}(\Delta t^4), \\ \vec{r}_{n+1} &\approx 2\vec{r}_n - \vec{r}_{n-1} + \vec{a}_n \Delta t^2, \end{aligned} \quad (3.10)$$

which is the Verlet algorithm. We have found that it allows us to calculate the position of the particles for the next time step, with an error of fourth order with respect to the time step Δt .

Often, one would like to measure physical quantities indirectly through the velocities of the system as it evolves, for instance kinetic energy and temperature, and the Verlet algorithm just gives us the position. One solution is to find the velocity by numerical differentiation of the positions

$$\vec{v}_n = \frac{\vec{r}_n - \vec{r}_{n-1}}{\Delta t} + \mathcal{O}(\Delta t^2). \quad (3.11)$$

A very common variant of the Verlet algorithm is the velocity Verlet algorithm. This algorithm gives us the same positions as Verlet (down to machine-precision), and calculates the velocities as we go. Another likeable trait is the fact that we only need data from the current time step, not from the previous time step, to calculate the next. Unlike the Verlet algorithm, we do not have any terms with \vec{r}_{n-1} .

Derivation of the Velocity Verlet algorithm

Start with the Taylor expansion of the position and the velocity

$$\vec{r}_{n+1} = \vec{r}_n + \Delta t \vec{v}_n + \frac{\Delta t^2}{2} \vec{a}_n + \mathcal{O}(\Delta t^3), \quad (3.12)$$

$$\vec{v}_{n+1} = \vec{v}_n + \Delta t \vec{a}_n + \frac{\Delta t^2}{2} \dot{\vec{a}}_n + \mathcal{O}(\Delta t^3). \quad (3.13)$$

We find an expression for the term with the differentiated acceleration ($\frac{\Delta t^2}{2} \dot{\vec{a}}_n$) in eq. 3.13 by using the Taylor expansion of the acceleration

$$\begin{aligned} \vec{a}_{n+1} &= \vec{a}_n + \Delta t \dot{\vec{a}}_n + \mathcal{O}(\Delta t^2), \\ \Rightarrow \frac{\Delta t^2}{2} \dot{\vec{a}}_n &= (\vec{a}_{n+1} - \vec{a}_n) \frac{\Delta t}{2} + \mathcal{O}(\Delta t^3). \end{aligned}$$

Insert this into 3.13 and get

$$\vec{v}_{n+1} = \vec{v}_n + \Delta t \vec{a}_n + (\vec{a}_{n+1} - \vec{a}_n) \frac{\Delta t}{2} + \mathcal{O}(\Delta t^3) \quad (3.14)$$

$$= \vec{v}_n + (\vec{a}_{n+1} + \vec{a}_n) \frac{\Delta t}{2} + \mathcal{O}(\Delta t^3). \quad (3.15)$$

We truncate the expression and get the velocity Verlet algorithm:

$$\vec{r}_{n+1} \approx \vec{r}_n + \vec{v}_n \Delta t + \frac{1}{2} \vec{a}_n \Delta t^2, \quad (3.16)$$

$$\vec{v}_{n+1} \approx \vec{v}_n + \frac{\Delta t}{2} (\vec{a}_n + \vec{a}_{n+1}). \quad (3.17)$$

We want to show that this scheme gives us the same positions as the Verlet algorithm. We start by solving eq. 3.16 with respect to \vec{v}_n :

$$\vec{v}_n = \frac{1}{\Delta t} (\vec{r}_{n+1} - \vec{r}_n) - \frac{1}{2} \vec{a}_n \Delta t,$$

and insert this for the two velocity terms in eq. 3.17 (we just increase all indices by one to get an expression for \vec{v}_{n+1}). Eq. 3.17 becomes

$$\begin{aligned} \frac{1}{\Delta t}(\vec{r}_{n+2} - \vec{r}_{n+1}) - \frac{1}{2}\vec{a}_{n+1}\Delta t &= \frac{1}{\Delta t}(\vec{r}_{n+1} - \vec{r}_n) - \frac{1}{2}\vec{a}_n\Delta t + \frac{\Delta t}{2}(a_n + a_{n+1}), \\ \Rightarrow \vec{r}_{n+2} &= 2\vec{r}_{n+1} - \vec{r}_n + \Delta t^2\vec{a}_{n+1}, \\ \Rightarrow \vec{r}_{n+1} &= 2\vec{r}_n - \vec{r}_{n-1} + \vec{a}_n\Delta t^2, \end{aligned}$$

which we recognize as the basic Verlet algorithm (eq. 3.10).

This means that the position error of the velocity Verlet algorithm is the same as for the basic Verlet algorithm: $\mathcal{O}(\Delta t^4)$, even though it appears to be $\mathcal{O}(\Delta t^3)$ in eq. 3.12. During computation we divide the velocity Verlet algorithm into four steps:

$$\vec{v}_{n+1/2} = \vec{v}_n + \frac{1}{2}\vec{a}_n\Delta t,$$

$$\vec{r}_{n+1} = \vec{r}_n + \vec{v}_{n+1/2}\Delta t,$$

Calculate \vec{a}_{n+1} according to the new position \vec{r}_{n+1} :

$$\vec{a}_{n+1} = \vec{F}_{n+1}/m,$$

$$\vec{v}_{n+1} = \vec{v}_{n+1/2} + \frac{1}{2}\vec{a}_{n+1}\Delta t.$$

Summary of the errors

- The local error for the position with the basic Verlet algorithm is $\mathcal{O}(\Delta t^4)$.
- The local error for the position with the velocity Verlet algorithm is $\mathcal{O}(\Delta t^4)$.
- The local error for the velocity with the basic Verlet algorithm is $\mathcal{O}(\Delta t^2)$.
- The local error for the velocity with the velocity Verlet algorithm is $\mathcal{O}(\Delta t^3)$.

3.3 Periodic boundary conditions

The particles in a MD system is contained inside a box, and we need to decide what will happen when a particle moves outside the boundaries of our system. Will it disappear, bounce back or appear somewhere else in the system? We choose the latter. More specifically, we choose our system to have periodic boundary conditions. This means that if a particle moves out of the boundary on one side, it reappears inside the box on the opposite side of the system from where it disappeared, with unchanged velocity. Another way to look at it is to imagine an infinite system with identical replicas of our system lying next to each other. Figure 3.2 illustrates this in two dimensions for a system of two particles. The purple particle has a velocity towards the left border of the system, and we see that in one time step from now it will enter the neighboring box. So, when a particle moves out of the box it enters another, and so does the replica-particle in the neighbor box on the opposite side and the net number of particles in the box is constant.

When measuring distances in periodic systems, we use the *minimum image convention*. Say we want to measure the distance between the purple and the gray particle. From the illustration in figure 3.2 we understand that there are infinite number of distances to choose from, and the dashed lines pointing from one of the purple particles show us seven

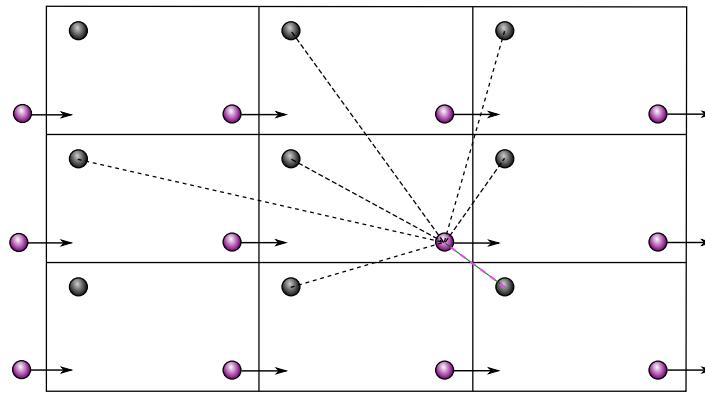


Figure 3.2: Illustration of periodic boundary conditions and minimum image convention. Each box is an identical replica of the system with two particles. The purple particle has a velocity toward the boundary.

of these options. The *minimum image convention* tells us to choose the smallest one, which is marked with a purple dashed line. To find the smallest distance we only need to check the distance to the gray particle in the 27 closest neighbor boxes (nine in the 2D case), since the particles in the other boxes are necessarily further away. The easiest way to do this is to find the vector between the particles, \vec{r} , and make sure each component of \vec{r} is as small as possible by adding or subtracting (or doing nothing) the whole system size L . The vector \vec{r} is the vector pointing from the gray particle to the purple particle

$$\vec{r} = \vec{p}_{\text{purple}} - \vec{p}_{\text{gray}}.$$

We find the components of the minimum image vector \vec{r}_{MI} by

$$\vec{r}_{MI}[i] = \min \begin{cases} |\vec{r}[i] + L| \\ |\vec{r}[i]| \\ |\vec{r}[i] - L| \end{cases}. \quad (3.18)$$

The distance between the gray and the purple particle is, according to the minimum image convention, the length of \vec{r}_{MI} .

3.4 Thermostats

In a MD system, the temperature depends on the kinetic energy which is directly connected to the velocities of the particles. We want to be able to control the temperature in the system, as we would be in the laboratory. Maybe we want to heat up a crystal to get a liquid, or cool down a liquid into a glass. This is done by the help of thermostats, that either insert or remove energy from the system. This subsection describes three types of thermostats. Two of the thermostats, Andersen and Berendsen, regulate the energy of the system in a intuitive and simple way. The last thermostat, called the Nosé-Hoover thermostat is a bit more complicated, but it samples the canonical ensemble much better than the Andersen and Berendsen thermostat does.

A Molecular Dynamics system on the computer will evolve in the microcanonical ensemble. The system is isolated with a constant number of particles and volume of our choosing, and the energy is conserved. In the laboratory, it is difficult to keep the energy in a sample constant, but we can maintain a constant temperature. This corresponds to the canonical ensemble, where the temperature is held constant by connecting the system to a heat bath, and allowing fluctuations of energy in the system. If we want to perform Molecular Dynamics simulations in the canonical ensemble, and thereby have similar conditions as in the laboratory, the Nosé-Hoover thermostat is the best choice.

All the thermostats connects the system to an imaginary heat bath, and thereby mimic the canonical ensemble. For this thesis we use the thermostats only in the preparation phase of the simulations, which means that we work in the microcanonical ensemble during the measuring phase. Still, we choose to present three thermostat, including the Nosé-Hoover for completeness.

3.4.1 Andersen Thermostat

The Andersen thermostat mimics the canonical ensemble, by letting a fraction of the particles in the system interact with an imaginary heat bath that has the desired temperature, T_{hb} . The interaction is a hard collision between a particle and the heat bath that results in a new velocity for the particle, independent of the pre-collision velocity. Each component of the new velocity, $v_{\text{new}} = [v_x, v_y, v_z]$, is drawn from the distribution

$$P(v_i) = \sqrt{\frac{m}{2\pi k T_{\text{hb}}}} e^{-mv_i^2/(2kT_{\text{hb}})},$$

where m is the mass of the particle and k is the Boltzmann constant. This leads to the Maxwell-Boltzmann distribution for the new particle speeds

$$P(v) = \sqrt{\left(\frac{m}{2\pi k T_{\text{hb}}}\right)^3} 4\pi v^2 e^{-mv^2/(2kT_{\text{hb}})}, \quad (3.19)$$

where v is the magnitude of the velocity, or the speed ($v = \sqrt{v_x^2 + v_y^2 + v_z^2}$). This is the same distribution of speeds we have in a canonical ensemble.

The fraction of the particles that get new velocities is often chosen as $\frac{\Delta t}{\tau}$, where Δt is the time step of the simulation and τ is the average time between collisions for one particle. τ is often chosen as $\sim 10\Delta t - 20\Delta t$ which means that each particle on average collide once every 10-20 time step. The Andersen thermostat sample the canonical ensemble quite well. It can be used when measuring time independent properties, but should not be used when measuring properties directly connected to the paths of the particles because they are changed by the thermostat. For example, the diffusion constant will give different results with different thermostats.

3.4.2 Berendsen Thermostat

This thermostat scales the velocities of the system, by multiplying all velocities in the system by a scaling factor λ . The value of λ is chosen so that the change in temperature

of time is proportional to the temperature difference between the system and the heat bath. Written mathematically

$$\frac{dT(t)}{dt} = \frac{1}{\tau}(T_{\text{hb}} - T(t)), \quad (3.20)$$

where T_{hb} is the temperature of the heat bath and also our desired temperature, $T(t)$ is the temperature of the system at time t . The proportionality constant $1/\tau$ gives the coupling between the heat bath and the system. The larger τ , the weaker the coupling and the longer it takes to reach the wanted temperature. For our simulation which is discrete in time, we write eq. 3.20 as

$$\begin{aligned} \frac{T_{j+1} - T_j}{\Delta t} &= \frac{1}{\tau}(T_{\text{hb}} - T_j), \\ \Rightarrow T_{j+1} - T_j &= \frac{\Delta t}{\tau}(T_{\text{hb}} - T_j), \end{aligned} \quad (3.21)$$

where T_j is the temperature at time step j , and T_{j+1} is the temperature at the next time step which is changed due to the velocity changes in the system.

To find the scaling factor λ , we use the equipartition theorem that gives us the change in kinetic energy as we change the temperature

$$\begin{aligned} \Delta E_{\text{kin}} &= \frac{3NK\Delta T}{2}, \\ E_{j+1, \text{kin}} - E_{j, \text{kin}} &= \frac{3NK(T_{j+1} - T_j)}{2}, \\ \Rightarrow T_{j+1} - T_j &= \frac{2}{3NK}(E_{j+1, \text{kin}} - E_{j, \text{kin}}), \\ T_{j+1} - T_j &= \frac{2}{3Nk} \sum_i^N \frac{m_i(\lambda v_i)^2}{2} - \frac{2}{3Nk} \sum_i^N \frac{m_i v_i^2}{2}, \\ T_{j+1} - T_j &= \frac{2}{3Nk} \sum_i^N \frac{m_i v_i^2}{2} (\lambda^2 - 1), \\ T_{j+1} - T_j &= T_j(\lambda^2 - 1). \end{aligned} \quad (3.22)$$

We compare the two expressions (eq. 3.21 and eq. 3.22) for ΔT and end up with a formula for the scaling factor λ ,

$$\begin{aligned} \frac{\Delta t}{\tau}(T_{\text{hb}} - T_j) &= T_j(\lambda^2 - 1), \\ \lambda^2 &= \frac{\Delta t}{\tau} \left(\frac{T_{\text{hb}}}{T_j} - 1 \right) + 1, \\ \lambda &= \sqrt{\frac{\Delta t}{\tau} \left(\frac{T_{\text{hb}}}{T_j} - 1 \right) + 1}. \end{aligned}$$

For every time step with the thermostat on, we calculate the value of λ , and multiply the velocities in the system with it. As the temperature of the system approaches T_{hb} , the scaling factor approaches 1.

The Berendsen thermostat helps the system reach the wanted temperature rather quickly, but it does not sample the canonical ensemble. This thermostat is a good tool when initializing systems, but should be turned off when measuring. The Andersen thermostat samples the canonical ensemble better because we simulate collisions with the heat bath, and just a fraction of the particles get a new velocity. When using the Berendsen thermostat, we change the velocity of all the particles in the system, which is not realistic.

3.4.3 Nosé-Hoover Thermostat

The Nosé-Hoover thermostat gives results that best samples the canonical ensemble. The idea behind the method is to view the heat bath as a part of the system, by adding a degree of freedom to the Lagrange function. We say we have two systems, one is our system and the other is our system plus the heat bath, called the extended system. The added degree of freedom corresponds to an artificial variable s , and a mass Q which determines the coupling between the heat bath and the system. The artificial variable has a velocity \dot{s} and an acceleration \ddot{s} . You could look at s as a time scaling factor between the system and the extended system (system + heat bath). The time in the extended system is

$$d\tilde{t} = sdt,$$

where dt is the time in the not-extended system. The coordinates are the same in the two systems (extended and not-extended); $\vec{r} = \vec{r}$ and $\tilde{s} = s$. We find the derivatives by using the chain rule,

$$\ddot{\tilde{s}} = \frac{\ddot{s}}{s^2} - \frac{\dot{s}^2}{s^3}, \quad (3.23)$$

$$\dot{\tilde{s}} = \frac{\dot{s}}{s} (= \gamma). \quad (3.24)$$

The parenthesis show a substitution we shall use later,

$$\ddot{\tilde{r}} = \frac{\ddot{\vec{r}}}{s^2} - \frac{\vec{r}\ddot{s}}{s^3}, \quad (3.25)$$

$$\dot{\tilde{r}} = \frac{\dot{\vec{r}}}{s}. \quad (3.26)$$

We choose the Lagrange function as

$$L = \sum_i \frac{m_i}{2} \tilde{s}^2 (\dot{\tilde{r}}_i)^2 - U(\vec{r}) + \frac{1}{2} Q \dot{s}^2 - 3NkT_{hb} \log \tilde{s}.$$

The two first terms are kinetic energy minus potential energy in our system (the not-extended one). The two last terms are the kinetic energy of \tilde{s} and a potential chosen so that the result samples the canonical ensemble. We solve the Lagrange equation

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = 0,$$

where $q_j \in [\vec{r}_j, s]$ and get

$$\begin{aligned}\ddot{\vec{r}}_i &= \frac{\vec{F}_i}{m_i \tilde{s}} - \frac{2\dot{\tilde{s}}\dot{\vec{r}}_i}{\tilde{s}}, \\ \ddot{\tilde{s}} &= \frac{1}{Q\tilde{s}} \left(\sum_i m_i \tilde{s}^2 (\dot{\vec{r}}_i)^2 - 3NkT_{\text{hb}} \right).\end{aligned}$$

Here the force is $\vec{F}_i = -\frac{dU(\vec{r})}{dr_i}$. If we insert eq. 3.23, 3.24 and eq. 3.25, 3.26 into these two equations, and do the substitution $\gamma = \frac{\dot{s}}{s}$ we get

$$\begin{aligned}\vec{F}_i^{tot} &= m_i \ddot{\vec{r}}_i = \vec{F}_i - \gamma m_i \dot{\vec{r}}_i, \\ \dot{\gamma} &= \frac{-3NkT(t)}{Q} \left(\frac{T_{\text{hb}}}{T(t)} - 1 \right),\end{aligned}$$

where \vec{F}_i^{tot} is replacing $m\ddot{\vec{r}}_i$, which is the force acting on particle i when the thermostat is on. The variable \vec{F}_i is the force acting on particle i due to the interatomic potential, and if the thermostat was turned off, this would be the total force acting on the particle. We see that all we need to do to implement the Nosé-Hoover thermostat is to change the force acting on each particle by subtracting the term $\gamma m_i \dot{\vec{r}}_i$. To find γ in our numerical simulations we approximate $\dot{\gamma}$ and get

$$\begin{aligned}\frac{\gamma_n - \gamma_{n-1}}{\Delta t} &= \frac{-3NkT(t)}{Q} \left(\frac{T_{\text{hb}}}{T(t)} - 1 \right), \\ \gamma_n &= \frac{-3NkT(t)\Delta t}{Q} \left(\frac{T_{\text{hb}}}{T(t)} - 1 \right) + \gamma_{n-1}.\end{aligned}$$

With the Nosé-Hoover thermostat, each particle gets a term subtracted from the force acting on it, which alters the equations that is solved each time step. The subtracting term is dependent on the current temperature of the system, and the wanted temperature, among other properties.

3.5 Thermalization

The system is started out in an initial state, and we evolve the system in time by using the Verlet-integrator, see section 3.2. Eventually, we would want to measure physical properties in the system, but before we do that we should simulate for a while. This is called *thermalizing*. In most cases we want to measure physical properties on a system that is in thermal equilibrium, so we simulate with the thermostats turned off until a steady state is reached. A steady state is reached when the temperature is stable and temperature fluctuations are small.

Of course, in order to know how stable the temperature is, we need to measure it, so the temperature is measured during the thermalization phase, but the result of the measurement is only used to determine if the system has reached equilibrium or not.

3.6 Defining neighbors

Often, during the preparation phase or when measuring physical quantities on the molecular dynamics system, we need to find an atoms neighboring atoms. To be able to determine if two atoms are neighbors, we need to define what it means to be a neighbor.

There are multiple choices, some more complicated than others. One that quickly comes to mind is the radial cut-off definition.

Definition 3.6.1. The radial cut-off.

The neighbors of particle i are all the particles, except particle i , within a sphere where the position of particle i is the sphere center and the radius r_{cut} .

Another possible definition is the n closest particles.

Definition 3.6.2. The n closest particles

Imagine a system of N particles, where each particle has a position \vec{r}_j where $j \in [1, N]$. The neighbors of particle i is defined as follows: Make a list, L , of the distances from particle i to all the particles in the system

$$L[j] = |\vec{r}_j - \vec{r}_i|. \quad (3.27)$$

Sort the list from smallest to largest, $L \rightarrow L_{sorted}$. The neighbors of particle i is the particles that was involved in making the n first elements of L_{sorted} .

A third possibility is to use the voronoi-neighbors. To be able to define the voronoi-neighbors, we need to discuss voronoi diagrams. Voronoi diagrams will also be used when we measure the local density of the water.

3.6.1 Voronoi diagrams

A general voronoi diagram is generated from a set of points in space called sites. Each site in the diagram is contained in a polyhedron (or a polygon in two dimensions) called a voronoi cell. The voronoi cell for site i is the points in space closer to site i than any other site. In figure 3.3 we see a voronoi diagram for some uniformly placed sites in two dimensions.

The voronoi cell for a site i is made by drawing line segments between i and all the other sites. The vertex of the voronoi cell is then found by drawing the mid-point normals, to all these line segments. This process is illustrated in figure 3.4, where the grayed out area is the voronoi cell of particle i . We notice that the mid-point normal of particle i and j does not contribute to particle i 's voronoi cell.

Definition 3.6.3. The voronoi neighbors.

We define the voronoi neighbors of particle i as the particles that contribute to the vertices that is a part of the voronoi cell.

With this definition we see that particle j in figure 3.4 is not a part of the neighborhood of particle i .

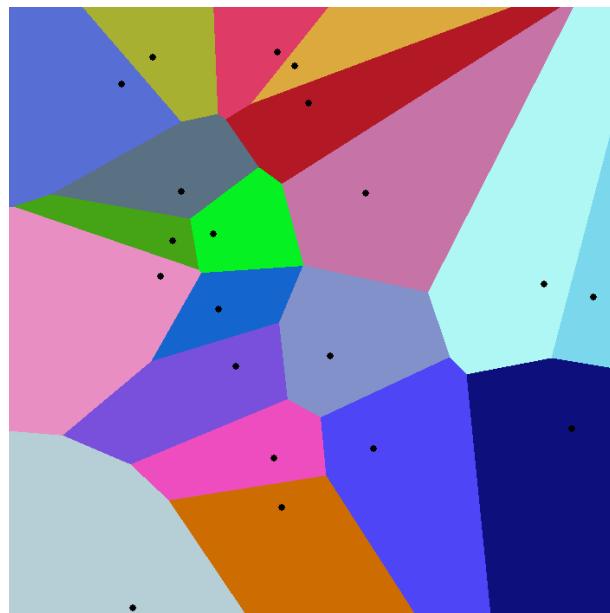


Figure 3.3: A two-dimensional voronoi diagram. This picture is taken from [24].

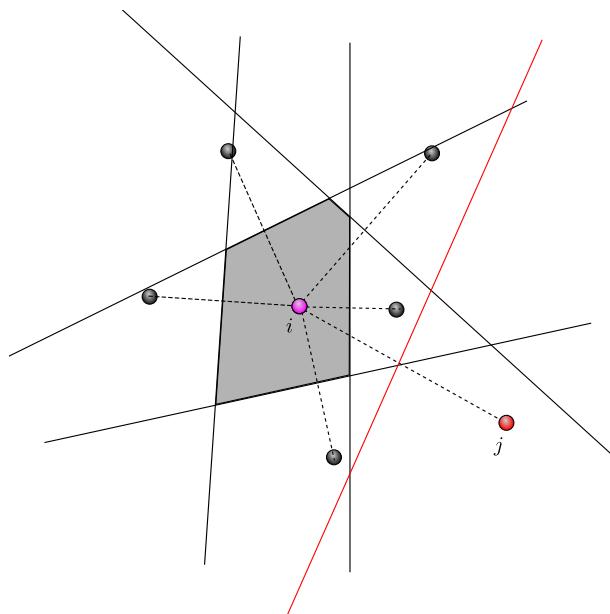


Figure 3.4: Illustration of how the voronoi neighbors are found using a voronoi diagram. Particle j is not part of the neighborhood of particle i , because their voronoi cells do not share a vertex.

We now have three neighbor definitions; the radial cut-off, the n closest particles and the voronoi neighbors. The cut-off and voronoi neighbors can be used if we, for instance, wanted to count the number of neighbors, while the n closest particles are best if we would need exactly n neighbors. If a particle was all alone in an area of the sample, we might want to reflect that in the number of neighbors. The cut-off definition would give zero neighbors, but the two other definitions would give us a list of neighbors at about the same size as the size of the neighbor lists for all of the other particles in the system. The advantage of using voronoi neighbors over the radial cut-off neighbors, is the fact that we do not have to choose a cut-off radius r_{cut} . This is also the case for n closest particles, but then here we have to choose the number n . The choice of definition is determined by our needs in the specific situation.

3.7 Physical quantities

This section lists common physical quantities we can measure on a molecular dynamics system. There are many interesting quantities to measure, and we have chosen out a few which are relevant for our investigations.

For this thesis we measure additional physical quantities to those in this section, but they are not so commonly measured and do not belong in this introduction to molecular dynamics chapter. These quantites will be described in chapter 6 Measurements.

3.7.1 Temperature

According to the equipartition theorem, the average total kinetic energy of the ideal gas of N particles, at temperature T is

$$\langle E_k \rangle = \frac{3}{2} N k_B T, \quad (3.28)$$

where k_B is the Boltzmann constant. In Molecular Dynamics, we use eq. 3.28 to measure the temperature of our system,

$$T = \frac{2\langle E_k \rangle}{3Nk_B}. \quad (3.29)$$

Since the velocity of each particle, v_i , is a variable calulated in each time step of the simulation, we can easily calculate the average kinetic energy of the particles in the system by

$$\langle E_k \rangle = \frac{1}{N} \sum_{i=0}^N \frac{1}{2} m_i v_i^2, \quad (3.30)$$

where v_i is the speed of particle number i , m_i is the mass and N is the total number of particles in the system. The temperature of the system at a given time step is

$$T = \frac{1}{3N^2 k_B} \sum_{i=0}^N m_i v_i^2. \quad (3.31)$$

3.7.2 Density

The total density of the system is just the total mass divided by the volume of the system. It is predetermined from the initialization of the system, where we choose a volume and insert an appropriate amount of particles. Since no particles are added to- or removed from the system during simulation, and the system size stay fixed, the total density will be unchanged. Still, the local density varies. We are interested in measuring the local density at different points in the system to map how it varies spatially and in time. Once we have a method for finding the local densities, we can find the density of different subsets in the system, for example a particular area or all particles with a given kinetic energy etc.

The total density of the system is

$$\rho = \frac{M_{\text{total}}}{V_{\text{total}}} = \frac{M_{\text{total}}}{\bar{V}_{\text{particle}} N_{\text{total}}} = \frac{\bar{m}}{\bar{V}_{\text{particle}}}, \quad (3.32)$$

where M_{total} is the total mass of the system, V_{total} is the total volume, $\bar{V}_{\text{particle}} = \frac{V_{\text{total}}}{N_{\text{total}}}$ is the average volume per particle and $\bar{m} = \frac{M_{\text{total}}}{N_{\text{total}}}$ is the average mass of one particle. From now on we assume that we have a system consisting of only one type of particles, with the same mass.

We define the local density as the density at each particle position, which is just the mass of the particle divided by the volume corresponding to the particle (similar to the total density in eq. 3.32),

$$\rho_{\text{particle}} = \frac{m_{\text{particle}}}{V_{\text{particle}}}. \quad (3.33)$$

We know the mass of the particle, but what is the corresponding volume? One option is the voronoi volume. The voronoi volume is found for each particle by creating a voronoi diagram where each site is the position of the particle. The voronoi diagram is explained in section 3.6.1 where it is used to find the neighboring atoms. A two dimensional diagram is illustrated in figure 3.3.

In the voronoi diagram each particle is surrounded by a polyhedron where all points inside the polyhedron are closer to that particle than any other particle in the system. We use the volume of the polyhedron as the volume corresponding to our particle.

If we want the density of a subset of the system, we use the sum of the volumes:

$$\rho_{\text{subset}} = \frac{N_{\text{subset}} m_{\text{particle}}}{\sum_{i=0}^{N_{\text{subset}}} V_{\text{polyhedron}}^i} \text{ for } i \in \text{subset} \quad (3.34)$$

where N_{subset} is the number of particles in the subset and $V_{\text{polyhedron}}^i$ is the volume of the polyhedron surrounding particle i . We notice that if we let the subset be the entire system, we end up with the total density

$$\rho_{\text{total}} = \frac{N_{\text{total}} m_{\text{particle}}}{\sum_{i=0}^{N_{\text{total}}} V_{\text{polyhedron}}^i} = \frac{M_{\text{total}}}{V_{\text{total}}}.$$

3.7.3 Diffusion

The word diffusion comes from the latin verb *diffundere*, and means *to spread out*. Usually, the word is used about the movement of particles due to differences in concentration at different regions. It is also used when describing heat exchange between two systems of different temperatures; the heat flows from the system with high temperature to the system with low temperature. When we talk about diffusion in this thesis we mean the concept of *self-diffusion*. Self-diffusion is the movement in a substance that has no concentration differences.

Diffusion is described by the diffusion equation

$$\frac{\partial C(\vec{r}, t)}{\partial t} = D \nabla^2 C(\vec{r}, t) \quad (3.35)$$

where $C(\vec{r}, t)$ is the concentration at position \vec{r} at time t , and D is the diffusion constant. When we talk about self-diffusion, we talk about how an arbitrary particle in the system move through a substance, which is the other particles in the system. In that case, the C in eq. 3.35 should not be interpreted as a concentration, but as a probability distribution for the position of the particle at a given time. It is possible to show that the concentration/probability distribution for the positions in a model system of random walks will follow the diffusion equation (eq. 3.35), so we will start there.

(The following derivations are collected from section 7.3 in *Statistical Physics - a second course* [25])

Imagine a system of one particle acting like a random walk in one dimension, meaning that the position of the particle at a time t is a sum of random step lengths drawn from a distribution with mean $\langle x \rangle = 0$ and a finite variance σ^2

$$x(t) = \sum_{i=1}^N \Delta x(t_i). \quad (3.36)$$

We can find the probability distribution for $x(t)$ by using the central limit theorem:

Definition 3.7.1. The central limit theorem (from section 7.2 in Statistical Physics compendium[25])

Assume a variable X that is the sum of a large number (N) of random numbers y_i

$$X = \sum_{i=1}^N y_i,$$

where y_i are drawn from a distribution $p(y)$ that has $\langle y \rangle = 0$ and a finite variance $\sigma^2 = \langle y^2 \rangle$. Then the distribution for X is the Gaussian distribution, with mean $\langle X \rangle = 0$ and a variance $\sigma_G^2 = \langle X^2 \rangle = N\sigma^2$:

$$P(X, t) = \frac{1}{\sqrt{2\pi N\sigma^2}} \exp\left(\frac{-X^2}{2N\sigma^2}\right)$$

The central limit theorem tells us that the probability distribution for our random walk is the Gaussian distribution

$$P(x, t) = \frac{1}{\sqrt{2\pi N\sigma^2}} \exp\left(\frac{-x^2}{2N\sigma^2}\right). \quad (3.37)$$

If we substitute $2N\sigma^2 \rightarrow 4Dt$ where D is the diffusion constant and t is time, we see that the probability distribution satisfies the diffusion equation (eq. 3.35). The probability distribution for the position after the substitution is

$$P(x) = \frac{1}{\sqrt{4Dt\pi}} \exp\left(\frac{-x^2}{4Dt}\right).$$

Because this is the Gaussian distribution, we can immediately say that the variance is

$$\sigma_x^2 = \langle x^2 \rangle = 2Dt, \quad (3.38)$$

where $\langle x^2 \rangle$ is the mean square displacement after a time t .

For random walks in three dimensions, where the position of each random walk is $\vec{r} = [x(t), y(t), z(t)]$, we get the mean square displacement

$$\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle = 3\langle x^2 \rangle = 6Dt. \quad (3.39)$$

Eq. 3.39 tells us that the mean square displacement (MSD) is a linear function in time with the slope $6D$. We transfer this result for the random walks to our Molecular Dynamics system. The particles in the Molecular Dynamics system collide with each other which changes the trajectories in an seemingly random way (even though the evolution of the system is deterministic), which could be modeled with random walks. Therefore, we use the results for the random walks to find the diffusion constant in the Molecular Dynamics systems. We calculate the MSD for a particular system at different sampling times by averaging the square displacement, $r^2(t) = [\vec{r}(t) - \vec{r}(0)]^2$, for all the particles in the system. Here \vec{r} is the position of the particle, and $r^2(t)$ is the squared distance between a particle's current position and the its position at a time origin, $t = 0$. According to eq. 3.39, the result of these calculations should be a linear function in time with a slope equal to six times the diffusion constant. If it turns out not to be linear in time, the result transfer from the random walk to Molecular Dynamics was not appropriate. If it is linear, we find the diffusion constant by dividing the slope of the linear function by six. The computational details are described later, in section 6.3.1.

3.7.4 The radial distribution function

When we use the radial cut-off definition for the neighbors, what is an appropriate choice for the cut-off radius, R ? In order to make a good choice for R , we need to investigate the distances between atoms. The radial distribution function, $g(r)$, is a tool suited for this investigation.

Assume the system consists of N particles, and choose an arbitrary particle a . The radial distribution function is then the relative density as seen from particle a 's point of view. We place imaginary, thin spherical shells around particle a , see figure 3.5, and let $g(r)$ be the density, $\rho_{\text{shell}}(r)$ of particles in the shell at distance r from particle a . This is found by counting the number of particles in the shell and dividing by the shell-volume

$$g(r) \propto \rho_{\text{shell}}(r) = \frac{\bar{N}_{\text{shell}}(r)}{V_{\text{shell}}(r)} = \frac{3\bar{N}_{\text{shell}}}{4\pi[(r+dr)^3 - r^3]}.$$

To get good statistics, we usually time average the shell density, and that is why the number of particles in a shell is denoted by \bar{N}_{shell} . A common way to normalize the radial distribution is to let it go towards 1 as the radius increases

$$\lim_{r \rightarrow \infty} g(r) = 1. \quad (3.40)$$

Say the average density of the system is $\rho = \frac{N}{V}$ where N is the total number of particles and V is the volume of the system, then we would calculate $g(r)$ by

$$g(r) = \frac{\rho_{shell}(r)}{\rho} = \frac{\bar{N}_{shell}(r)V}{V_{shell}(r)N}, \quad (3.41)$$

which would give us 1 in the limit $r \rightarrow \infty$.

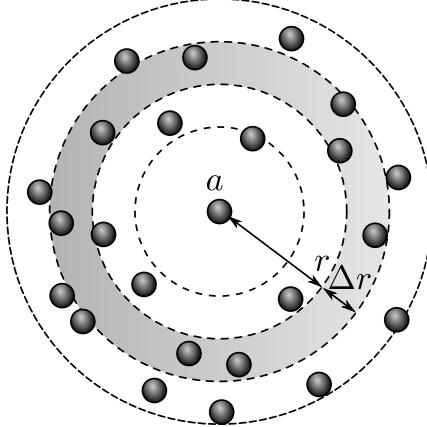


Figure 3.5: This figure illustrate the idea behind the radial distribution function, $g(r)$. Here we have chosen an arbitrary particle in the system, and placed imaginary spherical shells. The radial distribution function for distance r is the density of the grayed out area when we take the limit $\Delta r \rightarrow 0$.

When measuring $g(r)$ we could just choose an arbitrary particle a , and find the densities as described. We get much better statistics, however, if we find the function for all the particles in the system, $g_i(r)$ where $i \in [1, N]$, and use the average: $g(r) = \frac{1}{N} \sum_{i=0}^N g_i(r)$.

The computational process of finding $g(r)$ consists of two stages: *the counting stage* and *the normalizing stage*. It is possible to first do the counting stage, and be completely done with that before starting the normalizing stage.

The counting stage produces an array which is a not-normalized histogram of the distances that appear between particle pairs in the system. This means that the value of element i in the histogram is the average number of particles in the spherical shell with inner radius $= \Delta r(i - 0.5)$ and outer radius $= \Delta r(i + 0.5)$ surrounding an arbitrary particle. In order to make array-element i we need to count the number of particle pairs that have a distance $r \in [\Delta r(i - 0.5), \Delta r(i + 0.5)]$. The histogram corresponds to the function $\bar{N}_{shell}(r)$ in the definition of $g(r)$, eq. 3.41, which should also be time-averaged as well. To time-average the histogram means to take the average of many such histograms found for the same system at different times.

The normalizing stage receives the time-average histogram, $\bar{N}_{shell}(r)$, from the counting stage. On this stage we run through all the elements in $\bar{N}_{shell}(r)$ and multiply with

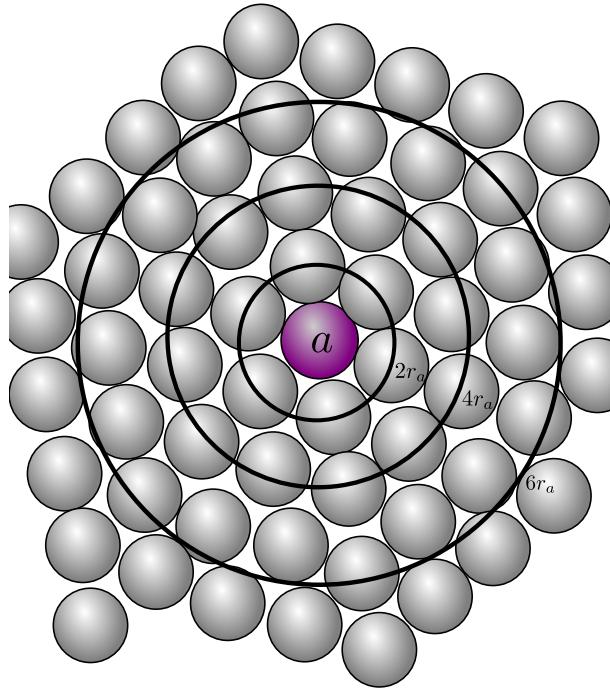


Figure 3.6: A monoatomic system with spherical particles with radius r_a . The particle a is arbitrarily chosen and is used to calculate $g(r)$. This figure is included illustrate how an arbitrary particle “sees” the rest of the system. If distances are measured between the center of particles, there are no particles closer than $2r_a$, and at $2r_a$ there is a ring of neighboring particles (spherical shell in the 3D case). At $r = 4r_a$ another ring appear, though a bit more distorted than the first ring. Between the two first rings there are few particles because most of the particle centers lie on the circles drawn. The first ring of particles is called first shell, and the second ring of particles is called second shell. In the $g(r)$ function the rings will appear as peaks (local maxima) and the empty regions between the rings as local minima. As r increases, the less ringlike-structured the system appears. The values of $g(r)$ at the minima increases and the values at the maxima decreases; the ring structure is washed out and $g(r)$ flattens out.

$V/(V_{shell}(r)N)$ in order to meet eq. 3.40 and eq. 3.41. Notice that the factor we multiply by is different for each element. In fact, the factor decreases as r is increased because it is inversely proportional to the volume of the shell, which of course increases as the radius increases (we assume fixed Δr).

In figure 3.7 we see examples of $g(r)$ measured on simulated bulk water. A typical $g(r)$ has several maxima (peaks) and minima for small values of r , and flattens out for large values of r . Typically, the first maxima is a large peak compared to the other maxima. This characteristic shape can be explained. In figure 3.6 we see a dense system of particles that each cover a spherical area of radius r_a . This means that no pair of particles can come closer than $2r_a$ leading to $g(r) = 0 | r < 2r_a$. At $2r_a$ we get a peak in $g(r)$ due to the nearest neighbors. Notice that the nearest neighbors form a ring around particle a (this would be a spherical shell in the 3D case). The nearest neighbors take up some space where no other particles can exist, which means that $g(r)$ drops again and we get another

local minimum. Outside the ring of the nearest neighbors another ring appears, though a bit more distorted than the first ring. In $g(r)$ the rings will appear as local maxima and the space between as local minima. As r increases, the ringlike structures are washed out and the values of $g(r)$ at the minima increases and the values at the maxima decreases which leads to flattening of $g(r)$.

MD systems often contain more than one type of particles, for instance water has both hydrogen and oxygen and silica has both silicon and oxygen. $g(r)$ can be found for different types of atom pairs. We can choose to only look at $g(r)$ for oxygen-oxygen pairs, or oxygen-hydrogen pairs, and use $g(r)$ to determine the average distance between oxygen and hydrogen in a water molecule. In figure 3.7, $g(r)$ is plotted for the three possible atom pairs in bulk water: oxygen-oxygen, oxygen-hydrogen and hydrogen-hydrogen. The position of the oxygen atom in the water molecule is often used as the position of the water molecule it self, so we may use the plot for oxygen-oxygen to find the distance between the nearest neighbors of water molecules by finding the distance r at the first minima. The first minima in oxygen-hydrogen $g(r)$ gives an upper limit for the distance between oxygen and hydrogen in a water molecule.

In figure 3.8 we see $g(r)$ for the three possible atom pairs in bulk silica. We see that the curves have the characteristic peaks and minima. In silicon-oxygen $g(r)$ we can find an upper limit for the distance from a silicon atom to its nearest oxygen neighbors, at the first minima. This distance is called the first shell distance and is the R used in the passivation procedure.

If we use the radial cut-off definition, two atoms are considered neighbors if the distance between them is smaller than some radius R . Sometimes when deciding a value for R , we want to use a distance corresponding to a certain amount of neighbors. For instance, we may want to choose R so that the average number of neighbors in the system is 4. A function, $N_{neigh}(R)$ telling us the number of neighbors as function of R can be found by integration of $g(r)$

$$N_{neigh}(R) = \int_0^R g(r) \rho A_{shell}(r) dr, \quad (3.42)$$

where $A_{shell}(r)$ is the area of a spherical shell at r , and ρ is the number density of the entire system.

Numerically we start by extracting $\bar{N}_{shell}(r)$ from $g(r)$ by multiplying $g(r)$ with ρ and $V_{shell}(r)$. In figure 3.9 the histogram/function $\bar{N}_{shell}(r)$ is plotted. $\bar{N}_{shell}(r)$ tells us the number of particles in a spherical shell with width Δr at r , so in order to find the average number of neighbors inside R we just need to find the sum of all the shells inside R . The average number of neighbors for a radius R is

$$N_{neigh}(R) = \sum_{r_i=0}^R \bar{N}_{shell}(r_i) = \sum_{r_i=0}^R g(r_i) \rho V_{shell}(r_i),$$

where $V_{shell}(r_i)$ corresponds to $A_{shell}(r)dr$ from eq. 3.42.

$g(r)$ for bulk water

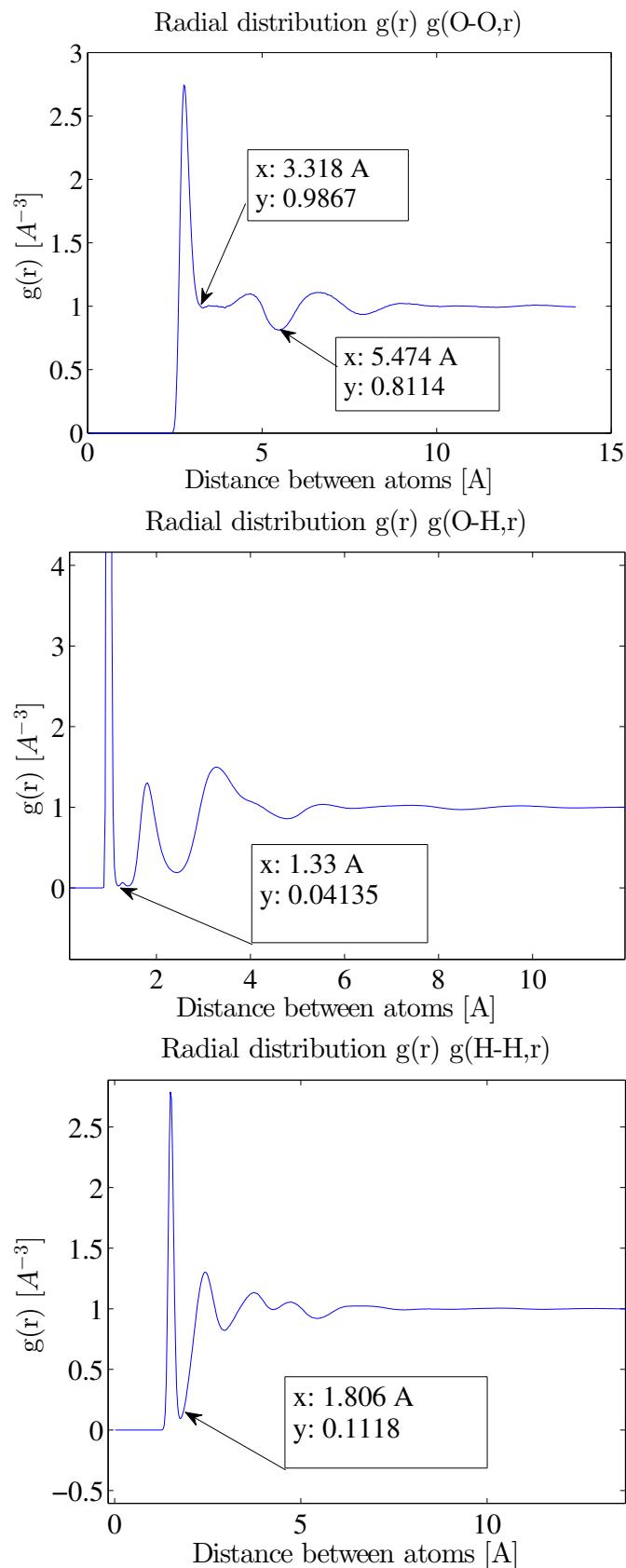


Figure 3.7: The radial distributions for bulk water.

$g(r)$ for bulk silica

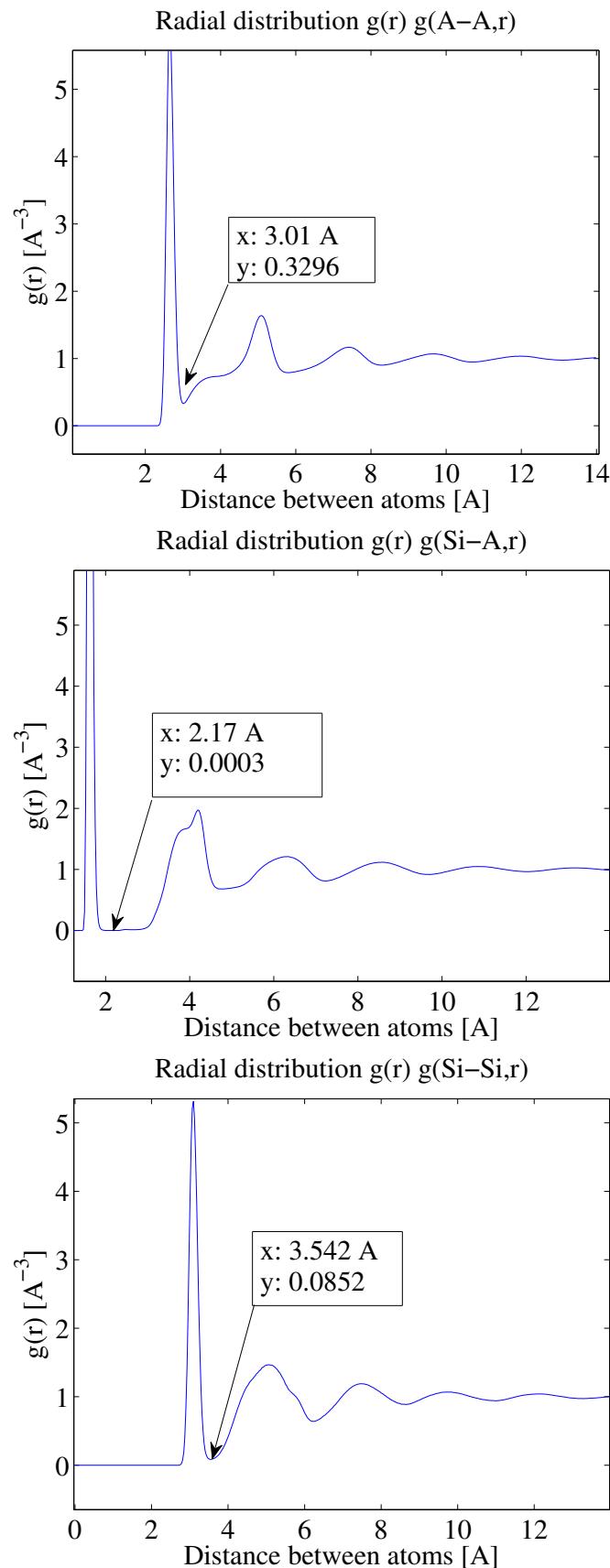


Figure 3.8: The radial distributions for bulk silica.

$g(r)$ integrated to find the number of neighbors

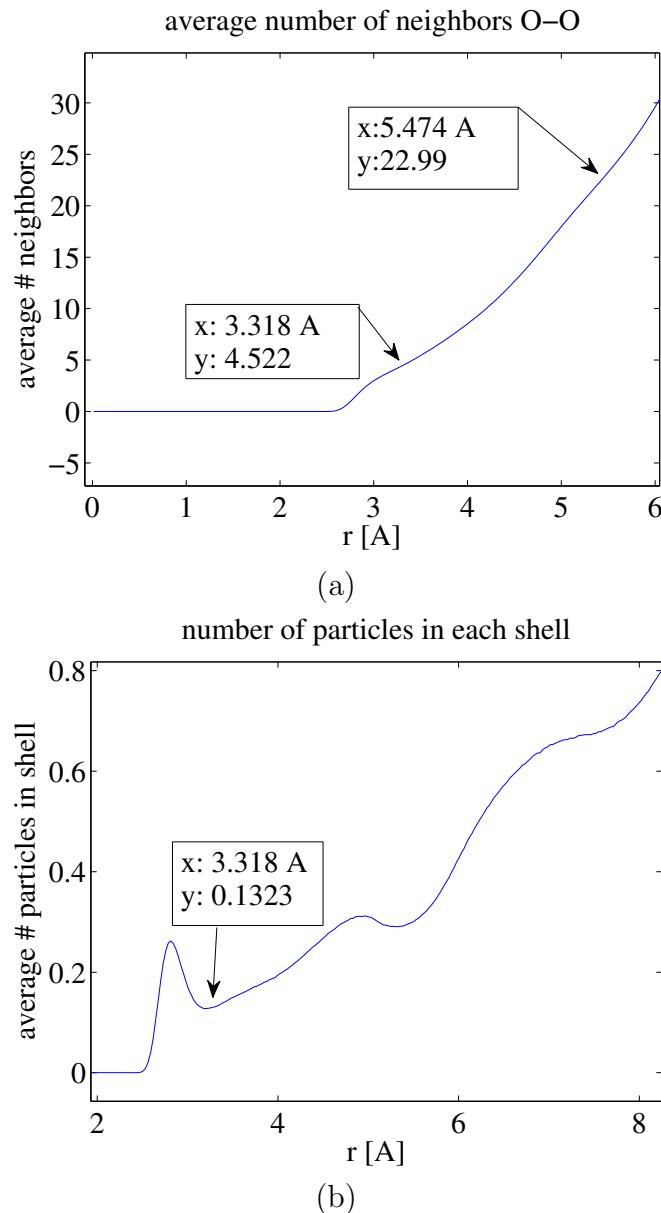


Figure 3.9: (a) The function plotted here tells us the number of neighbors of an arbitrary water molecule in the system, if the neighborhood was a sphere with radius, r . Found by integration of $g(r)$. The marked points are at the radius found for first and second shell in $g(r)$. Inside the first shell we see that the average number of neighbors is 5.5, and inside the second shell it is 23.

(b) This function is the number of water molecules in a spherical shell with radius r and width $\Delta r = 0.028$ Å, around an arbitrary water molecule in the system. It is found by $\rho g(r) V_{\text{shell}}(r)$.

4

Generating a porous media

In this chapter we will talk about how we build up and prepare a nanoporous MD system of silica and water. Some of the topics are specific for silica and water systems, while other topics apply to generic porous systems, for example the section about different ways to make pores in a sample. The main features of preparing a MD-system of a porous media, are to start with a non-porous sample of a substance/material of our choice, which for us is the amorphous silica (SiO_2) glass, and then make small cavities or pores in it. We choose to fill the cavities with water, but it is of course possible to fill them with other substances or gases, for example carbon dioxide (CO_2) or methane (CH_4).

We want our porous media to consist of silica in an amorphous glass state, so our first topic is on how to create amorphous glass, and then we will discuss different methods for creating pores. After the pore making section we will discuss a process called passivation, which is the process of sealing the freshly made pore surface. One of the goals for this thesis was to further develop the method for passivation, so this chapter contains the method we created and a little about how we ended up with this method. Eventually, there is a section about the way we inject water into the pores.

4.1 Creating amorphous glass

The goal is to make a silica glass which will be the basis for the porous media. We start by placing silicon and oxygen atoms in the crystalline structure of the β -cristobalite, as seen in figure 4.1. Figure 4.1 (b) is the unit cell that builds up the crystal, and figure 4.1 (a) is the crystal build up of 1728 unit cells, 12 unit cells in each direction. Each silicon atom is surrounded by four oxygen atoms, tetrahedrally, and each oxygen is shared by two silicon atoms, which leads to twice as many oxygen atoms as silicon atoms in the system (SiO_2).

To obtain a glass state, the crystal is heated up and then cooled down quickly, or quenched. The sample is cooled down quickly so that the atoms do not have time to come back to the lattice locations of the crystalline structure. When quenching the complex configuration from the heated stage is conserved and we have obtained amorphous glass. Each of the two temperature changes are done stepwise, meaning that if we have temperature T_{current} and want T_{wanted} , we do not set the thermostat to T_{wanted} immediately, but do iterations where we set the thermostat to intermediate temperatures that

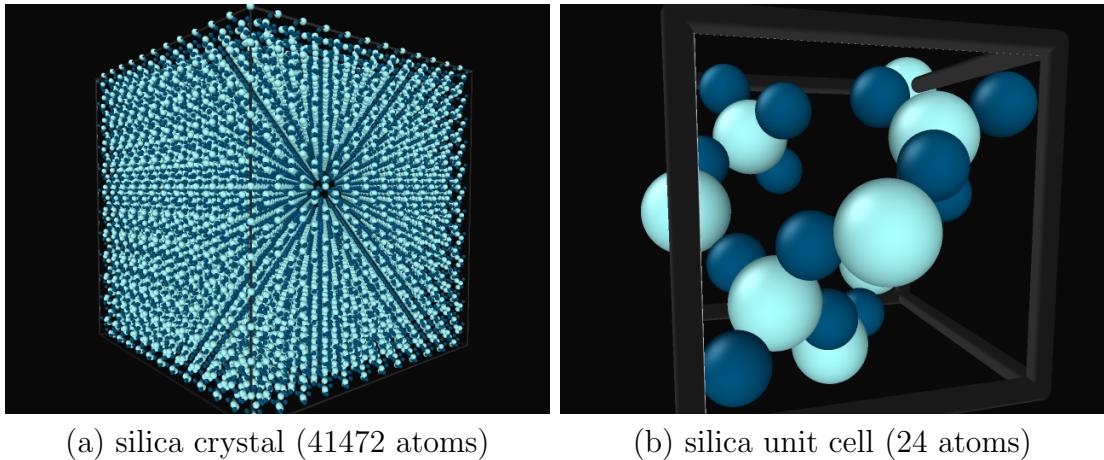


Figure 4.1: A silica β -cristobalite crystal and a unit cell it is built from. Colors: oxygen (dark blue), silicon (light blue). These systems are generated from our MD-code, and visualized with the open visualization tool OVITO [26].

approaches T_{wanted} . For each iteration we have a phase where the thermostat is turned on, and another phase where we let the system thermalize (thermostat is turned off).

In figure 4.2 we see the system in the glass state, after the heating and the quenching.

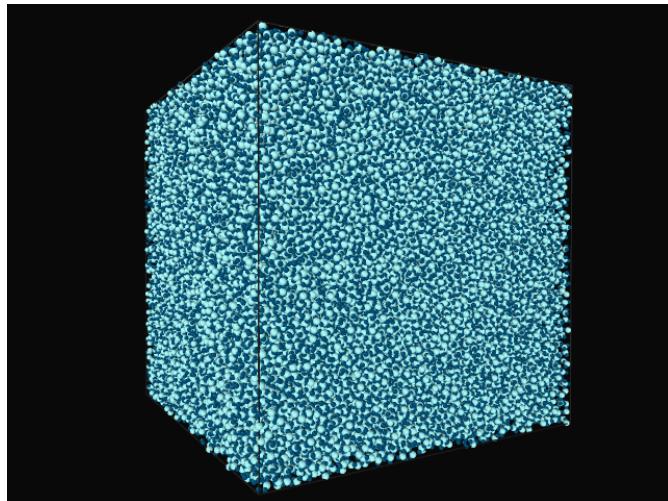


Figure 4.2: A silica glass. Colors: oxygen (dark blue), silicon (light blue).

4.2 Methods for creating pores

The method we choose for creating pores depends on what we want to do research on. For this thesis, we want to investigate behaviors inside pores with simple geometric shapes (spherical, cylindrical, plane), and we choose the method that gives us that. Others might want to look at more heterogeneous pore structures, and they will typically choose a different method.

The method used for this thesis

The method for making pores in a homogeneous sample, is based on a process where we remove atoms that occupy the wanted pore space. We say we cut out the pore. If we want a spherical pore with radius R and a center in the middle of the system, \vec{r}_{center} , we remove all atoms inside this sphere. In practice, it would mean running through all atoms in the system and for each atom check if its position is closer to the pore center than the distance R . If it is, the atom is removed from the system. The test each atom has to pass in order to be removed looks like this

$$(x - x_{\text{center}})^2 + (y - y_{\text{center}})^2 + (z - z_{\text{center}})^2 < R^2,$$

where x_{center} is the first coordinate of the center of the pore, and x is the first coordinate of the position of the atom.

When we remove atoms from the silica, we should make sure to remove twice as many oxygen atoms as silicon atoms, so that the chemical formula for the remaining substance is SiO_2 . We say we cut out the pore neutrally. When we run through the atoms and remove those inside the pore, we also store the ids of the atoms that are closer to the pore wall than some distance ϵ . For the spherical pore, this means to store all the ids of the atoms that were not removed and are inside the sphere with radius $R + \epsilon$. At the end of the procedure we count the number of removed silicon and oxygen, and if there are too few oxygen atoms removed compared to silicon, we randomly pick out as many oxygen atoms as we need from the stored ids, and remove those as well, and vice versa.

Methods for complex pore structures

We may use the same cutting strategy to make a complex pore space, by letting the cutting region be complex. For instance, we could remove all atoms inside spheres of random positions and radii. These spheres could overlap. If we choose many spheres with a small maximum radius, we will typically get a more complex system than if we choose to have few spheres with large maximum radius.

Another way of getting a complex system is by removing a certain fraction of the system, in a random way. For instance, if the fraction is p , then we could draw a random number between zero and one for each atom, and those atoms with number smaller than p would be removed.

The last method we mention here does not include removing atoms. We start with a substance that has the appropriate density according to the potential used for interaction between the atoms, meaning that the average distance between the atoms when the atoms are uniformly distributed in the volume, match a potential well distance, and that perturbing the distances in the system would not lower the energy of the system. Then we stretch the system by increasing all distances in the system (kind of like how the universe expands according to Hubble, only we do it instantly), and use this new volume as the basis for simulations. We have increased the volume, but let the number of atoms stay the same. We expect that the energy of the system would decrease if the distances between the atoms were smaller than it is immediately after the stretching. If we are lucky, the atoms will try to lower the energy of the system by gathering somewhere in the system, with a higher local density than the total density of the system, and leave other regions empty which gives us pores.

4.3 Passivating (hydroxylate) the porous media

We have generated a glass and, by one method or the other, made pores in it. In the pore-making process we may have broken covalent bonds between atoms, or removed atoms that used to make covalent bonds with the atoms on the pore wall, leaving the atoms on the pore wall with dangling bonds.

We want to 'seal' the surface by inserting hydroxyl-groups and hydrogen-groups at the position of the dangling bonds and by that making the surface less reactive. This is called passivation or sometimes to hydroxylate the surface.

The idea is that if we had injected the water right away, and simulated long enough, some of the H_2O molecules would split up to OH-groups and H-groups and attach to the pore wall at the dangling bonds. Unfortunately, the thermalization time for such a system is too long for us to wait for, so we speed up the process by inserting the OH- and H-groups ourselves before we start our simulation.

After passivation, the pore walls will not react as heavily with the water inside the pore, as it would without passivation, and we reach a stable state much quicker.

Since the process we are trying to copy is the splitting of water into OH - and H -groups it is important that we insert the same amount of OH-groups as H-groups; we need to passivate neutrally (we do not want an alkaline/acidic system). If a silicon atom is missing an oxygen neighbor, we insert an OH-group close to this atom. If an oxygen is short one silicon neighbor, we insert a H-group close to the oxygen atom.

4.3.1 Developing a method for passivation

The chemical formula for the glass is SiO_2 , which means that there are twice as many oxygen atoms as silicon atoms in the system. Oxygen has six electrons in the valence shell, and can therefore form two covalent bonds. Silicon has only four electrons in the valence shell, so silicon can form four covalent bonds. The most common structure of silica is one silicon bonded to four oxygen, which means that the coordination number of silicon in silica is four. The oxygen atoms are placed at the corners of a tetrahedron surrounding the silicon. Each oxygen forms covalent bonds with two different silicon atoms, which means the silica shares its four oxygen neighbors with four other silicon atoms and that the coordination number of oxygen in silica is two. [27]

The atoms on the surface that should be passivated are under-coordinated, meaning the coordination number exceeds the number of nearest neighbors. This might occur inside the bulk of the glass as well as on the pore surface, but it rarely does. In order to find all under-coordinated atoms, we need to count the number of neighbors of all atoms in the system. The natural definition of neighbors to choose here is the radial cut-off definition, see definition 3.6.1, where all particles within a given radius, R , of particle i are its neighbors. To save some computation time on the neighbor counting process, we first do a rough classification where we find potential neighbors which are the particles that are close to particle i , and later pick the actual neighbors from the roughly classified neighbors. We split the system into equally sized voxels with volume R^3 . A voxel is the 3D-equivalent of a pixel in a digital picture. To find the neighbors of particle i , we first find the voxel the particle is contained in, and call it voxel v_i . Voxel v_i is the grayed out area of figure 4.3 where we also see particle i and the potential neighbors. As the first

rough classification, we say that all particles in the voxel v_i and the neighboring voxels, are neighbors of particle i . To store this information we have a list as long as the number of particles in the system, let's call it *voxel_neighbor_list*. Each element is a list of the indices of the neighboring particles, for example: element number i (*voxel_neighbor_list*[i]) is a list of the indices of the neighbors we found for particle i . We then run through the list in *voxel_neighbor_list*[i] and pick the particles that *actually* has a distance to particle i smaller than the radius R , and store them in a new list *neighbor_list*[i].

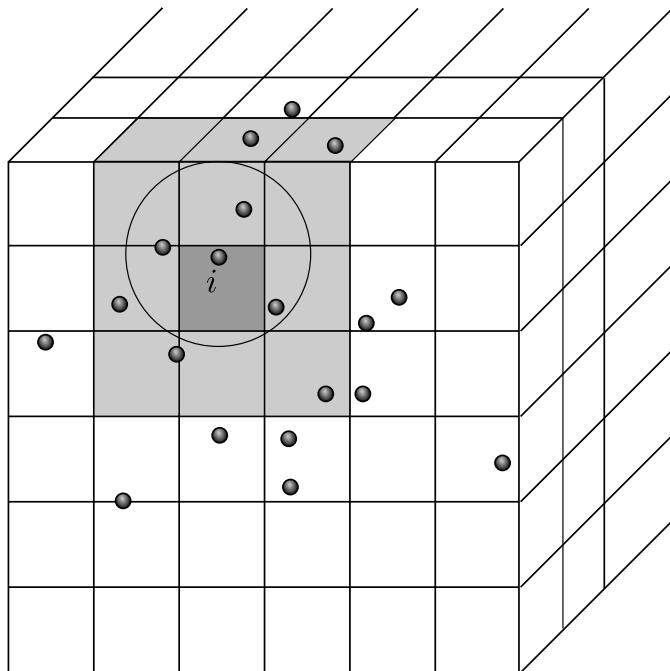


Figure 4.3: The system is divided into equally sized voxels. The dark grayed out area is the voxel in which particle i is contained, and the area with a lighter shade of gray is the neighboring voxels. As a first crude classification all particles in the gray area (light and dark) are considered particle i 's neighbors. The circle marks the sphere in which the actual neighbors of particle i is contained.

Now, we use *neighbor_list* to find the under-coordinated particles(just by counting), and insert the appropriate groups (OH or H). But where should the groups be positioned? Close to the particle with missing neighbors for sure, but there are infinite possibilities and we need to make sure we do not place the group too close to other particles. One way is to use the voxel-system we made before. Say particle i is a silicon atom with only three oxygen neighbors. The appropriate group is then an OH-group. We could just put it in the middle of one of the empty neighboring voxels, with the hydrogen pointing away

from the voxel containing particle i . That way the group is within the neighborhood of particle i , and it is not too close to other particles. This is just an approximation for the position which will be adjusted during the steepest descent stage of the process (more about this later, section 4.3.3). Still, we wanted to explore other and better options than the voxel method for inserting the groups. We wanted to exploit the fact that the four oxygen atoms often are placed on the corners of a tetrahedron around the silicon in bulk. This is not always the case, and our potential does not explicitly tell us that the oxygen atoms should behave like this, but one of the crystalline forms of silicon dioxide in nature is the quartz. The building blocks of quartz is silicon with oxygen on the corners of the tetrahedron [27].

So, the idea is to place the oxygen in the OH-group so that the silicon with its four neighbors is as close to a tetrahedron as possible. The hydrogen in the OH-group and in the H-group is just placed near the corresponding oxygen atom at an angle with the line that goes through both the oxygen and the silicon in question. There are four outcomes for each silicon with number of oxygen neighbors less than four: one missing oxygen, two missing oxygen, three missing oxygen or four missing oxygen. If we are in the situation where a silicon lacks all its oxygen atoms, we insert an oxygen at an appropriate distance, and use the strategy for three missing neighbors. We developed three methods for the remaining three outcomes:

- One missing: We find the normal vector, \vec{n} , to the plane formed by the three existing oxygen atoms, and insert the OH-group at $\vec{r}_{new} = \vec{r}_{Si} + D_{Si-O}\vec{n}$, where \vec{r}_{Si} is the position of the silicon in question and D_{Si-O} is an appropriate distance between silicon and oxygen. This distance should be smaller than the radius R that determines the neighbors, but not much smaller. See the left figure in figure 4.4 for a visual explanation.
- Two missing: We find the plane formed by the two existing oxygen atoms and the silicon atom. The two inserted oxygen atoms should be placed on opposite sides of this plane in a symmetric manner. We use the plane normal \vec{n} and the vector \vec{O} which is made by adding the two vectors between the silicon atom and the two existing oxygen atoms, and normalized. We insert the OH-group at $\vec{r}_{new}^{\pm} = \vec{r}_{Si} \pm \omega_n \vec{n} - \omega_O \vec{O}$, where $\omega_n = D_{Si-O} \sin(\frac{\alpha}{2})$ and $\omega_O = D_{Si-O} \cos(\frac{\alpha}{2})$ (α is the tetrahedron angle $\alpha = 109.5^\circ$). See the right figure in figure 4.4; for the two missing oxygen case the bold line is the plane through the two oxygen atoms and the silicon.
- Three missing: We find the vector between the silicon and the existing oxygen atom and use this as the normal to the plane, \vec{n} , the three inserted OH-groups should be on (\vec{n} is normalized). The three OH-groups can be placed anywhere along a circle on the plane, as long as they are placed in the vertices of a equilateral triangle. We randomly choose the position of one of the new oxygen atoms along the circle, and the other two positions are given by the equilateral triangle. We find the midpoint of this circle in the plane by $\vec{r}_{mid} = \vec{r}_{Si} + \cos(\frac{\alpha}{2})\vec{n}$. See figure 4.4; for the three missing oxygen case the line is just the vector between the existing oxygen and the silicon atom.

We now have a desired position for the group(s) we want to insert, but if this position is very close to other particles we skip inserting it. At the end, if there is imbalance in

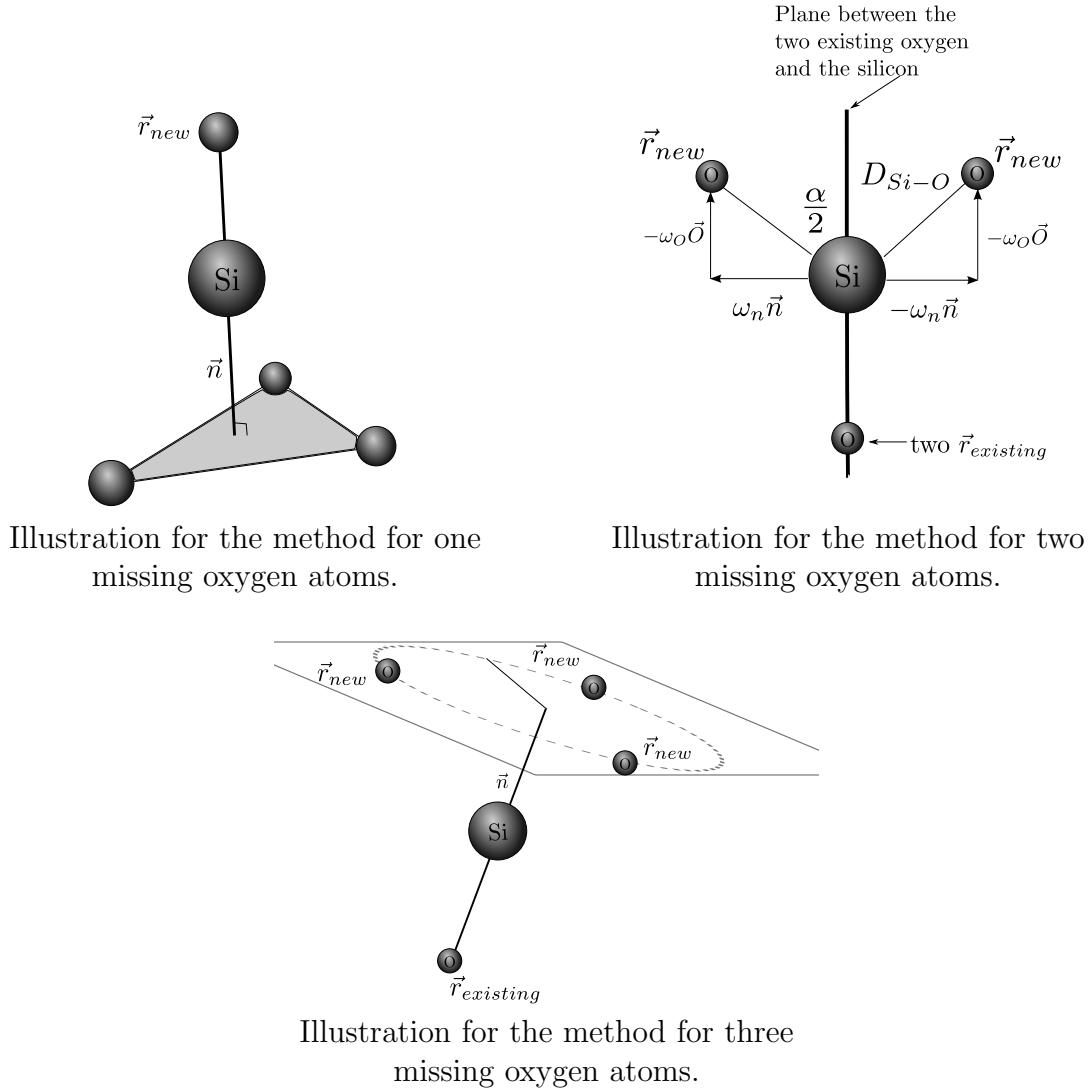


Figure 4.4: Illustrations to the tetrahedron method.

the amount of inserted OH-groups and H-groups we simply remove the excess groups by picking randomly.

The code that carry out the tetrahedron method is much more intricate than the code for the voxel method and more time consuming, but the result is better with the tetrahedron method judged by visual inspection.

As mentioned earlier, we might find atoms with missing neighbors inside the bulk of the glass. To avoid this we may use a technique where we find the voxels on the surface of the pore, and only look for missing neighbors for atoms contained in these voxels.

4.3.2 Before and after passivation

In figure 4.5 we see the passivated surface of a pore in a β -cristobalite sample. For this type of system, we expect hydrogen and hydroxyl groups to cover the surface of the pore and nothing inserted into the bulk silica, because the structure is very regular and the atoms should be coordinated. In the picture to the left we see a slice of the system that has a regular crystallic configuration, and there is a spherical hole in the middle. In the picture to the right we see the same slice of the system after it has been through the passivation procedure. The surface of the pore has been covered with red and white particles, which is oxygen and hydrogen. In the bottom picture we see the same system, zoomed in on the pore surface. Inside the red circle we see how the OH group is placed on the tetrahedral structure. Note that in this figure the system has not been through the steepest descent algorithm.

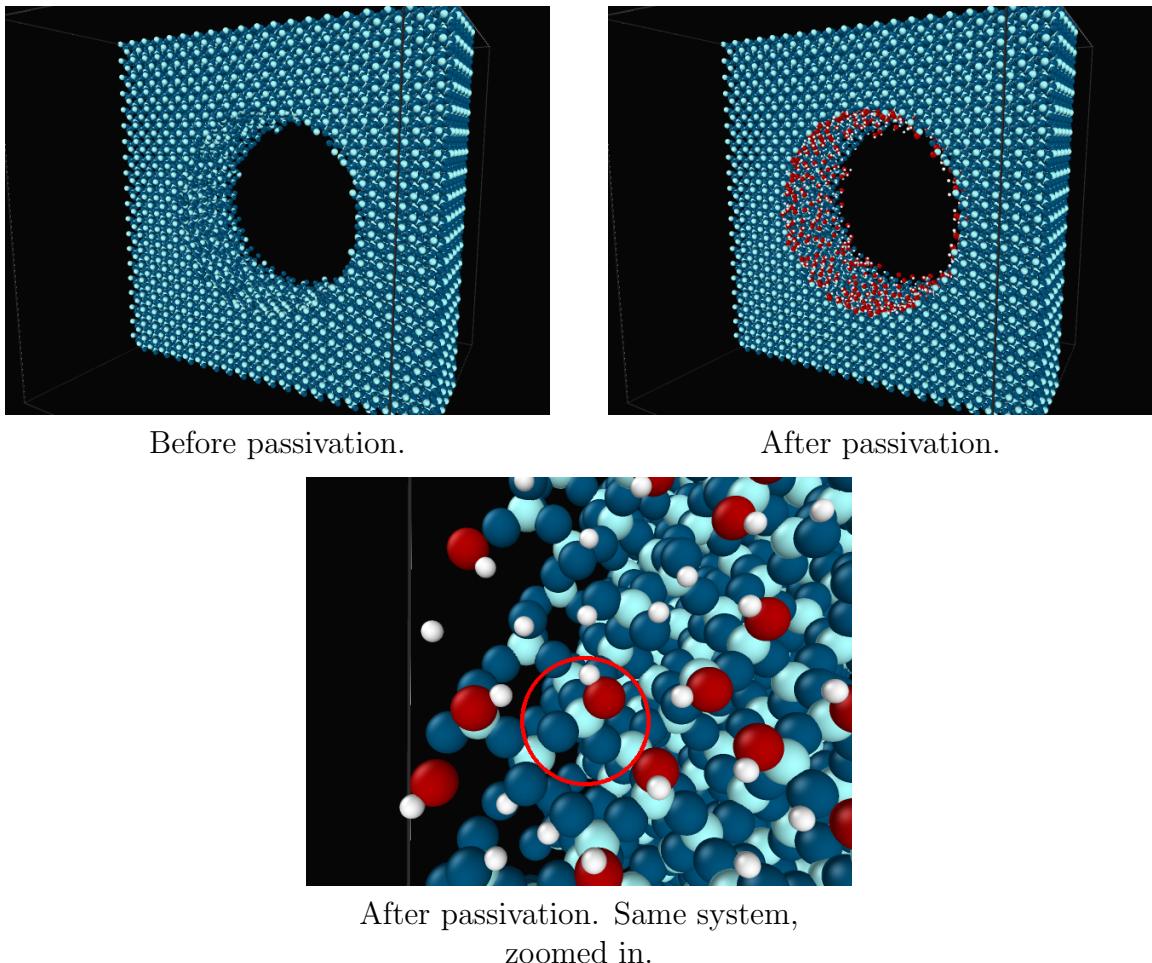


Figure 4.5: Crystallic system with spherical pore, before and after passivation. Colors: light blue (silicon), dark blue (silica-oxygen), white (inserted hydrogen), red (inserted oxygen). The hydrogen atoms are placed at an angle with the oxygen and the silicon.

4.3.3 The steepest descent method

After the passivation stage, it is important to do the steepest descent procedure on the inserted atoms. This is to make sure that none of the inserted atoms are too close to another atom, which could lead to unphysical large forces. The steepest descent method moves the particles away from regions where the potential is steep, where the forces would be large. It also reduces thermalization time, because we only calculate the forces on the inserted atoms. The steepest descent method is primarily used to find local minima of functions.

The idea of the method is to move each particle in the opposite direction of the gradient of the potential. We have defined a variable to this quantity, namely the force the particle feels from the other particles. The magnitude of the movement is proportional to the size of the force. The method is an iterative method. Let \vec{r}_i be the position of an arbitrary inserted particle in our system and $\vec{F}_i(\vec{r}_i)$ is the force working on this particle when the particle is in position \vec{r}_i , at iteration i . The position at iteration $i + 1$ is given by,

$$\begin{aligned}\vec{r}_{i+1} &= \vec{r}_i + \alpha \vec{F}_i(\vec{r}_i), \\ \vec{F}_i(\vec{r}_i) &= -\nabla U_i(\vec{r}_i)\end{aligned}$$

where α is a proportionality constant which is the same for all particles. We have $3N_{\text{inserted}}$ such calculations to do, three for each particle inserted into the system. For every iteration we need to calculate the force again, since some of the particles have moved and changed the potential.

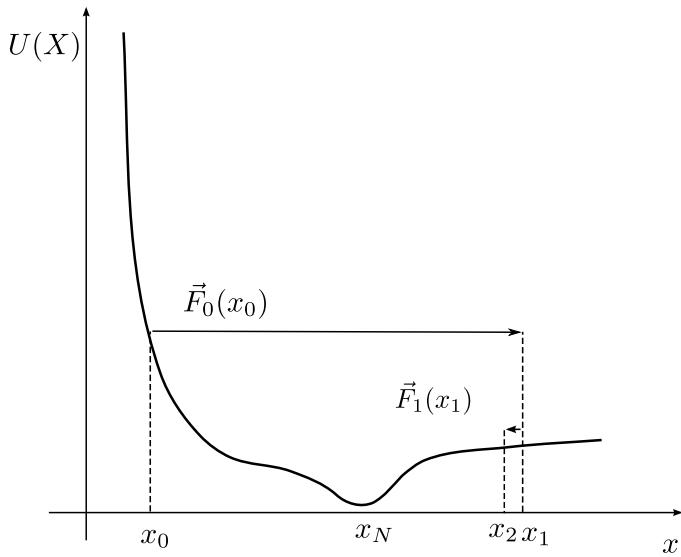


Figure 4.6: A made up potential for one dimension.

To illustrate the method, we have made up a potential for the one dimensional case, which can be seen in figure 4.6. The potential has a very steep part, some parts with more gentle slopes and a minimum. Say we insert a particle at position x_0 , and we use the

method of steepest descent on this particle. The final destination after N iterations will be the position marked x_N , that is, if we do things correctly. We notice that the particle is placed in a steep part of the potential. This means the gradient, or force is large in magnitude, illustrated by the large arrow. The next position, x_1 , is found by going a distance proportional to the magnitude of the force, in the direction of the force. In this example we chose $\alpha = 1$ for simplicity. With this α we passed over the local minimum at the first step, but we ended up in a part of the potential that gives force in the opposite direction.

The new position, x_1 , is in a very shallow part of the potential, which means that the magnitude of the force is small and the next position, x_2 , will be close to the previous position, x_1 . The idea of the method is that the force will push the particle in the direction of the closest minimum, and the particle gets closer to its goal. In theory we can get as close as we want, but the chance of hitting the ideal spot exactly is very small (sometimes a minimum does not exist, and in those cases we would iterate forever), so we need to decide when to stop iterating. In lack of a better solution we choose to stop when the magnitude of the force is reduced to ϵ times the starting force (we stop when $\frac{|\vec{F}_N|}{|\vec{F}_0|} < \epsilon$). If the particle ends up in a part of the potential similar to a parabola, we might get oscillatory behavior, and never reach a point where $\frac{|\vec{F}_N|}{|\vec{F}_0|} < \epsilon$, and we will iterate forever. To guarantee termination of our program, we need to set a limit to the number of iterations, a maximum number of steps. Even though the program has to stop before we find the ideal spot for our particle, we probably have found a better starting point for simulations than if we did nothing.

α can change between iterations. The choice of α determines the rate of convergence, and there exists methods for finding a good choice. The brute force way is to let α be constant. It is not a very bad way to do it, we may just converge at a poorer rate than with other methods. The steepest descent part of an experiment is not very important, just a push in the right direction you could say, so for our use the brute force way is good enough.

When we later inject water into our pore, we will use the method of steepest descent on the water molecules, before we start our simulations.

4.4 Injecting water

The procedure for injecting water is built upon a previously used code, but we have added a few alterations and implemented it in C++. One of the alterations was that the physical orientation of the inserted water molecules were chosen at random, as opposed to the old program that left all the water molecules pointing in the same direction. The procedure goes as follows:

The system is divided into equally sized cubic voxels, and the water molecules are placed in the middle of each empty voxel. The size of the voxels determines the obtained density, or the other way around, we choose the size of the voxels according to the density we wish to obtain. Say we want the density ρ , the volume of the voxels would be

$$V_{\text{ideal}}^{\text{voxel}} = \frac{m_{\text{water}}}{\rho},$$

where m_{water} is the mass of a water molecule. With this voxel volume, the length of the voxel in each direction would be

$$d_{\text{ideal}} = \sqrt[3]{V_{\text{ideal}}^{\text{voxel}}} = \sqrt[3]{\frac{m_{\text{water}}}{\rho}},$$

and the number of voxels in each direction is

$$N_{\text{ideal}} = \frac{L}{d_{\text{ideal}}},$$

$$N_{\text{obtained}} = \text{round} \left(\frac{L}{d_{\text{ideal}}} \right),$$

where L is the length of the system in each direction if we assume it is cubic. Since the number of voxels can not be a decimal number, we approximate to the nearest integer. This means we have to adjust the size of the voxel, which lead to a slightly different density than the one we ask for. A solution to this is to always round up when calculating N_{obtained} , and only fill an appropriate fraction of the voxels. This solution was implemented after the simulations for this thesis was completed, which means that all systems for this thesis have been using the method where we approximate to the nearest integer.

When inserting the water molecule into the voxel, the oxygen is placed in the middle of the voxel and the two hydrogen atoms are placed at a distance $r_{\text{OH}} = 0.9584 \text{ \AA}$ from the oxygen atom, and the angle created by the hydrogen - oxygen - hydrogen is set to 104.45 degrees. First, the water molecule is inserted so that one hydrogen atom is pointing in the x direction, and then the whole molecule is rotated with randomly chosen angles. In figure 4.7 we see a system with a spherical pore just after the water molecules are injected. We see structures in the water, and this is due to the use of voxels during injection. The picture is taken immediately after inserting, and steepest descent has not yet been performed. The silica is removed from the picture in order to see the water molecules better.

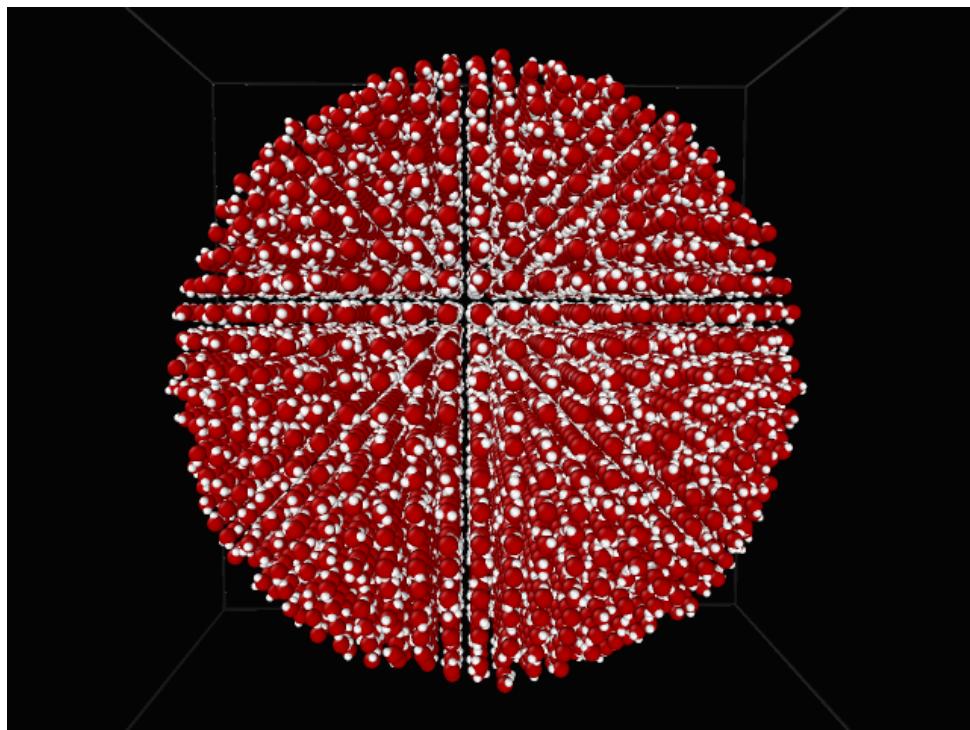


Figure 4.7: Water injected into a spherical pore, immediately after inserting the molecules. The structures we see are due to the use of voxels in the injection procedure, and they will be gone once we have run steepest descent on the inserted atoms. The silica is removed in order to better visualize the water molecules. Colors: oxygen (red), hydrogen (white).

5

Systems

This chapter contains descriptions of all of the systems we have simulated for this master thesis. Some of the information is common to all of the systems, while others are specific for each system.

There is also a section about how the systems evolve during the preparation period and after. At the end of this chapter there is a small section about an interesting phenomenon we encountered during the preparation phase of the system with the plane pore; we had created a nanobubble.

5.1 Description of the different systems

In this thesis we investigate the water inside single pores in silica. There are three different types of pores: the spherical pore, the cylindrical pore and the plane pore, all of which have simple geometric shapes for which there exist mathematical formulas. In addition to the pore systems, we have two systems containing bulk substances. One with bulk water and one with bulk silica. The one with bulk water is used for comparison with the water inside the pores, and the one with bulk silica exist so that we could measure the radial distribution function, $g(r)$. The total amount of systems is ten.

5.1.1 Spherical pores

We performed six simulations where the pore in the silica was spherically shaped. We chose to focus on spherical pores because all points on a spherical surface have the same curvature, which is convenient, and it has a simple expression for the Gaussian curvature, which for the surface of a sphere with radius R is

$$\kappa = \frac{1}{R^2}, \quad (5.1)$$

see section 2.2 for details.

Figure 5.1 shows us the water inside the six spherical pores of different sizes, and table 5.1 contains size related properties. The radius of the pore was originally chosen as $R_{\text{cut}} = 0.3L$, where L is the length of the cubic system in each direction (the entire system has volume $V_{\text{box}} = L^3$), which means that the system size determines the radius

Table 5.1: Table showing the size related properties for the systems with spherical pores. L is the length of the system in each direction (cubic systems), R_{cut} is the radius of the sphere that was cut out of the silica, while measured R_0 is the radius measured right after the cut was made and R_{eq} is the radius measured after thermalizing the system. V/V_{smallest} is the relative volume of the pore compared to the smallest pore simulated.

L [Å]	R_{cut} [Å]	measured R_0 [Å]	measured R_{eq} [Å]	V/V_{smallest}	# H ₂ O	# atoms
42.96	12.89	13.41	13.39	1	299	5484
64.44	16.11	16.41	16.45	1.85	563	18045
64.44	19.33	19.60	19.42	3.05	991	18471
85.92	25.78	25.98	25.75	7.11	2404	43947
114.56	34.37	34.45	34.14	16.57	5533	103791
143.20	42.96	43.02	42.58	32.17	10752	202380

of the pore. There is one exception; the system with the second smallest pore has radius $R_{\text{cut}} = 0.25L$. This system was created later, after the five other systems, because we needed a pore slightly larger than the smallest pore we already had.

We know the radius of the spheres we cut out of the silica, but there is no guarantee that the pore radius does not change during thermalization and simulation, so we decided to measure the pore radius immediately after cutting the silica, R_0 , and after injecting water and properly thermalizing, R_{eq} . The radius was measured by taking the average of the distances between the center of the system and the $n = 50$ silicon atoms closest to the center. This measure depends on the number of silicon atoms, n , so it is not surprising that we do not get the exact same value for R_{cut} and R_0 . We see in table 5.1 that the measured radius changes slightly from R_0 to R_{eq} , but the change is not bigger than 1%.

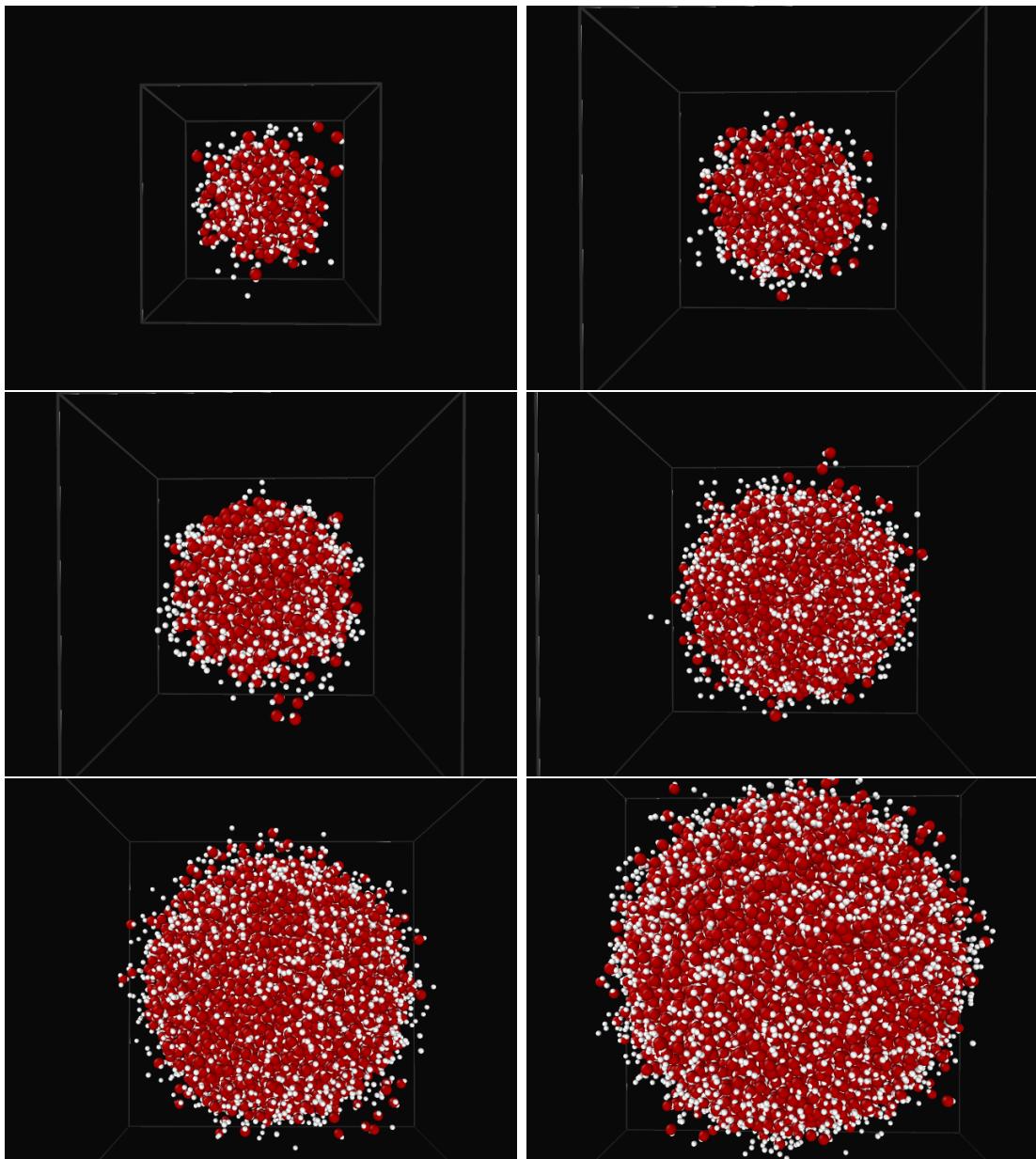


Figure 5.1: The water in the systems with spherical pores. The red atoms are oxygen and the white are hydrogen, which together makes water. We notice the single atoms surrounding the pores. These are water-atoms that have migrated into the silica during the simulation.

5.1.2 Different pores with the same volume

The curvature of the spherical pores depends on the radius of the sphere (eq. 5.1), but so does the volume, and if our results vary as a function of curvature, we can not know if the variations in the results are due to the curvature or the volume. For this reason, we wanted to look at three different systems with different curvatures, but with the same volume. We landed on adding two systems to our ensemble, one with a tunneling cylindrical pore and one with a plane pore which consists of a flat ceiling and a flat floor and no walls (periodic boundary). The latter system is actually just water between two slabs of silica, referred to as a *plane pore*, see figure 5.2.

The two non-spherical pores were constructed to have the same volume as one of the medium sized spherical pores ($R_{\text{eq}} = 25.75 \text{ \AA}$), which means we now have three different systems with the same volume.

Like the sphere, the cylinder and the plane are simple geometric shapes which both have the property that the curvature does not change when moving on the surface. Both have Gaussian curvature equal to zero, but the principal curvatures of the cylinder are not both zero,

$$\begin{aligned} k_{\parallel} &= 0, \\ k_{\perp} &= \frac{1}{R}, \\ \kappa_{\text{cylinder}} &= k_{\parallel} k_{\perp} = 0, \end{aligned}$$

where k_{\parallel} is the principal curvature parallel to the axis of the cylinder, and k_{\perp} is the other principal curvature perpendicular to the cylinder length, see section 2.2 for details. This means that a water molecule can move along the surface in one direction without changing its momentum, but moving along the surface in the perpendicular direction means it has to constantly change its momentum. A water molecule moving along the surface of a sphere has to change its momentum no matter the direction, while a water molecule moving along a plane surface can move freely along any direction without changing its momentum. We chose the plane pore shape because we wanted to investigate a system with no curvature, and we chose the cylindrical pore because the shape is a cross between the plane and the sphere.

Table 5.2 contains size related properties for the three systems with the same volume. From the table we see that the measured value for the length/radius of the pore is slightly larger than the length/radius used to cut out the pore, but if we look at the length/radius measured after thermalizing the system we see that all three pores have shrunk a bit, and the lengths/radii are quite close to the values we wanted in the first place. Lucky!

From the volume and number of water molecules in table 5.2 we can calculate the number density of the water. Comparing the number density for the three pores, we see that $\rho_{\text{planes}} > \rho_{\text{sphere}} > \rho_{\text{cylinder}}$, which is later reflected in figure 6.2.

Table 5.2: Table showing size related properties for the three systems with different pores but the same volume. The values for the spherical pore is the same as the values in table 5.1. L is the length of the cubic system. R is the length that characterizes the pore (radius for sphere and cylinder, height between the two planes in the plane pore). R_{cut} is the length used when cutting the pore, m. R_0 is the length measured immediately after the cutting, and m. R_{eq} is the length measured on the thermalized system.

type	L [Å]	R_{cut} [Å]	m. R_0 [Å]	m. R_{eq} [Å]	V/V_{smallest}	# H_2O	# atoms
plane	57.28	21.86	22.18	21.94	7.16	2440	14949
cylinder	71.60	17.86	18.00	17.79	7.08	2233	26013
sphere	85.92	25.78	25.98	25.75	7.11	2404	43947

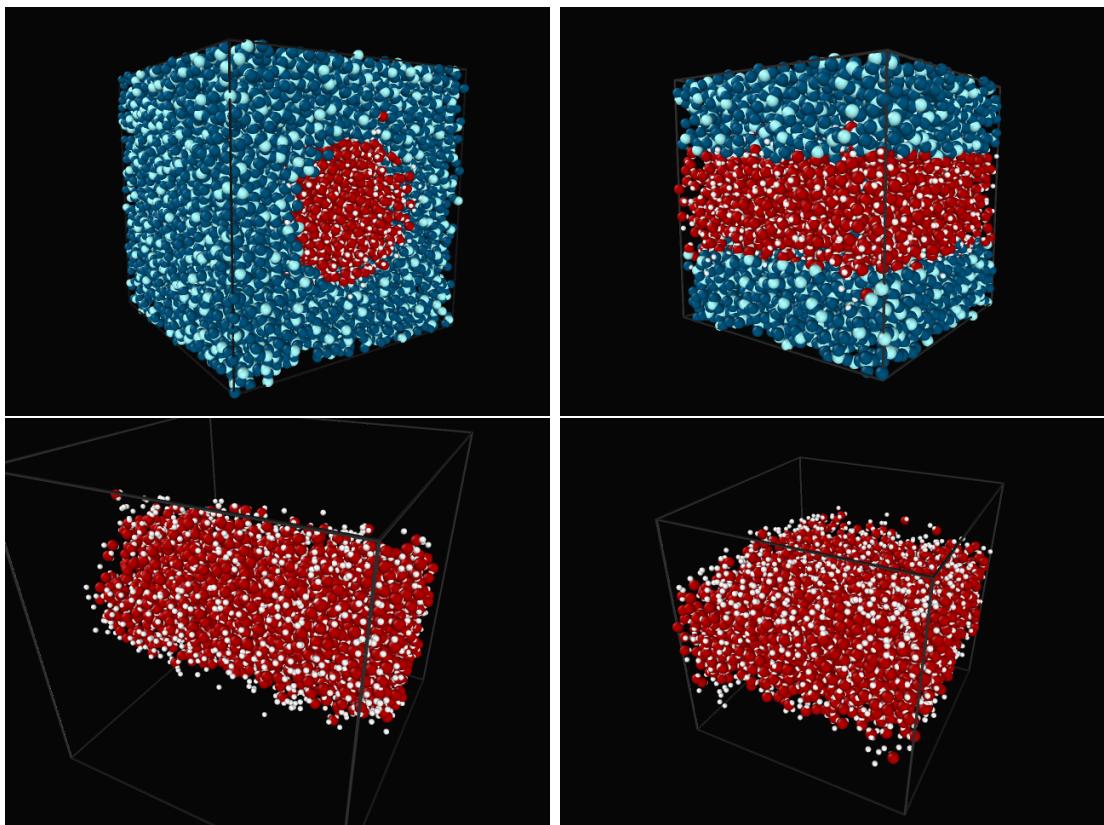


Figure 5.2: The content of the different pores with the same volume. The red atoms are oxygen and the white are hydrogen, which together makes water.

5.1.3 Bulk systems

In order to compare our pore systems with not-confined water we simulated bulk water, which is a periodic system consisting only of water molecules. In addition, we needed a sample of bulk silica in order to measure the radial distribution function, $g(r)$. The parameters of the two bulk systems are displayed in table 5.3, and the systems are visualized in figure 5.3.

Table 5.3: Table showing the size related properties of the systems with bulk water and bulk silica. L is the size of the cubic system with volume L^3 . *Number of* is denoted by $\#$.

Type	L [Å]	$\# \text{ H}_2\text{O}$	$\# \text{ SiO}_2$	# atoms	water density [kg/m ³]
Water	85.48	21952	-	65856	1051.6
Silica	114.56	-	32768	98304	-

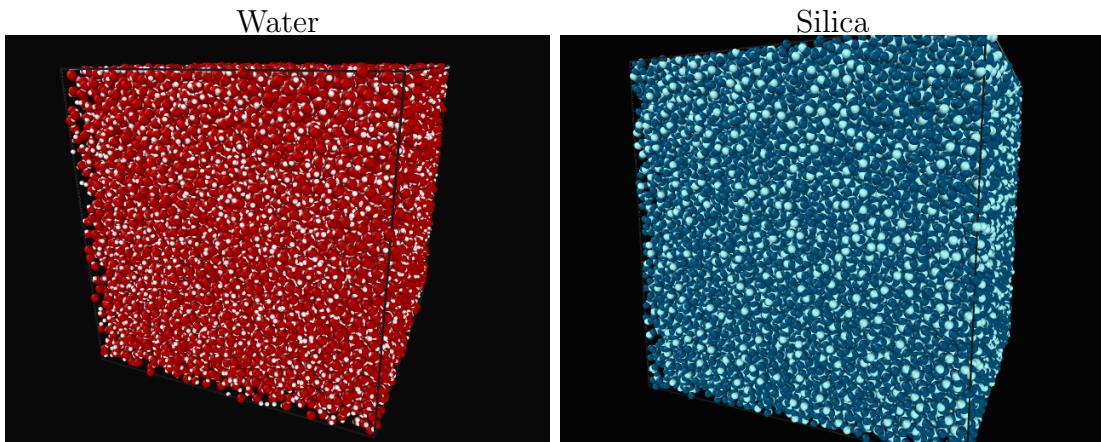


Figure 5.3: Bulk water and bulk silica.

5.2 System evolution

5.2.1 Pore evolution

This section is included because we were curious about how the size of the pores evolves right after the pore is created. We want the silica to behave like a solid material, which means that the pore size should not change much during the short simulation time, but we can not know for sure until we have tested it.

We measured the radius of the largest spherical pore, and we tested for four scenarios. We let the pore stay empty, meaning we did not inject the water, with and without temperature control prior to the measurement, or we injected the water as usual, with and without temperature control. The temperature is controlled using the Berendsen thermostat. The evolution of the radius of the empty pore is shown in figure 5.4, with and without temperature control prior to the measurement. We see that it is clearly shrinking, at approximately the same pace. The radius measured after the temperature control

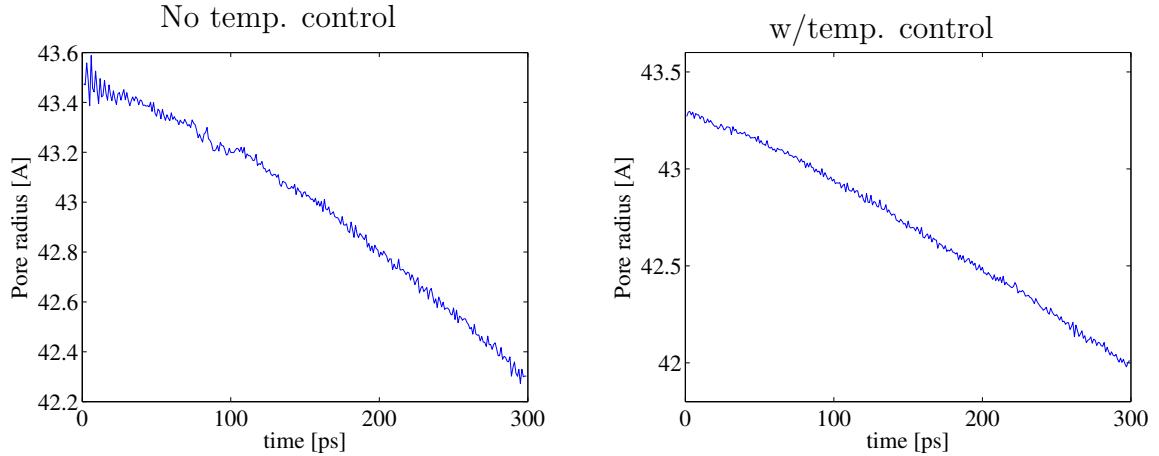


Figure 5.4: The evolution of the pore size of the largest spherical pore (14 nm system), if we did not inject water. Left: No temperature control was used. Right: The system was under temperature control (300 K) before measuring the pore radius.

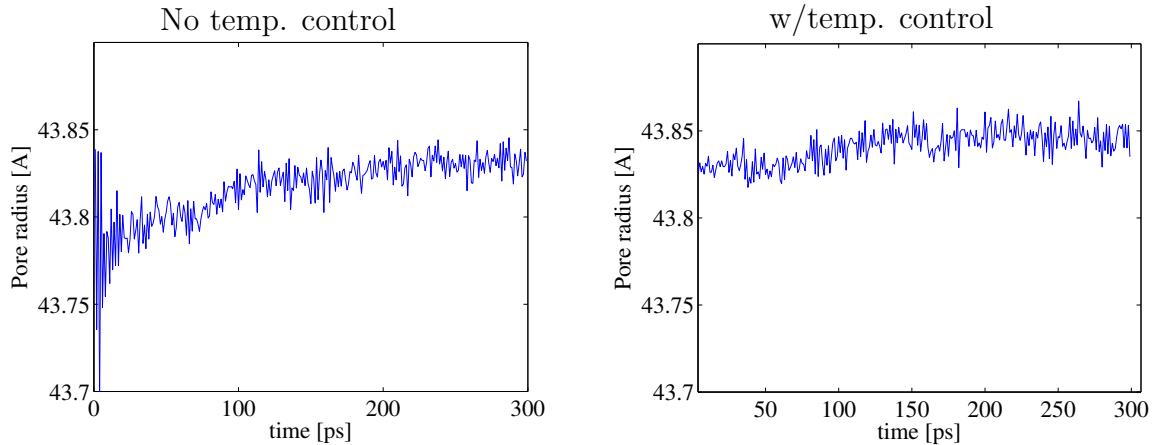


Figure 5.5: The evolution of the pore size of the largest spherical pore (14 nm system), when the pore contains water. Left: No temperature control was used. Right: The system was under temperature control (300 K) before measuring the pore radius.

reaches a lower value during the 300 ps, than the radius measured without any temperature control, but it also starts lower. The radius without temperature control decreases with approximately 1.2 Å, while the radius with temperature control decreases with approximately 1.3 Å, during 300 ps. The reason why the pores are shrinking, is tension in the glass making the glass expand and fill the pore space. The tension comes from the fact that we have created amorphous glass in the same volume and with the same number of atoms as the beta cristobalite we started with. When the atoms are in a glass state they are all slightly shifted away from the potential minimum in phase space that was the crystal structure. To relieve the tension, a real-life sample might expand and get a lower density than the crystal sample, but the MD system does not have that option, so the tension remains, until we cut out pores and the glass have space into which it can expand.

The evolution of the radius of the filled pores is shown in figure 5.5, with and without prior temperature control. For both cases, the radius is rather stable, it does not decrease

the way the radius of the empty pores did. In fact, the pore without temperature control seem to increase a little bit the first 120 ps.

5.2.2 Nanobubbles

During the preparation of the system with water between two slabs of silica (the plane pore), something peculiar happened. The mechanism for injecting water into the system was not optimal, and the water injected into the pore ended up with a lower density than anticipated. The system solved the problem by nucleating a nanobubble. The bubble can be seen in figure 5.6. The bubble is present for a long time, we found it was still there after 500 000 timesteps ($\Delta t = 0.5$ fs), which means that the system is in a stable state. In figure 5.7, the empty space of the system is filled with yellow 'particles', to visualize the shape of the bubble. Such nanobubbles (not necessarily empty, may be filled with gas) can cause

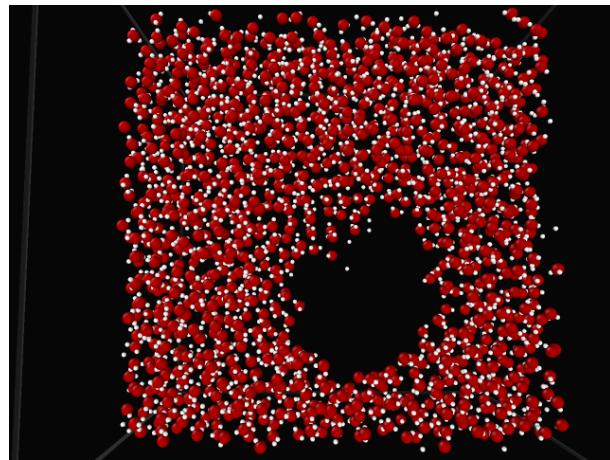


Figure 5.6: A nanobubble seen in a slice of the system with water between two slabs of silica.

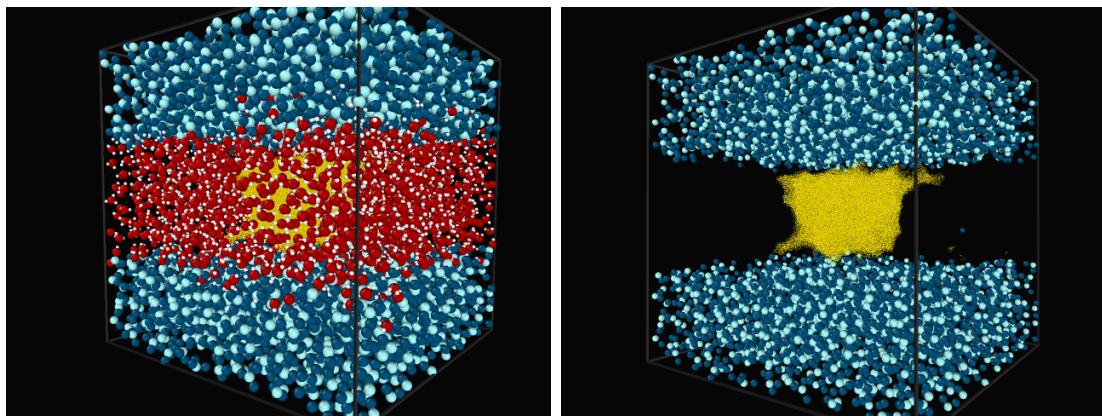


Figure 5.7: Shape of the nanobubble. Left: The entire system with the empty space in the nanobubble filled with tiny yellow dots. Right: The same system where the water molecules are made invisible.

damage in materials when they collapse (one can induce collapse in the MD system by

letting the bubble interact with a shock wave), which leads to disintegration of pipelines and ship propellers [28]. The field of medicine, on the other hand, takes advantage of the collapsing bubbles and uses them for targeted drug delivery[28][29] and extracorporeal shock wave lithotripsy[28][30].

Although this is an interesting case, it is not ideal for the purpose of this assignment. We want to investigate the effect the wall-curvature has on the water, and a big hole might ruin our results. This particular system will be left alone from now on, while systems with a little higher density will be the main focus.

6

Measurements

This chapter contains all of the results generated for this master thesis. We are going to look at measurements addressing the the structure and the dynamics of the water, and see if they are affected by the geometry of the pore. The measurements addressing the structure of the water are the density and the tetrahedral order parameter. To investigate the dynamics of the water we will measure the diffusion constant, the cage correlation function and life span. The methods for measuring the density and the diffusion constant were described in the chapter 3, because they are common to measure during a molecular dynamics simulation no matter the type of system. The other measurements are more specific for this master thesis, and they are described here.

6.1 Density

We choose a density for the water when generating a $\text{SiO}_2 + \text{H}_2\text{O}$ system. Still, we can not be sure that the density obtained is the density requested. This is due to the algorithm used for inserting water molecules to the system. The algorithm divides the system into equally sized voxels, and insert water molecules in the middle of each empty voxel. The problem is that the number of voxels has to be an integer, and to obtain this we have to adjust the size of the voxels which leads to a slightly different density than we ask for. As previously stated, there is a solution to this problem, but the solution was implemented after the simulations for this thesis were completed. The solution was to make the voxels smaller, and only insert the water molecules in an appropriate fraction of the empty voxels. This is one of the reasons why we want to measure the density.

Another more interesting reason is because we are curious to see how the density varies in the pore. We measure the local density using voronoi cells as described in section 3.7.2. The equation for the density with voronoi cells is

$$\rho_{\text{subset}} = \frac{N_{\text{subset}} m_{\text{particle}}}{\sum_{i=0}^{N_{\text{subset}}} v_{\text{polyhedron}}^i} \text{ for } i \in \text{subset}, \quad (6.1)$$

which is the same as equation 3.34 from section 3.7.2. N_{subset} is the number of particles in the subset, m_{particle} is the mass of one particle and $v_{\text{polyhedron}}^i$ is the volume of the voronoi cell surrounding particle i .

6.1.1 Water density as function of distance to the pore wall

We want to find the density of water as a function of distance to the pore wall. We use an algorithm outlined by eq. 6.1 where we let the subset be all the water molecules at a distance $d \pm \Delta d$ from the pore wall. The distance between a water molecule and the pore wall is defined as the distance between the center of said water molecule and the center of the closest silicon atom. When we find the center of the water molecules, we actually just find the center of the oxygen atoms in the water and assume that they are part of water molecules.

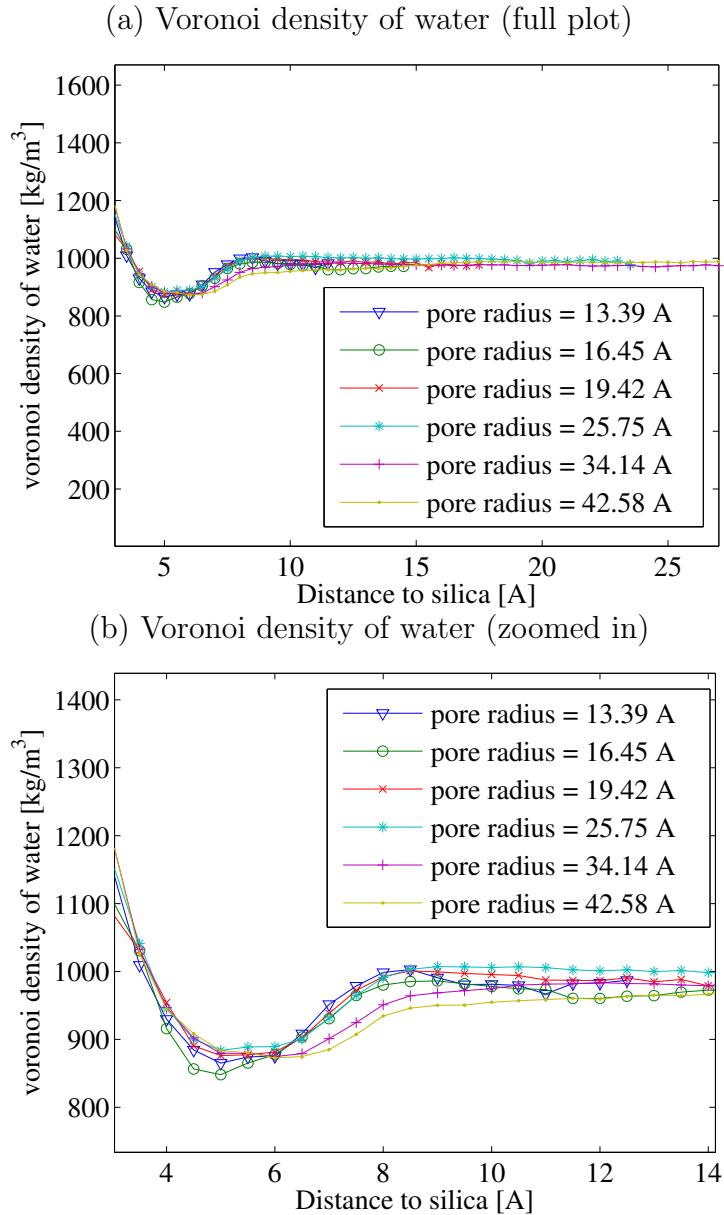


Figure 6.1: The density of water as a function of distance to the pore walls.

In figure 6.1 (a) and (b) we see the measured density at different distances for our six spherical pore systems. All of the density functions have the same shape: far away from the pore wall they all have roughly the same density as liquid water under normal

pressure, which is 1000kg/m^3 . When we get closer to the pore wall, the density drops and reaches a minimum at $d = 4.5 - 5.5 \text{ \AA}$ before it starts increasing again and reaches a higher value than the water far away from the wall. This value is between $\rho \approx 1100 - 1200 \text{ kg/m}^3$. Figure 6.1 (b) gives us a closer look on the of the density functions close to the silica. We see that the functions for the two largest pores grow slower after the minimum as the distances increase. The two smallest pores, $R = 12.89 \text{ \AA}$ and $R = 16.11 \text{ \AA}$, both have a maximum at approximately $d = 8 \text{ \AA}$ and decrease a little before they flatten out. The pore with $R = 25.776$ have the largest density overall. The different density functions in figure 6.1 are not identical, but the differences does not seem to follow any patterns according to the pore size or curvature. The functions in figure 6.1 are the results of averaging density functions from multiple states or snap-shots at different times, but the data for each function is collected from the same system. This means that the functions are dependent on the initial configuration of the system. To achieve more reliable results we should create multiple equally sized systems with different (randomly created) initial configuration of the water molecules, and average over measurements on all these systems. Unfortunately, the simulations are time consuming and expensive, and for this assignment we have only one system per pore size.

Compare the densities of different pore shapes. Plane - Cylinder - sphere

In figure 6.2 we see the density functions for the three different pore shapes, two planes, cylinder and sphere, all pores with approximately same volume. The plot on the left shows the actual density and the plot to the right shows the densities relative to the density in the center of the pore. We see the same density profile as for the spherical pores with higher density close to the pore wall, a minimum point around $d = 5 \text{ \AA}$ and at large distances it flattens out. The plane pore and the spherical pore have approximately the same density as liquid water under normal pressure far away from the pore wall. The cylindrical pore, on the other hand, stabilizes on a lower density, around 950 kg/m^3 . This is due to the average density of the water in the cylinder being lower than the density in the two other pore types because of the not optimal procedure for injecting water. In the plot showing the relative density we see more clearly that the density profiles are approximately equal for the three pore shapes, even though the average density in the systems differ.

The density profile

We have seen that all the density functions follow the same profile: a high value near the wall, than a dip before the function grow up to the flat part. Both Bonnaud et. al [31] and Milischuk and Ladanyi [32], who investigated structure and dynamics of computer simulated water confined in silica nanopores, found that the water forms two molecular layers at at the surface of the silica, which is represented in the density profiles as two peaks. The density profiles from the two articles are in appendix 8.3, in figure 8.1 and figure 8.2. The two peaks are within 5 \AA of the silica surface, but we have only measured for distances larger than 3 \AA , which may mean that we only see one of two peaks. Both articles show a small dip after the second peak, which is compatible with our results.

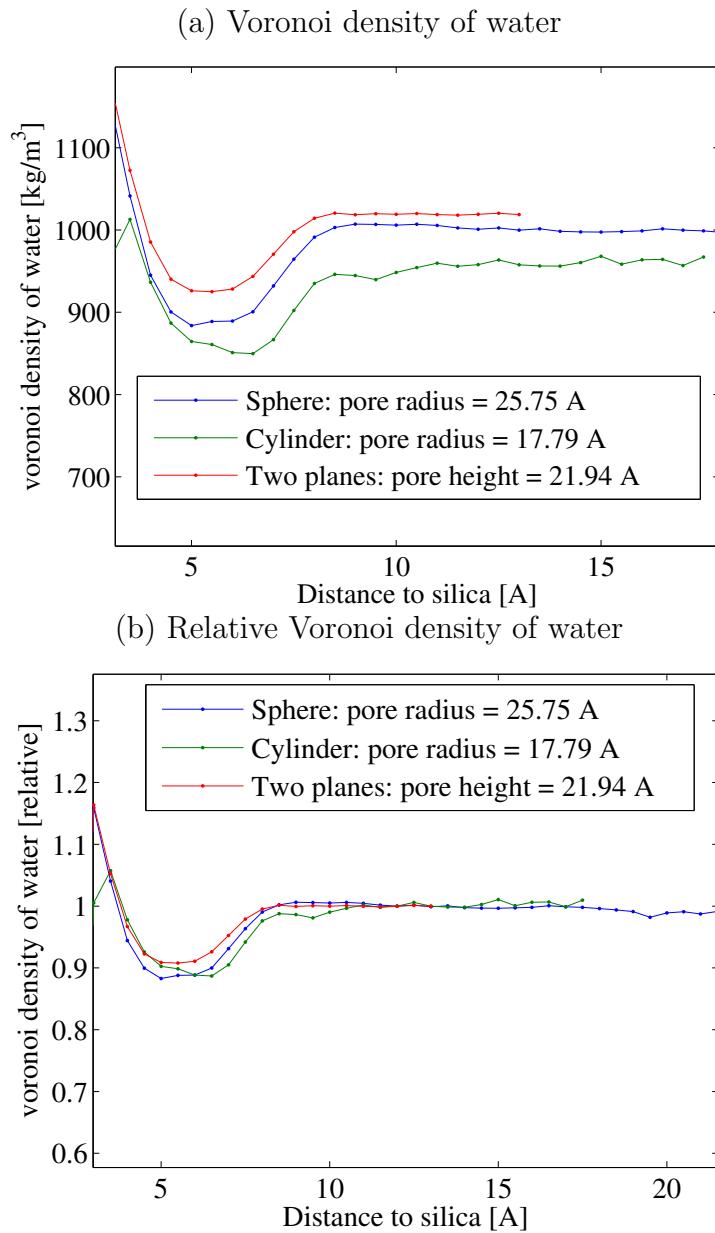


Figure 6.2: (a) Density as a function of distance to pore surface for three pore types.
 (b) Density as a function of distance to pore surface for three pore types, relative to the density in the center of the pore.

6.2 Tetrahedral order parameter - TOP

We want to investigate the local structure, or arrangement, of the water molecules inside the pores, and see how the structure depends on the distance to the pore wall. The water molecules in ice are organized in a tetrahedral manner [33][34], due to hydrogen bonds between the oxygen and hydrogen in neighboring water molecules. Figure 6.3 illustrates how each water molecule (ice) has four neighboring water molecules, making a tetrahedron surrounding the first water molecule. As the temperature is increased, and the ice converts to liquid water, the degree of tetrahedrality in the water decreases. Even though liquid water does not have the perfect tetrahedral structure that ice has, the structure has tetrahedral properties to some extent. We can measure the degree of tetrahedrality by assigning each water molecule with a *tetrahedral order parameter*. The tetrahedral order parameter (TOP) of molecule k can be written as

$$Q_k = 1 - \frac{3}{8} \sum_i^3 \sum_{j=i+1}^4 \left[\cos \psi_{ikj} + \frac{1}{3} \right]^2, \quad (6.2)$$

(from article [34]) where molecule i and j are the nearest neighbors of molecule k and ψ_{ikj} is the angle between the vector pointing from k to i and the vector pointing from k to j , illustrated in figure 6.3. The double sum uses all angles that can be made between

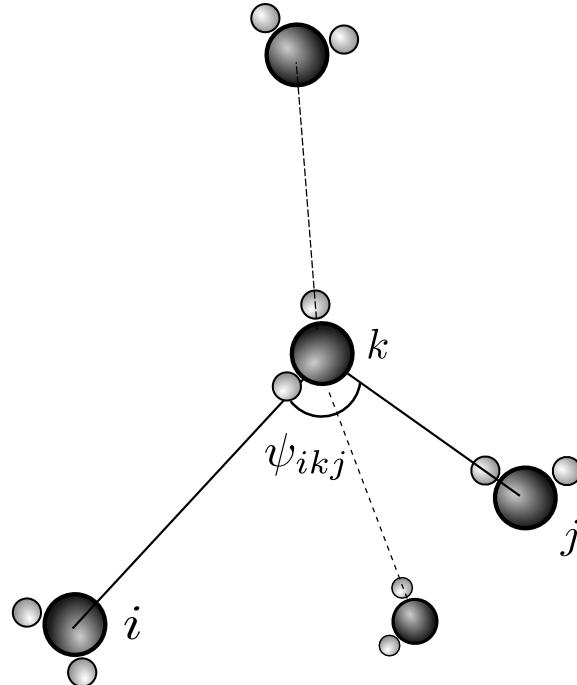


Figure 6.3: An illustration of the angle between three neighboring water molecules in a semi-tetrahedral structure.

the molecule in consideration and two of the four nearest neighbors. We notice that the maximum value of the TOP is 1 since the double sum is never negative. The TOP can be negative. When the neighboring water molecules form a perfect regular tetrahedron

(all angles are $\arccos(-1/3) \approx 109.47$), the TOP of the molecule in the center is 1. The more the positions of the neighboring molecules deviates from the vertices of a regular tetrahedron, the smaller the TOP.

Figure 6.4 shows an example of a probability density function, $P(Q, T)$, for TOP (denoted Q) at different temperatures. For the lower temperatures we have a peak between $Q = 0.8$ and $Q = 1$, and as the temperature increases the height of the peak reduces and another peak at a smaller Q -value emerges and the probability density function becomes bimodal.

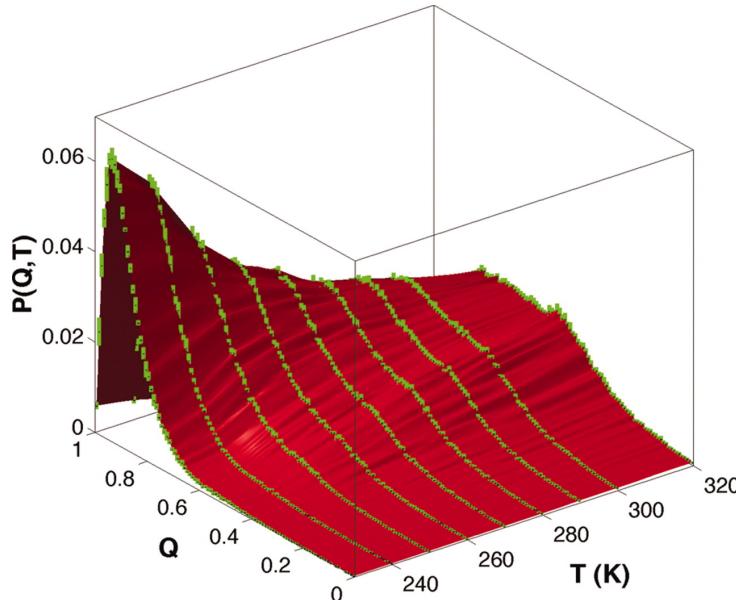


Figure 6.4: This is a figure collected from the article by Kumar et al. [34] where the TOP was defined. It shows the probability density function, $P(Q, T)$, for TOP (Q) in bulk water at different temperatures. The function $P(Q, T)$ is shown in red, while the green lines are errorbars. For high temperatures the TOP is bimodal, which means it has two peaks; two values of the TOP which has high likelihood of appearing in the water.

6.2.1 TOP in the pores

The tetrahedral order parameter was measured for the water molecules inside shells at different distances to the silica walls, for all six systems. By shell we mean all space at a distance $d \pm \Delta d$ from the pore surface. In figure 6.5 the average values for all the particles in a shell and histograms of the TOP of the spherical pores are plotted. The TOP histograms are the probability densities for different values of Q . We observe that the probability density has a bimodal shape, which is in agreement with $P(Q, T = 300)$ in figure 6.4. We also see that the shape of the probability density function is dependent on the distance to the pore wall for distances smaller than 8 Å. The average value of Q decreases as we approach the pore wall, which means that the tetrahedrality of the surface water is distorted compared to the bulk water. The water is affected by the SiO_2 structure close to the surface.

In figure 6.6 the TOPs of different pore shapes are compared. The probability densities for distance 10 Å are indistinguishable, while the probability densities for distance 4 Å

differ a little bit. The flat pore has a bit higher average TOP than than the spherical pore, and the spherical pore has a bit higher average TOP than the cylindrical pore, but these differences are very small, and could be due to differences in the density of the water.

It seems like the TOP is not affected by the curvature of the pore, but it do depend on the distance to the pore wall.

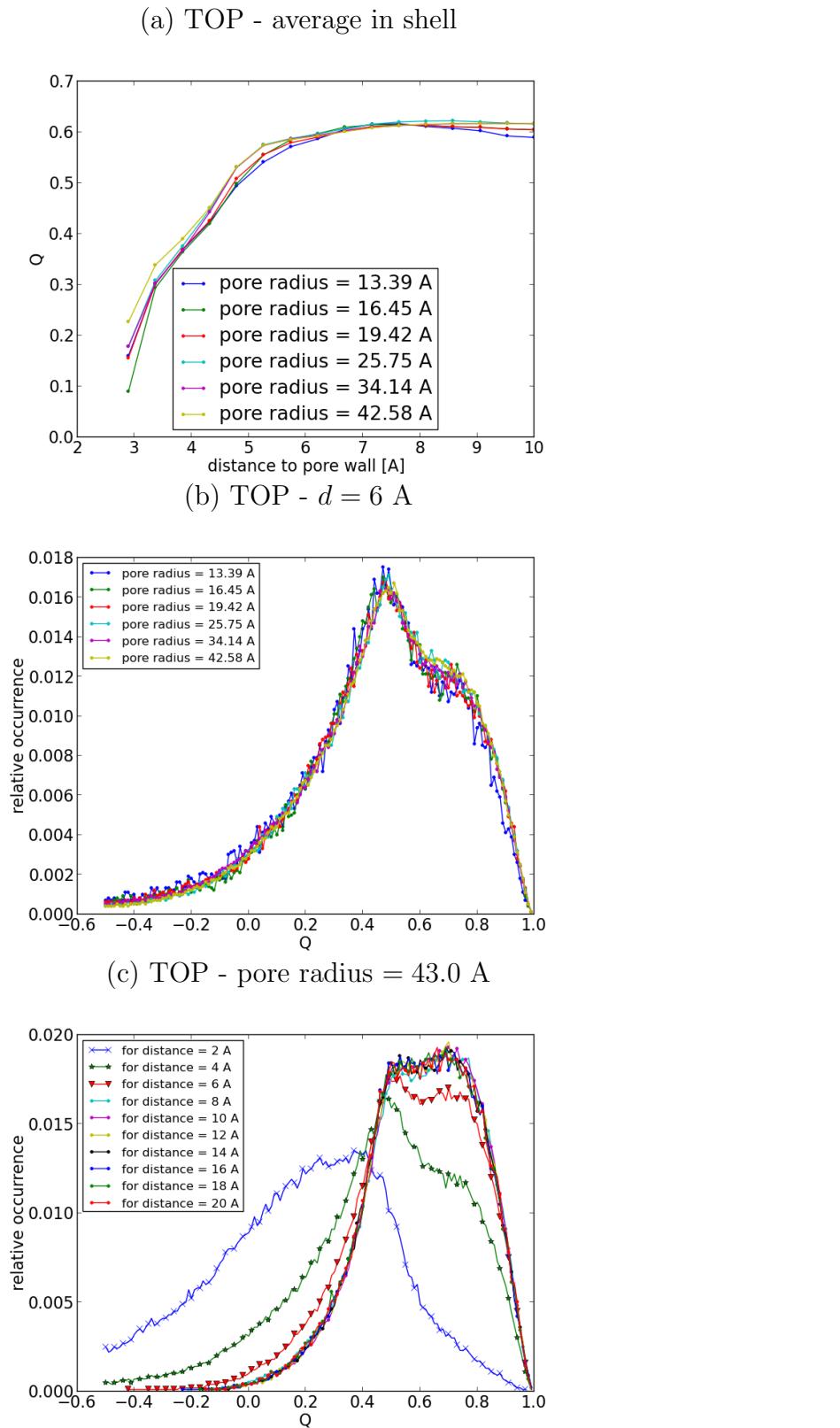


Figure 6.5: (a) The average value of the TOP as a function of distance to the pore wall. There is one plot for each system with a spherical pore.

(b) Probability density of the TOP for the water molecules in the distance 6 Å to the pore wall. There is one plot for each system with a spherical pore. Notice that the probability densities are basically identical, which is also the case for other distances.

(c) All probability densities generated for the largest spherical pore. Each histogram comes from the water molecules at different distances to the pore wall. For distances larger than 6 Å the functions are indistinguishable.

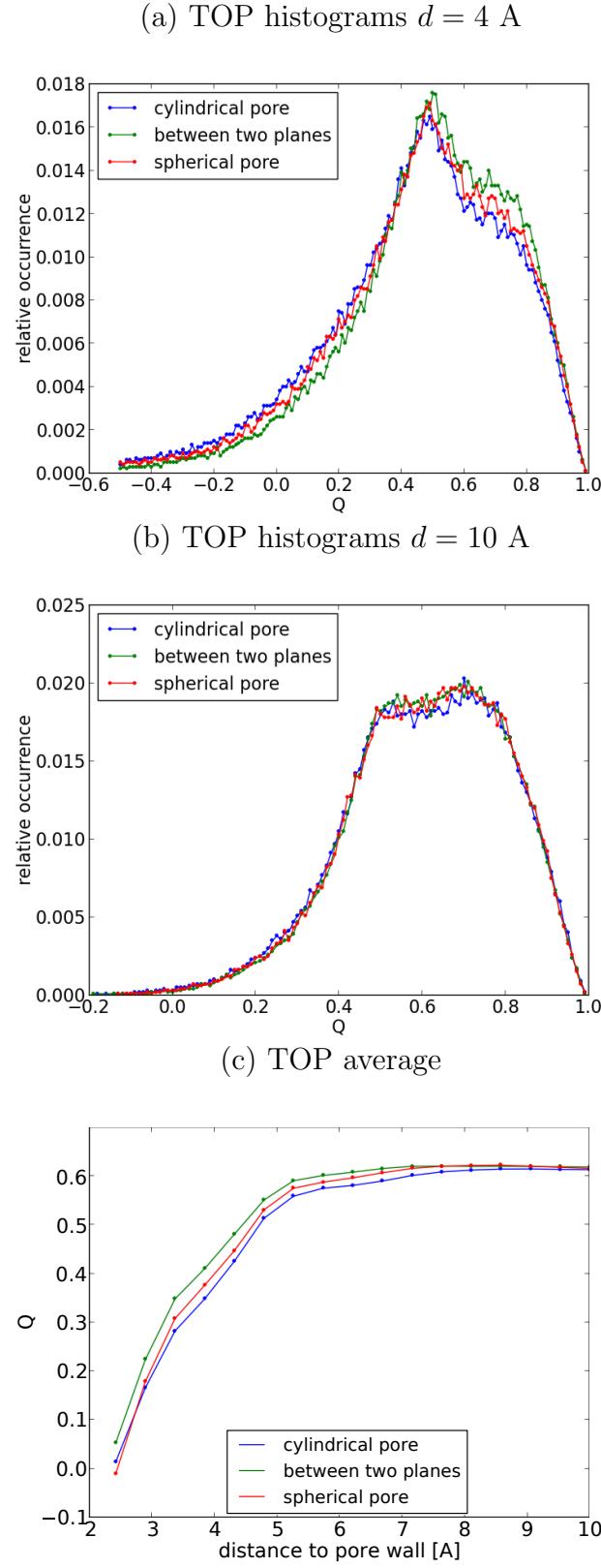


Figure 6.6: (a) The probability density of the TOP for molecules in the distance 4 \AA to the pore wall, for three different pore shapes.

(b) The probability density of the TOP for molecules in the distance 10 \AA to the pore wall, for three different pore shapes..

(c) The average TOP for three different pore types with the same volume.

6.3 Diffusion

We want to investigate the diffusion (or self-diffusion) of the water inside the pores. It will be interesting to see how the diffusion constant for the water inside the pores compares to the diffusion constant of bulk water. Since the water molecules have less space to move around in, it would be reasonable to expect that the diffusion constant would be smaller than for the bulk water. We will also investigate the effect the pore wall curvature has on the diffusion constant, and see how molecules at different distances from the pore wall contribute to the diffusion constant. We will measure the diffusion constant as a function of distance, d , to the pore wall for two different classifications of the water; surface water and shell water. First we divide the pore into surface and confined water, where the surface water is the water closer to the surface than the distance d . We measure the diffusion constant of the surface water for various distances d . Second we divide the pore into shells of equal width, at various distances d to the pore wall. In order to have many particles to average over, and at the same time get many measuring distances, we choose the width of the shells to be quite large ($\sim 4 \text{ \AA}$) and allow the shells to overlap.

Mean square displacement (MSD)

As described in section 3.7.3, we find the diffusion constant by measuring the mean square displacement of the water molecules. In that section we found that the mean square displacement would be proportional with time, if the particles acted as random walks. The proportionality can be written as

$$\langle r(t)^2 \rangle = 6Dt,$$

where D is the diffusion constant, t is the time difference between the current time t_{current} and the starting time t_0 , $t = t_{\text{current}} - t_0$, and the $r(t)$ is the distance between the current position of a particle and the position at t_0 .

When measuring the $\langle r(t)^2 \rangle$, we loop through all particles/molecules in consideration, and for each particle we find the distance from the particle's current position to the position it had at the starting time, t_0 . $\langle r(t)^2 \rangle$ is the average of all these distances

$$\langle r(t)^2 \rangle = \frac{1}{N} \sum_{i=1}^N (\vec{r}_i(t_{\text{current}}) - \vec{r}_i(t_0))^2,$$

where $\vec{r}_i(t_{\text{current}})$ is the position of particle i at time t_{current} , N is the number of particles and t_0 is the starting time. Note that if a particle moves through the periodic boundary, the position changes dramatically from one time step to the next because the particle reappears on the opposite side of the simulation box, while the true displacement was small. For the MSD calculations we want the small displacement, and for that reason we check if the change in position between time steps is larger than half the simulation box, $L/2$. If that is the case, we assume the particle have moved through the boundary and we use the position it would have had if the boundary was not periodic to calculate the MSD. We guess that the particle have moved through the boundary, but that is very likely if the position suddenly changes with half the system length. If it was so that the particles did move more than half a system length between sampling times, the data is sampled too rarely.

To get better statistics, we use a technique we call *to move the time origin*. This means that we repeat the measurement for multiple starting times, t_0 , and use the average of these measurements as our result. The technique is illustrated in figure 6.7, where Δt_{origin} is the time between the different time origins, and t_{MPEO} is the measuring period for each origin. We see from this figure that it is possible to let the measuring periods overlap. With $\Delta t_{\text{origin}} > t_{\text{MPEO}}$ the time periods would not overlap. All MSD measurements performed in this thesis is with $t_{\text{MPEO}} = 16$ ps and $\Delta t_{\text{origin}} = 5.5$ ps, which leads to overlapping time periods. We repeat the measurements 16 times, which means that the total period of measurement was $16 \cdot \Delta t_{\text{origin}} = 88$ ps.

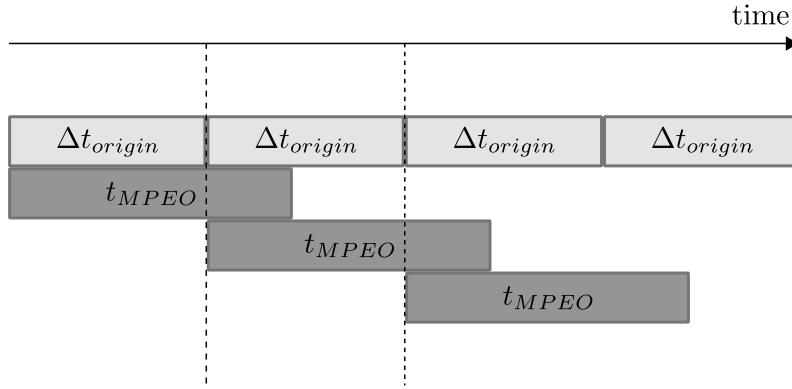


Figure 6.7: A diagram of the measuring periods when we move the time origin. Δt_{origin} is the time difference between each time origin, and t_{MPEO} is the measuring period for each time origin. We notice that the measuring period overlap.

MSD results

Figure 6.8 (a) shows the mean square displacement found for all the water inside the spherical pores. Based on visual inspection, the MSD looks rather linear in time if we ignore the first few points, so the results agree well with the random walk theory. Still, the linear functions do not go through the origin, which means that $\langle r(t)^2 \rangle$ is not proportional with time; there exists a small constant term $\langle r(t)^2 \rangle = 6Dt + b$. The reason why we need this small constant term b is that the MSD is not a linear function for short times, as we can see in figure 6.8. We do not expect to see diffusive behavior for short times, because this is where thermal vibrations and ballistic motion dominates over diffusive motion.

Because of the non-diffusive effects that occur for the short times, we skip the first few data points, and use the slope of the remaining data points when finding the diffusion constants. The slope is found by making a linear approximation, $y = at + b$, of all data points where $t \geq 5$. The linear approximation is found with the aid of matlab's polyfit[35]. We take the diffusion constant as $D = a/6$ where a is the slope of the linear approximation delivered by matlab.

We see in figure 6.8 (a) that the slopes of the MSD for the water inside the spherical pores increase with the radius of the pore, and the slope of the bulk water MSD exceeds them all. The diffusion constants corresponding to figure (a) is plotted as a function of pore radius in figure 6.8 (b). Here we see that the diffusion constant increases rather linearly if we exclude the end points.

In figure 6.8 (c) we see the MSD for the three systems with different pore shapes, but same volume. The pore shapes are spherical, cylindrical and plane (see section 5.1.2). We see that it is nearly impossible to tell the MSD functions for the spherical pore and for the plane pore apart. This indicates that the size of the pore has a larger effect on the MSD of the water, than the curvature, at least when *all* the water in the pore is included in the measurement. But, the MSD for the cylinder has a steeper slope than the two other types. This could be due to the density differences we saw in section 6.1 (fig. 6.2). Here, we saw that the average density of the water inside the cylinder was lower than the average density of the two other types.

Diffusion constants - different distances - surface water

The next step is to investigate how the different parts of the pore water contributes to the diffusion constant, or simply the diffusion constant in different parts of the pore. First we look at the water molecules near the pore wall, the surface water. The surface water is defined as all the water molecules closer to the pore wall than some distance d . For measuring the MSD this definition is a bit vague, because we have to measure over a time period, and the molecules do not simply stay at the same region throughout the entire measurement. We could say that we would only measure the molecules that stay within the range the entire measurement period, but that means excluding molecules that move at a high speed (they flee the region quickly), and only measuring on the slow molecules leading to a lower diffusion constant. If we choose a long measurement period, we could end up with no molecules to measure, because all molecules had enough time to escape the region. Another option is to only let the molecules contribute while they are within the region, but that would lead to favoritism of the molecules moving parallel to the pore wall, because they typically stay inside the region longer than the molecules moving in the perpendicular direction. Also, the shape of the pore wall would affect the result. For example, the area of the surface the molecules can escape through would be dependent on the distance d for the spherical pore while it would not be dependent of the distance for the plane pore. This needs to be taken in consideration when interpreting the results of such measurements. We ended up measuring the molecules that are within the region at time t_0 . We do not require that they stay in the region, we just follow them where ever they may go. You could say that we choose a set of *molecules* and see how *they* contribute to the diffusion constant, instead of saying that we choose a *region* and see how *it* contributes to the diffusion constant. If we only measure for a short period of time, these two interpretations are approximately equal.

Figure 6.9 shows the measured diffusion constant as a function of distance to the wall, d , for the spherical pores and for the three pore types. All diffusion constant functions in figure 6.9 have the same shape. They grow steeply from about $d = 4 \text{ \AA}$, and then the growth decelerate. The deceleration seem to be dependent on the radius of the pore for the spherical pores (figure 6.9 (a)), at least for the pores with radius smaller than $R = 34.14 \text{ \AA}$. The diffusion constants of the two largest pores are approximately equal, but differ a bit when the distance d reaches the second larges pores radius.

The diffusion constant functions for the cylindrical pore and the plane pore in figure 6.9 (b), lie approximately on top of each other and reach higher values than the function for the spherical pore. Even though this seems to be conflicting with the MSD-functions in 6.8, where the cylindrical pore have a larger diffusion constant than the two other

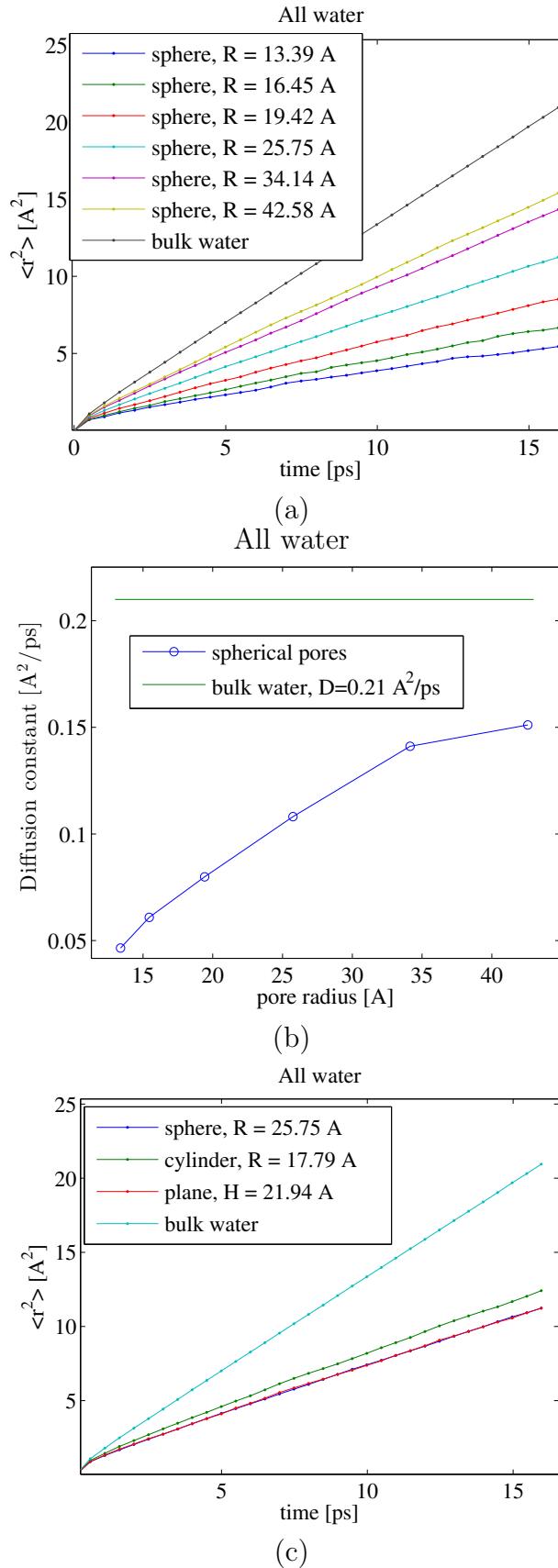


Figure 6.8: Diffusion in the spherical pores.

- (a): $\langle r^2 \rangle(t)$ averaged over all water molecules in the spherical pores, plus bulk water.
- (b): The diffusion constants from the MSD functions in (a), as a function of the radius of the pore.
- (c): $\langle r^2 \rangle(t)$ averaged over all water in the three different pore types with the same volume.

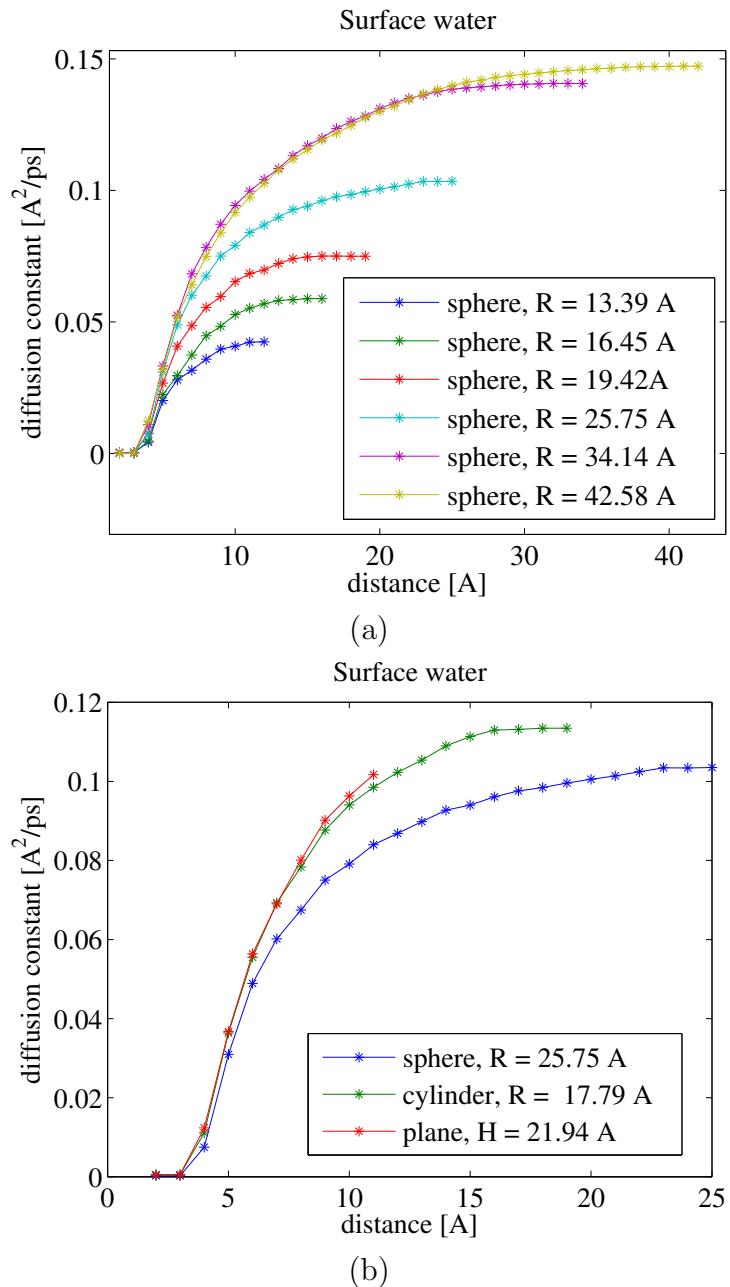


Figure 6.9: Diffusion in the spherical pores.

(a): The diffusion constant for the water molecules within a distance d from the pore wall, the surface molecules, for the spherical pores.

(b): The diffusion constant for the water molecules within a distance d from the pore wall, the surface molecules, for the three different pore types of the same volume.

shapes, don't worry, there is no conflict. The MSD-functions in 6.8 were measured for *all* the water inside the pore, which corresponds to the *last* measuring point in figure 6.9 (last, meaning the highest d -value). If we compare the last points for the three functions in figure 6.9(b), we see that the sphere and the plane land on approximately the same value, while the cylinder lands a little bit above. If we look at the diffusion constant function found with the surface water (fig. 6.9 (b)) for the plane pore, it looks like it could grow further if the distance between the two pore walls were larger. If that was the case, the slopes of the MSD function in figure 6.8 (c) would be largest for the plane, second largest for the cylinder and smallest for the sphere. That would fit better with our intuition, because molecules close to a curved surface should intuitively be more trapped, and therefore move less, than molecules close to a plane surface.

It is important to remember that we have used all molecules within the distance d of the pore wall, not the molecules *at* distance d . This means that all the molecules contributing to the value for distance $d = 15 \text{ \AA}$, also contribute to the value for distance $d = 20 \text{ \AA}$. When the distance d reaches the pore radius, we have included all the water molecules in the pore. When trying to interpret the results we see in figure 6.9 it is important to have in mind that the results are affected by the shape of the pore. When we increase the distance d , it is not guaranteed that the number of contributing molecules is increased linearly. Linearity only happens in the plane pore. If n_d is the number of molecules contributing to the value at d , then the difference $\Delta n_{5 \text{ to } 6} = n_6 - n_5$ is the same as the difference $\Delta n_{12 \text{ to } 13} = n_{13} - n_{12}$. This means that for a given d , all molecules at inboard distances contribute equally to the MSD. For the spherical pore, this is not the case. When we start measuring for the current distance d_{current} we use the same molecules as used for measuring at the previous distance d_{prev} and add the molecules that lie within a spherical shell with width $\Delta d = d_{\text{current}} - d_{\text{prev}}$. The width Δd is constant and independent of the distance d . The spherical shell added at $d = 5 \text{ \AA}$ has a larger volume, and therefore more molecules, than the spherical shell added at $d = 10 \text{ \AA}$. This means that there are more molecules contributing to the MSD for the smaller distances than for the larger distances. The trend is that the diffusion constant for shorter distances is smaller than for larger distances, due to the hydrophilic surface. We believe this is the main reason why the diffusion constant of the surface water for the spherical pore in figure 6.9 (b) reach a smaller value than for the two other pore types; the molecules at shorter distances, that contribute with small diffusion constants, outnumber the molecules at longer distances, which means the smaller diffusion constants are weighted heavier than the larger ones.

Shell water

Another way of getting the diffusion constant as a function of distance to the pore wall, is to measure the MSD for the molecules inside shells where d is the distance from the pore wall to the middle of the shell. We let the shell have a constant width, l , which is the same for all distances d . In order to have enough particles to average over, and at the same time get many measuring distances, we choose the width of the shells to be quite large ($l \sim 4 \text{ \AA}$) and allow the shells to overlap.

For the spherical pores, we get a problem similar to the problem for the surface water discussed above. The volume of the shells decrease as the distance to the pore wall increases. Still, the consequence is not as severe as for the surface water, as it will not

give interpretation issues but lead to poorer statistics because the number of molecules contributing to the result decreases.

Again, we choose to measure the molecules that are present inside the region of our choice (the shell) at the first time step, and follow these molecules the entire measurement period even though they move out of our region. Figure 6.10 shows the diffusion constant as a function of the distance to the pore wall, for the different pore shapes and sizes. We see the same trend as in figure 6.8, where the diffusion constants grow rapidly for the first few Angstroms, and they decelerate. The functions that actually do flat out for the spherical pores (fig. (a)), flattens out on more similar values than in figure 6.9 (a). They also go higher, which is expected since the values at the larger distances are not dragged down by the values at smaller distances. In figure 6.8 (b) we see that the diffusion constant of bulk water is approximately $D_{\text{bulk}} = 0.21 \text{ A}^2/\text{ps}$. In figure 6.10 (a) we see that the second largest pore reaches values higher than that, the highest being $D = 0.25 \text{ A}^2/\text{ps}$. The largest pore's values flattens out at approximately the bulk value. It reaches the bulk value at $d = 15 - 20 \text{ \AA}$.

For diffusion constants found with the shell water, the differences between the functions in figure 6.10 (b) are smaller than for the surface water in figure 6.9. We see that the cylinder has the highest diffusion constant for distances larger than 10 \AA (approximately bulk value), while for distances smaller than 7 \AA , the functions look almost identical. The plane pore is unfortunately a bit too small, so we can not say anything about diffusion at distances larger than 10 \AA for plane pores.

It appears that the diffusion in the water closer than 10 \AA of the pore wall is affected by the presence of the wall. At distances larger than 10 \AA the water diffusion is similar to the diffusion of bulk water. At distances between $5-10 \text{ \AA}$ can we see a slight difference in the development of the of the diffusion constant. The diffusion constant for the ones with larger radius, and smaller curvature, grow more quickly than for the smaller pores, larger curvature. At the end of the functions we see some weird effects. For example, the diffusion constant for the spherical pore with $R = 19.33 \text{ \AA}$ start to drop at $d = 14 \text{ \AA}$, while the diffusion constant for the spherical pore with $R = 25.78 \text{ \AA}$ has a growth spurt after $d = 20 \text{ \AA}$. This may be because there are few molecules in the shell, which gives us poor statistics, or it could be local irregularities in the different pores. To get a more general picture of the diffusion in pores such as ours, we should simulate more systems with the same parameters of the six systems we already have (size, temperature etc.), but with different initial configurations of the water molecules inside the pore, and measure all of them. When we only have one version of each system, it is hard to tell if the results apply to a generic system, or just this system in particular.

6.3.1 Model for the diffusion constant

The data from the shell water suggest that the diffusion constant is a function of distance to the pore wall, $D(d)$, regardless of the geometry of pore. A crude model for the behavior of the diffusion constant could be a twofold function, where the model is a linear function for small distances and a constant function for larger distance, for example

$$D(d) = \begin{cases} d \frac{D_\infty}{A_0} & \text{if } d \leq A_0, \\ D_\infty & \text{if } d > A_0, \end{cases} \quad (6.3)$$

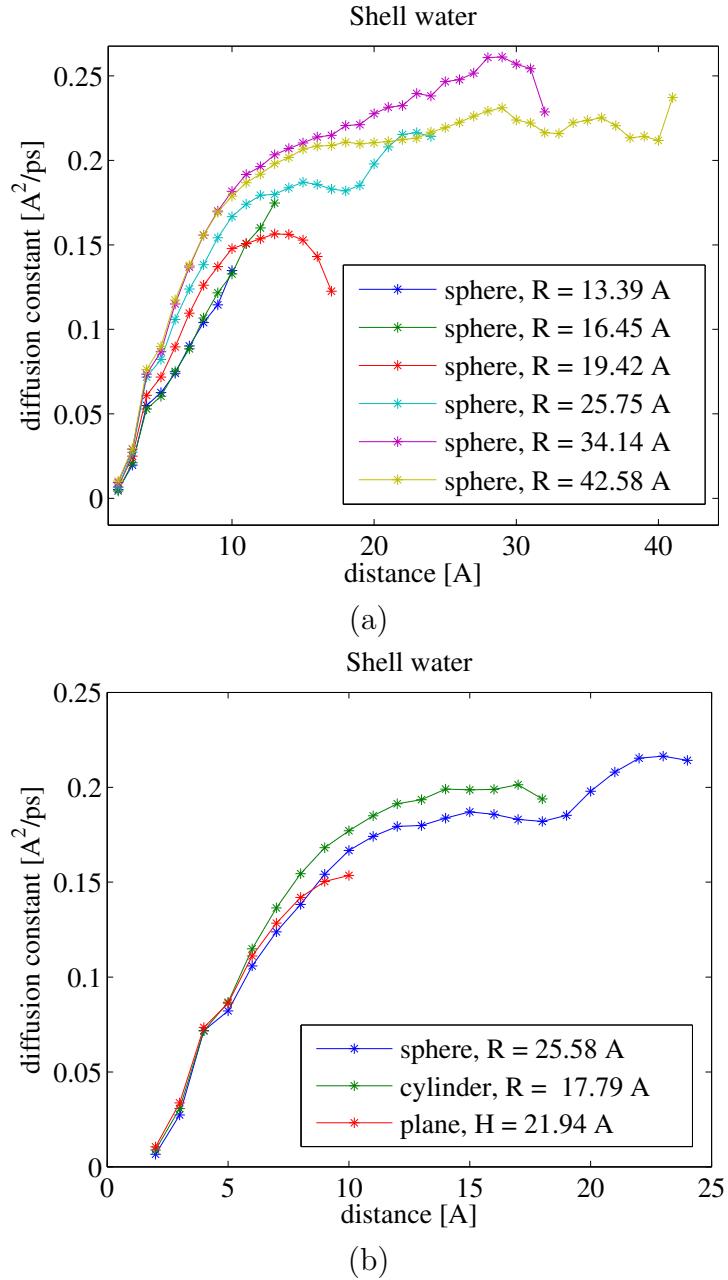


Figure 6.10: Diffusion in the spherical pores.

- (a): The diffusion constant for the water molecules inside a shell at distance d from the pore wall, for the spherical pores.
- (b): The diffusion constant for the water molecules inside a shell at distance d from the pore wall, for the three different pore types of the same volume.

where A_0 is the distance where the model changes from ascending linear to constant and D_∞ is the diffusion constant far away from the pore wall. Figure 6.11 shows the model on top of the measured diffusion constants for the three pore shape, where we have chosen $A_0 = 10 \text{ \AA}$ and $D_\infty = 0.185$. Since it seems that the diffusion is only dependent on the distance d , we can use such a model to find approximated values for the diffusion constant in a system with generic pore geometry, for any point where we know the distance to the pore wall.

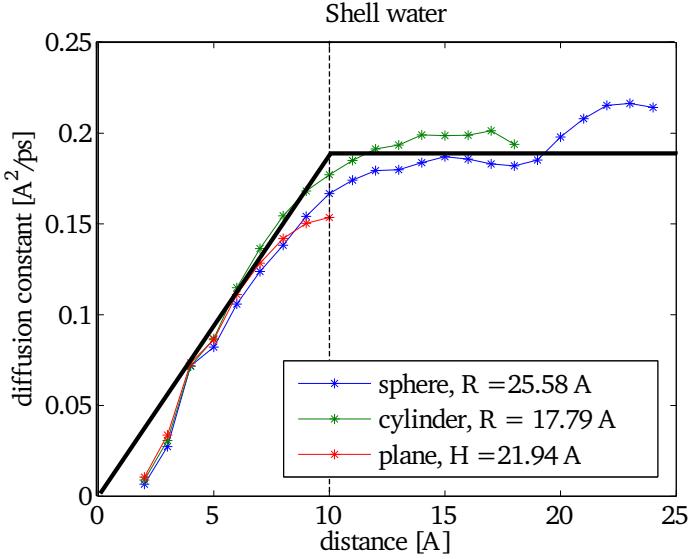


Figure 6.11: The diffusion constant for the three pore shapes, together with the crude model, where $A_0 = 10 \text{ \AA}$ and $D_\infty = 0.185$.

The model in eq. 6.3 is a very crude representation of the measured data, but it is easy to use and all we need to provide is the distance, A_0 , and the value of the diffusion constant at long distances, D_∞ . Another suggestion that is less crude, but more sensitive and harder to handle, is the logistic growth model. This function is often used to describe population growth, but is present here just because we found similarities between the shape of the function and the shape of the measured data. The function is

$$D(d) = \frac{D_\infty}{1 + (\frac{D_\infty}{D_0} - 1)e^{-rd}}, \quad (6.4)$$

where D_∞ is the diffusion constant far away from the pore wall, r is a parameter deciding how fast the function grows and D_0 is the value of the diffusion constant at $d = 0$. We do not know what the parameters, r and D_0 , should be, so we perform a curvefit and find that the values $r \approx 0.5 - 0.6$ and $D_0 \approx 0.008 - 0.1$ were appropriate to fit the measured data. The model is very sensitive to the choice of these parameters, which makes it harder to use than the crude model. Figure 6.12 shows the model on top of the measured diffusion constants for the three pore shapes, where $D_\infty = 0.197$, $D_0 = 0.0085$ and $r = 0.56$.

Let's see if we can reproduce the functions displayed in figure 6.9 (a), using these two models. The functions we want to reproduce are the diffusion constants in surface water as a function of distance d for the spherical pores. To calculate the diffusion constant at a given distance, d_{surf} , we have to sum up all the diffusion constants for all the inboard

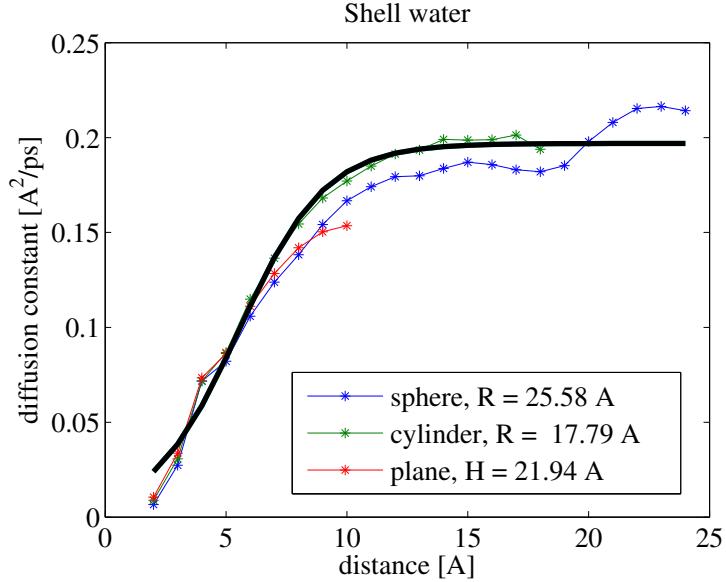


Figure 6.12: The diffusion constant for the three pore shapes, together with the logistic model, where $D_\infty = 0.197$, $D_0 = 0.0085$ and $r = 0.56$.

distances, and we have to weigh them according to the volume of their shell and of course divide by the total volume. When we let the shells become infinitesimally thin, this sum becomes the integral

$$D_{\text{surf}}(d_{\text{surf}}) = \frac{1}{V_{\text{tot}}} \int_0^{d_{\text{surf}}} D_{\text{model}}(d) A_{\text{shell}}(R_{\text{pore}} - d) \delta d, \quad (6.5)$$

where $A_{\text{shell}}(R_{\text{pore}} - d) \delta d$ is the volume of the infinitesimally thin shell with radius $R_{\text{pore}} - d$, R_{pore} is the radius of the pore, V_{tot} is the volume of the spherical shell with outer radius R and inner radius $R - d_{\text{surf}}$ and $D_{\text{model}}(d)$ is the diffusion constant at distance d from the pore wall given by the model. We integrated numerically for various d_{surf} , and all our spherical pore sizes, for both models. The result is in figure 6.13. The models reproduce the shape of the functions very well. We notice that the crude model creates higher values than the logistic model, and if we compare the functions for the smallest radius with the measured function in figure 6.9, we see that the values of the crude model is a bit too high and the values of the logistic model are quite close.

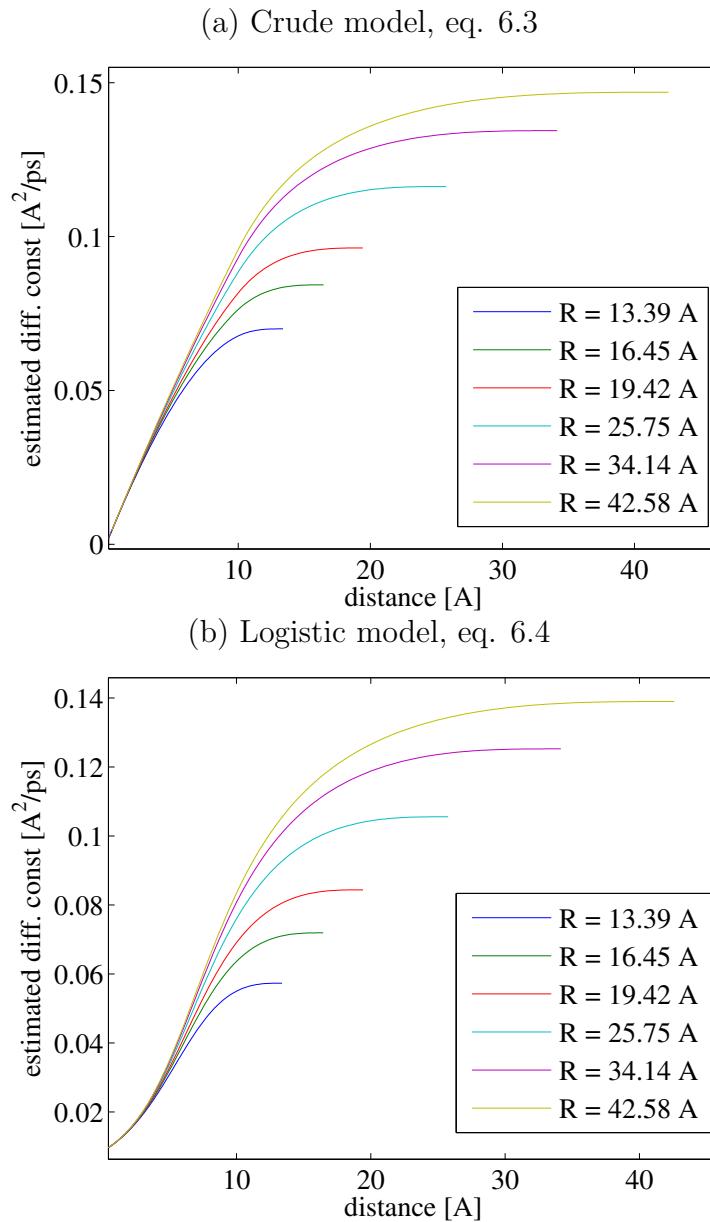


Figure 6.13: Reproduction of the result in figure 6.9 (a), with the two models. To create figure (a) we used $D_\infty = 0.2 \text{ A}^2/\text{ps}$ and $A_0 = 10 \text{ \AA}$. To create figure (b) we used $D_\infty = 0.2 \text{ A}^2/\text{ps}$, $D_0 = 0.0085 \text{ A}^2/\text{ps}$ and $r = 0.56$.

Compare results

Milischuk and Ladanyi [32] performed an analysis similar to ours, where they simulated water confined in three cylindrical nanopores with varying sizes. Like we did, they also found that the diffusion constant decreased when approaching the silica from the center of the pore, and that the dependence of pore size was weak. Their results on the diffusion can be found in appendix 8.3, figure 8.3 and figure 8.4. Unlike us, they found two diffusion constants, D_{\perp} and D_{\parallel} , which are the diffusion constants in the radial and axial directions of the cylinder, respectively. For future studies, it could be interesting to investigate these two constants for our eight pore systems, and compare them for the different pore shapes. We would expect to find D_{\parallel} smaller for the spherical pores than for the plane pore.

The value of their diffusion constants are somewhat higher than ours even though we have approximately the same temperature, for instance they find the the axial diffusion constant far away from the pore wall in a 40 Å pore to be 0.3 Å²/ps, while our bulk value is 0.21 Å²/ps and our highest measured value is 0.25 Å²/ps. They also had non-diffusive effects for short times similar to ours.

6.4 Cage correlation function

6.4.1 Defining the cage correlation function

The goal of this thesis is to determine the effects the curvature of the pore walls has on the water confined in the pore. Essentially see if some properties of the water are dependent on the curvature of the pore. One property we want to investigate is the cage correlation function.

We want to investigate some diffusive properties of the water, because we think that the diffusion process is affected by the shape of the pore. What a water molecule experiences when it diffuses through the pore, is how the surroundings changes. By the surroundings we mean the particles that exists in the immediate neighborhood, the neighbors. The cage correlation function gives a measure of the changes in the neighborhood for an arbitrary particle. We grant each particle (or water molecule) with an integer between zero and the number of particles in the system minus one ($N - 1$), and we remember this marking throughout the simulation. The concepts in this section are from the article where the cage correlation function was first introduced by Rabani et al. [36].

Definition 6.4.1. We define the neighbor vector ($L_i(t)$) for particle i at time t (as in [36]) as a vector of size N (number of particles) where each element is defined by

$$L_i(t)[j] = \begin{cases} 1, & \text{particle } j \text{ and } i \text{ are neighbors at time } t, \\ 0, & \text{otherwise.} \end{cases}$$

Example 6.4.2. Imagine a system consisting of three particles: 0, 1 and 2, where (at a time t) particle 0 and 1 are neighbors, and 0 and 2 are neighbors, but particle 1 and 2 are not neighbors. The three neighbor vectors of the system is then

$$L_0(t) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, L_1(t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, L_2(t) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (6.6)$$

Aided by the neighbor vector we can calculate the number of particles that have left particle i 's neighborhood after a time t' , and the number of particles that have entered the neighborhood. We find the neighbor vector at a wanted start time step, and call it $L_i(t = 0)$, and evolve the system a time t' and find the neighbor vector for this time step, $L_i(t')$. The number of atoms that have left the neighborhood of particle i during the time t' can be found by

$$n_i^{out}(0, t') = |L_i(0)|^2 - L_i(0) \cdot L_i(t').$$

The first term $|L_i(0)|^2$ gives us the number of neighbors for particle i at the start time step (finding $|L_i(0)|^2$ is essentially just counting the number of elements with value 1 in $L_i(0)$, i.e. counting the neighbors). The last term, $L_i(0) \cdot L_i(t')$, is the number of neighbors that did not move out of the neighborhood of particle i . We can find the number of particles that have entered the neighborhood in the same manner by

$$n_i^{in}(0, t') = |L_i(t')|^2 - L_i(0) \cdot L_i(t'),$$

where $|L_i(t')|^2$ is the number of neighbors for particle i at the time t' .

We define a variable c to be the number of particles that has to leave or enter a neighborhood in order for us to call it a change of surroundings. Now, there exists two definitions of the cage correlation function:

Definition 6.4.3. The first definition takes into account both the particles leaving the neighborhood, and the particles entering.

$$C_{cage}^{in-out}(t) = \langle \Theta(c - n_i^{out}(0, t)) \Theta(c - n_i^{out}(0, t)) \rangle \quad (6.7)$$

where Θ is the unit step function

$$\Theta(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases} \quad (6.8)$$

Definition 6.4.4. The second definition is only concerned with the particles leaving the neighborhood.

$$C_{cage}^{out}(t) = \langle \Theta(c - n_i^{out}(0, t)) \rangle \quad (6.9)$$

According to [36] the second form prevents us from counting too many of the vibrational motions. We therefore choose to use the second form, definition 6.4.4.

Computation techniques

Even though the mathematical definition of the cage correlation function (def. 6.4.4) is elegant, we can quickly get in trouble and lack memory for large systems due to long neighbor vectors. The neighbor vectors are just long lists of mostly zeros and few ones, so we can store the same information with less bytes by storing the indices where $L_i(t) = 1$, let's call it the reduced neighbor vector $L_i^r(t)$.

Example 6.4.5. The reduced neighbor vector in example 6.4.2 is

$$L_0^r(t) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, L_1^r(t) = \begin{bmatrix} 0 \end{bmatrix}, L_2^r(t) = \begin{bmatrix} 0 \end{bmatrix}. \quad (6.10)$$

We need to figure out how to compute n_i^{out} from the reduced neighbor vectors. We saw earlier that

$$\begin{aligned} n_i^{out} &= \text{number of neighbors at } t = 0 \\ &\quad - \text{number of particles that are neighbors} \\ &\quad \text{of particle } i \text{ at both } t = 0 \text{ and } t = t. \end{aligned}$$

The first term is easy to find, it is just the number of elements of the reduced neighbor vector length($L_i(0)^r$). The second term can be found by first finding the intersection of the two neighbor vectors, and then counting the number of elements of the result, length($L_i(0)^r \cap L_i(t)^r$). We have found our strategy in computing n_i^{out} (and hence the cage correlation function def. 6.4.4) with the reduced neighbor vector,

$$n_i^{out} = \text{length}(L_i(0)^r) - \text{length}(L_i(0)^r \cap L_i(t)^r).$$

Stretched exponential function

As we will soon see, the cage correlation function seem to follow the stretched exponential function (the abbreviation SER is often used, where the R stands for relaxation). We are going to fit SER functions to all of our cage correlation results, so it is therefore a good idea to get to know the function first. We will first define the function, and then take a short review of the SER functions appearances through history, before we discuss the shape of the function and the meaning of the parameters.

Definition 6.4.6. The stretched exponential function is

$$S(t; \beta, \tau) = \exp \left[-\left(\frac{t}{\tau} \right)^\beta \right], \quad (6.11)$$

where t is time, τ is the mean lifetime and β is the stretching exponent.

The stretched exponential function is often referred to as the Kohlrausch relaxation function. Rudolph Kohlrausch (1809-1858) introduced the function in 1854 when he studied the discharge of the *Leyden jar* capacitor [37], and saw that the usual description of relaxation, the exponential, did not fit his results unless he included a power β . Berberan-Santos, Bodunov and Valeur state in “*History of the Kohlrausch (stretched exponential) function: Focus on uncited pioneering work in luminescence*” [38] that the second documented use of the Kohlrausch function was by Werner in 1907, who used it to describe short-time luminescence decay of an inorganic phosphor [39]. In 1969 G. Williams and D.C. Watts re-introduced the Kohlrausch as the *empirical decay function* [40]. The article William and Watts presented in 1969 has been very popular, with 2143 citations (anno October 2007) [41]. This popularity is due to the fact that many scientists today get results that show good fit with the stretched exponential function. One example is in the Grassberger and Procaccia’s article about diffusion in a medium with randomly distributed static traps [42]. In this article the diffusive particles can be absorbed by static particles, the traps. They argue that the density of the diffusive particles decays slower than the exponential, actually it decays as the SER, $\exp(-t^{d/(d+2)})$, where d is the dimensions of the system.

Mostly, the Kohlrausch function is used as an empirical tool: it is useful to describe phenomena in studies of relaxation in complex systems, but why we see such phenomena is not yet fully understood. M.D Ediger explains in his review paper on Supercooled liquids [43] that a nonexponential relaxation has two different explanations: either each element in a homogeneous system relaxes in the same nonexponential way or that the relaxation is nearly exponential but with different relaxation time for different parts of a heterogeneous system, adding up to a nonexponential result.

In figure 6.14 (a) the stretched exponential function has been plotted with different values of β . Notice that when $\beta > 1$ the name *stretched* exponential function is not appropriate, because the function is actually compressed. While the functions with $\beta < 1$ are approaching zero slower than the exponential (because they are *stretched*), the functions with $\beta > 1$ approach it faster. All functions plotted in fig 6.14 (a) intersect at $t = \tau$, where τ is the mean lifetime. If each function had a different value for τ , they would not all intersect at the same point, but each function would have the same value at each respective τ : $S(\tau; \beta, \tau) = \exp(-1)$ as they do in fig 6.14 (b). The value of τ tells us how fast the function decays.

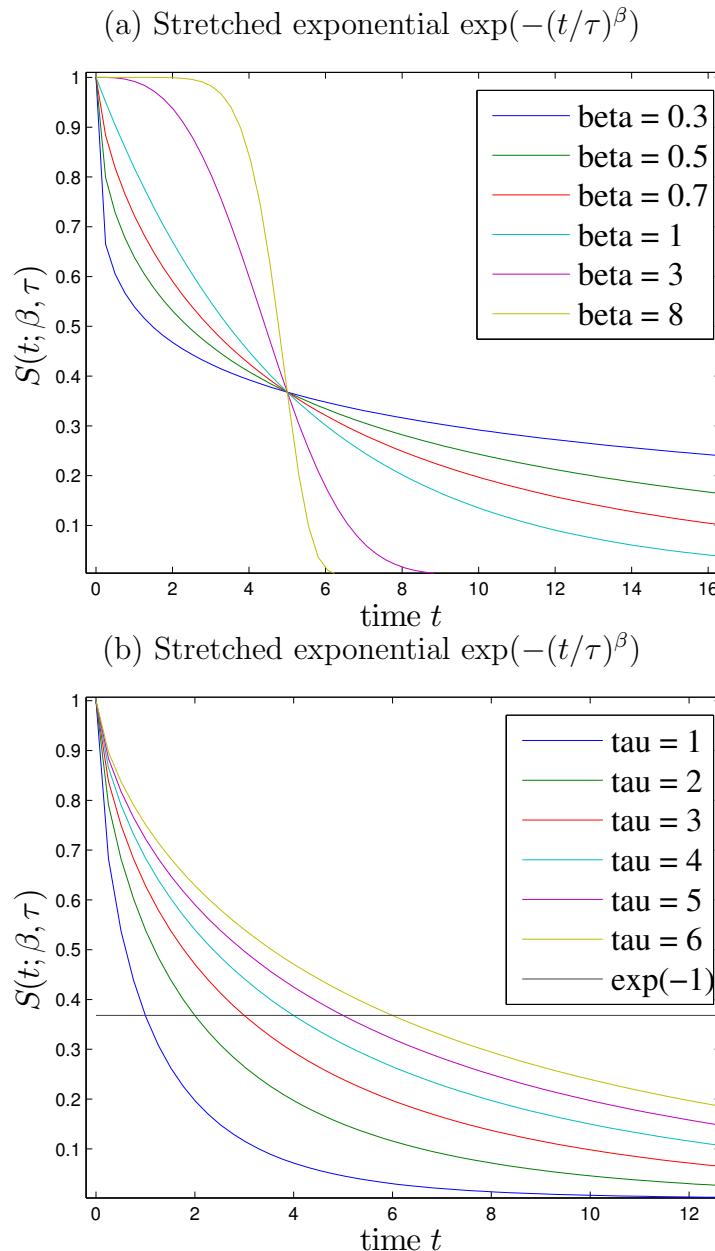


Figure 6.14: The shape of the stretched exponential function for different values of β and τ . (a) β varies and $\tau = 5$ (b) τ varies and $\beta = 0.7$.

6.4.2 Bulk water

The point of this section is to see how cage correlation function (CC) behaves for two different neighbor definitions; voronoi and radial cut-off, and for different values of c . The measurements will be performed on simulated bulk water, to avoid disturbances from the silica and to have something to compare with when we later measure on the pore systems. The CC functions measured seem to follow the stretched exponential function (SER) described in the previous section, see eq. 6.11. The SER function has two parameters that can be adjusted; the mean lifetime τ and the stretching exponent β . For each CC function we measure, we find a fitted SER function, and thereby describe the measured curve with the two parameters; τ and β .

The error of a curvefit can be measured with the mean square error

$$err = \sqrt{\sum_i [y_i - S(t_i; \tau, \beta)]^2}, \quad (6.12)$$

where t_i and y_i is the data set, and $S(t_i; \tau, \beta)$ is the SER function. The curvefitting is done with the aid of matlab's lsqcurvefit [44] (see appendix 8.1 for details), which can be used to fit user defined functions to data sets.

The matlab function, lsqcurvefit needs a guess for the parameters; $p_0 = [\beta_0, \tau_0]$, a starting point where the algorithms starts to look for the parameters $p = [\beta, \tau]$ that gives the smallest error as it is defined in eq. 6.12. If p_0 is placed near a local minimum in the error surface in the β, τ coordinate system, we may end up with a β and τ corresponding to this local minimum. There is no guarantee that we end up in the global minimum. To increase our chances of reaching the global minimum, we have made a function that runs through multiple suggestions for p_0 , and makes a curvefit for each suggestion. The curvefit that has the smallest error is chosen as our curvefit.

In addition to β and τ , we have introduced a third parameter, B , to the SER function $S(t; \beta, \tau, B) = B \exp [(t/\tau)^\beta]$. The parameter B decides the value of the SER function in $t = 0$, and even though all measured CC functions have the value 1 in $t = 0$, it may be that the rest of the function fits better to a SER function which is not exactly 1 at $t = 0$.

When we want to measure the cage correlation function, we first need to decide the value of c , and second we need to decide what definition to use for the neighbors. If we choose the radial cut-off definition, we also have to choose the value of the radial cut-off, R . The parameter space is large (not β or τ , but c and R) and the CC measurements on the bulk water might give us an indication to which parameter combinations would be interesting for our systems. If we wish to use β and τ to describe the CC functions, it might be a good idea to choose a combination of c and possibly R that gives the smallest error for the curvefit.

Another thing to consider is what values of β and τ do we wish to have for the bulk water. For instance, it is not straightforward to interpret what various values of the parameter β means, but if the value of β is 1 in bulk water, then the value of β in another system would directly tell us how much the system configuration contributes to the stretching of the CC function.

We start measuring with the voronoi definition, and move on to measuring with the radial cut-off definition.

Voronoi method

In figure 6.15 (a) we see the cage correlation function found with the voronoi method, for different values for c . Figure 6.15 (b) tells us the fitting parameters (B, τ, β), and the error as defined in eq. 6.12 is visualized as a function of c in figure 6.15 (c). With the voronoi method, the average number of neighbors is 15.6, which means we have many options for the value of c .

We see that $c = 1$ gives the smallest error, but the function drops quickly and is zero most of the time which is not very useful. The second smallest error is for $c = 2$, but again the function drops to quickly. The third best fit is for $c = 5$ which is marked in the error plot, and this function falls at an appropriate pace. The error for $c = 6$ is larger than for $c = 5$, but it is of the same order of magnitude while it has a slower decay. The distance between the first and second point is shorter than for $c = 5$ which gives the function a more smooth shape.

For the voronoi method it seems that $c = 5$ and $c = 6$ are both good choices, when considering the error of the curvefit. We should choose $c = 5$ if we need a good fit with $F(t; B, \tau, \beta)$ and $c = 6$ if we want a slower decay. We notice that $c = 5$ gives a β value very close to 1.

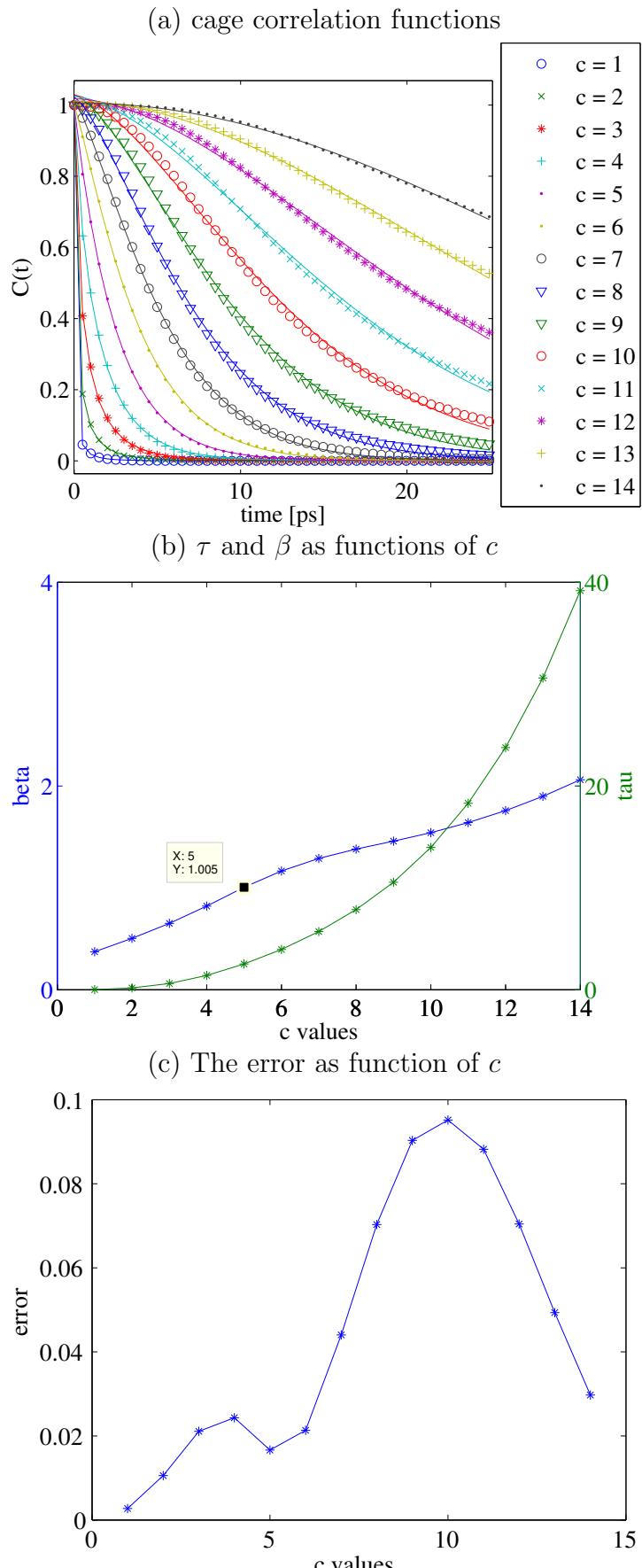


Figure 6.15: (a) The cage correlation functions of bulk water found with the voronoi method for various values for c . (b) The parameters to the curvefit found in (a). (c) The error in the curvefits in (a)

Radial method

Figure 6.16 and 6.17 show the cage correlation found with the radial method for the distance corresponding to the first and second shell, respectively. The first shell distance is the distance where $g(r)$ for oxygen-oxygen has its first local minimum from the left, which is $R = 3.318 \text{ \AA}$, and the second shell distance is where $g(r)$ has its second local minimum, which is at $R = 5.474 \text{ \AA}$, see figure 3.7.

The integrated $g(r)$ function in figure 3.9 tells us that the first shell corresponds to an average of 4.5 neighbors and the second shell corresponds to 23 neighbors.

The first shell-distance gives us few neighbors, which limits our options for the value of c . The parameter c is the number of neighbors that need to leave the neighborhood before we can say that there has been a change in the neighborhood, so when the average number of neighbors is smaller than 5, it does not make sense to choose c larger than 5. In figure 6.16 (a) we can see that the cage correlation function for $c \geq 5$ falls slowly and for the largest values, $c \geq 7$, the correlations are horizontal lines with value 1. Similar to the CC functions with the voronoi method, the smallest c value had the best fit with $F(t; B, \tau, \beta)$, and even though it is not as bad as with the voronoi method, it drops a bit too quickly for us. The next value $c = 2$ gives an appropriate fit-error and does not fall too quickly, even though the distance between the first and second point is quite large. For $c = 3$ this distance is smaller, but the fit-error is too large. It can be seen with the naked eye that the $F(t; B, \tau, \beta)$ found by matlab for $c = 3$ was not a good fit. We see from the error plot that the error actually drops to its lowest value at $c = 7, 8$, but these functions are not interesting to us because they are virtually horizontal. The conclusion is that for the radial method with the first shell-distance, it seems that $c = 2$ is the best choice.

The figure 6.17 (a), with cage correlation for the second shell-distance, is much more similar to the voronoi method figure (fig. 6.15) than the figure for the first shell. Again, the first c -value $c = 1$ gives a very good fit, but drops to quickly. When the average neighborhood has 23 neighbors, one moving away is not really a big change. Neither is $c = 2$ or $c = 3$. For $c \geq 7$ the fit-errors are too large compared to the smallest errors for this method. This leaves us with $c = 4, 5, 6$ which all give small errors. It does not really matter which one of these we choose, but since it is beneficial to have many non-zero values we may choose the c which gives the slowest decay, which is $c = 6$. Or, we could argue that the β value for $c = 5$ is the value closest to 1.

We choose three different parameter combinations for measurements on the pore systems; the voronoi neighbor definition with both $c = 3$ and $c = 5$, as well as the radial cut-off definition with neighbor radius equal to the first shell distance, $R = 3.318 \text{ \AA}$, and $c = 1$. The reason why we choose $c = 1$ instead of $c = 2$, which we saw was a better option for the bulk water, is because we will look at CC functions for the molecules close to the pore surface, where there is less movement than in bulk water, and the molecules do not lose their neighbors as often. We tried measuring for $c = 2$, but some of the functions for the water close to the surface did not decay fast enough to tell us anything, but with $c = 1$ the decay rate was appropriate.

Number of neighbors

Before we start looking at the results for the cage correlation (CC) function for the water inside the pores, we will look at something that affects the CC results. As stated earlier,

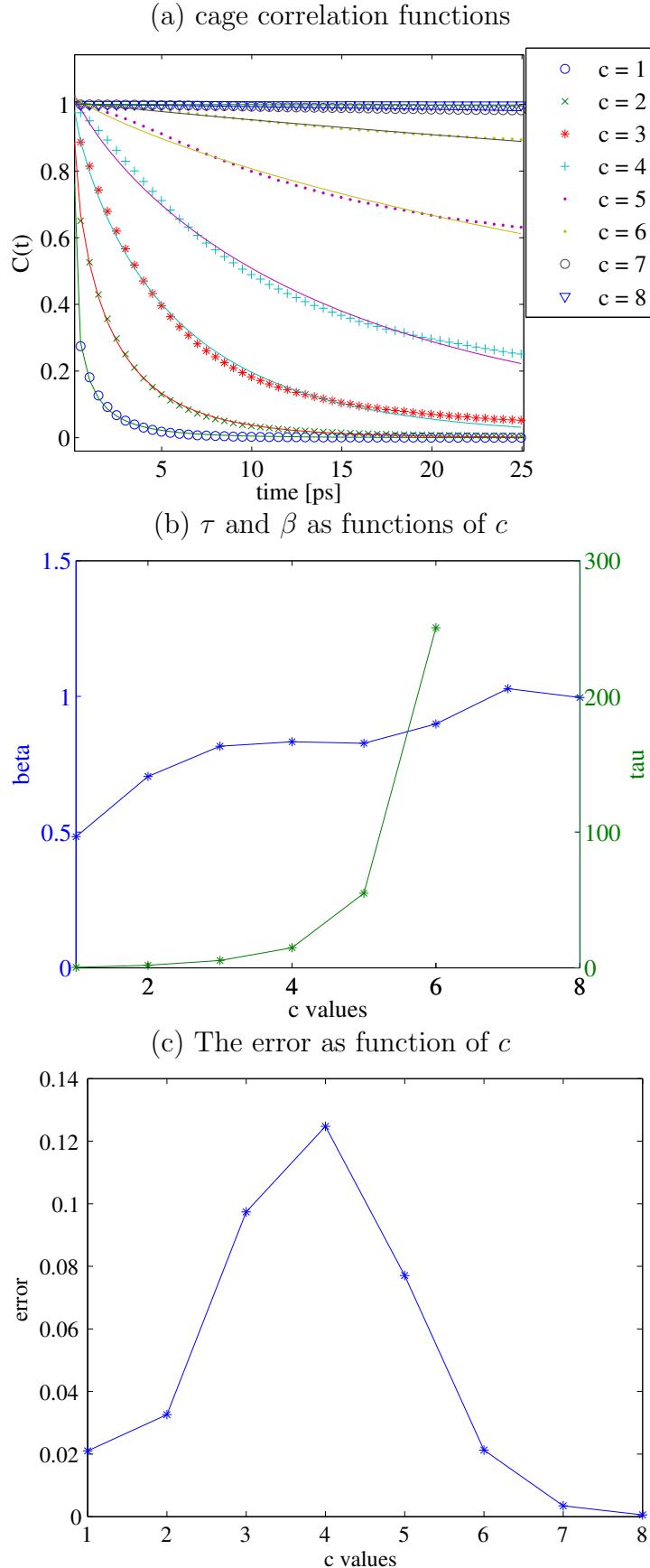


Figure 6.16: (a) The cage correlation functions of bulk water measured with the radial method, for the first shell, and various values for c .(b) The parameters to the curvefit found in (a).(c) The error in the curvefits in (a)

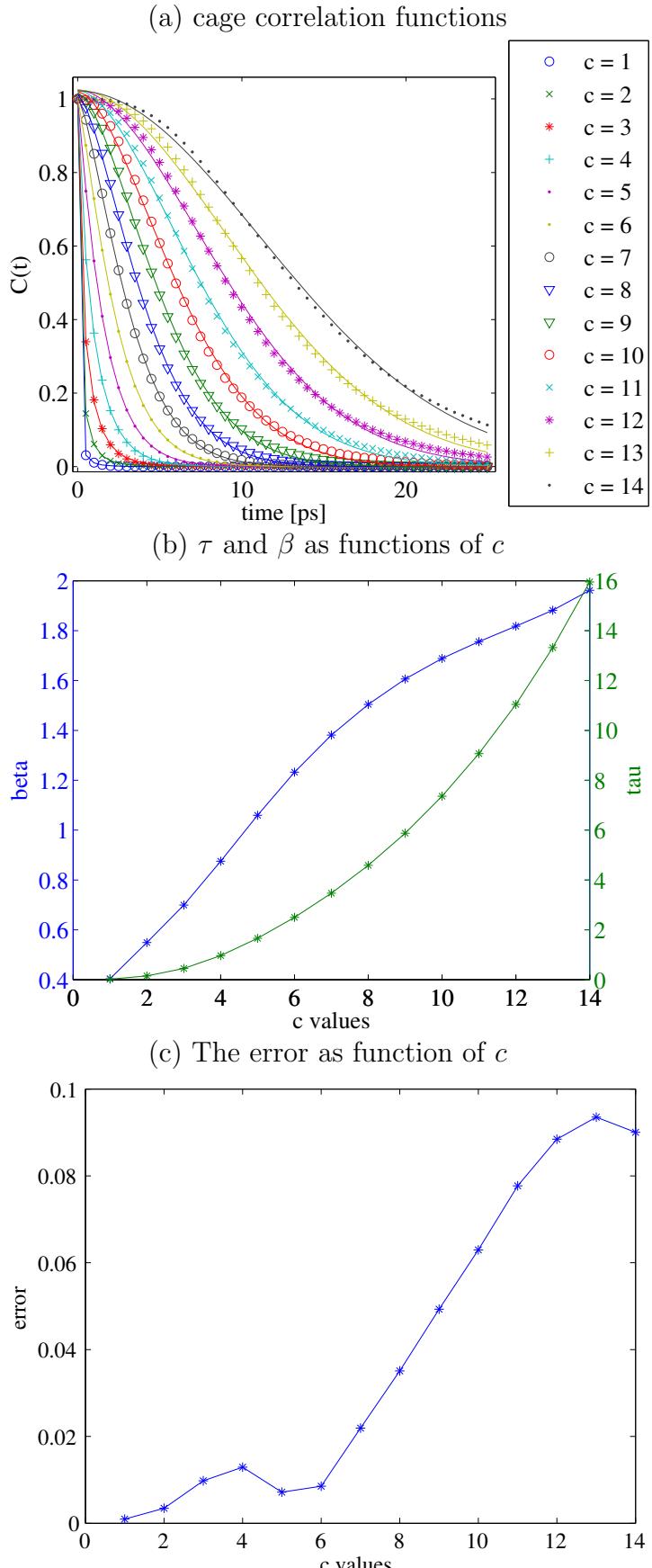


Figure 6.17: (a) The cage correlation functions of bulk water found with the radial method with R equal to the second shell distance, for various values for c . (b) The parameters to the curvefit found in (a). (c) The error in the curvefits in (a)

the parameter c is the number of molecules that have to leave a molecule's neighborhood before we say that the surroundings for that molecule have changed. When we choose the value of c we have to consider how many neighbors we expect each molecule to have, and how large portion of these has to be removed before we can say that the surroundings have changed. Maybe a good rule of thumb is to choose c to be a third or a fourth of the original number?

The problem is that the number of neighbors may vary in the system, especially in porous systems where there are walls. Before we start interpreting the CC results, we will check how the number of neighbors vary as a function of distance to the pore wall. For each distance we find the neighbors for the molecules inside a shell at the current distance of width $l = 3 \text{ \AA}$. Figure 6.18 shows the number of neighbors in one of the spherical pores,

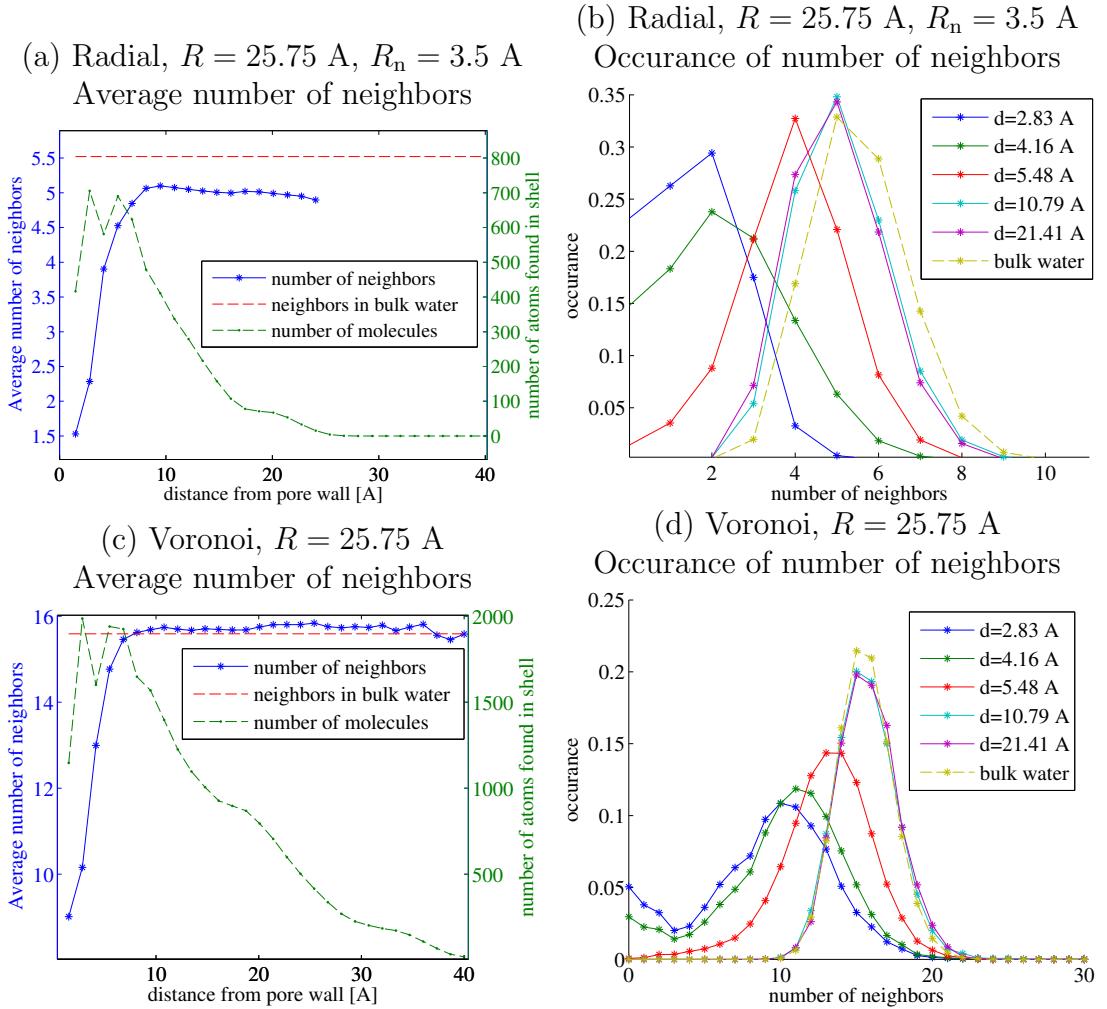


Figure 6.18: The neighbors found with the radial method for the first shell. (a) The number of radial neighbors as a function of distance to the pore wall for a spherical pore. The green line is the number of water molecules found in the shell at distance d . (b) The occurrence of number of neighbors in the spherical pore from (a). Each distribution corresponds to a distance from the pore wall. (c) The number of Voronoi neighbors as a function of distance for the largest spherical pore. (d) The occurrence of number of neighbors in the spherical pore from (c).

for both the radial cut-off definition and the Voronoi definition. We only count the water molecules neighbors, and ignore silica neighbors. Figure (a) and (c) show the average number of neighbors as a function of distance, together with the number of molecules found inside the shells at these distances and the average number of neighbors in bulk water. Figure (b) and (d) show the distribution of number of neighbors for different distances to the pore wall, together with the distribution in bulk water.

We see that for both definitions, the first 10 Å we have fewer neighbors than for the rest of the pore. For the radial definition, the average number of neighbors start at 1.5 for the smallest distance, and grow until it reaches 5 for distance $d = 10$ Å. For distances larger than 10 Å, the average number of neighbors stay at approximately this value. We notice that the average number for larger distances is approximately 10% lower than the number in bulk water. The neighbors for the Voronoi definition follow the same pattern, but start at 9 and grow up to bulk value. Actually, the number of neighbors are slightly above the bulk number of neighbors. This is also reflected in the distribution in figure (d). The distribution for the largest distance $d = 21.41$ Å has its peak at the same amount of neighbors as the bulk water does, but the peak for the bulk water has a higher occurrence than for the pore, and this is because the pore has slightly higher occurrence for the number of neighbors that are larger than the peak value.

The number of molecules found at each distance is largest for the smallest distances (except the first), and decreases as the distance increases, which is due to the fact that we are looking at a spherical pore. This means that a large portion of the contributions to the CC function is from molecules with few neighbors.

For two of the distributions we see something a bit weird. We see that the occurrence of water molecules with zero neighbors is relatively large. When we say that we measure on water molecules, we actually mean that we measure on the oxygen atoms marked as water-oxygen (from the injection phase). Sometimes, it happens that a water-oxygen switches places with a silica-oxygen, but it is still marked as a water-oxygen. In figure 6.19 we see a visualization of the water inside a spherical pore, where the water-oxygen that has no water-oxygen neighbors are marked in yellow. The blue atoms that surround the yellow ones are the silica neighbors of the yellow atoms. We see that those who had no neighbors were all trapped in the silica, none of them were in a nanobubble. The water-oxygen that has no neighbors are ignored in the cage correlation measurements.

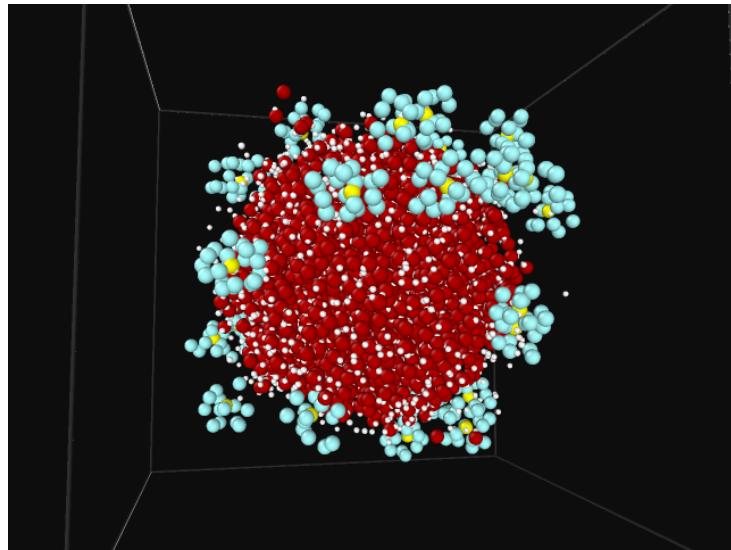


Figure 6.19: The water inside a spherical pore, with the water-oxygen atoms registered with no water neighbors is marked in yellow. The blue atoms surrounding the yellow atoms are the registered neighbors of the yellow atoms. The blue ones are either silicon or silica-oxygen.

6.4.3 Spherical pores

This section contains all the results associated with the Cage correlation (CC) functions for the spherical pores. We measure CC for the three different combinations of parameters: Voronoi neighbors with $c = 3$ and $c = 5$, and radial neighbors with the first shell distance (3.318 Å, see section 3.7.4) and with $c = 1$.

Figure 6.20 contains the CC functions for the three combinations, measured on all the water inside the six spherical pores. The three subfigures are very similar, as they all show that the CC functions fall slower the smaller the radius of the pore, or equivalently, the larger the curvature of the pore surface is. The three subfigures differ by how fast the CC functions fall in general. We see that the Voronoi $c = 5$ functions fall slower than the Voronoi $c = 3$, which is to be expected. The CC functions for the radial neighbor definition and $c = 1$ fall faster than the functions for the two other combinations (reach a small value in very few time steps). The mean life time parameter τ from the SER approximation is a measure of how fast the CC function falls. Figure 6.21 shows τ found for the CC functions in figure 6.20, as a function of pore radius. Here we see that $\tau_{\text{voro},c=5} > \tau_{\text{voro},c=3} > \tau_{\text{first shell},c=1}$ for all pore sizes. All three functions, $\tau(R_{\text{pore}})$, have the same shape. The highest value appears for the smallest pore radii, and decays towards a steady value as the pore radius increases.

Similar to the other measurement types, we wish to see how the different parts of the pore water contribute to the results we see in figure 6.20. We split the pore into surface and confined water with the distance d . The surface water is then all the water closer to the pore wall than the distance d , while the confined water is all the water further away from the pore wall than the distance d .

Figure 6.22 shows an example of CC functions for surface and confined water, where $d = 7$ Å, for the three different combinations. In the confined water plots we have included the CC function measured on simulated bulk water. We see that the surface CC functions

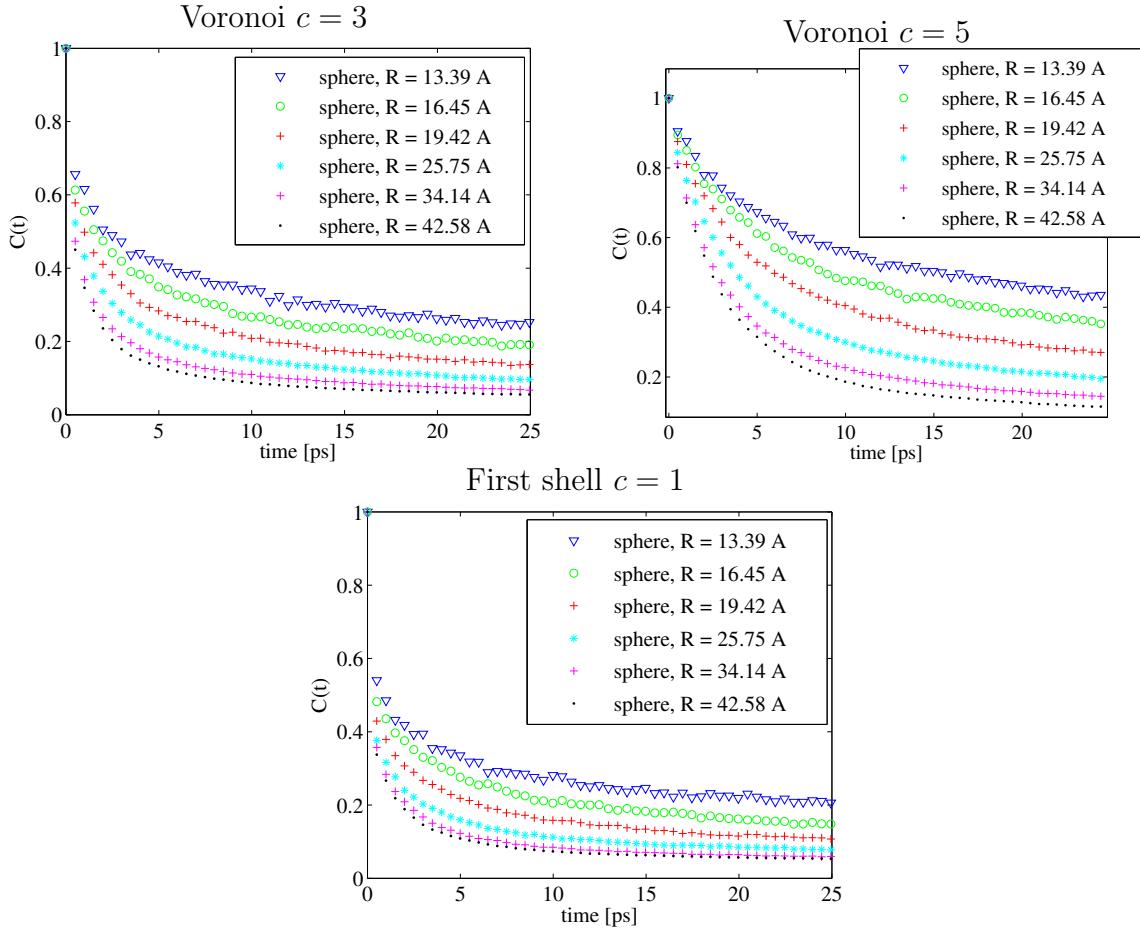


Figure 6.20: CC functions measured with three different combinations of parameters, for all the water inside the pore, for six different pore sizes.

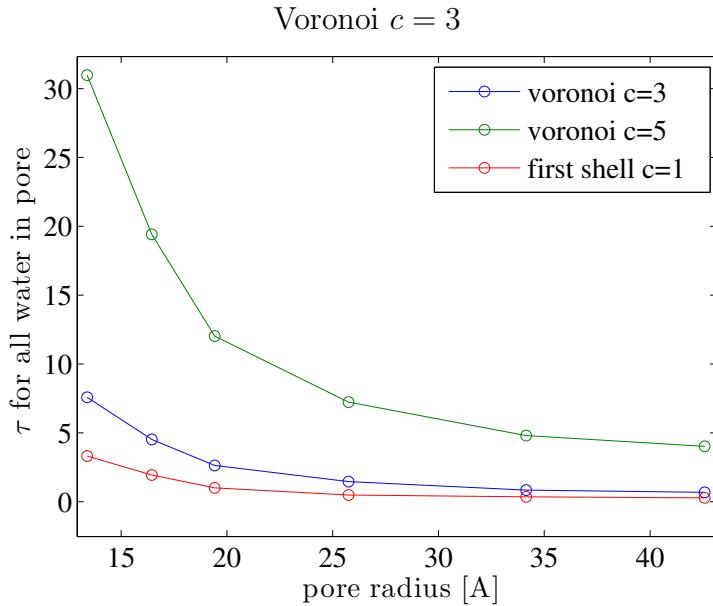


Figure 6.21: The mean lifetime of the CC functions for all the water inside the pore, figure 6.20, as a function of pore radius.

fall at a slower pace than the confined CC functions, and that the surface CC functions differ more for the different pore radii than the confined CC functions do. The confined CC functions lie close to the CC function for bulk water, but not perfectly on top. It seems that the water at distances larger than $d = 7 \text{ \AA}$ from the pore wall, is also affected by the wall. In particular, for the $c = 5$ confined CC functions, we see that the pores with large radii (small curvature) lie closer to the bulk water function, than the functions from pores with smaller radii (larger curvature). These functions lie a little above the bulk function. The surface functions are more similar to the functions in figure 6.20 than the confined functions are. As in figure 6.20, the functions from small pore radii fall slower than the functions from larger pore radii. This is what we just noticed for the confined functions as well, but for the surface functions the differences between the functions are much larger. In addition, the pace the surface functions fall with are more similar to the functions made with all the water inside the pore (figure 6.20), than the confined functions are.

Until now we have been looking at CC functions for surface and confined water for *one* value of distance d . Now, we will look at the CC functions created with various distances d . To avoid an overcrowding number of figures, we will only include plots for the second largest pore in our ensemble of pores, shown in figure 6.23. This figure is quite similar to the figure 6.22 which compares the curves for different pore radii. Again, the confined curves lie close to the bulk water CC function, and close to each other, while the surface curves are more spread out.

When we increase the distance d , we include more and more water molecules to the surface part of the pore, and remove more and more molecules from the confined part of the pore. We see that when the distance d is increased, both the surface functions and the confined functions fall quicker and quicker. This must mean that the molecules closer to the wall lose their neighbors less frequently than the molecules further away. When we include more and more molecules to the surface water, we increase the number of molecules who tend to lose their neighbors rapidly, and the average goes down. When we remove more and more from the confined water, we gradually remove those molecules who tend to hold on to their neighbors for a long time, and the average goes down.

If it was so that the confined functions did not change when we increase the distance d , but the surface functions did change, it would mean that there were a number of molecules very close to the surface that very rarely lose their neighbors, and draw the average up. These molecules would never be in the confined part of the pore, and always included in the surface part of the pore. This is not the case. Since we see that the confined functions gradually approaches the bulk water function for distances smaller than 7 \AA , it means that the differences we see in figure 6.20 is not due to a few trapped molecules very close to the pore surface, but the water molecules all the way out to 7 \AA from the wall. What is surprising with this figure is the fact that the confined curves for $d > 7 \text{ \AA}$ (Voronoi) are actually below the bulk water curves. When we increase d , the confined functions approaches the bulk water function and then move past it.

We have found by visual inspection that the stretched exponential function (SER function) is a good fit to our CC functions for the simulated bulk water. Before we go any further, we wish to check if the SER function is a good fit for the pore water as well. We make data collapse plots for the measured CC functions, where we let the x-axis be $(t/\tau)^\beta$. The parameters τ and β are found with the same tool used to find the parameters of the

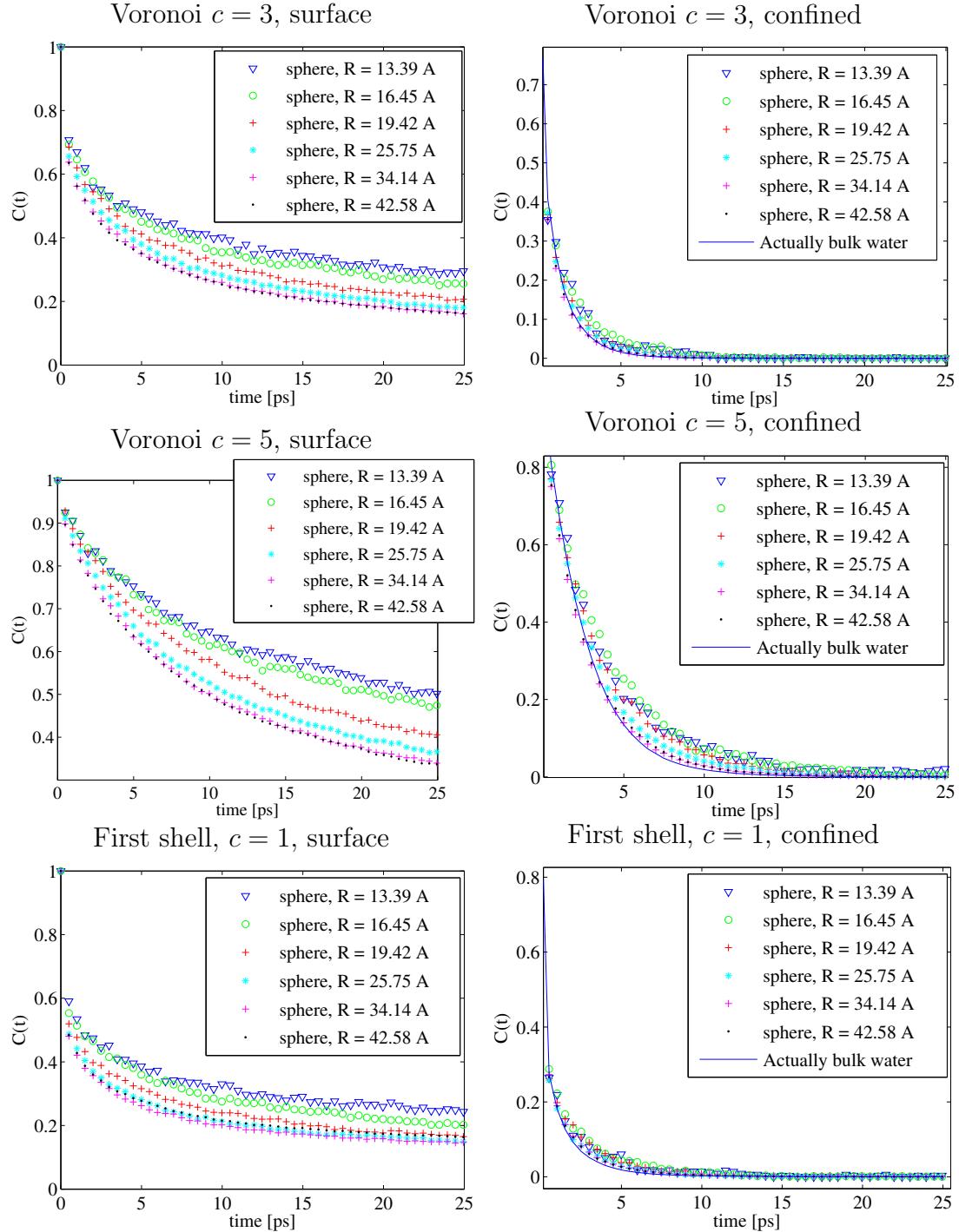


Figure 6.22: Surface and confined Cage correlation functions for all the spherical pores. The pores are divided into surface and confined at $d = 7 \text{ \AA}$.

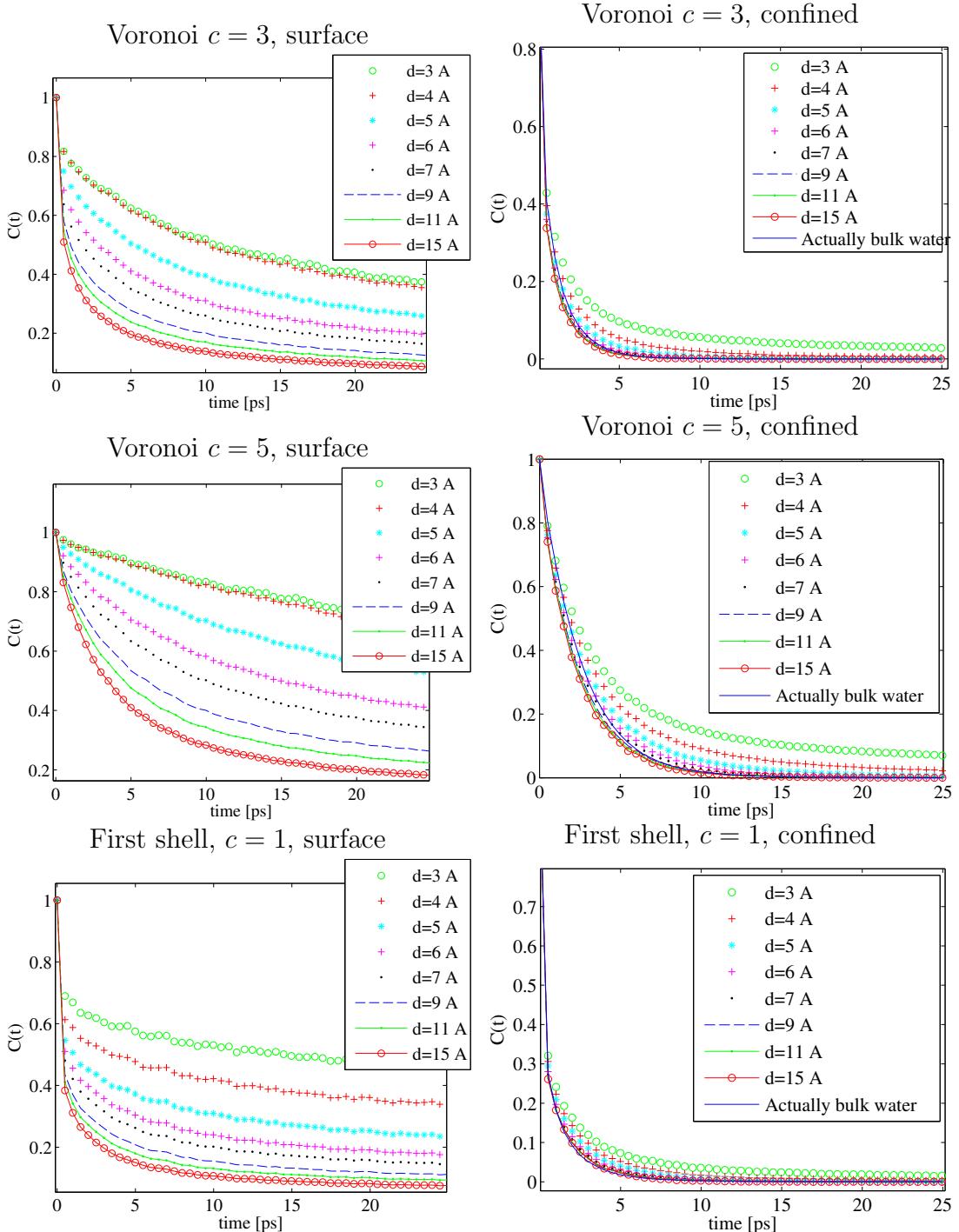


Figure 6.23: Surface and confined Cage correlation functions for the second largest spherical pore. The pore is divided into surface and confined at various d values.

bulk water CC functions. If a CC function really follows the SER function, the function in the data collapse plot should follow the exponential function: $\exp(-(t/\tau)^\beta) \rightarrow \exp(-x)$. Each CC function has its own pair of parameters, τ, β , so each CC function is plotted towards its own set of x-values, even though the original time values, t , were equal.

Figure 6.24 shows an example of a data collapse plot for the CC functions from the pore water. Here we are looking at the surface water for $d = 7$ Å and $c = 1$, and compare the results with the exponential function. We see that the data points lie close to the exponential curve.

Another way to check if the SER is a good fit is to plot $\log(|\log(C)|)$, where C is the CC function, towards $\log(t)$ where t is time, and see if the data points create a straight line. If the CC function actually was a SER function, this would happen because

$$C = e^{-(t/\tau)^\beta},$$

$$\log(C) = -\left(\frac{t}{\tau}\right)^\beta, \text{ takes the logarithm,}$$

$\log(|\log(C)|) = \beta \log(t) - \beta \log(\tau)$, takes absolute value, and the logarithm again,

$\log(|\log(C)|) = \beta x + \text{const}$, recognizes a linear function,

where $x = \log(t)$. We see that $\log(|\log(C)|)$ is a linear function of x , where β is the slope.

Figure 6.25 shows this test for the voronoi and radial cut-off definition, for both surface and confined water. For the surface water it seems like the fit gets worse for larger distance d , but for shorter d the plots are rather linear especially for $d = 5$ Å. For the confined water it is the other way around, the function for the smallest distance $d = 3$ Å does not appear very linear, while it seems the linearity improves when d is increased. For the voronoi CC function, it is the function for the largest distance that has the most linear shape, but for the radial CC function it is the function for $d = 5$ Å that is best. For the latter definition, it looks like the $d = 3$ function and $d = 14$ function bend in opposite directions. The reason why the confined figures only have CC functions for distances $d = 3, 5, 14$ Å is that it was difficult to tell the functions for $d \geq 8$ Å apart, so we only included one of them. We have learned that the SER is not always a perfect fit, but for most cases it is adequate, and we choose to characterize the CC functions using the two SER parameters τ and β .

The first thing we want to look at is τ and β as functions of distances to the pore wall for surface and confined water. Figure 6.26 shows $\tau(d)$ for surface and confined water for the three combinations $c = 3$, $c = 5$ and $c = 1$. We see that, in general, the value of τ decreases as d is increased. Since τ is a measure of how much time it takes for the functions to fall to certain value, this fits with our observations for the CC functions for the second largest pore (figure 6.23). There we saw that the CC functions fell slower for smaller distances d than for larger d . We also observed that the confined CC functions went past the bulk water function and ended up falling a little bit more rapidly than the bulk water function. This is also represented in $\tau(d)$ for the CC functions for the confined water, where we see that the value of τ becomes lower than the bulk water value after a certain distance. For $c = 3$, this distance is approximately 7 to 8 Å. We see that $\tau(d)$ for confined water with $c = 1$ has a few exceptions from the general behavior of $\tau(d)$. Here $\tau(d)$ for the three smallest pores behave very differently from the other curves. Two of them suddenly start to increase while the last one decreases. This is somewhat surprising

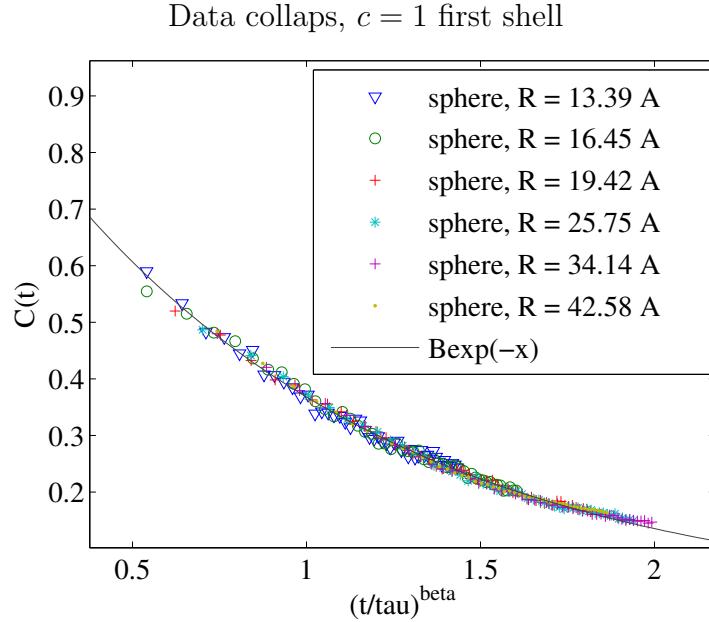


Figure 6.24: Data collaps of the surface Cage correlation, $d = 7$ Å.

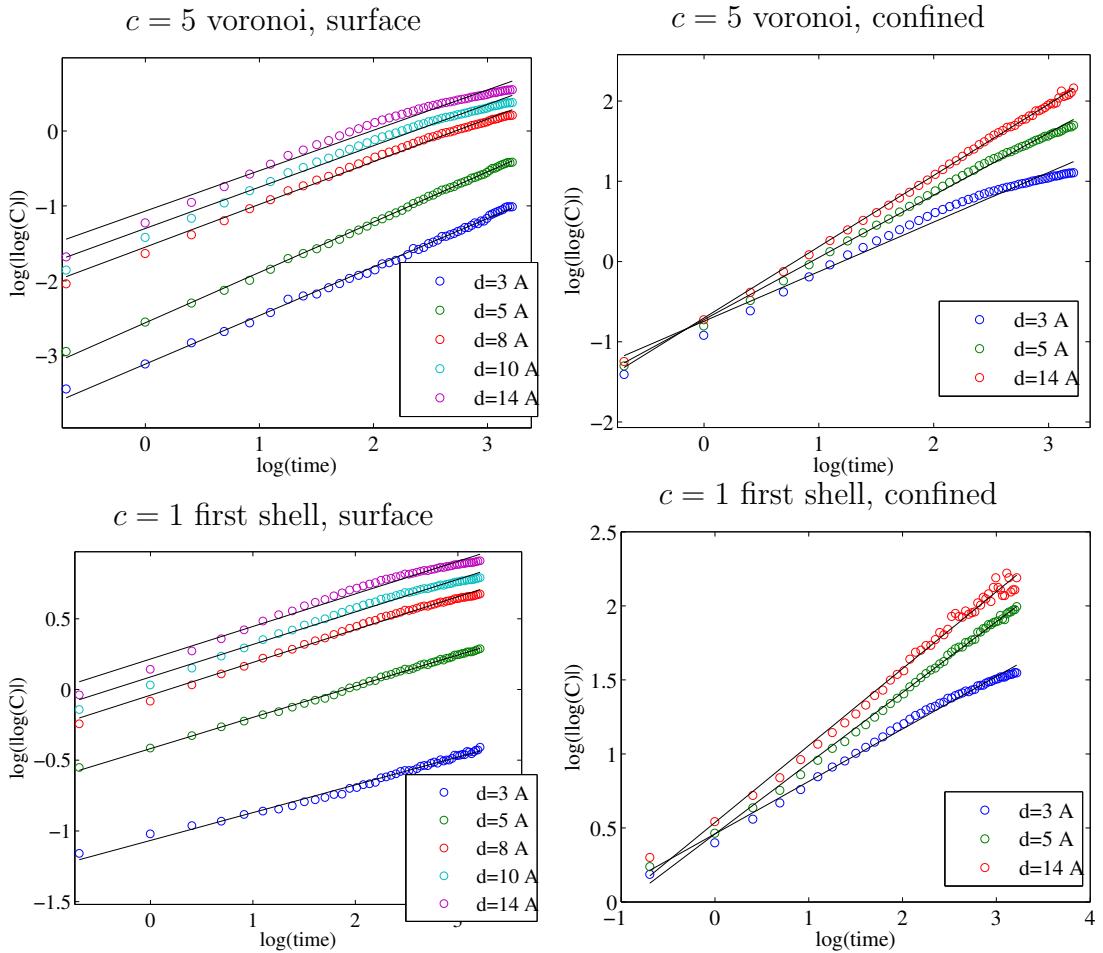


Figure 6.25: Test to see if the SER function is a good fit for the CC functions. If the plotted data points lie on a straight line, the SER function is a good fit.

considering we do not see this sort of behavior for $c = 3$ and $c = 5$, and they are measured on the same system. To investigate this we looked at the CC function for the confined water in the smallest pore ($R = 1.34$ nm) for $d = 5$ Å and $d = 10$ Å, because these two functions have very similar values for τ (these plots are not included in the thesis, only described). For $d = 5$ Å, the data points lie nicely on top of its SER function, while for $d = 10$ Å the data points are spread out around its SER function. This is probably caused by the number of measured molecules being very small. We also saw that the $d = 10$ Å data points were closer to the bulk water function than the $d = 5$ Å, which is not clear from just looking at the $\tau(d)$ plots. The functions for confined water $c = 1$ does fall very rapidly, so that the functions are down at $\exp(-1)$ already at the second or third data point from the left. So, we blame the abnormalities on too few molecules in the region we measure on, and not frequent enough samplings for a function decreasing this rapidly.

We see that the value of τ for small distances d is dependent on the curvature/radius of the pore. For most cases we see that the smaller the radius (larger the curvature), the higher is τ . We have some exceptions here as well, for instance $c = 5$ and $R = 1.65$ nm or $c = 1$, $R = 4.26$ nm.

In figure 6.27 we see β as a function of d for surface and confined water. For all the confined water plots we see a weird behavior for the three smallest pores, which we again blame on too few molecules in the region of measurements. In general, for the confined water, the functions appear to start at a low value and grow until they reach a higher value where they flatten out. The value they flatten out on is a bit lower than the value we find for bulk water, it looks like the value is approximately 90% of the bulk value.

For the two Voronoi figures for surface water we see that β starts on a value lower than the value measured on bulk water, and that it decreases, which means that it is moving away from the bulk value, before it flattens out. For $c = 1$ it is the other way around, meaning that β starts at a low value and increases (just a little bit) before it flattens out. It looks like β flattens out on approximately 50% of the bulk value, for all configurations.

The next thing we want to look at is how τ behaves as a function of pore radius. These are the same values found in figure 6.26, only we want to turn it around, and let the x-axis in those plots be our captions, and let the captions be our x-axis. We have seen that small radius often means higher τ than for a large radius. Figure 6.28 shows $\tau(R_{\text{pore}})$ for surface water for a few selected distances d . For $c = 3$ we see that τ falls when the pore radius is increased. For $c = 5$ we see the same thing, if we ignore the first points (smallest radius) of the two functions with $d = 3, 4$ Å. For $c = 1$ we see that τ actually goes up for the largest pore radius. This is seen in all functions up until $d = 8$ Å. A hint of this can also be seen in $c = 3$ and $c = 5$ for the smallest distance d . Why this happens is not easy to explain, and might be system specific for the largest pore.

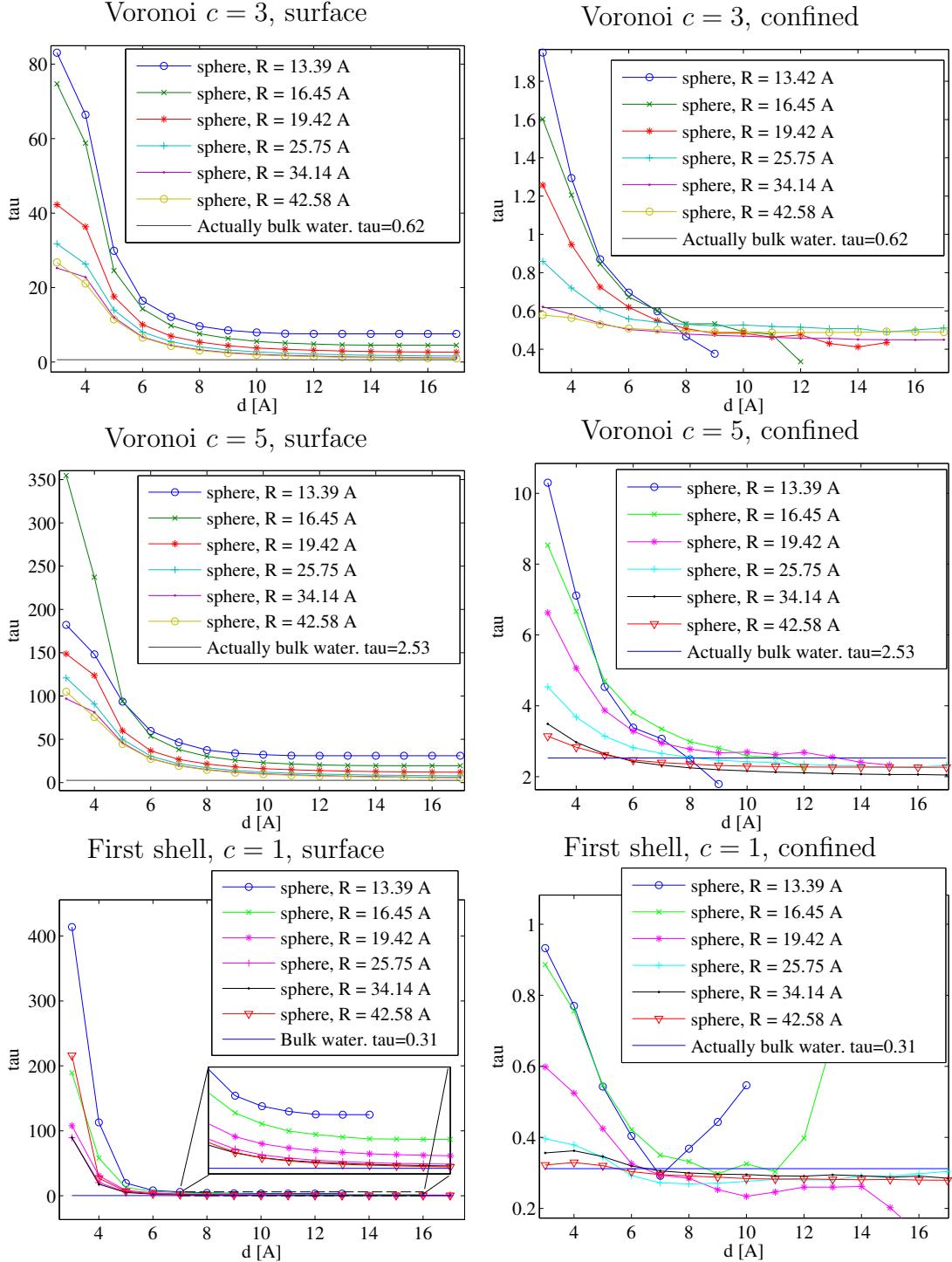


Figure 6.26: τ as functions of d for surface and confined CC functions.

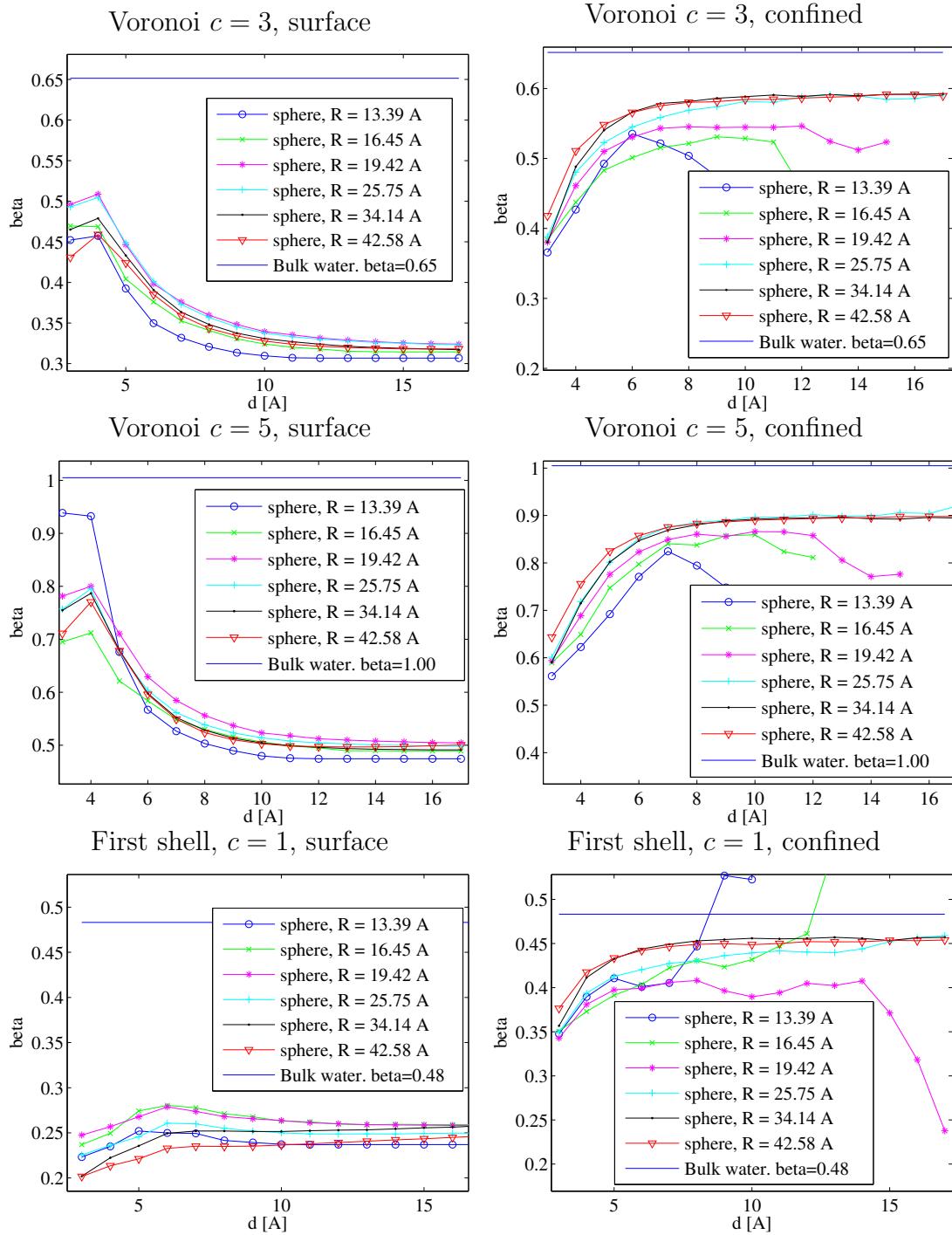


Figure 6.27: β as functions of d for surface and confined CC functions.

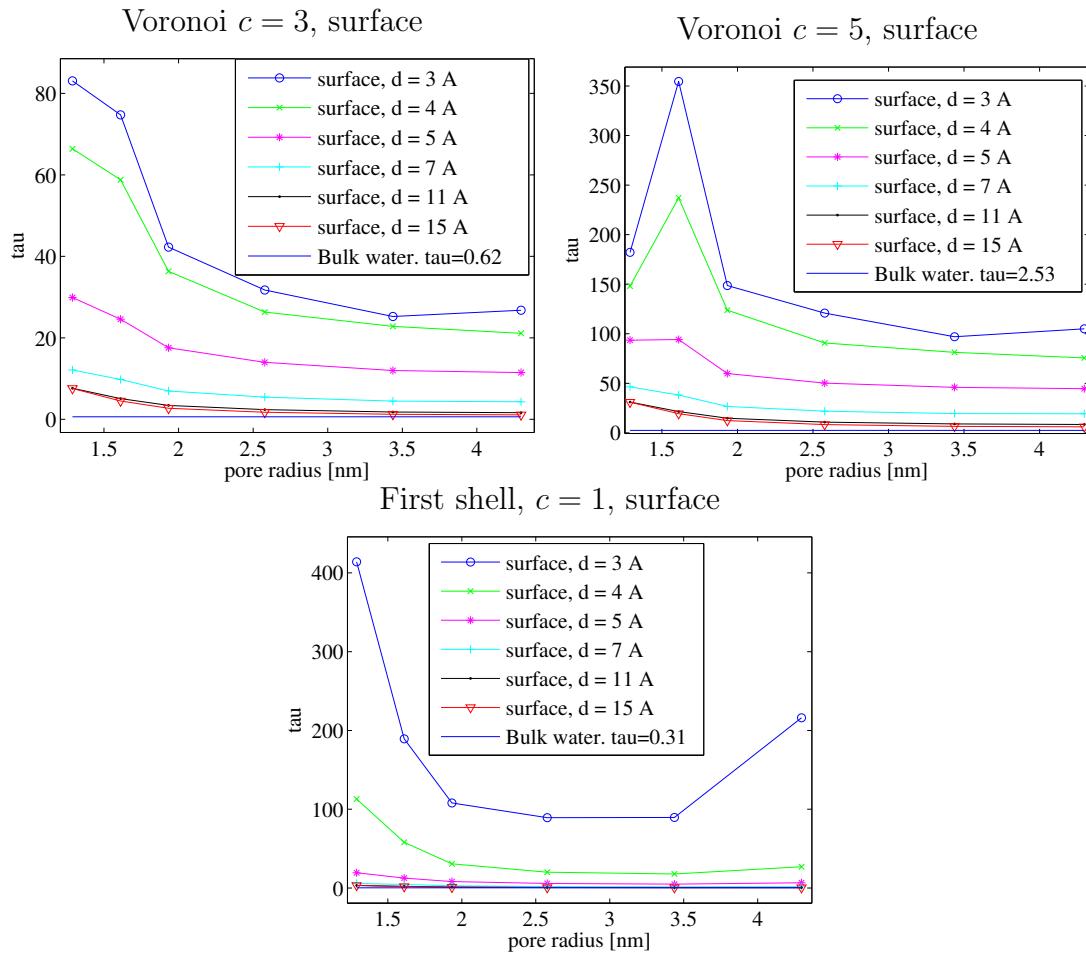


Figure 6.28: τ as functions of pore radius for surface CC functions.

6.4.4 Same volume pores - three pore shapes

The last CC related topic we are going to look at is the parameter τ for surface water for the three different pore shapes, the sphere, the cylinder and the plane. The differences we see related to different radii of the spherical pores may be due to both the volume of the pore and the curvature of the surface. These pores of different pore shapes are constructed to have the same volume, but different curvature.

Figure 6.29 shows τ as a function of distance d for the three pore shapes. For the two voronoi functions we see that $\tau_{\text{sphere}} > \tau_{\text{cylinder}} > \tau_{\text{plane}}$ at all distances, but the differences decreases as the distance d increases. For $c = 1$ the plane pore has the lowest value of τ , while τ for the sphere and the cylinder lie very close to each other above the τ for the plane pore. Although it is a bit hard to say for the plane pore since $d = 10 \text{ \AA}$ is the last data point of the plane pore function, we find that the functions for the three pore shapes have flattened out within 10 \AA away from the pore wall.

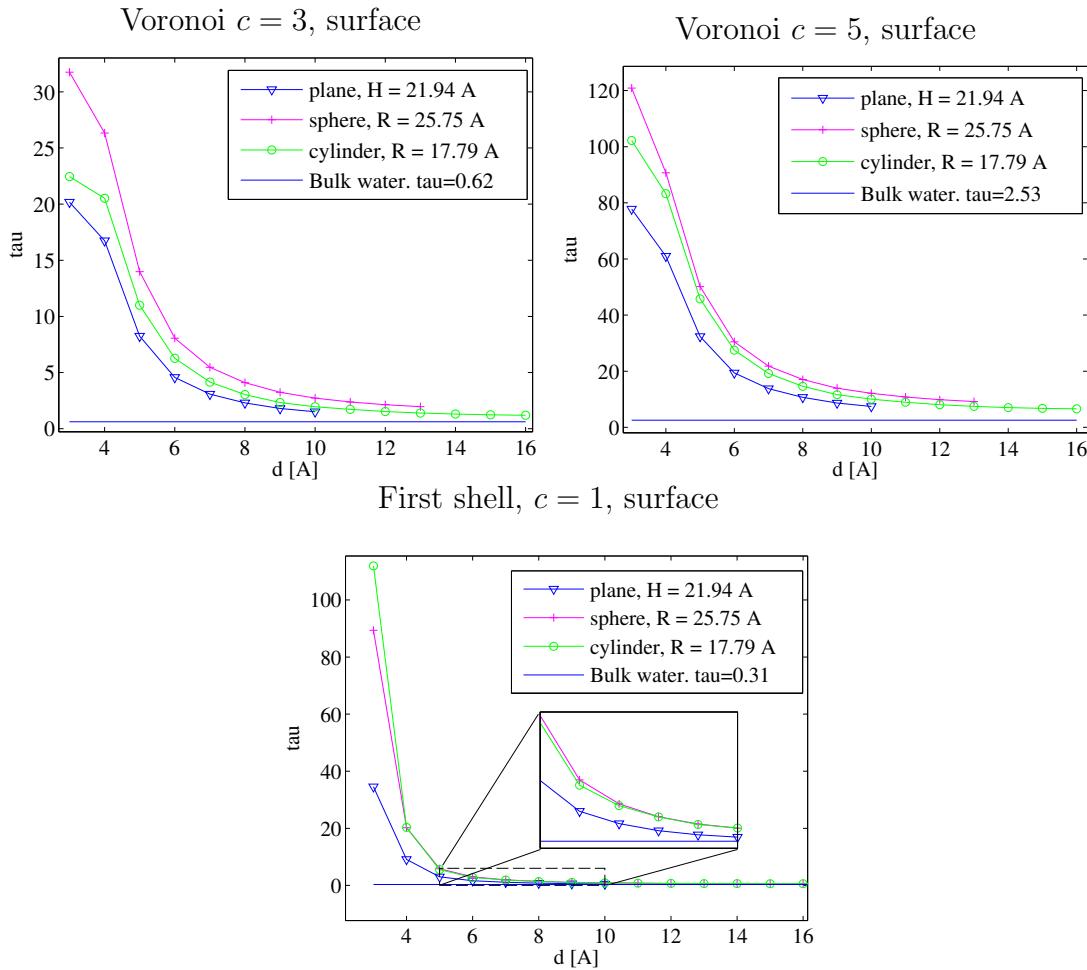


Figure 6.29: τ as functions of distance d for surface CC functions for the three pore shapes, compared with the bulk value.

6.5 Life span

We want to investigate how long an arbitrary water molecule stays inside a given region of the pore, for instance how long it stays near the pore surface. We want a characteristic time for the different distances to the pore wall, and compare this for the different pore sizes/curvatures. We have seen that the water behaves differently for different distances to the pore wall, and also to some degree, it behaves differently for different sizes and curvatures of the pore. It will be interesting to get a time perspective for the molecules in the specific regions where we see non-bulk behavior, to see how long the molecules actually stay inside the region and behave differently than they would have in other parts of the pore or in bulk water.

Instead of directly measuring the time each molecule spends in a region of the pore, we choose to find the time it takes for a fraction of the water molecules that were in the region at a time origin, t_0 , to leave the region. We call these molecules the original water molecules. This is measured by counting the number of original water molecules that still is inside the region at each time sample, and use the time where that number has fallen to $\exp(-1)$ of its original value as the characteristic time, the life span. We work with the relative amount instead of the actual amount to easier compare the different regions. If an original water molecule has escaped once, it will be registered as escaped even if it reenters the region.

We suspected that the relative amount would decay exponentially, and that is the reason why we chose this definition for the life span. When the measurement was done, we discovered that most of the amounts followed the stretched exponential function, $S = \exp(-(t/\tau)^\beta)$ better than the exponential function, $E = \exp(-t/\tau)$. The parameter τ is defined exactly how we defined life span for both functions, S and E , and β is a stretching parameter that we are not very interested in here.

Not all of the relative amount functions we measured, decayed quickly enough for us to directly measure the life span, because the relative amount did not reach $\exp(-1)$ during the measuring time. Since the measured functions seem to follow the stretched exponential, we can use the same tools as for the cage correlation to calculate the life span, τ , which will work even for the slowly decaying functions.

Results

The measurements are divided into two categories: surface water and shell water. Surface water is the set of water molecules that are closer to the pore wall than a given distance, d , while the shell water is the set of water molecules that are inside a shell of width l at a distance d from the pore wall. The measurements are performed with different values of d for both categories.

Figure 6.30 (a) shows the relative amount of surface water for a medium sized spherical pore. Figure 6.30 (b) is the same as figure 6.30 (a) plus the fitted stretched exponential curve. We see that the fit might not be perfect for all of the functions, but based on visual inspection they are alright enough for us, and we use the parameter τ as our life span.

Figure 6.31 shows the logarithm of the estimated life span (the parameter, τ , from the curve fit) of the surface water in all of our systems, collected from the relative amount functions similar to those in figure 6.30. In these plots they are shown as functions of distance to the pore wall.

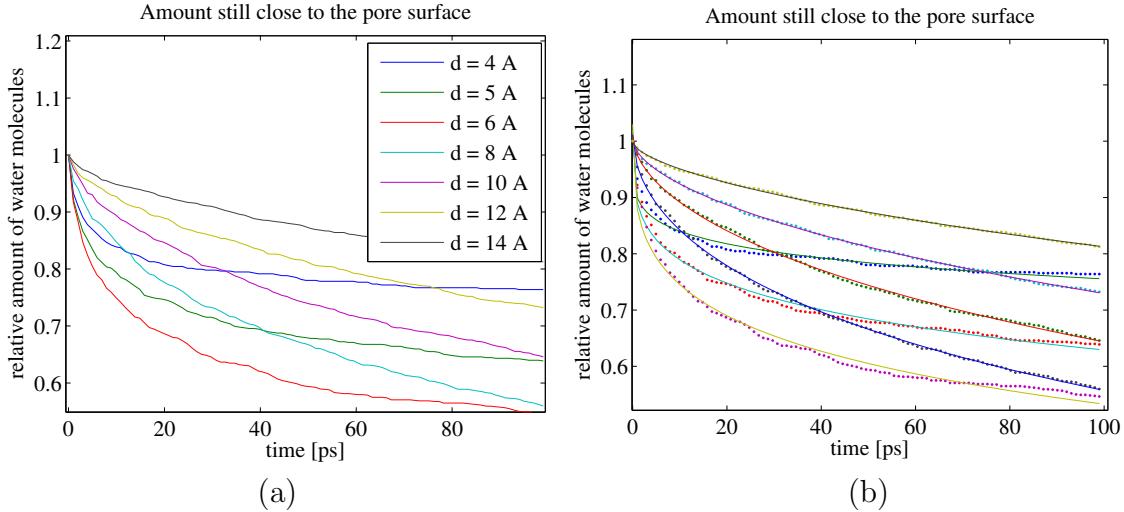


Figure 6.30: The relative amount of original surface water atoms after a time t . We fit to the function $B \exp(-(t/\tau)^\beta)$. The mean lifetime is plotted as a function of distance to the surface in figure 6.31.

We had to take the logarithm because the values for the shortest distances are much larger than the values at the longer distances, and without the logarithm, the plots were unreadable; the values for longer distances looked like they were zero. We see that the τ value for small distances d has a very high value, and that the value decreases quickly when d increases. A high value of τ means that the water molecules exit the region at a slow rate, they stay inside the region for a long time, while a low value of τ means that the water molecules leave quickly. Of course, while the molecules we measure leave, other molecules enter, but we are only concerned with the ones that were inside the region at $t = t_0$, the original molecules. At ca. 8 Å the value of τ starts increasing again, which means the exit rate is slowing down. It is important to remember that we are dealing with surface water, and that all the molecules that contributed at $d = 5$ Å, also contribute at $d = 7$ Å. When we increase d , the distance the inner molecules have to move in order to escape the region increases, so it is logical that the value of τ increases. In the section about the diffusion measurement we saw that the diffusion constant was lower close to the wall, compared to further away (see figure 6.10), which can be interpreted as the molecules closer to the wall move less than the molecules far away from the wall. When we increase d , we include more and more moving molecules to the measurement. The molecules that move more are quicker to escape than those who move less, and that is why the life span τ decreases for the smallest distances. We say we measure the life span of the water molecules, but in reality we only follow the oxygen atom in the water molecules, and assume that the hydrogen atoms comes with the oxygen atom. At the pore surface it may happen that the oxygen in the water switches places with the oxygen in the silica, so that the oxygen that belonged to the water now belongs to the silica. When this happens, the oxygen is still labeled water-oxygen, and is therefore included in our measurements. These few oxygen atoms may be the cause of the extremely high value of τ for the smallest d values, because they are more trapped inside the solid silica than inside the water. Other contributors to the high τ values may be the oxygen atoms placed out during the passivation of the pore wall.

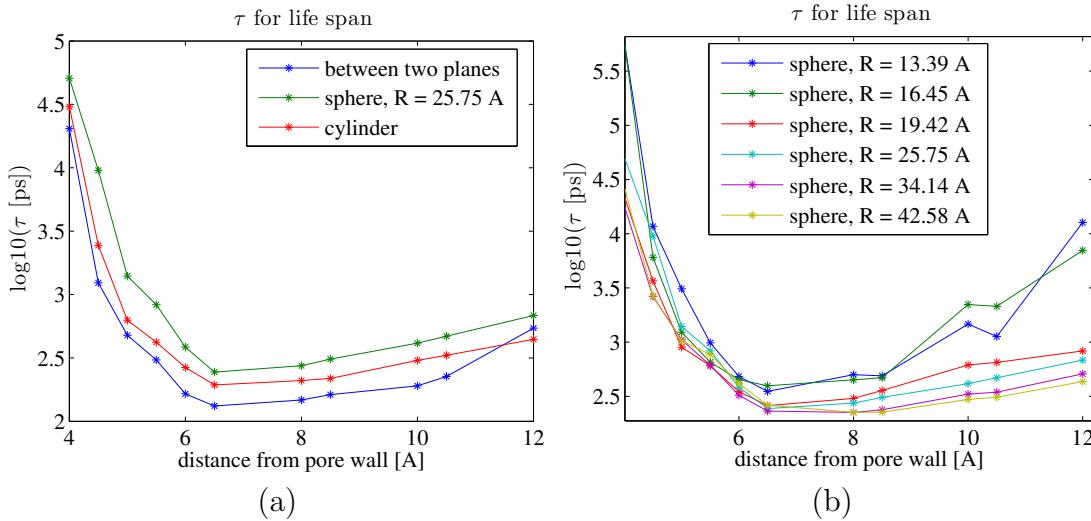


Figure 6.31: The logarithm of the mean lifetime, or τ , for the surface water as a function of distance to the pore wall, for (a) the three pore types: sphere, cylinder and two planes, (b) all spherical pores.

We understand that the functions in figure 6.31 are a result of two effects; when d increases, the number of movable molecules also increases, which pulls the value of τ down, but at the same time, when d increases the length each molecules has to move in order to escape also increases. The first effect is most dominant in the first half of the measured distances ($d > 8 \text{ \AA}$), while the second effect is most dominant in the last half. If we had continued measuring for larger and larger values of d , τ would reach infinity when d reaches the radius of the pore, because that would mean the entire pore was the region and there would be no escape.

We see that the sphere overall has a higher τ than the cylinder, and the cylinder has a higher τ than the plane curve, except for the last distance $d = 12 \text{ \AA}$. These differences can be due to the pore shape, which we also saw for the diffusion in surface water. When d increases, the number of molecules is not increased linearly, which means that the ratio between movable and less movable molecules for the spherical and cylindrical pores are lower than for the plane pore. The life span τ for the plane pore starts to increase for the larger d values because we are getting close to the height of the pore, while the two other pores have more to go on.

For the spherical pores, we see the same effect, the ones with higher curvature (smaller radius) have higher life span, τ , than the ones with smaller curvature. We see that the life span for the two smallest pores increase more rapidly than the other spherical pores, and this is probably because d is getting close to the pore radius, and the volume of the space that is not a part of the region decreases.

This measure for the life span gives a lot of effects that we need to take into account when interpreting the result. The shell water method is easier to understand.

Figure 6.32 (b) shows the relative amount of shell water as a function of time, for a medium sized spherical pore. Each function is made up by water molecules inside a shell of width $l = 3 \text{ \AA}$, at a distance d from the pore wall, which is specified by the legend of the plots. Figure 6.32 (a) shows the relative amount inside shells that have the same shapes as the shells used for plot (b), but these shells are in bulk water. The relative

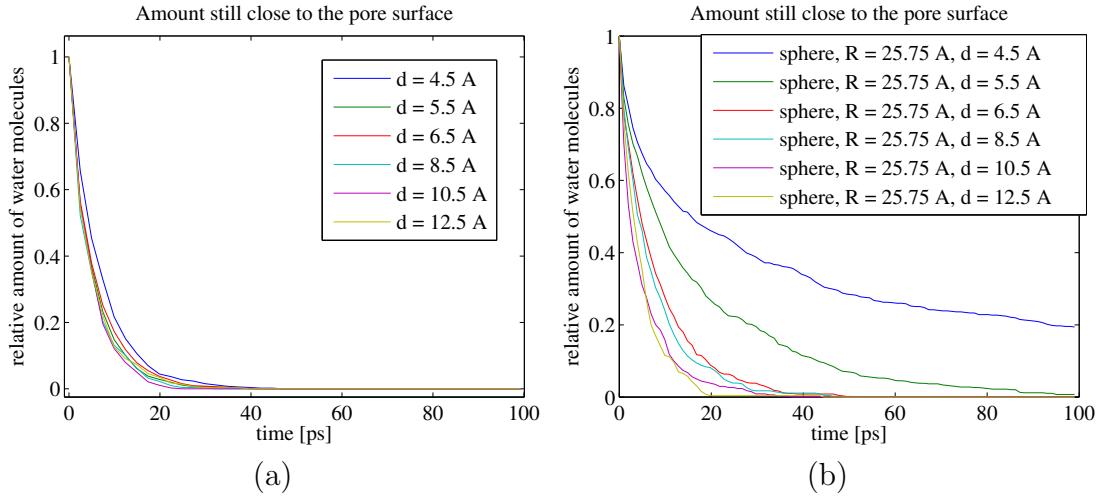


Figure 6.32: The relative amount of original water atoms in different shells after a time t . (a) The result for spherical shells in bulk water (b) The result for spherical shells in a spherical pore ($R = 25.75 \text{ \AA}$).

amount in bulk water was measured to check whether it is the shape of the shells that affects the results or if it is the distance distance d . The functions from the bulk water are not identical, but at the same time it is difficult to see a connection between the shape of the functions and the distance d . The fact that the function for $d = 4.5 \text{ \AA}$ lies somewhat higher than the function for $d = 10.5 \text{ \AA}$, makes it look like a larger d -value means shorter life span, τ , but the function for the largest d lies in between the $d = 4.5 \text{ \AA}$ and $d = 10.5 \text{ \AA}$ functions, which undermines that hypotheses. Regardless, the differences between the functions for the shell water inside a pore is much larger than the differences for the shell water in the bulk water, at least for $d < 6.5 \text{ \AA}$.

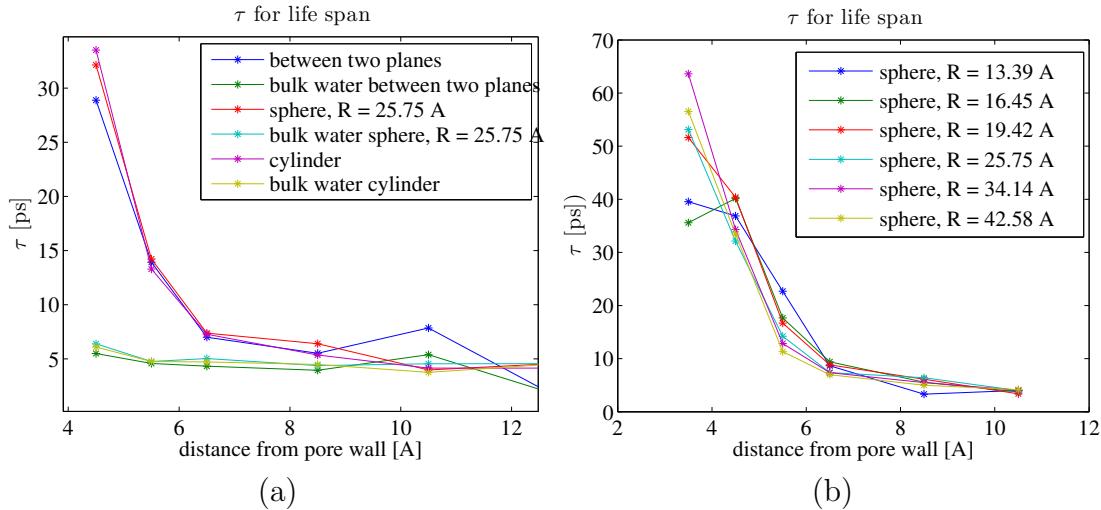


Figure 6.33: The mean life time of the shell water. (a) τ for cylindrical, two planes and spherical pore, compared to τ for same shells in bulk water. (b) τ for all the spherical pores. The mean life time for the same shells in bulk water was between 3.5 and 6.5.

The same tools were used on the relative amount functions, similar to the ones in

figure 6.32, to find the life span, τ , for multiple shells in all of our systems. In addition, the shell shapes from the three different pore shapes was measured on in bulk water. The results are shown in figure 6.33. In this case it was not necessary to take the logarithm of τ to make reasonable plots to look at. The values of τ are not extremely high like we saw in the surface water results, and that is probably because the oxygen atoms closer than $d = 3 \text{ \AA}$ to the wall are excluded. The exclusion happens because we are looking at shells of width $l = 3 \text{ \AA}$, and our smallest distance is $d = 4.5 \text{ \AA}$, which means that the minimum distance a molecule/atom can have to the pore wall and still contribute to the measurement is $4.5 \text{ \AA} - 3/2 \text{ \AA} = 3 \text{ \AA}$. The oxygen atoms that are within 3 \AA of the silica are typically those who have merged into the silica and the passivation atoms. We see that there is a large difference between the τ values for $d < 6 \text{ \AA}$ for the water inside the pore, and τ for the same shells found in the bulk water. The bulk water has a life span at $\tau \approx 5 \text{ ps}$ more or less for all the distances, while the water in the shell closest to the pore wall ($d = 4.5 \text{ \AA}$) has a life span of $\tau \approx 30 \text{ ps}$. As the distance to the wall increases, the τ value reaches the bulk $\tau \approx 5 \text{ ps}$. The water at $d = 4.5 \text{ \AA}$ stay inside their region six times longer than the water further inside the pore, and it is not due to oxygen trapped inside the silica or passivation atoms. It is not a large difference between the life spans in the three different pore shapes, as opposed to what we saw for the surface water. The life span for the spherical pores are also very similar to each other, which indicates that the life span is not dependent on the curvature of the pore wall or the size of the pore.

We see that the spherical pores also end up with a bulk value for the life span, after ca $d \approx 9 - 10 \text{ \AA}$. For the spherical pores, we have one more measuring point, $d = 3.5 \text{ \AA}$, which means that the smallest distance to the wall is $3.5\text{\AA} - 3\text{\AA}/2 = 2 \text{ \AA}$. τ is pretty large for the shell at $d = 3.5 \text{ \AA}$, but nothing like the enormous value we saw for the surface water. Apparently, we still have managed to exclude the silica-trapped water-oxygen. If a molecule/atom move out of the region, we say it has escaped even if it is back in the region the next time sample. This means that the molecules that vibrates right at the border of the region, is measured as escaped rather quickly, and the oxygen that are trapped inside the silica, has the opportunity to be measured as escaped. That was not the case with the surface water measurements.

Discussion and conclusions

7.1 Revisiting the goals

In section 1.4 we established the goals for this master project, and in the current section we will revisit them and discuss how the goals were met.

1. **Simulate systems of single pores of different geometric shapes such as spheres, cylinders and plane pore walls.**

We performed large scale computer simulations for a total of ten molecular dynamics systems: six systems with spherical pores of various radii, one system with a cylindrical pore, one with a plane pore, one with bulk water and one with bulk silica. The code which performs the heavy integrator tasks was developed by Vashishta et al, while most of the code used for preparation of the systems and for analysis is developed/further developed by the author, in cooperation with master student Filip Sund.

2. **Improve already existing methods for preparing a silica/water MD-system.**

This includes:

- **Developing C++ code for making pores in silica.** C++ code for cutting pores neutrally in the silica was created. This code is specialized to cut out different geometric shapes. The available shapes are spheres, boxes, cylinders and ellipsoids.
- **Further developing the method for passivating the pore walls, and implementing the method in C++ code.** We developed a method and corresponding code for passivation of a generic pore surface. Instead of placing the OH/H groups in an available voxel close to the Si/O with missing neighbors (old method), the placement of the group is based on the tetrahedral structure of the silica. With this method the groups are placed closer to the minima of the potential compared to the voxel method, which may reduce the time needed to spend on the steepest descent part of the preparation. Drawbacks to the tetrahedral method are that the new code is much more intricate than the old code, and the execution time is longer. The code, as it is constructed now, is very strict when it comes to inserting the groups. If there is lack of space at

the calculated position, we skip inserting at that site, even though there might have been room right next to the calculated position, which may lead to too few OH groups.

- **Develop a better method than the method used today for injecting water molecules into the system, and implementing the method in C++ code.** The method used to inject water for this thesis is based on a previously used code. The method was implemented in C++, with a few small alterations compared to the old code. One alteration was to insert the water molecules with random orientation, whereas the old code inserted all the water molecules pointing in the same direction. A weakness of the method is that the number of voxels in each direction has to be an integer. This means that the size of the voxels are slightly different from the ideal size and the density is slightly different from the requested density. This makes it more complicated to make conclusions based on the results, because we have to keep in mind that the densities differ from system to system. After the simulations was completed, a solution to this problem was discovered. Instead of approximating to the nearest integer we always round up when calculating the number of voxels and only insert molecules in a fraction of the voxels.

3. **Measure both diffusive and structural properties of water confined in pores and on bulk water. See if/how the measured results change with the curvature of the pores, and compare them with measured results for bulk water.** We measured the local density of the water, the tetrahedral order parameter, the diffusion constant, the cage correlation function and the life span on all eight pore systems. In addition, the radial distribution function was measured on bulk water and bulk silica. All code used for these measurements are either made by the author, by Filip Sund or by both. The different measurements were measured as a function of distance to the pore wall, either as surface/confined water or as shell water, or both. The results of these measurements will be discussed in the next section.

7.2 Summary of the results and discussion

In this thesis we have done a systematic investigation of how pore geometry affects the behavior of confined water, which, to the best of our knowledge, has not been done before. In this section we compare our results for the different properties measured on the water confined in the different pores. We start by looking at how the properties vary with the distance to the pore wall, before we discuss how the curvature, pore shape and size affect the different properties.

Dependence on the distance to the pore wall

The structure related properties:

The structure related properties are the local density and the tetrahedral order parameter (TOP).

We saw that the local density functions all follow the same density profile: a peak near the wall, a dip at ~ 5 Å from the wall before the function flattens out at a distance around 8 Å from the pore wall. These results suggest that a pore can be divided into two regions, the surface region and the confined region. The surface region is where the density varies due to interactions with the hydrophilic silica surface, while the confined region is where the density is constant. According to our results, the separation between the surface and the confined region is 8 Å. Compared to results from similar simulations [31][32], we learned that the density peak near the wall might be the second peak from a molecular layering structure that occur near the wall. For the tetrahedral order parameter (TOP), we saw that both the average value and the shape of the probability density function depend on the distance to the pore wall, for distances smaller than 7 Å. For distances larger than 7 Å, the TOP was constant and did not dependent on the distance to the pore wall. Similar to the local density results, the TOP results indicate that the pore can be divided into a surface region and a confined region, separated at the distance $d = 7$ Å. We saw that when we increased the distance to the pore wall, the probability density function of the TOP grew into a bimodal shape, which was in agreement with the bulk water results by Kumar et al. [34].

The diffusive properties.

The diffusive properties measured are the diffusion constant, the cage correlation function (CC function) and the life span. The diffusion constant had a low value near the pore wall, and as we increased the distance d to the pore wall, the diffusion constant grew until it reached a steady value at approximately 10 Å from the pore wall. We found that the diffusion constant could be crudely described with a twofold linear model

$$D(d) = \begin{cases} d \frac{D_\infty}{d_{\text{surf}}} & \text{if } d \leq d_{\text{surf}}, \\ D_\infty & \text{if } d > d_{\text{surf}}, \end{cases} \quad (7.1)$$

where d_{surf} is the distance that for our results could be 10 Å, and D_∞ is the diffusion constant far away from the pore wall, or the bulk value.

All cage correlation functions measured was fitted to a stretched exponential function, $S = e^{-(t/\tau)^\beta}$, and thereby described with the stretching parameter β and the mean lifetime parameter τ . We saw a trend that near the pore wall, the mean lifetime τ had a high value, and decreased towards a steady value for increasing distances to the wall. The steady value was reached within $d_{\text{surf}} = 10$ Å. The mean lifetime τ is a measure of how rapidly the cage correlation function fall, and can be interpreted as how rapidly the water molecules lose their neighbors. This means that the molecules close to the wall, where the τ was high, hold on to their neighbors longer than the molecules further inside the pore. This fits with the results for the diffusion constant, because molecules that contribute to a high diffusion constant tend to move around more which will make them lose their neighbors quicker than molecules that moves less and contribute to a low diffusion constant.

The life span is a measure of how long it takes for a fraction of the water molecules to leave a given region. It gives an estimate of how long the average water molecule stay inside a region. We measured this on shells, at various distances from the pore wall. The life span had a high value for the shells close to the pore wall, and it decreased towards a steady value which was reached within 9 – 10 Å. The life span maintained this steady value for distances larger than 10 Å from the pore wall, and it was very close to the bulk value. Comparing the life span at 4.5 Å away from the pore wall with the steady value,

we learned that the molecules close to the wall stay inside their region six times longer than the molecules in the middle of the pore. We believe this is mainly due to interactions with the hydrophilic silica surface. This measure is strongly connected to the diffusion constant measure, and we see that there are no contradictions when comparing the results for the two measures. It is natural that the molecules stay longer inside their region at shorter distances, when the diffusion constant here is smaller than further inside the pore.

The results suggests that a pore can be divided into two regions of different behaviors, a surface region and a confined region. The surface region is the region closer to the pore wall than a distance d_{surf} . In this region the properties of the water vary with the distance to the pore wall due to interactions with the hydrophilic surface. The confined region is the region of the pore that is further away from the pore wall than the distance d_{surf} . The properties in this region are constant for different distances to the pore wall. For the structure related properties we found this distance as $d_{\text{surf}} \sim 7 - 8 \text{ \AA}$, while for the diffusive properties it was $d_{\text{surf}} \sim 9 - 10 \text{ \AA}$.

Dependence on the curvature, pore shape and size

The purpose of this thesis was to determine how the curvature of the pore surface and the pore shape affect properties of the water. For the structure related properties we saw no dependence on the shape of the pore. We suspected that a curved surface would frustrate the packing or configuration of the water molecules, which would affect the structure related properties of the water, and that the degree of this frustration would depend on the curvature of the surface.

The structure related properties:

As previously discussed, the structure related properties was affected by the pore wall. However, the effect did not vary across the different pore shapes or the different curvatures. For the local density, there were differences between the three pore shapes, $\rho_{\text{cylinder}} < \rho_{\text{sphere}} < \rho_{\text{plane}}$, but this was due to the weakness in the water injection mechanism. When we looked at the relative density, we saw that the local density functions for the three pore shapes were approximately equal.

The diffusive properties:

For two of the diffusive properties, the diffusion constant and the cage correlation function, we saw a dependency on the radius of the pore, while the life span showed no dependence on the pore size. When all the water inside the pore was included in the measurement, the diffusion constant was approximately linearly dependent on the pore radius. However, the studies on the diffusion constant for water inside shells at various distances from the pore wall, revealed that there was no dependence on the curvature, only the distance to the pore wall. We believe the reason for the dependence of diffusion constant on the pore radius is related to the fact that as the radius is increased, the fraction of water affected by the pore wall decreases. How this fraction decreases is dependent on the shape of the pore.

When we plotted the mean lifetime parameter, τ , for the CC functions as a function of the pore radius for various distances d , we saw that it decreases as the pore radius increases. This can be interpreted as the water inside the larger pores hold on to their neighbors longer than the water inside the smaller pores. For the three pore shapes we had $\tau_{\text{plane}} < \tau_{\text{cylinder}} < \tau_{\text{sphere}}$, which means that the water molecules in the plane pore escapes

their neighbors more often than the molecules in the cylinder and that the molecules in the cylinder escapes their neighbors more often than the molecules in the sphere. Since the CC measurements were done with surface/confined water, and not with the water inside shells, we can not know if these results are due to the curvature of the pore, or if they are due to the shape of the pore, like we saw for the diffusion constant.

7.3 Future work

For future studies it would be advisable to optimize and make parallel versions of the most computationally heavy tools used to analyze the systems, particularly the cage correlation program which is very time consuming for large systems. In addition, it would be wise to improve the water injection mechanism before performing future simulations, to make sure the density received is the density requested.

In this thesis we had multiple spherical pores with various radii, but only one version of the cylindrical and the plane pore each. It would be useful to simulate a few additional systems with cylindrical and plane pores of varying size. For instance, we would have liked to find the diffusion constant as a function of d for distances larger than 10 Å for the plane pore, but we could not since our plane pore was narrow.

It would also be interesting to simulate systems with pores of other curvatures than we have here. We could simulate a system where the pore wall had inverted curvature, for instance by letting the silica be a spherical mass and have the water fill the rest of the cubic system.

Now that we have investigated physical properties in pores of varying curvature, the next step could be to investigate flow in such pores. Encapsulated water, like the water in a spherical pore, is not suitable to induce flow in, but the water inside the cylindrical pore, the plane pore and the future inverted pore can be forced to flow. If we strongly wished to investigate flow in the spherical pores nevertheless, we could create a hole or a cylindrical tunnel in the pore surface where water could flow.

We motivated the choice we made to only look at single pores of simple shapes, by stating that such pores can be viewed as components of models for complex pore space. It would have been interesting to simulate systems composed of multiple pores, and maybe even investigate the flow in such a system.

8

Appendix

8.1 Outline of the matlab lsqcurvefit

The matlab-function `lsqcurvefit` is designed to solve the problem of fitting a function, $f(\boldsymbol{\beta}, x)$, to a set of data points \mathbf{x} and \mathbf{y} , so that the least-squares is obtained. The mathematical formulation of the problem is to find the parameters $\boldsymbol{\beta}$ that minimizes the function

$$S(\boldsymbol{\beta}) = \sum_{i=0}^N (f(\boldsymbol{\beta}, \mathbf{x}_i) - \mathbf{y}_i)^2. \quad (8.1)$$

The vector $\boldsymbol{\beta}$ is the collection of all the parameters needed in the function $f(\boldsymbol{\beta}, x)$.

Examples of functions:

Quadratic functions: $f(\boldsymbol{\beta}, x) = \beta_2 x^2 + \beta_1 x \beta_0$, where $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2]$.

Exponential functions: $f(\boldsymbol{\beta}, x) = \beta_0 \exp(x/\beta_1)$, where $\boldsymbol{\beta} = [\beta_0, \beta_1]$.

The matlab-function `lsqcurvefit` has multiple options for the algorithm used for solving the problem. One of these algorithms is the Levencberg–Marquards algorithm.

Levenberg–Marquardt algorithm (LMA)

The derivation of LMA displayed in this section is highly influenced by an article about the algorithm from wikipedia [45].

The LMA is an iterative algorithm. It starts with an initial guess for the parameters, $\boldsymbol{\beta}^0$, and calculates a new suggestion for a solution, $\boldsymbol{\beta}^1$, that is hopefully closer to the real solution than $\boldsymbol{\beta}^0$ was. For the next iteration, the found suggestion $\boldsymbol{\beta}^1$ is used as the initial guess, and we obtain another suggestion for a solution, $\boldsymbol{\beta}^2$. Iteration number n takes $\boldsymbol{\beta}^{n-1}$ as an initial guess and calculates $\boldsymbol{\beta}^n$. This continues until the difference between $\boldsymbol{\beta}^{n-1}$ and $\boldsymbol{\beta}^n$ is smaller than a predefined limit, or the number of iteration has exceeded a predefined maximum.

The problem to solve was to minimize eq. 8.1, and at the minimum of eq. 8.1 the gradient with respect to $\boldsymbol{\beta}$ would be zero

$$\nabla_{\boldsymbol{\beta}} S(\boldsymbol{\beta}_{\min}) \equiv 0. \quad (8.2)$$

Solve eq. 8.2 and we have a solution.

Now we want to include the initial guess, β^0 , so we change the variable β to $\beta = \beta^0 + \Delta\beta$, which translates eq. 8.1 into

$$S(\beta^0 + \Delta\beta) = \sum_{i=0}^N (f(\beta^0 + \Delta\beta, \mathbf{x}_i) - \mathbf{y}_i)^2, \quad (8.3)$$

and realize that the gradient of eq. 8.1 with respect to β is the same as the gradient of eq. 8.3 with respect to $\Delta\beta$. We get a new equation to solve, which is equivalent to eq. 8.2

$$\nabla_{\Delta\beta} S(\beta^0 + \Delta\beta_{\min}) \equiv 0, \quad (8.4)$$

which means we need to find $\Delta\beta_{\min}$ instead of β_{\min} . In order to get a linear matrix-vector equation, which we are able to solve, we need to do the linear approximation

$$f(\beta^0 + \Delta\beta, \mathbf{x}_i) \approx f(\beta^0, \mathbf{x}_i) + \mathbf{J}_i(\beta^0)\Delta\beta, \quad (8.5)$$

where $\mathbf{J}_i(\beta^0)$ is the gradient of $f(\beta, \mathbf{x}_i)$ with respect to β evaluated at the initial guess, β^0 :

$$\mathbf{J}_i(\beta^0) = \nabla_{\beta} f(\beta^0, \mathbf{x}_i). \quad (8.6)$$

We insert the approximation into eq. 8.3 and get an approximation for the function we want to minimize

$$S(\beta^0 + \Delta\beta) \approx \sum_{i=0}^N (f(\beta^0, \mathbf{x}_i) + \mathbf{J}_i(\beta^0)\Delta\beta - \mathbf{y}_i)^2,$$

or in vector notation:

$$S(\beta^0 + \Delta\beta) \approx |\mathbf{f}(\beta^0, \mathbf{x}) + \mathbf{J}(\beta^0)\Delta\beta - \mathbf{y}|^2.$$

We find the gradient in eq. 8.4 with the approximated S and solve eq. 8.4. We end up with:

$$\mathbf{J}^T \mathbf{J} \Delta\beta_{\min} = \mathbf{J}^T (\mathbf{y} - \mathbf{f}(\beta^0)), \quad (8.7)$$

where $\mathbf{J} = \mathbf{J}(\beta^0)$. The right hand side of eq. 8.7 is a vector we can calculate, and the left hand side has a matrix vector product between the matrix $\mathbf{J}^T \mathbf{J}$ and the unknown vector $\Delta\beta_{\min}$, so eq. 8.7 is basically the equation $\mathbf{Ax} = \mathbf{b}$ which we are able to solve. Equation 8.7 was the basis Levenberg-Marquardt algorithm, to which they added a damping part to the left hand side. Levenberg added the damping part, and Marquardt improved it. The Levenberg damped algorithm looks like this

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \Delta\beta_{\min} = \mathbf{J}^T (\mathbf{y} - \mathbf{f}(\beta^0)),$$

where the damping factor λ is adjusted each iteration. When the solution approaches the minimum, λ is decreased, and if the solution gives a bad approximation, λ is increased.

Marquardt substituted the identity matrix, \mathbf{I} by a diagonal matrix where the diagonal elements are the same as the diagonal elements of $\mathbf{J}^T \mathbf{J}$ [46]. The result was the Levenberg-Marquardt algorithm:

$$(\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J}))\Delta\beta = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\beta)],$$

where λ is a damping factor and \mathbf{J} is the Jacobian matrix where the i^{th} row is

$$\mathbf{J}_i = \frac{\partial f(x_i, \beta)}{\partial \beta}(\beta_0).$$

8.2 Differential geometry

8.2.1 Proof: Curvature of regular curves

We will prove that the curvature of regular curves with arbitrary parametrization has the following form

$$\kappa(t) = \frac{|\alpha' \times \alpha''|}{|\alpha'|^3}. \quad (8.8)$$

Let $\alpha : I \rightarrow R^3$ be a regular parametrized curve and $\beta : J \rightarrow R^3$ be a reparametrization of α by the arc length s measured from t_0 . Let $t(s)$ be the inverse function of s , which gives us $\beta(s) = \alpha(t(s))$.

The derivatives written in short-hand notation

$$\begin{aligned}\dot{\alpha} &= \frac{d\alpha}{ds}, \\ \ddot{\alpha} &= \frac{d^2\alpha}{ds^2}, \\ \alpha' &= \frac{d\alpha}{dt}, \\ \alpha'' &= \frac{d^2\alpha}{dt^2}.\end{aligned}$$

The definition of arc length is

$$s(t) = \int_{t_0}^t |\alpha'(t)| dt.$$

We will use relations from vector calculus

$$\begin{aligned}|a \times b|^2 &= |a|^2 |b|^2 - (a \cdot b)^2, \\ |a - b| &= \sqrt{|a|^2 + |b|^2 - 2(a \cdot b)}, \\ \frac{d|v(t)|}{dt} &= \frac{(v \cdot v')}{|v|}.\end{aligned}$$

First we need to find the first and second derivative of t as a function of s . Start with the definition of arc length

$$s(t) = \int_{t_0}^t |\alpha'(t)| dt.$$

Since we have a regular curve we know that $|\alpha'(t)| \neq 0$ and we can write $\frac{ds}{dt} = |\alpha'|$. But we wanted the derivative of t , so we use the physician way of finding it, namely the inverse of $\frac{ds}{dt}$ giving us

$$\frac{dt}{ds} = \frac{1}{|\alpha'|}.$$

The second derivative

$$\frac{d^2 t}{ds^2} = \frac{d\left(\frac{1}{|\alpha'|}\right)}{dt} \frac{dt}{ds} = \frac{1}{|\alpha'|} \left(-\frac{1}{|\alpha'|^2} \frac{d|\alpha'|}{dt} \right) = \frac{-1}{|\alpha'|^3} \frac{(\alpha' \cdot \alpha'')}{|\alpha'|} = -\frac{(\alpha' \cdot \alpha'')}{|\alpha'|^4}.$$

Let's get to it. The definition of curvature for parametrized

$$\kappa(s) = \left| \frac{d^2 \alpha(t(s))}{ds^2} \right| = |\ddot{\alpha}|.$$

Start by differentiating the vector α , and later find the length.

$$\dot{\alpha} = \frac{d\alpha}{ds} = \frac{d\alpha}{dt} \frac{dt}{ds} = \frac{\alpha'}{|\alpha'|}, \tag{8.9}$$

$$\begin{aligned} \ddot{\alpha} &= \frac{d\dot{\alpha}}{ds} = \frac{d\dot{\alpha}}{dt} \frac{dt}{ds} = \frac{d\dot{\alpha}}{dt} \frac{1}{|\alpha'|} \\ &= \frac{\alpha''|\alpha'| - \frac{(\alpha' \cdot \alpha'')}{|\alpha'(t)|} \alpha'}{|\alpha'|^3} \\ &= \frac{\alpha''|\alpha'|^2 - (\alpha' \cdot \alpha'')\alpha'}{|\alpha'|^4}, \end{aligned} \tag{8.10}$$

$$\begin{aligned} |\ddot{\alpha}| &= \frac{|\alpha''|\alpha'|^2 - (\alpha' \cdot \alpha'')\alpha'|}{|\alpha'|^4} \\ &= \frac{1}{|\alpha'|^4} \sqrt{|\alpha''|^2|\alpha'|^4 + (\alpha' \cdot \alpha'')^2|\alpha'|^2 - 2|\alpha'|^2(\alpha' \cdot \alpha'')(\alpha'' \cdot \alpha')} \\ &= \frac{\sqrt{|\alpha''|^2|\alpha'|^2 - (\alpha' \cdot \alpha'')^2}}{|\alpha'|^3} = \frac{\sqrt{(\alpha' \times \alpha'')^2}}{|\alpha'|^3} \\ &= \frac{|\alpha' \times \alpha''|}{|\alpha'|^3}. \end{aligned} \tag{8.11}$$

We have now found the expression for curvature as

$$\kappa(t) = |\ddot{\alpha}| = \frac{|\alpha' \times \alpha''|}{|\alpha'|^3},$$

QED.

8.2.2 Proof: Curvature of plane curves

We will prove that for plane curves parametrized as $\alpha(t) = [x(t), y(t), 0]$ the curvature can be found by

$$\kappa(t) = \left| \frac{x'y'' - x''y'}{[(x')^2 + (y')^2]^{3/2}} \right|. \quad (8.12)$$

This is just a matter of inserting into eq. 8.8. We need the derivatives of $\alpha(t)$

$$\alpha'(t) = [x', y', 0], \quad \alpha''(t) = [x'', y'', 0].$$

According to the formula we need the cross product of the derivatives

$$\alpha' \times \alpha'' = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x' & y' & z' \\ x'' & y'' & z'' \end{vmatrix} = [0, 0, x'y'' - y'x''].$$

The length of this vector is easily found as

$$|\alpha' \times \alpha''| = |x'y'' - y'x''|.$$

We also need the length of the first derivative for the formula 8.8

$$|\alpha'| = [(x')^2 + (y')^2]^{1/2},$$

and we are ready to find the curvature of plane regular curves

$$\kappa(t) = \frac{|\alpha' \times \alpha''|}{|\alpha'|^3} = \frac{|x'y'' - y'x''|}{[(x')^2 + (y')^2]^{3/2}}.$$

We notice that it is possible to keep the sign (remove the absolute value in eq. 8.12. This only works for plane curves. So, we do that and call the result the signed curvature.

$$k(t) = \frac{x'y'' - x''y'}{[(x')^2 + (y')^2]^{3/2}}, \quad (8.13)$$

QED.

8.3 Figures

Results from article

This section contains results from an article called “Molecular simulation of water confined in nanoporous silica”[31], where they look at two small systems with a silica slit nanopore, which is the same as we denote a plane pore, of height $H = 10, 20$ Å. In figure 8.1 we see the density profiles of the oxygen atoms and the hydrogen atoms in the water as a function of distance to the pore wall.

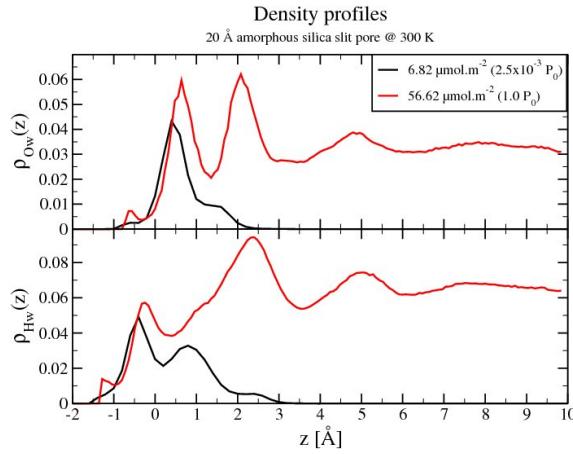


Figure 8.1: (Color online) Density profiles for oxygen and hydrogen in the water inside a silica slit nanopore (plane pore) of height $H = 20 \text{ \AA}$, collected from the article [31]. The red function is from a system similar to the plane pore system, meaning the entire pore is filled with water molecules. The black function is from a system that has fewer molecules.

Results from another article

This section contains results from an article called “Structure and dynamics of water confined in silica nanopores” [32]. Figure 8.2 shows density profiles of distance to the pore center (!) of cylindrical pores. Figure 8.3 contains MSD functions for the water inside the cylindrical pores. Figure 8.4 has the parallel and perpendicular diffusion constants as a function of distance to pore center.

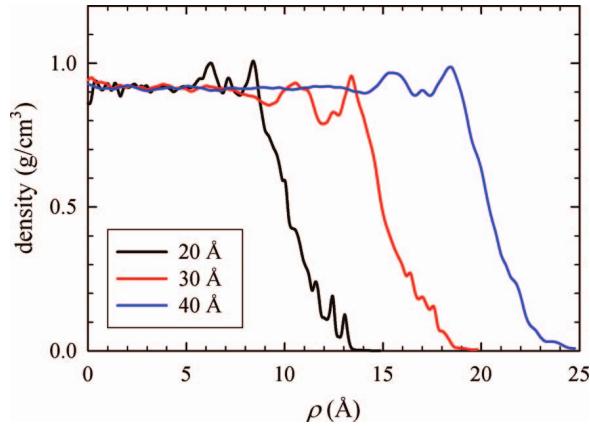


Figure 8.2: (Color online) Density profiles of water inside cylindrical silica nanopores, collected from the article [32].

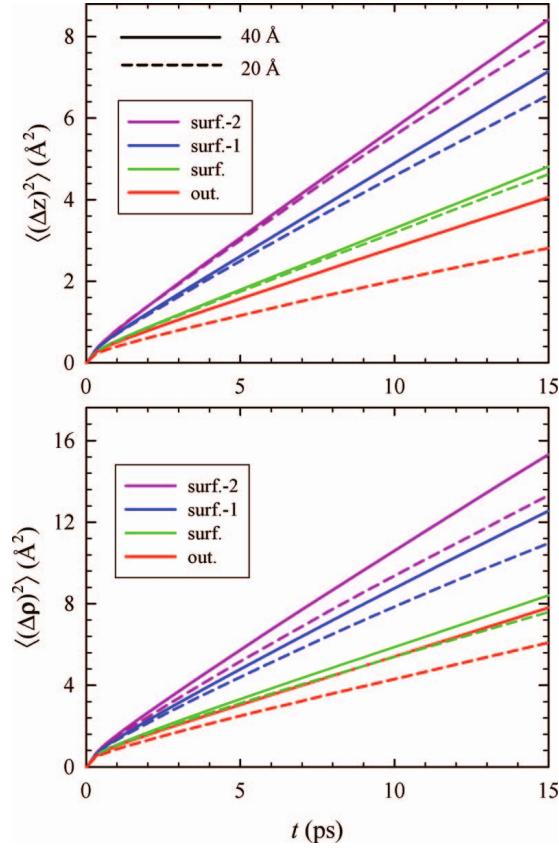


Figure 8.3: (Color online) (top) Mean square displacement in the axial direction of the cylinders, (bottom) mean square displacement in the radial direction, collected from the article [32].

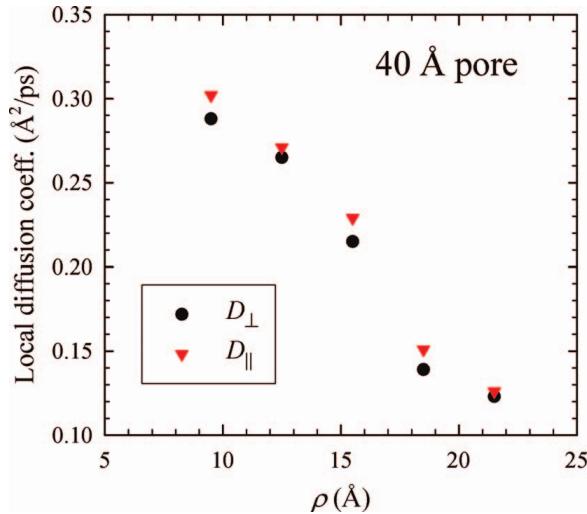


Figure 8.4: (Color online) Diffusion constants in the radial and axial direction of the cylinder shaped pore, as a function of distance to the pore center, collected from the article [32].

Bibliography

- [1] KU Leuven MTM. *PICTURE: porous ceramic*. URL: <https://www.mtm.kuleuven.be/Onderzoek/Ceramics/Materials> (visited on 05/27/2014).
- [2] University of Washington Rob Harrill. *PICTURE: Swiss Cheese Polymer*. URL: <http://www.washington.edu/news/2000/05/22/uw-partners-with-advanced-tissue-sciences-and-others-in-10-million-grant-to-grow-human-heart-tissue> (visited on 05/27/2014).
- [3] Frederick A. Mumpton. “La roca magica: uses of natural zeolites in agriculture and industry”. In: *Proceedings of the National Academy of Sciences* 96.7 (1999), pp. 3463–3470.
- [4] Store Norske Leksikon. *Silikatmineraler*. URL: <http://snl.no/silikatmineraler> (visited on 05/27/2014).
- [5] Wikipedia. *PICTURE: Quartz*. URL: http://commons.wikimedia.org/wiki/File:Pure_Quartz_at_Senckenberg_Natural_History_Museum.jpg (visited on 05/27/2014).
- [6] Wikipedia. *PICTURE: Silica gel beads*. URL: <http://en.wikipedia.org/wiki/File:SilicaGel.jpg> (visited on 05/27/2014).
- [7] desiccantpacks. *PICTURE: Silica gel packets*. URL: <http://www.desiccantpacks.net/silica-gel-packets> (visited on 05/27/2014).
- [8] Shuichi Takahara et al. “Neutron scattering study on dynamics of water molecules in MCM-41”. In: *The Journal of Physical Chemistry B* 103.28 (1999), pp. 5814–5819.
- [9] Limei Xu et al. “Appearance of a fractional Stokes–Einstein relation in water and a structural interpretation of its onset”. In: *Nature Physics* 5.8 (2009), pp. 565–569.
- [10] F Mallamace et al. “The fragile-to-strong dynamic crossover transition in confined water: nuclear magnetic resonance results”. In: *The Journal of chemical physics* 124.16 (2006), p. 161102.
- [11] P. Vashishta et al. “Interaction potential for SiO₂: A molecular-dynamics study of structural correlations”. In: *Phys. Rev. B* 41.17 (1990), pp. 12197–12209.
- [12] UiO. *Abel Computing Cluster*. URL: <http://www.uio.no/english/services/it/research/hpc/abel> (visited on 05/27/2014).

- [13] Manfredo Perdigao Do Carmo. *Differential geometry of curves and surfaces*. Prentice-Hall Englewood Cliffs, 1976, pp. 1–151.
- [14] Wolfram Mathworld. *Gaussian Curvature*. URL: <http://mathworld.wolfram.com/GaussianCurvature.html> (visited on 11/02/2013).
- [15] Wolfram Mathworld. *Developable surface*. URL: <http://mathworld.wolfram.com/DevelopableSurface.html> (visited on 11/01/2013).
- [16] Martin Karplus, Gregory A Petsko, et al. “Molecular dynamics simulations in biology”. In: *Nature* 347.6294 (1990), pp. 631–639.
- [17] Berni J. Alder and T.E. Wainwright. “Studies in molecular dynamics. I. General method”. In: *The Journal of Chemical Physics* 31.2 (2004), pp. 459–466.
- [18] A. Rahman. “Correlations in the motion of atoms in liquid argon”. In: *Physical Review* 136.2A (1964), A405–A411.
- [19] Adri C.T. Van Duin et al. “ReaxFF: a reactive force field for hydrocarbons”. In: *The Journal of Physical Chemistry A* 105.41 (2001), pp. 9396–9409.
- [20] P. Vashishta et al. *Interatomic potential for SiO₂ and H₂O*. URL: <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.111.184503#supplemental> (visited on 06/05/2014).
- [21] W. Wang et al. “Interatomic potentials for molecular dynamics simulations of hydrolysis and stress corrosion cracking of silica glass”.
- [22] Loup Verlet. “Computer ‘experiments’ on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules”. In: *Physical review* 159.1 (1967), pp. 98–103.
- [23] Daan Frenkel and Berend Smit. *Understanding Molecular Dimulation*. second edition. Computational Science series. Academic Press, 2012, p. 638.
- [24] Wikipedia. *PICTURE: Voronoi Diagram*. URL: http://en.wikipedia.org/wiki/File:Euclidean_Voronoi_Diagram.png (visited on 11/05/2013).
- [25] Finn Ravndal and Eirik Grude Flekkøy. *Statistical Physics - a second course*. Sept. 2008, p. 213.
- [26] Ovito. URL: <http://www.ovito.org> (visited on 06/02/2014).
- [27] Mindat. *mineral database*. URL: <http://www.mindat.org/min-3337.html> (visited on 11/05/2013).
- [28] Mohammad Vedadi et al. “Structure and dynamics of shock-induced nanobubble collapse in water”. In: *Physical review letters* 105.1 (2010), p. 014503.
- [29] Tetsuya Kodama, Michael R. Hamblin, and Apostolos G. Doukas. “Cytoplasmic molecular delivery with shock waves: importance of impulse”. In: *Biophysical Journal* 79.4 (2000), pp. 1821–1832.
- [30] Tetsuya Kodama and Kazuyoshi Takayama. “Dynamic behavior of bubbles during extracorporeal shock-wave lithotripsy”. In: *Ultrasound in medicine & biology* 24.5 (1998), pp. 723–738.
- [31] P.A. Bonnaud, B. Coasne, and R. J-M. Pellenq. “Molecular simulation of water confined in nanoporous silica”. In: *Journal of Physics: Condensed Matter* 22.28 (2010), p. 284110.

- [32] Anatoli A. Milischuk and Branka M. Ladanyi. "Structure and dynamics of water confined in silica nanopores". In: *The Journal of chemical physics* 135.17 (2011), p. 174709.
- [33] <http://www.chem1.com>. *About Water*. URL: <http://www.chem1.com/acad/sci/aboutwater.html> (visited on 05/05/2014).
- [34] Pradeep Kumar, Sergey V. Buldyrev, and H. Eugene Stanley. In: *Proceedings of the National Academy of Sciences* 106.52 (2009), pp. 22130–22134.
- [35] MathWorks. *Polyfit, Matlab*. URL: <http://www.mathworks.se/help/matlab/ref/polyfit.html> (visited on 05/27/2014).
- [36] Eran Rabani, J. Daniel Gezelter, and B.J. Berne. "Calculating the hopping rate for self-diffusion on rough potential energy surfaces: Cage correlations". In: *The Journal of chemical physics* 107.17 (1997), pp. 6867–6876.
- [37] Rudolph Kohlrausch. "Theorie des elektrischen Rückstandes in der Leidener Flasche". In: *Annalen der Physik* 167.2 (1854), pp. 179–214.
- [38] M. Berberan-Santos, E.N. Bodunov, and B. Valeur. "History of the Kohlrausch (stretched exponential) function: Pioneering work in luminescence". In: *Annalen der Physik* 17.7 (2008), pp. 460–461.
- [39] Alfred Werner. "Quantitative Messungen der An-und Abklingung getrennter Phosphoreszenzbanden". In: *Annalen der Physik* 329.11 (1907), pp. 164–190.
- [40] Graham Williams and David C. Watts. "Non-symmetrical dielectric relaxation behaviour arising from a simple empirical decay function". In: *Trans. Faraday Soc.* 66.0 (1970), pp. 80–85.
- [41] Manuel Cardona, Ralph V. Chamberlin, and Werner Marx. "The history of the stretched exponential function". In: *Annalen der Physik* 16.12 (2007), pp. 842–845.
- [42] Peter Grassberger and Itamar Procaccia. "The long time properties of diffusion in a medium with static traps". In: *The Journal of Chemical Physics* 77.12 (1982), pp. 6281–6284.
- [43] M. D. Ediger. "Spatially heterogeneous dynamics in supercooled liquids". In: *Annual review of physical chemistry* 51.1 (2000), pp. 99–128.
- [44] MathWorks. *Curvefit matlab*. URL: <http://www.mathworks.se/help/optim/ug/lsqcurvefit.html> (visited on 04/09/2014).
- [45] Wikipedia. *Levenberg Marquardt algorithm*. URL: http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm (visited on 04/09/2014).
- [46] Donald W. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the Society for Industrial & Applied Mathematics* 11.2 (1963), pp. 431–441.