Calculation of the Capacitance of Single Ice Crystals and Aggregates of Crystals via a Monte-Carlo Walk-on-Spheres Algorithm

Thomas Hayes
School of Physics and Astronomy,
University of Manchester,
Oxford Road,
Manchester
M13 9PL
Email: thomas.hayes-3@student.manchester.ac.uk

This investigation was performed in collaboration with Zach Waller

***Abstract*—A Monte-Carlo 'walk-on-spheres' algorithm for calculating the capacitance of ice crystals was implemented in MATLAB. The code was tested by calculating the capacitance of a unit cube which was found to be 0.6612 ± 0.0055, in good agreement with previously calculated values. The dependence of the capacitance of hexagonal plates and columns on their aspect ratio A was investigated and was found to vary as $C = 0.58(1 + 0.936A^{0.76})$. The capacitances of aggregates of hexagonal plates of aspect ratio 0.25 were determined. The normalised capacitance C/D_{max} (D_{max} is the maximum dimension in the crystal) was found to decrease as more plates are added to the chain. We expect it to reach a constant value in the range 0.10-0.13 as the chains increase in length.**

The code written during this investigation may now be used to calculate the capacitance of any ice crystal geometry or indeed other quantities such as the electrostatic capacitance or hydrodynamic friction of objects.

I. Aims

The aims of this project were:

1) To understand the growth of ice crystals by vapour diffusion and the importance of capacitance to this process.
2) To understand the 'walk-on-spheres' Monte Carlo method of calculating the capacitance of arbitrarily shaped ice crystals.
3) To implement the 'walk-on-spheres' algorithm in MATLAB.
4) To test the implementation of the algorithm by calculating the capacitance of shapes for which the capacitance is known such as cubes.
5) To calculate the capacitance of realistic ice crystal habits including hexagonal plates with varying aspect ratios and scalene hexagonal plates.
6) To calculate the capacitance of more complex aggregates of hexagonal ice crystals.

II. Introduction

Many types of cloud exist in the atmosphere. Some, such as stratus, are composed of water droplets suspended in the air, others, such as cirrus, are composed of ice crystals while still more are a mixture of the two phases. The main process by which ice crystals in clouds grow is by diffusion of water vapour onto their surfaces. It is important for meteorologists and other scientists to know how quickly ice crystals will grow as this affects many aspects of weather, such as cloud cover and precipitation rates. Unfortunately it is extremely difficult to directly solve the equations the equations which govern vapour diffusion growth due to the geometry of most realistic ice crystal types which include sharp corners and edges and other features which complicate the calculation. Due to a mathematical correspondence between vapour diffusion and electrostatics the growth rate of a crystal is proportional to a factor called the capacitance which is equal (up to a numerical constant) to the electrical capacitance of a conductor of the same size and shape as the crystal. The capacitance may only be calculated analytically for a few simple geometries such as spheres

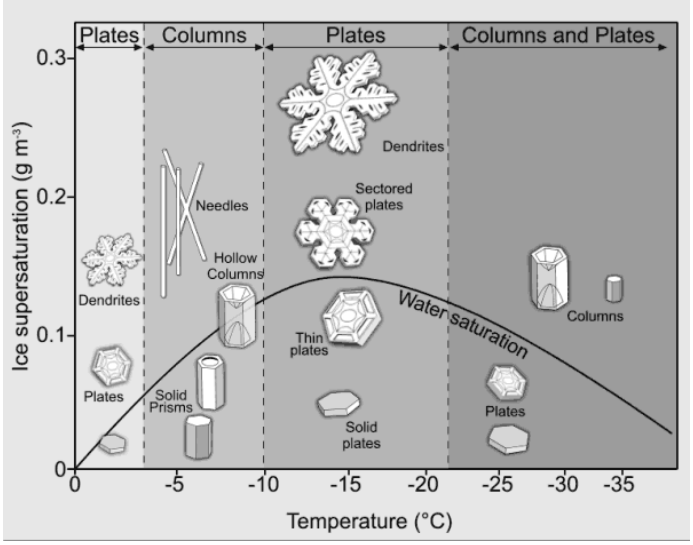

Figure 1. Diagram showing how common ice crystal morphologies form as a function of temperature and supersaturation [3].

so it is necessary to rely on alternative methods to determine it. Previously meteorologists had used the capacitance of similiarly shaped conductors for which the capacitance had been measured experimentally. For example a conducting cylinder was used in place of a hexagonal column. However this an unwieldy method as it involves making a physical model of every ice crystal geometry for which we desire to know the capacitance. Since the advent of modern computation the capacitance may instead be calculated using numerical methods such as finite-element or Monte-Carlo methods.

This report describes the calculation of the capacitance of various ice crystal geometries using a 'walk-on-spheres' Monte-Carlo method. The main aim of the project was to calculate the capacitance of complex aggregates of hexagonal ice prisms. These chain shaped aggregates are commonly found in anvil clouds in thunderstorms where the electric field causes the crystals to line up and fuse together along the field lines [1]. The method followed is largely based on that of Westbrook et al. [2] who implemented the 'walk-on-spheres' algorithm and calculated the capacitances of many of the most common types of ice crystal shapes.

III. Theoretical Background

A. Vapour diffusion growth of ice crystals and capacitance

Ice crystals in clouds may grow via a number of processes. One of the most important of these processes is the sublimation of water vapour in the air directly onto the ice crystal. We will now discuss the theory behind this process. The vapour density of water at any point, $\rho_v(\boldsymbol{r})$, and the current density, $\mathbf{j}(\boldsymbol{r})$, are related by Fick's law of diffusion:

$$\mathbf{j} = -D\boldsymbol{\nabla}\rho_v \tag{1}$$

where D is the diffusion constant of water vapour in air at the temperature in question. $\mathbf{j}$ and ρ_v both obey the continuity equation:

$$\frac{\partial \rho_v}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{j} = \sigma \tag{2}$$

where σ represents any sources or sinks of vapour. As there are no sources or sinks except at the surface of the crystal and we may assume that the crystal is moving slowly enough relative to the background that we may use the steady-state approximation and set the time-derivative equal to zero, we find that ρ_v obeys Laplace's equation in the space around the crystal:

$$\boldsymbol{\nabla}^2 \rho_v = 0 \tag{3}$$

The mass growth rate of a crystal is equal to the total flux of water vapour through the surface, S, of the crystal:

$$\frac{dm}{dt} = \int_S D(\boldsymbol{\nabla}\rho_v) \cdot d\boldsymbol{S} = 4\pi C D(\rho_{v,\infty} - \rho_{v,S}) \tag{4}$$

where $\rho_{v,\infty}$ and $\rho_{v,S}$ respectively represent the vapour density an infinite distance away from the crystal and its surface and C is a numerical factor which we will call the capacitance [2]. C is defined as

$$C = \frac{\int_S (\boldsymbol{\nabla}\rho_v) \cdot d\boldsymbol{S}}{4\pi(\rho_{v,\infty} - \rho_{v,S})}. \tag{5}$$

C is called the capacitance as it is mathematically equivalent to the electrostatic capacitance of a perfect conductor of the same size and shape as the ice crystal as we shall now see [4]. In the absence of charge (equivalent to sources or sinks of electric field) the electrostatic potential, Φ, follows Laplace's equation:

$$\boldsymbol{\nabla}^2 \Phi = 0. \tag{6}$$

Using the definition of capacitcance $Q = C_e V$ and Gauss' law, the charge on a conductor with surface S is given by:

$$\int_S (\boldsymbol{\nabla}\Phi) \cdot d\boldsymbol{S} = -\frac{Q}{\epsilon_0} = -\frac{1}{\epsilon_0} C_e(\Phi_S - \Phi_\infty) \tag{7}$$

where C_e is the electrostatic capacitance of the conductor, ϵ_0 is the permittivity of free space and Φ_∞ and Φ_S are the values of the potential an infinite distance away from, and at the surface of, the conductor respectively. Rearranging we obtain

$$C_e = \frac{\epsilon_0 \int_S (\boldsymbol{\nabla}\Phi) \cdot d\boldsymbol{S}}{(\Phi_\infty - \Phi_S)}. \tag{8}$$

Thus, if Φ and ρ_v satisfy certain boundary conditions there is an equivalence between C and C_e up to a numerical factor of $4\pi\epsilon_0$:

$$C_e = 4\pi\epsilon_0 C. \tag{9}$$

The first boundary condition that must be satisfied for 9 to hold are that the vapour density, $\rho_{v,\infty}$, and the electrostatic potential, Φ_∞, an infinite distance from the capacitor must be equal to zero. The other condition is that the vapour density $\rho_{v,S}$ and the potential Φ_S at the surface must be constant at all points on the surface. The potential is constant at every point on the surface due to the fact that we are discussing a perfect conductor and therefore the internal distribution of charges will cause the potential to have a uniform value over the surface. The uniformity of the vapour density over the surface is actually dependent on the fact that the air just above the surface is at the saturation vapour density of water above an ice surface and that there is a uniform temperature across the whole surface, as may be revealed by a detailed analysis of latent heat and heat conduction around the crystal [4]. To simplify the equations further we set $\rho_{v,S}$ and Φ_S equal to 1 and obtain:

$$C = -\frac{1}{4\pi} \int_S (\boldsymbol{\nabla}\rho_v) \cdot d\boldsymbol{S} \tag{10}$$

and:

$$C_e = -\epsilon_0 \int_S (\boldsymbol{\nabla}\Phi) \cdot d\boldsymbol{S} \tag{11}$$

B. Overview of the Monte-Carlo algorithm for calculating capacitance

We used a Monte-Carlo algorithm, based on an ensemble of randomly diffusing particles, to calculate the capacitance of ice crystals of various geometries. Let us start by discussing the statistics of particles diffusing towards an object with surface S. The diffusing particles begin an infinite distance away from the object with a uniform probability density which we set equal to 1. The total flux of particles through the surface S is given by :

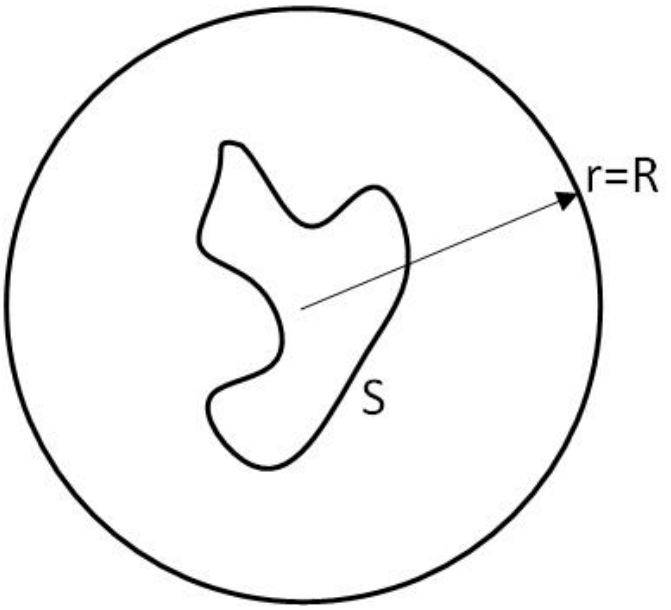

Figure 2. Schematic diagram illustrating the crystal with surface S inside a sphere with radius R. f is the probability of a walker reaching S after starting on the sphere with radius R.

$$k_S = k_R f \tag{12}$$

where k_R is the overall inward flux of particles through a sphere of radius R surrounding S and f is the fraction of particles that start on the sphere R that go on to reach S and become absorbed [5]. The overall current of diffusing particles, $\boldsymbol{j}_{diff}$ is given by:

$$\boldsymbol{j}_{diff} = D\boldsymbol{\nabla}\rho_v \tag{13}$$

where $p(\boldsymbol{r})$ is the probability density D is a diffusion constant deriving from the statistics of the particles.

In the steady-state, given that the diffusing particles are conserved in the absence of sources or sinks, p follows the Laplace equation in the region surrounding the crystal:

$$\boldsymbol{\nabla}^2 p = 0. \tag{14}$$

The boundary conditions on $p(\boldsymbol{r})$ are that it is equal to one an infinite distance from the crystal and zero at the surface as all particles are absorbed. On both boundaries the vapour density, ρ_v, and probability density of diffusing random walkers p are connected by:

$$\rho_v = \rho_0 (1 - p) \tag{15}$$

where ρ_0 is the standard density of water (needed to match the dimensions on both sides of the equation). Since both ρ_v and $1 - p$ follow Laplace's equation and have the same boundary conditions, by the uniqueness theorem for Poisson's equation (of which Laplace's equation is a special case) they must have the same value at all points between the boundaries [6]. Thus 15 holds at every point in the domain. Combining equations 10 and 15 we obtain:

$$C = -\frac{1}{4\pi}\int_S (\boldsymbol{\nabla}\rho_v)\,.\,d\boldsymbol{S} = \frac{1}{4\pi}\int_S (\boldsymbol{\nabla}p)\,.\,d\boldsymbol{S}. \quad (16)$$

The relation between k and C is thus:

$$k = D\int_S (\boldsymbol{\nabla}p)\,.\,d\boldsymbol{S} = 4\pi DC. \quad (17)$$

Returning to 12 we can choose a radius R large enough so that the flux of diffusing particles through R is isotropic, in this case k_R is given by: [7]

$$k_R = 4\pi DR. \quad (18)$$

Using 12 and 17 we finally obtain:

$$C = fR. \quad (19)$$

Thus the problem of calculating the capacitance of a crystal is reduced to that of calculating the fraction of random walkers which begin on a sphere of radius R that are absorbed by the crystal. Given that the rate of mass growth of the crystal (which is proportional to the capacitance and is the reason we are interested in the capacitance anyway) is essentially the rate at which water molecules diffuse onto the surface and undergo sublimation this is quite an intuitive and physically satisfying method of solving this problem.

Another advantage of this approach is that there is no intrinsic error the calculation of the capacitance, unlike other approaches such as finite-element methods. The only source of error is the sampling error of the Monte Carlo method and this decreases as $\mathcal{O}(N^{1/2})$ where N is the number of walkers being sampled. Thus by increasing the number of walkers (letting the simulation run for longer) the uncertainty can be made arbitrarily small [8].

C. Details of the Monte-Carlo 'walk-on-spheres' algorithm

The above outline of the model suggests the following implementation of the Monte-Carlo algorithm. A random walker is placed somewhere on a large sphere of radius R with R being sufficiently large that the vapour density would be spherically symmetric at this distance. The point is then released from its starting position and moves in a random direction with a step size much smaller than the the length scale of the crystal being investigated (say $\frac{1}{1000}$ times the smallest dimension of the crystal). The algorithm is called a 'walk-on-spheres' algorithm because the random walker moves to a random point on the surface of a sphere with its radius equal to the length of the step. This process is then repeated with the particle taking further steps until it has been judged to either have hit the crystal or else escaped to an infinite distance. Of course it is impossible to simulate the motion of a particle out to an infinite distance (unless one has infinite time to perform the simulation, which we certainly did not) so a cutoff point must be chosen. We choose to consider the walker as having escaped to infinity when it passes beyond a much larger sphere of radius R_∞. If we choose R_∞ to be a large multiple of R, say $500 \times R$, then it is reasonable to assume that the probability of the walker coming back and colliding with the crystal is vanishingly small and thus it can be considered to have escaped to an infinite distance away (for illustration, even if the crystal fully filled the volume inside R, for $R_\infty = 500 \times R$ then the crystal only occupies $1/500^3 = 8 \times 10^{-9}$ of the volume inside R_∞ so it is very unlikely to come back and strike the crystal after reaching such a distance so our assumption is sensible). We repeat this process for a large number of random walkers and calculate f as:

$$f = \frac{\text{number of walkers which hit the crystal}}{\text{total number of walkers}} \quad (20)$$

and use 19 to calculate C.

In practice however, this approach is extremely inefficient and thus inordinately slow on all but the fastest computers. The volume in which the walker moves is gargantuan in comparision to the tiny step sizes it takes and so a very large number of steps must be taken before the walker either is absorbed by the crystal or escapes past R_∞. A number of modifications may be made to reduce the computational load of the algorithm [2].

The first of these is that R may be made small enough to only just enclose the whole crystal (it is vital that it does fully enclose the crystal however!). We may do this because of the isotropic nature of

random walks. If we consider R_0 to be the previous, larger value of R, and R_1 to be the newer, smaller value, the distribution of walkers passing through the sphere R_1 *for the first time* is isotropic as there are no sources of sinks of walkers (or water vapour) between R_0 and R_1. Thus we may start the walkers off from a sphere with any smaller radius $R = R_1$ as long as it fully encloses the crystal. The *total* distribution of walkers (or water vapour) is not isotropic as the rates at which walkers diffuse out past R_1 again depend on the geometry of the crystal and how it absorbs walkers but this is irrelevant to the walkers on their first pass through R_1 [2].

Another improvement may be made to the algorithm by varying the size of the steps taken by the walkers. As long as the walker does not meet a source or sink during a step the step size taken by the walker is actually irrelevant as any large step may be made up from a series of shorter jumps taken over a long period of time. In the absence of sources and sinks using a larger step size does not bias the statistics in any way. As there are no sources or sinks of vapour between R and R_∞ we may use a step size of $|r - R|$ where r is the current distance of the walker from the centre of the spheres. Likewise, when the walker is inside R the shortest distance to a source or sink is the distance from the walker to the surface of the crystal so this distance may be used as our step size [2].

The last modification adds another complication however, if we take the step size to be the distance to the surface of the crystal it is almost impossible for the walker to ever actually reach the crystal as this would require the walker to move in the precise direction which corresponds to the shortest distance to the crystal. The probability of choosing this precise direction (by choosing the correct θ and ϕ in polar co-ordinates from the choice of real numbers available) is zero (neglecting the fact that computers have limited precision and thus we are not quite choosing from the real numbers). To resolve this we judge the walker to have been absorbed by the crystal when it is less than a very small distance ϵ away [2]. The use of the absorbing layer adds an error to the value of the capacitance and thus ϵ must be minimised in order to minimised the error induced. If ϵ is chosen to be $< 0.1\%$ of the smallest dimension of the crystal the values found for capacitance are accurate to within the sampling accuracy for $N \lesssim 1,000,000$. This error may be removed using analytically using Green's function first-passage algorithms [9] but that is beyond the scope of this investigation.

D. Sampling error in the Monte-Carlo algorithm

There is a small amount of uncertainty due to the sampling error in the Monte-Carlo method. To calculate f, the fraction of random walkers which hit the crystal, we count the number n of walkers which hit the crystal and divide it by N the total number of random walkers. Whether a walker hits the crystal or not is described by a Poisson process with a probability p of success on each trial. For large N the fraction $f = n/N$ of walkers which hit the crystal approaches p. The uncertainty on n, σ_N is given by $\sqrt{N}$, thus the uncertainty on f, σ_f is:

$$\sigma_f = \frac{1}{\sqrt{N}}. \tag{21}$$

By equation 19, σ_C, the uncertainty of the capacitance obtained by this method is given by:

$$\sigma_C = \frac{R}{\sqrt{N}}. \tag{22}$$

This uncertainty may be made arbitrarily small by increasing N. As the available time and computing power were limited we generally used $N = 100,000$ which gives, for a typical value of $R \sim 5$, gives $\sigma_C \sim 0.015$.

IV. Gilbert-Johnson-Keerthi (GJK) algorithm

The above discussion of the capacitance calculation algorithm glosses over a couple of vital points. Namely, how do we check if the walker has been absorbed by the crystal, and how is the distance from the walker to the surface of the crystal calculated? The answer to both of these question lie in an algorithm widely used in robotics and computer graphics known as the Gilbert-Johnson-Keerthi algorithm, so-named after its discoverers [10] [11].This algorithm can check whether two convex polyhedra are intersecting and calculate the distance between them if they are not (we consider a polyhedron A convex when a line segment drawn between any two points in A never leaves A at any point, a convex polyhedron may be specified by its vertices alone). There is no restriction on using a point as one of the polyhedra and this is what

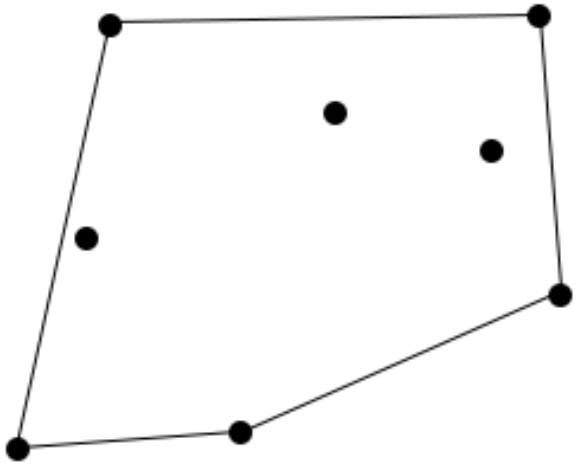

Figure 3. The area bounded by the lines is the convex hull of the points marked as dots.

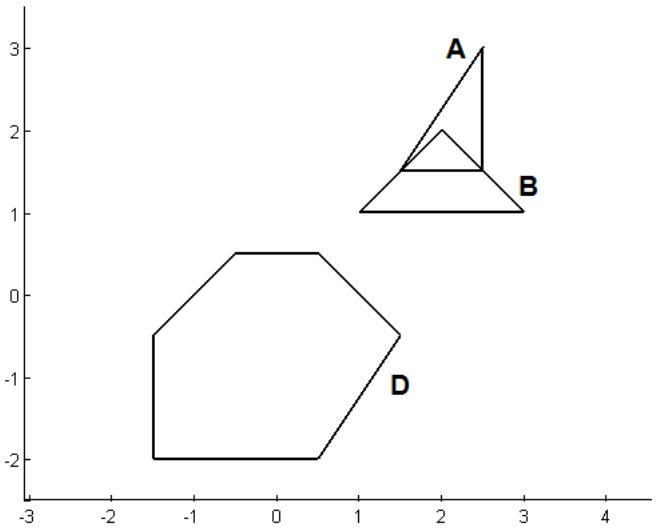

Figure 4. Polygon D is the Minkowski difference of A and B. As A and B intersect D contains the origin

we do when we calculate the distance between a walker and the crystal. The algorithm only works for convex polyhedra however it may be made to work for any shape, even a concave one, if that shape may be composed into a series of convex polyhedra and the algorithm may be applied to each of them in turn.

We will first give an overview of the mathematical concepts behind the GJK algorithm and then will discuss the details of the implementation.

A. Important concepts behind the GJK algorithm

The most important concepts behind the GJK algorithm as as follows:

1) Simplex: In the process of calculating the distance between two objects the GJK algorithm iteratively creates a set of points known as a simplex. The number of points in an n-dimensional simplex is equal to n+1. Thus in 2 dimensions the simplex is a triangle and in 3 dimensions the simplex is a tetrahedron.

2) Convex hull: For a finite set of points A, the convex hull of A is the smallest convex set of points that fully encloses A. In other words the convex hull is a outline of A. This is illustrated in fig. 3.

3) Support function: The support function is an important subroutine in the GJK algorithm. This takes a direction (specified by a vector $\boldsymbol{v}$) and a convex polyhedron (specified by a set of points A) and returns the point in the polyhedron that is "furthest" in the direction $\boldsymbol{v}$. This is achieved by returning the point in A which has the maximum dot product with $\boldsymbol{v}$. The support function is used to add points to the simplex.

4) Minkowski difference: The Minkowski Sum C of two convex polyhedra (specified by sets of points A and B) is defined as the union of the points formed by adding every point in A to every point in B. Mathematically:

$$C = A + B = \{x + y : x \in A, y \in B\}. \qquad (23)$$

The GJK algorithm uses a related concept called the Minkowski difference. This is simply the set of all points in one shape subtracted from all points in the other. For the sets A and B above, the Minkowski difference, D, of A and B is:

$$D = A - B = \{x - y : x \in A, y \in B\}. \qquad (24)$$

If two polyhedra intersect then their Minkowski difference will contain the origin and the distance between the two polyhedra will be the distance from their Minkowski difference to the origin [10]. See figure 4 for an example.

B. Overview of the algorithm procedure

The GJK algorithm works by iteratively constructing simplices from the vertices of the Minkowski difference of A and B that lie closer and closer to the origin until it has found the closest simplex to the origin. The distance from this closest

simplex to the origin is the distance between the two polyhedra, altenatively if the simplex contains the origin then the two polyhedra intersect.

Let A and B be sets of vertices that specify two polyhedra and let C be the Minkowski difference of A and B. On the k-th iteration of the algorithm let W_k be the set of vertices that specifies the simplex and let v_k be the point on the convex hull of the simplex which is closes to the origin. Let $s_C(\boldsymbol{v_k})$ denote the vertex of C returned by the support function using the search direction $\boldsymbol{v_k}$. The GJK algorithm works as follows:

1) : We start with an empty simplex $W_0 = \emptyset$ and add the point v_0 returned by the support function using a random search direction. This is our first simplex W_1.

The following steps are looped through repeatedly until we find the solution.

2) : We use the support function to find the point $w_k = s_C(-\boldsymbol{v_k})$ of C which is furthest in the direction $-\boldsymbol{v_k}$ pointing from the simplex to the origin.

3) : If w_k is not further in the direction $-v_k$ than v_k: return $\|v_k\|$ as the distance between A and B.

4) : Add w_k to the current simplex W_k to form $W_k \cup \{w_k\}$. If $W_k \cup \{w_k\}$ contains the origin then then A and B intersect, terminate the routine and return a distance of zero and that they intersect.

5) : Find the point v_{k+1} on the convex hull of $W_k \cup \{w_k\}$ (how this is done has not been explained yet).

6) : Make the next simplex (W_{k+1}) the smallest subset of $W_k \cup \{w_k\}$ which still contains v_{k+1}. This will involve dropping the point furthest from the origin from W_k if W_k already contains 4 points.

7) : Return to step 2) and continue.

C. Voronoi regions and distance calculation

One thing that was not made clear in the above overview of the algorithm is how v_k, the point on the convex hull of the simplex W_k closest to the origin, and thus the distance between A and B, is found. This involves finding out which Voronoi region of the convex hull the origin, O, is in. We say a point is in the Voronoi region of an element of the convex hull when the closest point on the hull is part of that element of the hull. An element of the convex hull may be a vertex, a line segment (edge) or a face. What this means and how it is determined should become clearer after a few illustrative examples.

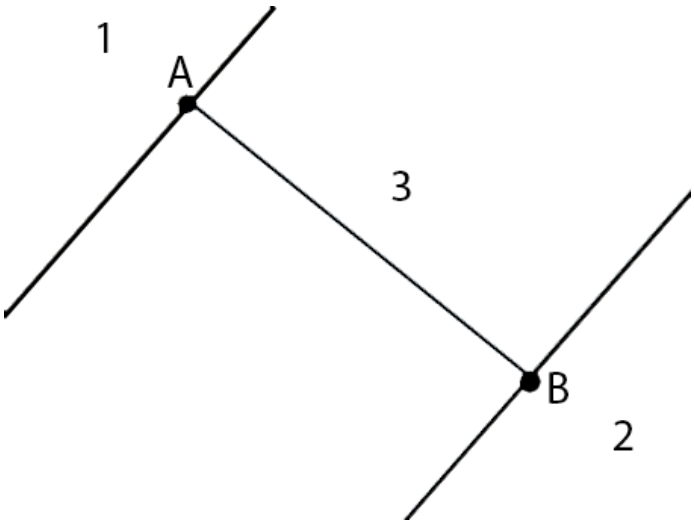

Figure 5. Diagram showing the three regions the origin may be in relative to a 1-D simplex.

1) 1-D simplex: First we'll deal with the case of a 1-dimensional simplex, i.e. a line. There are three possible regions in which O may be relative to the line as shown in figure 5. We take the dot products $\underline{AB}.\underline{AO}$ and $\underline{BA}.\underline{BO}$ where $\underline{AB}$ is the line segment from A to B et cetera. There are 3 cases:

- $\underline{AB} \cdot \underline{AO} < 0$ and $\underline{BA} \cdot \underline{BO} > 0$: the origin is in region 1.
- $\underline{AB} \cdot \underline{AO} > 0$ and $\underline{BA} \cdot \underline{BO} < 0$: the origin is in region 2.
- $\underline{AB} \cdot \underline{AO} > 0$ and $\underline{BA} \cdot \underline{BO} > 0$: the origin is in region 3.

2) 2-D simplex: A 2-dimensional simplex is a triangle. Given a triangular simplex ABC, O can either be closest to one of the vertices (regions V_A, V_B and V_C), closest to one of the edges (regions E_{AB}, E_{AC} and E_{BC} or inside the triangle (region E). These cases should be dealt with in turn:

a. Vertices: To check whether O is nearest to a particular edge we need to take the dot products of the two line segments running to the vertex with the vector from the vertex to the origin. To illustrate if O is closest to the vertex B (shown in figure 6) then:

$$\underline{AB} \cdot \underline{BO} > 0 \quad \& \quad \underline{CB} \cdot \underline{BO} > 0. \tag{25}$$

b. Edges: To check if P is nearest to an edge we first need to define vectors perpendicular to the edge pointing out, these are:

$$V_{AB} = (\underline{AC} \times \underline{AB}) \times \underline{AB} \tag{26}$$

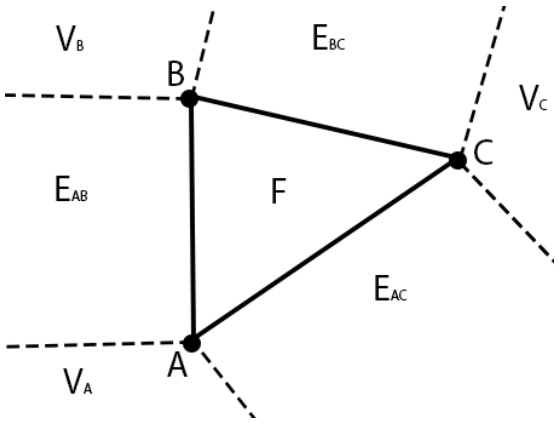

Figure 6. Diagram showing the the seven Voronoi regions the origin O may be in relative to a 2-D simplex

$$V_{AC} = (\underline{AB} \times \underline{AC}) \times \underline{AC} \tag{27}$$

$$V_{BC} = (\underline{BA} \times \underline{BC}) \times \underline{BC} \tag{28}$$

The conditions required for O to be in the region E_{AB} are:

$$\underline{AO} \cdot \underline{V_{AB}} > 0 \ \& \ \underline{AO} \cdot \underline{AB} > 0 \ \& \ .\underline{BO} \cdot \underline{BA} > 0. \tag{29}$$

The conditions for E_{BC} and E_{AC} are analogous.

c. Inside triangle: If O is inside the triangle, i.e. in F then:

$$\underline{AO} \cdot \underline{V_{AB}} > 0 \ \& \ \underline{AO} \cdot \underline{V_{AC}} > 0 \ \& \ .\underline{BO} \cdot \underline{V_{BC}} > 0. \tag{30}$$

3) 3-D simplex: A 3-D simplex, $ABCD$, is a tetrahedron. The process of figuring out which Voronoi region the origin is in is similiar to that for a 2-D simplex but with a few extra complications.

To check if O is nearest to a face ABC we must have:

$$(\underline{AB} \times \underline{AC}) \cdot AO > 0. \tag{31}$$

The points must be labelled so that $\underline{AB} \times \underline{AC}$ points out of the tetrahedron. We perform similar tests for the faces ABD, ACD and BCD. If just one test returns positive then O is closest to the face for which a positive result was obtained, if two tests return positive then O is closest to the edge separating them. If all tests return negative then O is inside the tetrahedron.

4) Distance calculation: The closest point on the simplex to the origin O is found using the above method. How the distance from the simplex to the point is calculated depends which part of the simplex is closest to the point. There are several possible cases:

a) The nearest point on the simplex is a vertex of the simplex.

This is the simplest case. If a vertex A is the closest point on the convex hull of the simplex to O then the distance d between A and O is simply:

$$d = |\underline{AO}|. \tag{32}$$

b) The nearest point on the simplex is part of an edge of the simplex.

If the nearest point on the simplex is part of an edge $\underline{AB}$ then:

$$d = \frac{\underline{AO} \cdot \left[(\underline{AB} \times \underline{AO}) \times \underline{AB}\right]}{|(\underline{AB} \times \underline{AO}) \times \underline{AB}|}. \tag{33}$$

c) The nearest point on the simplex is part of a face.

If the nearest point on the simplex is part of a face defined by the points A, B and C then:

$$d = \left| \frac{\underline{AO} \cdot (\underline{AB} \times \underline{AC})}{\underline{AB} \times \underline{AC}} \right|. \tag{34}$$

V. Simulation Procedure

All simulation code was written in MATLAB. In order to avoid having to rewrite the code that calculates capacitance every time we want to study a new crystal geometry, we decided to break the code up into different modules with different functions. These include programs which generate the co-ordinates of the vertices of the crystal being studied and store them in the MATLAB workspace, programs which calculate the capacitance of the shape stored in the workspace and programs which fulfil other functions like plotting crystals. To speed up the calculation the multi-threading features of MATLAB were used. 12 cores of the processor were used with each running a thread simulating the progress of a different random walker simultaneously.

A. Unit cube

To verify that our code works the first shape for which we calculated capacitance was a unit cube centred on the origin. Though there is no analytic solution for the capacitance of a cube, its value has been calculated numberically many times and is well known so it provides a benchmark to test the accuracy of our simulation against.

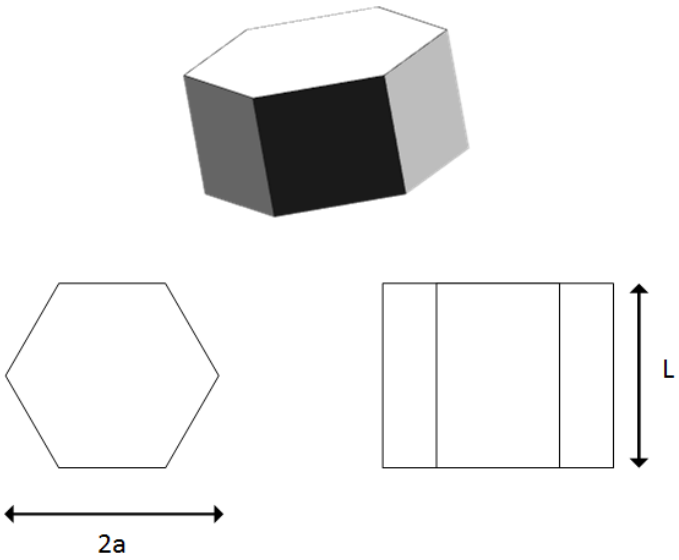

Figure 7. Illustration of the important dimensions of a hexagonal plate/column.

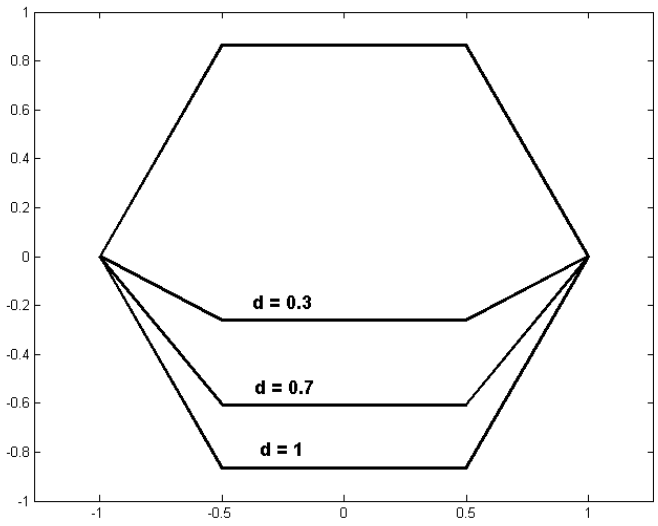

Figure 8. Diagram showing the effect of the "scalene-ness" parameter d on the shape of the crystal.

B. Hexagonal plates with varying aspect ratios

Almost all natural ice found on earth is in the I_h phase which has a hexagonal crystal structure [12]. Thus ice crystals formed in the atmosphere tend to form in shapes with a hexagonal symmetry and hexagonal prisms are one of the most common type of ice crystal. These can range from thin plates to more extended columns. The important parameters that describe the shape of a hexagonal prism are the lengths L and a illustrated in figure 7. We can define an aspect ratio, A, as:

$$A = \frac{L}{2a}. \tag{35}$$

Westbrook et al. (2008) [2] calculated the capacitance of hexagonal plates and columns using the same Monte-Carlo method. Therefore we have another benchmark to check the accuracy of our method against. To store the co-ordinates of a single hexagonal plate a matrix of size 12x3 is needed as there are 12 vertices in the prism and 3 spatial dimensions. We then used the third dimension of the array to store plates with different aspect ratios in the same array. We generated 50 hexagonal plates and columns with aspect ratios ranging from 0.05 to 65 and calculated their capacitance.

C. Scalene hexagonal plates

It is commonly obseved that ice crystals found in clouds have a distorted or scalene hexagonal shape [13]. A parameter d may be defined to control the "scalene-ness" of the crystal, we multiply the y-coordinates of two of the points of the base of the hexagon by a factor d which takes values between 0 and 1. The effect of d on the shape of the crystal is shown in figure 8. We calculated the capacitance of 20 scalene crystals all with $a = 1$ and an aspect ratio of 0.5 with d between 0 and 1. 10,000 random walkers were used for each crystal.

D. Aggregates of hexagonal plates

Complex aggregate crystals formed of multiple hexagonal plates or columns fused together are commonly observed in clouds, and are particularly common in cirrus clouds [14][1]. We investigated how the capacitance of aggregates depends on the number of plates in a chain and the degree of aggregation described by a parameter called the aggregation index. The aggregation index, AI, is introduced in Um and McFarquhar (2009) [14] and is defined as:

$$AI = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}}{Max\left(\sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}\right)} \tag{36}$$

where D_{ij} is the distance between plates i and j in the aggregate. The denominator is the maximum physically possible value of the sum, i.e. when all

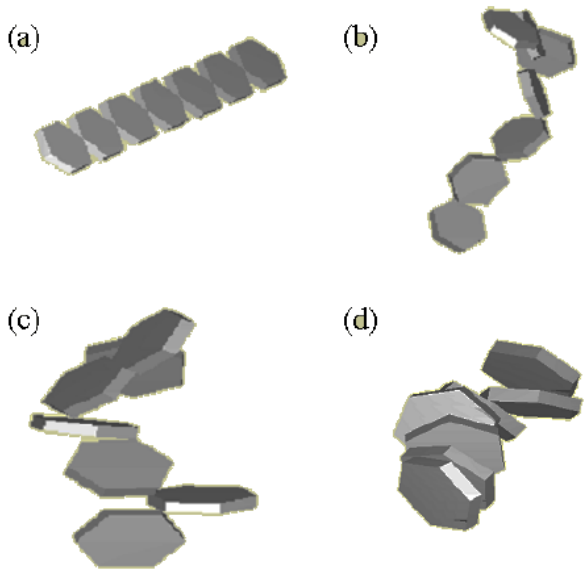

Figure 9. Illustration of idealized aggregates of plates with varying aggregation indices: (a) AI = 1.0, (b) AI = 0.81, (c) AI = 0.61, (d) AI = 0.40. This image is courtesy of Um and McFarquhar (2009) [14]

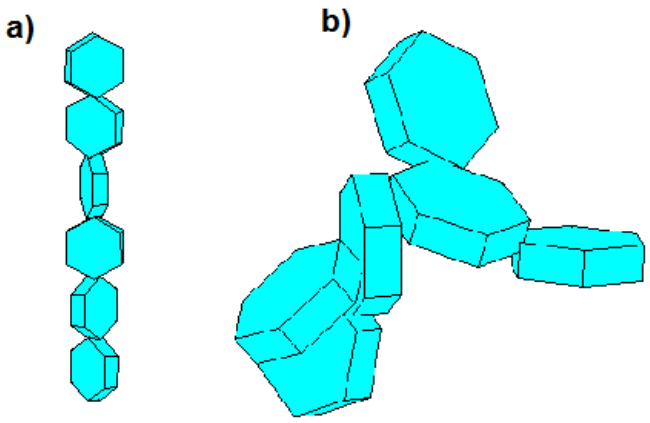

Figure 10. Examples of two chains generated by our program. The more extended chain, a), has all β values set close to 1 and little randomness. The more densely aggregated chain, b), has all β values close to one and the distribution of plates is almost completely random.

plates are arranged in a straight line. Examples of aggregates of plates with varying aggregation indices are shown in figure 9.

Each aggregate is a chain made up of n hexagonal plates. We investigated the properties of aggregates of plates where each single plate has an aspect ratio of 0.25. The co-ordinates of the vertices were stored in a multi-dimensional array. The first two dimensions had size 12 and 3 and were used to store the x, y and z co-ordinates of the 12 vertices of a single plate, the third dimension had size n and was used to store the co-ordinates of the n plates in an aggregate simultaneously. The fourth dimension was used to store a large number of aggregates of plates simultaneously.

The process by which the chains were generated is as follows. The first plate in the chain is placed centred on the origin. The spatial orientation of the next plate in the chain is chosen by rotating it about the x, y and z axes by angles between 0 and 2π randomly chosen from the output of a beta distribution multiplied by 2π.. A random azimuthal angle, ϕ, between 0 and 2π, is chosen from a uniform distribution and a random polar angle, θ between 0 and π is chosen from a beta distribution. A beta distribution is used because it allows us control over how densely bunched the aggregates are. A beta distribution is a probability distribution with the form:

$$P(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B} \tag{37}$$

where $P(x)$ is the probability of obtaining a value x (in the range 0 to 1), α and β are parameters of the distribution (in range 0 to 1 for our purposes) and B is a normalisation constant (dependent on the choice of α and β. If we set $\alpha = \beta$ we obtain a symmetrical distribution whose properties we can alter by altering β. Examples of the distribution for different values of β are shown in figure **??**. The parameters β_x, β_y, β_z and β_θ control the distributions for x, y, z and θ respectively. When these quantities are set to values near 0 the distributions are highly peaked at each end and the resulting chains are very extended with aggregation indices close to 1, when the β values are set close to 1 the distributions become more uniform and the resulting aggregates are more "bunched up" and have lower aggregation indices. Figure 10 shows examples of aggregates generated by this method.

Aggregates were generated and their capacitance calculated for values of n between 2 and 10. 130 aggregates with a wide variety of aggregation indices (controlled by varying the β parameters) were generated and 10,000 random walkers were used to calculate the capacitance of each. As the GJK algorithm can only operate on convex polyhedra and the overall aggregate is not convex, when finding the

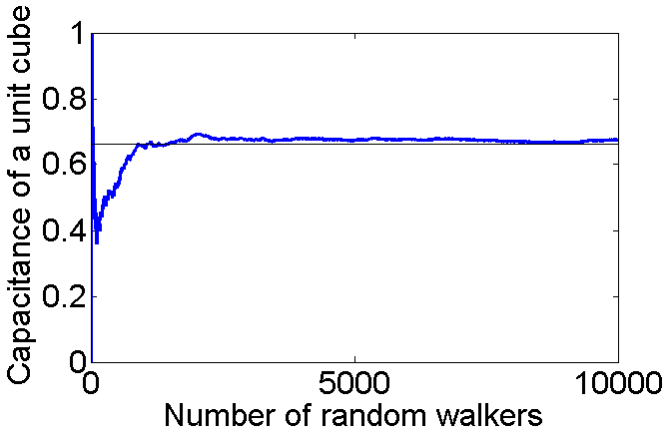

Figure 11. The capacitance of a cube with unit side length converges on a value of 0.6612 as the number of random walkers used increases.

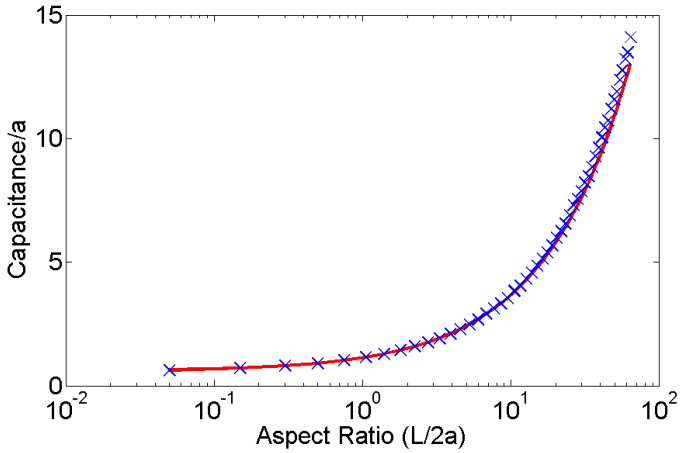

Figure 12. Plot of capacitance (C/a) versus aspect ratio for hexagonal plates and columns. The line shows the fit $C = 0.58(1+0.937A^{0.76})$. Error bars have been left off the figure as they are too small to be shown cleanly. The fitted curve passes within one standard deviation of every data point.

distance from the random walker to the aggregate, the distance to each of the plates was found (using the GJK on algorithm on each of them in turn) and the smallest one was selected as the distance to the aggregate. The normalised capcitance, i.e capacitance divided by the maximum dimension in the crystal D_{max} was calculated for every aggregate as Westbrook et al. [2] have shown that C/D_{max} is a function only of the shape of an ice crystal and not its size. Thus C/D_{max} is the important quantity when discussing the capacitance of aggregates. 10,000 random walkers were used for each aggregate.

VI. Results

A. Unit cube

After 100,000 random walkers we obtained a value of 0.6612 ± 0.0055 for the capacitance of the unit cube. The convergence of the capacitance on this value over the first 10,000 walkers is shown in figure 11.

B. Hexagonal plates of varying aspect ratios

Figure 12 shows our results for capacitance as a function of aspect ratio $A = L/2a$ for hexagonal prisms with $a = 1$. Sampling uncertainties on each value were calculated but are too small to comfortably show on the plot. The fit $C = 0.58(1+0.936A^{0.76})$ is shown as well. Westrook et al. (2008) [2] gives a fit of $C = 0.58(1+0.95A^{0.75})$, as our data shows a trend similiar to Westbrook's we decided a fit with of the same form, i.e. $C = X + YA^Z$ was appropriate. At lower values of A our data almost precisely matches Westbrook's so thus X was set as 0.58. Moving X to the other side of the equation and taking logs of both sides we obtain:

$$ln(C - X) = lnY + ZlnA. \qquad (38)$$

Y and Z were then found by performing a linear fit of $ln(C - 0.58)$ against lnA giving the values $Y = 0.543$ and $Z = 0.76$, leading to the overall fit $C = 0.58(1 + 0.936A^{0.76})$ which is shown in figure 12. The reduced χ^2 value for this fit is 1.153.

C. Scalene hexagonal plates

Figure 13 shows our results for the capacitance of scalene ice crystals with $a = 1$ and $L = 1$ as a function of the parameter d. It has been suggested that the capacitance of an ice crystal depends mainly on the the area covered by the plate and not on its fine structure [4], thus as the area depends linearly on d a straight line has been fitted to this data using a least-squares fitting routine. The fitted line has equation $C = 0.74 + 0.16d$. The reduced χ^2 value for this fit is 0.27.

D. Aggregates of hexagonal plates

The normalised capacitance, i.e. capacitance divided by D_{max}, the largest dimension in the crystal, was calculated for aggregates of hexagonal plates with 2 to 10 plates in the aggregate. For all lengths of chain, a general downward trend in normalised capacitance was observed as the aggregation index

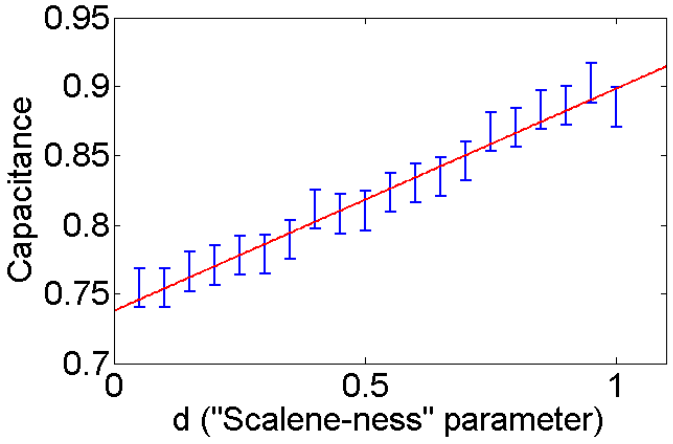

Figure 13. Results of the simulation of scalene hexagonal plates with 10,000 random walkers. A straight line with equation $C = 0.74 + 0.16d$ has been plotted through the data.

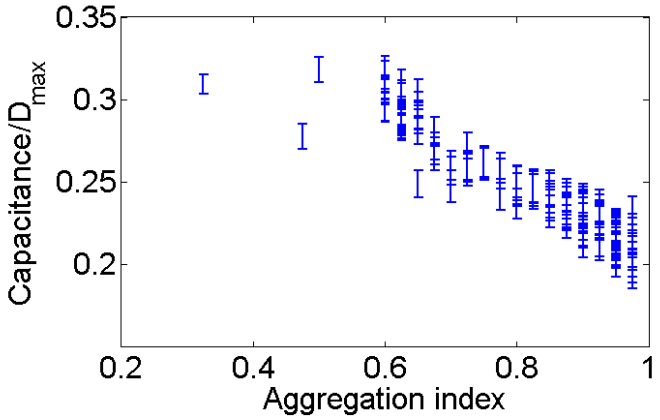

Figure 14. Plot of normalised capacitance against aggregation index for chains consisting of 2 hexagonal plates of aspect ratio 0.25.

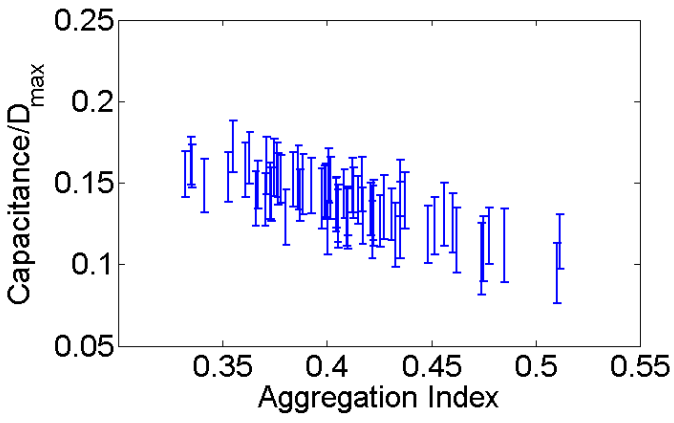

Figure 15. Plot of normalised capacitance against aggregation index for chains consisting of 9 hexagonal plates of aspect ratio 0.25.

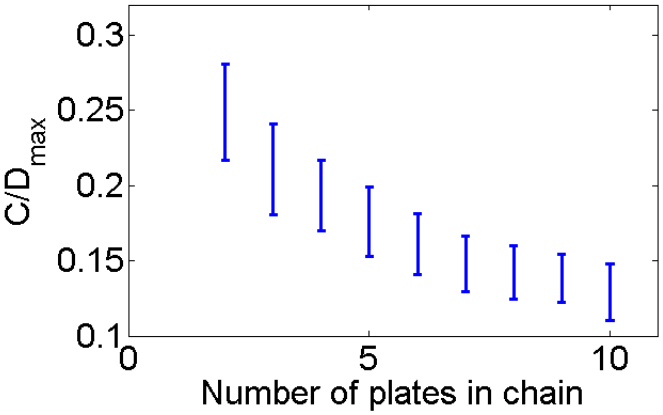

Figure 16. Normalised capacitance of aggregate versus number of plates in the aggregate. Each plate had an aspect ratio of 0.25.

increases. As it would be unnecessary to include 9 graphs (one for each length of chain) showing the same trend only 2 are included. Figure 14 shows normalised capacitance against aggregation index for aggregates consisting of 2 plates and figure 15 shows normalised capacitance against aggregation index for aggregates consisting of 9 plates.

We next investigated how the normalised capacitance depends on the number of plates in the aggregate, n. For each value of n the normalised capacitance was averaged across all aggregates and the standard deviation of these values was used as the uncertainty. The standard deviation of the capacitances was used as the uncertainty and not the weighted average of the uncertainties on each aggregate as the variation in capacitance from aggregate to aggregate is greater than the uncertainty due to sampling error on each data point. Also we are not comparing like with like when we average across the aggregates so it is nonsensical to say that the uncertainty should be the weighted average of the induvidual uncertainties.

A plot of normalised capacitance against number of plates is shown in figure 16. A general downward trend in C/D_{max} is observed as n increases.

VII. Discussion of Results

A. Unit cube

Hwang and Mascagni (2004) [8] have performed a detailed study of the capacitance of the unit cube which involved a review of the many methods by which capacitance has previously been calculated and their own calculations involving a walk-on-planes Monte-Carlo simulation and found a value of 0.6606782 ± 0.0000001. Given et al. (1997) [15] find a value of 0.660675 ± 0.00001 using first-passage Monte-Carlo methods. Our result of 0.6612 ± 0.0055

is consistent with these values and many others that have been calculated, indicating that our method of calculating the capacitance is accurate. We expect that our value would continue to approach these values as the number of random walkers used is increased.

B. Hexagonal plates with varying aspect ratios

Our results are very similiar to those found in Westbrook et al. (2008) [2]. We have more data points for larger values of the aspect ratio than Westbrook et al. so this would indicate that perhaps our fit of $C = 0.58(1+0.936A^{0.76})$ is more accurate, however the uncertainties on the capacitance are larger for the crystals with greater aspect ratios so both our fit and the fit found in Westbrook et al. are compatible with our results. The fact that the reduced χ^2 value for our fit is close to 1 also indicates that this is a good fit for the data. We cannot rule out that perhaps another type of fit would be more appropriate, however the fact that this simple 3-parameter fit matches the data closely over three orders of magnitude in the aspect ratio indicates that it is likely to be the correct form.

C. Scalene hexagonal plates

Due to time limitations only 10,000 random walkers were used, this led to a relatively large uncertainty on each data point. This makes it difficult to draw conclusions as to the proper form of the fit which should be performed on this data. The linear fit has some tenuous support from theory and fits fairly well to the data however several other types of fit could be performed on data with such large uncertainties. The small value of the reduced χ^2 (0.27) reinforces this point. The correct form of the data would become clear if a greater number of random walkers were used.

D. Aggregates of hexagonal plates

The observation that normalised capacitance decreases as the aggregation index increases and the chains become more extended in space may initially seem surprising. But as the capacitance is proportional to the fraction of random walkers which hit the crystal which is related to the volume of space filled by the crystal. As the aggregation index of a chain increases the volume of space filled relative to the maximum dimension of the aggregate decreases and thus the normalised capacitance drops.

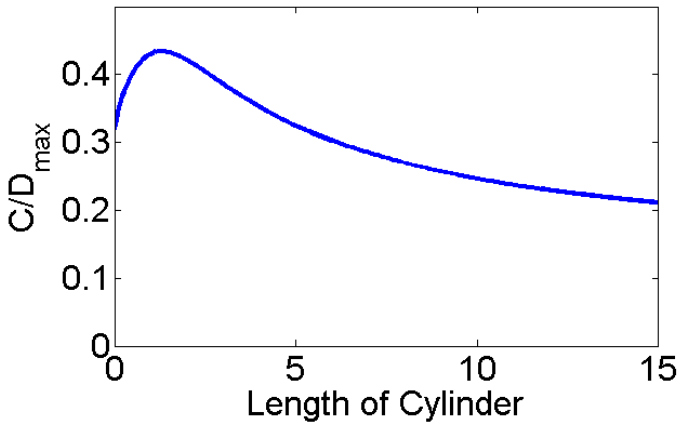

Figure 17. C/D_{max} for a cylinder of radius 1.

Our results show C/D_{max} decreasing as the number of plates in the aggregate, n, increases. This is in contradiction of the theoretical findings by Westbrook et al. [2] and experimental observations by Field and Heymsfield [16] that the capacitance of aggregates of chains approaches a constant value as n increases. The paper of Westbrook et al. does not provide information on how densely aggregated were the chains that were investigated, the chains investigated in this investigationincluded many that were quite extended and roughly cylindrically shaped in overall outline with a radius of about 1 as $a = 1$ for our induvidual crystals.. Figure 17 shows the normalised capacitance of a cylinder with radius 1, this is similiar in shape to the values we found for the normalised capacitance of aggregates. As our aggregates fill less space than a cylinder of the same length and diameter the normalised capacitance will be lower but will follow the same trend. We only calculated capacitance for aggregates with up to 10 plates per chain but we expect the normalised capacitance to asymptotically reach a value in the range of 0.10 - 0.13 as the number of plates increases judging from the trend in our data and comparing it to the results for a cylinder. That his is lower than the value of 0.25 - 0.28 found by Westbrook may be attributal to differences in the form of the aggregates investigated.

VIII. Conclusion

We have produced a piece of software which is capable of capable of calculating the capacitance of

any ice crystal geometry provided it is a convex polyhedron or may be decomposed into a series of convex polyhedra, which all polyhedra may be. Values of capacitance have been obtained for a cube, hexagonal plates and columns with various aspect ratios, scalene hexagonal plates and aggregates of hexagonal plates of varying lengths. The results obtained are similiar to those previously obtained by Westbrook et al. [2]. It would now be a simple matter to extend the investigation further and calculate the capacitances of many different crystal geometries and habits such as bullet-rosettes and the classic dendritic snowflake shape, however we have been unable to do this due to time limitations. Our main achievement has not been in the calculation of results but in the production of the code for carrying out the simulation. Though the outline of the Monte-Carlo algorithm for calculating capacitance had already been given in several papers, details of its implemetation including how the distance from the random-walker to the crystal is calculated and how the aggregates are generated have not been given. The solution of these problems, especially the implementation of the GJK algorithm (which had not been mentioned anywhere in the literature on random-walker methods and may be an innovation), constituted the main body of work. Due to the correspondences between the vapour diffusion growth of ice crystals, electrostatics, diffusion-influenced reaction theory [7] and Stokes flow in hydrodynamics [5], the code may also be used to calculate the electrostatic capacitance, reaction rate constants and hydrodynamic friction of arbitrary polyhedra once the results are multiplied by the appropriate numerical constant.

IX. Division of labour

I mainly worked on writing the pieces of code which run the random-walk simulations and the pieces that generate the crystal plates and chains of plates. My lab partner Zach Waller mainly worked on writing the code to implement the GJK algorithm.

References

[1] P. J. Connolly, C. P. R. Saunders, M. W. Gallagher, K. N. Bower, M. J. Flynn, T. W. Choularton, J. Whiteway, and R. P. Lawson, "Aircraft observations of the influence of electric fields on the aggregation of ice crystals," *Quarterly Journal of the Royal Meteorological Society*, vol. 131, no. 608, pp. 1695–1712, 2005.

[2] C. D. Westbrook, R. J. Hogan, and A. J. Illingworth, "The Capacitance of Pristine Ice Crystals and Aggregate Snowflakes," *Journal of Atmospheric Sciences*, vol. 65, p. 206, 2008.

[3] C. Emersic, "Light scattering by ice crystals - figure 2. url: http://www.cas.manchester.ac.uk/resactivities/cloudphysics/topics/lightscatter

[4] J. E. McDonald, "Use of the electrostatic analogy in studies of ice crystal growth," *Zeitschrift Angewandte Mathematik und Physik*, vol. 14, pp. 610–620, Sep. 1963.

[5] H.-X. Zhou, A. Szabo, J. F. Douglas, and J. B. Hubbard, "A brownian dynamics algorithm for calculating the hydrodynamic friction and the electrostatic capacitance of an arbitrarily shaped object," *The Journal of Chemical Physics*, vol. 100, no. 5, 1994.

[6] J. D. Jackson, *Classical Electrodynamics*. New York, NY, USA: John Wiley & Sons, 1962.

[7] S. H. Northrup, S. A. Allison, and J. A. McCammon, "Brownian dynamics simulation of diffusion-influenced bimolecular reactions," *The Journal of Chemical Physics*, vol. 80, no. 4, 1984.

[8] C.-O. Hwang and M. Mascagni, "Electrical capacitance of the unit cube," *Journal of Applied Physics*, vol. 95, no. 7, pp. 3798–3802, Apr 2004.

[9] M. Mascagni and C.-O. Hwang, "epsilon-shell error analysis for "walk on spheres" algorithms." *Mathematics and Computers in Simulation*, vol. 63, no. 2, pp. 93–104, 2003.

[10] E. Gilbert, D. Johnson, and S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE Journal of Robotics and Automation*, vol. 4, no. 2, pp. 193–203, Apr 1988.

[11] S. Cameron, "A comparison of two fast algorithms for computing the distance between convex polyhedra," *Robotics and Automation, IEEE Transactions on*, vol. 13, no. 6, pp. 915–920, Dec 1997.

[12] P. Hobbs, *Ice Physics*. Clarendon Press, 1974.

[13] M. Bailey and J. Hallett, "Growth rates and habits of ice crystals between -20°c and -70°c," *Journal of the Atmospheric Sciences*, vol. 61, no. 5, pp. 514–544, 2004.

[14] J. Um and G. M. McFarquhar, "Single-scattering properties of aggregates of plates," *Quarterly Journal of the Royal Meteorological Society*, vol. 135, no. 639, pp. 291–304, 2009. [Online]. Available: http://dx.doi.org/10.1002/qj.378

[15] J. A. Given, J. B. Hubbard, and J. F. Douglas, "A first-passage algorithm for the hydrodynamic friction and diffusion-limited reaction rate of macromolecules," *The Journal of Chemical Physics*, vol. 106, no. 9, 1997.

[16] P. R. Field and A. J. Heymsfield, "Aggregation and scaling of ice crystal size distributions," *Journal of the Atmospheric Sciences*, vol. 60, no. 3, p. 544.560, 2003.