

**Electronic Supplementary Information** for "Complex three-dimensional conformal surfaces formed by atomic layer deposition: Computation and experimental verification", *Journal of Materials Chemistry*

Andrew Brzezinski, Ying-Chieh Chen, Pierre Wiltzius, and Paul V. Braun  
University of Illinois at Urbana-Champaign, USA  
Email: pbraun@illinois.edu

### **The Resolution Parameter, Parallelepiped Voxels, and Error Bounds**

In order to discuss the scaling and error bounds of the algorithm it is useful to define  $N$  as a measure of the resolution of the computational grid. Also,  $N$  is the length, in voxels, of one side of the computational domain so the discussion below assumes the domain is  $N * N * N = N^3$  voxels in size. The arguments below will also apply to domains having different numbers of voxels on each side because these lengths can be set equal to some  $N$  scaled by a constant factor that is irrelevant to the asymptotic behavior of the scaling and error bounds.

The thickness of material deposited in a single iteration is limited by the Courant–Friedrichs–Lewy condition [1] to less than the distance across a single voxel. This ensures the proper flow of information between voxels during successive growth iterations so that the algorithm correctly identifies the nearest surface to each empty voxel. When the voxel shape is cubic the maximum thickness per iteration is a third the of the voxel diagonal. However, the algorithm is designed to work with voxels having an arbitrary parallelepiped shape as this facilitates the representation of a primitive unit cell of a Bravais lattice. In this case, it is necessary to limit the deposition per iteration to no more than a third of the minimum diagonal distance across the parallelepiped voxel.

Pinch-off may occur anywhere inside the voxel but a voxel's position is represented by its center. Here, the assumption is made that the error is due solely to the difference between the true pinch-off point and the center of the voxel. Thus, the maximum possible error is proportional to the longest diagonal of the voxel. Furthermore, the error bound is made tighter by choosing a voxel shape that minimizes the longest diagonal. Also, when the resolution  $N$  is increased, the length of each voxel becomes smaller and the error bound becomes tighter. It is easy to see that the relative error in the deposition thickness at pinch-off scales as  $N^{-1}$ . The relative volumetric error in the amount of deposited material would be the thickness error times the surface area divided by the volume, respectively  $1 * N^2 / N^3 = N^{-1}$ .

To estimate the size of the bounds on the volumetric error and compare it to the expected asymptotic behavior, we used the algorithm to calculate the filling fraction of infilled spheres on an fcc lattice. 1000 calculations were performed for each value of  $N$ , where  $N$  is either 64, 128, 256, or 512. The voxel at the origin of the sphere in the primitive fcc unit cell was uniformly spanned with 1000 points (a uniform grid, 10 points in each direction) which served as the exact position of the sphere center in each respective run of the algorithm. This has the effect of shifting the exact position of the true pinch-off point. Furthermore, the voxels that are initially set solid and empty (respectively inside or outside the sphere radius) may differ for each run. The filling fraction at pinch-off was calculated for each run of the algorithm and the

maximum and the minimum results were plotted (Fig. 1). This forms an empirical estimate of the error bounds and also demonstrating the apparent  $N^{-1}$  convergence to the value 0.9641, the analytical result for fcc spheres. Although larger deviations might be found at points in between the 1000 sampled origin-points, it is likely that the sampling is sufficiently dense to capture the majority of the error and its behavior.

### Calculating the Distance to a Surface

To calculate the distance from an empty voxel to a surface, the path is followed from the voxel, through the previously deposited voxels, to termination on an original surface voxel. This distance can be represented by an integer number of steps in the respective directions of the vectors which describe the parallelepiped voxel:

$$d = \lfloor \frac{d}{a} \rfloor a + n_a \lfloor \frac{d}{b} \rfloor b + n_b \lfloor \frac{d}{c} \rfloor c$$

As each of the six possible paths leading from an empty voxel are compared to find the shortest distance to a surface, it is possible that two different surface voxels are equidistant from the empty voxel. But the algorithm must choose one of these equidistant surfaces as being the nearest surface. In our implementation, the shortest equidistant path is that path that was probed first, while the order of probing (-a direction, +a, -b, +b, -c, and finally +c) remains constant. This can potentially lead to a slight anisotropy in the different directions of our computational domain. Furthermore, our implementation conveniently accumulates the distance vector,  $d$ , at each voxel of the traced out path to the surface. This can cause seemingly equidistant paths being unequal due to round off as the  $d$ -vector is accumulated. Thus, in the above sampling of the 1000 origin-points to estimate the error bounds, points that may have appeared degenerate due to the fcc symmetry are in fact distinct.

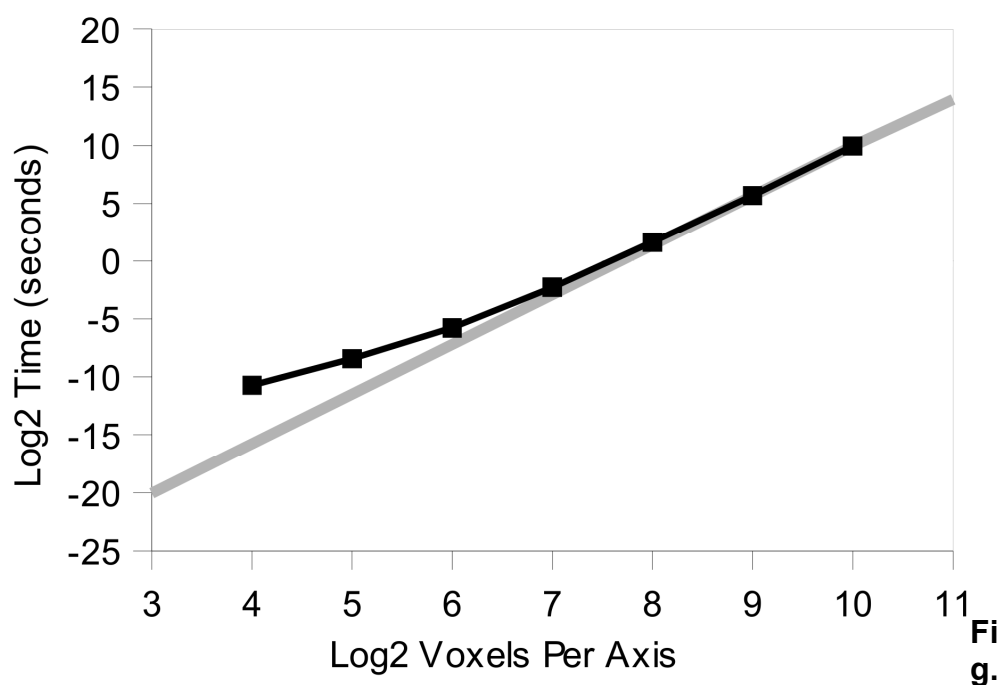
The origin of the larger error bounds for our improved  $N^4$  algorithm vs the straightforward  $N^5$  algorithm lies in the necessary choice between two ideally equidistant surfaces. Examples can be contrived where a certain choice prevents subsequently deposited voxels from finding the true nearest surface. However, the error cause by this occurrence is a fraction of a voxel length. Furthermore, as the deposition proceeds these errors tend to be reduced and even eliminated because the distance from these other empty voxels to alternate candidates for a nearest surface either become closer to the true nearest distance or they become the true nearest distance. The above empirical estimation of the error bounds suggests that the effect due to incorrectly identifying the true nearest surface is not significant.

### Scaling of the Computation Time

The straightforward brute-force approach to calculate each voxel's nearest surface requires calculating the euclidean distance between each empty voxel and each surface site. In a three-dimensional computational volume with  $N$  voxels per side,  $N^3$  voxels in total, the number of empty voxels is proportional to the total number of voxels,  $N^3$ , and the total number of surface voxels will be proportional to  $N^2$ . The complexity for testing  $N^3$  empty voxels against  $N^2$  surface voxels is  $N^3 * N^2 = N^5$ , which is the dominant contribution to the computation time.

In the presented algorithm the complexity is reduced to  $N^4$ . Each of the  $N^3$  empty voxels are tested against a constant number of paths to a potential nearest surface. These tests are only performed when the empty voxel in question is adjacent to the surface, which is only for a constant number of growth iterations. However, the path length will be proportional to  $N$ , giving  $N * N^3 = N^4$  computational cost. An alternate way to derive this result is that there are  $N^2$  empty voxels adjacent to the surface which must be tested at each growth iteration. The total number of growth iterations, like the path length, is proportional to the linear distance across the computational domain which is proportional to  $N$ , giving  $N^2 * N * N = N^4$  complexity.

The above analysis only accounted for the cost of finding the nearest surfaces. An additional cost is incurred by checking for a continuous path of empty voxels between the deposition source and the growth surface. The "flood-fill" type algorithm will visit each of the  $N^3$  empty voxels in every growth iteration and the total number of iterations is proportional to  $N$ , giving  $N^3 * N = N^4$  complexity. To further improve the computational cost would require improvements to both the nearest surface algorithm and continuous path algorithm which form the complete algorithm for tracking the evolving surfaces during conformal deposition.



**S1** The scaling of computation time versus the number of voxel per axis (black). The scaling approaches the expected  $N^4$  asymptotic behavior (grey), which is plotted with a slope of 4 and passes through the data point at 1024 voxels per side.

The  $N^4$  scaling of the improved algorithm is demonstrated in Fig. S1 where a primitive unit cell with the specified number of voxels per axis is used for the computation. At the last data point, with 1024 voxels per axis, the slope of computation time on the log-log plot is nearly 4, the

expected asymptotic value. In comparison, the straightforward  $N^5$  algorithm that we implemented required about 10 minutes at 128 voxels per axis and about 5 hours at 256 voxels per axis. This is consistent with  $N^5$  scaling and also requiring significantly more time compared to the few seconds used by the improved algorithm.

### Touching Spheres on an FCC Lattice

To find the analytical result for the filling fraction of conformal deposition, until pinch-off, on touching spheres arranged to form a fcc lattice, we divide the sum of the initial and grown material volume in one primitive unit cell by the total volume of that unit cell. Let  $R_1$  be the radius of the initial spheres, then it is trivial to derive the unit cell volume,  $V_{cell}$ , and the distance,  $R_2$ , from the sphere's center to the pinch-off point at the centroid of three mutually touching spheres (Fig. 1):

$$V_{cell} = 8R_1^3 / \sqrt{2}$$

$$R_2 = 2R_1 / \sqrt{3}$$

The interior of the fcc lattice of spheres becomes pinched-off after a total thickness of  $R_2 - R_1$  of material is deposited, as all paths to the surface become closed. The final volume of the initial and deposited material will be the volume of a sphere with radius  $R_2$  minus the overlapping volume of the neighboring spheres. It is straightforward to derive the volume of a spherical cap, the region bounded between a sphere of radius  $r$  and an intersecting plane, distance  $p$  from the sphere center.

$$V_{cap} = \frac{\pi}{3} [2r^3 - 3r^2 p + p^3]$$

As there are 12 nearest neighbors, 12 cap volumes must be subtracted, with  $r = R_2$  and  $p = R_1$ . The total filling fraction at pinch-off is

$$f_{pinch} = \frac{V_{sphere}(R_2) - 12 V_{cap}}{V_{cell}}$$

$$f_{pinch} = \frac{\pi}{\sqrt{2}} \left[ \frac{2}{6} \left( \frac{2}{\sqrt{3}} \right)^3 - 5 \frac{2}{\sqrt{3}} \right] \approx 0.964102939$$

### References

[1] Richard Courant, On the partial difference equations of mathematical physics, Courant Institute of Mathematical Sciences, New York University, (New York), 1956.