Geometric Advection Algorithm for Process Emulation

Xaver Klemenschits
Institute for Microelectronics
TU Wien
Vienna, Austria
Email: klemenschits@iue.tuwien.ac.at

Siegfried Selberherr
Institute for Microelectronics
TU Wien
Vienna, Austria
Email: selberherr@iue.tuwien.ac.at

Lado Filipovic
Institute for Microelectronics
TU Wien
Vienna, Austria
Email: filipovic@iue.tuwien.ac.at

***Abstract*—An algorithm is developed, which advects a material interface analytically, according to purely geometric considerations. This algorithm is implemented in ViennaLS, a sparse level set library and its applicability to common microelectronic fabrication processes is demonstrated. A pinch-off plasma CVD process is emulated using the presented algorithm. This algorithm is compared to common advection algorithms, showing a significant improvement in accuracy, with a performance penalty of a factor of about 2 when compared to simple advection schemes and a performance benefit of a factor of 6 when compared to more sophisticated schemes.**

I. Introduction

In topography simulations, implicit surface representations such as the level set (LS), offer significant advantages over explicit representations. These include robust merger and separation of surfaces, and simple extraction of geometrical properties like surface normals and curvature [1], which is especially important for the simulation of processes such as air-void creation with chemical vapour deposition (CVD). The complex merger of surfaces at the pinch-off point required for the accurate simulation of this process is critical. It is therefore important for the movement of surfaces to be as robust as possible, without losing information about the precise location of the surface. Implicit surface representations offer intuitive handling of such complex deformations and no additional correction steps are required, while they are essential when advecting explicit surfaces. An algorithm providing these important features was developed and implemented in ViennaLS [2], an open-source, high performance sparse level set library tailored towards the simulation of semiconductor fabrication processes.

II. Level Set Advection

Advancing the surface described by a LS is deemed advection and is achieved by solving the equation [3]

$$\frac{\partial \phi_{(\vec{x})}}{\partial t} + V_{(\vec{x})} |\nabla \phi_{(\vec{x})}| = 0 \quad , \tag{1}$$

where $\phi_{(\vec{x})}$ is the signed distance function represented by the LS, and $V_{(\vec{x})}$ is a scalar velocity field describing the movement of the surface. Since this is a Hamilton-Jacobi equation, several numerical schemes are available for its solution. However, they all have some restrictions on how much the values of $\phi_{(\vec{x})}$ are permitted to change in one advection step in order to achieve a desired accuracy. The largest step which is permitted, regardless of the numerical scheme applied to solve Eq. (1), is given by the Courant-Friedrichs-Lewy (CFL) condition [4]. Therefore, advection must be repeated several times in order to advance the surface corresponding to a single fabrication step. This leads to the propagation of numerical errors introduced during the first advection step, resulting in problems such as flattened corner geometries during deposition [5]. Therefore, in recent years, the concept of process emulation has grown in interest, where the surface is effectively re-drawn to its new state in a single geometric step, corresponding to a complete processing step. We present a novel geometric advection algorithm and its implementation in ViennaLS, allowing for fast and accurate surface advancement.

III. Geometric Advection Algorithm

If the exact geometry of an advected surface after a fabrication step is known in advance, there is no need to directly solve Eq. (1) repeatedly while adhering to the CFL condition. The geometric relationship between the initial surface and the known final surface can simply be applied in a single step. For example, for a highly conformal deposition process, we know from the onset, that the surface will be advanced outwards isotropically by a distance calculated by multiplying the deposition rate by the process time. Therefore, the new surface can simply be deduced from the initial surface and our knowledge of the applied process. Here, we propose a geometric advection algorithm which is able to advance the surface in a single step, given that the geometric properties of the fabrication process are known.

A. Geometric Distributions

The proposed algorithm is applicable, if the geometric properties of a process can be described by a distribution. This distribution defines how each point on the surface influences the final geometry. Conceptually, the final geometry is created by performing a union between the initial surface and the geometric distribution centred at each point on the surface. Creating these unions explicitly is not necessary, since the new level set values can be calculated directly. Referring to Fig. 1, first, the distance vector $\vec{v}$ between a point on the initial surface $\vec{P}_{contribute}$ and a point close to the new surface $\vec{P}_{candidate}$ is computed. $\vec{v}$ is then used as the position vector of the geometric distribution to calculate the new LS value at $\vec{P}_{candidate}$. As can be seen in Fig. 1, for a spherical distribution with radius r, which corresponds to a perfectly conformal deposition/etching process, the signed distance d_s for a candidate point is computed using

$$d_s = |\vec{P}_{candidate} - \vec{P}_{contribute}| - r \quad . \tag{2}$$

The signed distance value d_s can directly be applied to construct the LS describing the surface after the processing step.

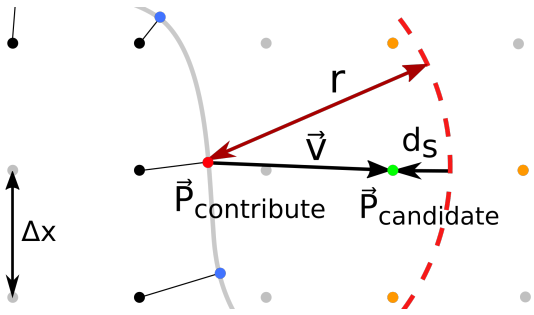

Fig. 1: Signed distance calculation for candidate point $\vec{P}_{candidate}$ (green) from an explicit point $\vec{P}_{contribute}$ (red) on the initial surface. The spherical distribution is depicted by the dashed red circle. d_s is then used as the new LS value for the grid point $\vec{P}_{candidate}$. The LS grid spacing is shown as Δx.

B. The Algorithm

The advection algorithm consists of the following steps, as shown in Fig. 2, where point colours refer to those given in Fig. 1:

1. Extract the explicit point cloud of the initial surface by shifting grid points by their LS value in the direction of the surface normal at that grid point [6] (blue points).
2. Identify candidate grid points which may be close to the new surface (orange points). In the simplest case, the bounding box of the initial surface is extended by the size of the geometric distribution in each dimension and all points in this new bounding box are checked. This simple approach was used in the presented implementation.
3. Iterate over candidate grid points while applying the following procedure:
   a) Identify initial surface points which are close enough to the current candidate point (green) to contribute to the signed distance calculation (contribute points).
      i) Iterate over contribute points: Find the signed distance using Eq. (2). If d_s is negative (inside the surface) and its absolute value is larger than Δx, discard the candidate point because it is too far away to improve the description of the surface.
      ii) Apply the d_s with the lowest value found in i) as the final signed distance value for this candidate point.
4. Construct the new LS from the final values of each candidate point which was not discarded.

After the algorithm has completed, all sparse LS points will be set with the correct signed distance, thus resulting in a valid sparse field LS. This algorithm combines the advantages of the explicit representation of the initial surface and the implicit representation of the new surface. The explicit initial surface can be accessed and reordered quickly, while the new surface, described by a sparse field LS, automatically discards self-intersecting or overlapping points.

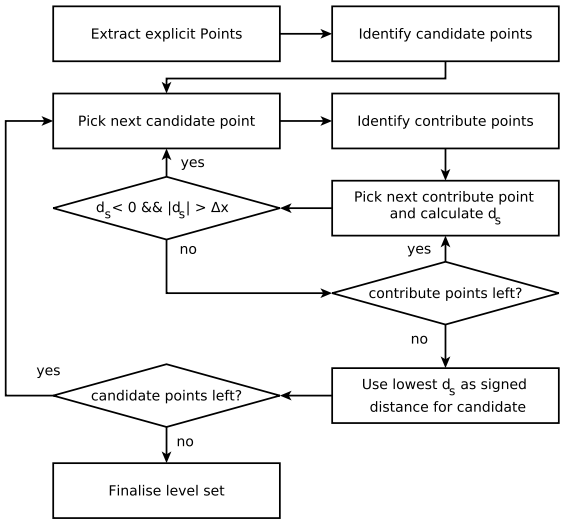

Fig. 2: Flowchart of the geometric advection algorithm.

C. Limitations

The algorithm, as presented above, is used if the surface is moved outwards (deposition), while inwards movement (etching) requires some changes to work correctly. During deposition, all level set values must become smaller, since the surface moves outwards. Therefore, the lowest level set value is used in Item 3.(a)ii. However, the opposite is the case for etching, as all level set values must become larger. Identification of candidate and contribute points remains the same, but due to the changed signs of the advection, Item 3.(a)i must be changed to discard level set values which are larger than Δx and positive, rather than negative. Similarly, in Item 3.(a)ii the d_s with the largest value is used as the final distance. These changes in the algorithm are applied automatically depending on the sign of the distribution. All other parts, as well as the robustness concerning self-intersections or overlapping phenomena remain unchanged.

The change in logic depending on the direction of movement of the surface is the only limitation of this algorithm compared to iterative advection schemes. It is not possible to advect the surface outwards at some points and inwards at others, as it is not clear whether the smallest or largest level set value should be chosen. In our implementation, this change in logic is applied automatically, based on the geometric distribution passed to the algorithm. If the distribution has a negative extent, it is treated as etching, while a positive extent means deposition. It is therefore not possible to define a distribution with a positive extent in one axis direction and a negative extent in another. Therefore, this technique may not be applicable for certain processes, where deposition and etching occurs at the same time, such as ion-enhanced plasma etching [7]. However, many of these processes could likely be approximated by splitting them each into a separate etching emulation followed by a deposition emulation.

D. Performance and Parallelisation

Perfectly isotropic deposition or etching results in a very simple signed distance calculation, as shown in Eq. (2). Since signed distance calculations are conducted for all combinations of candidate and contribute points, great care should be taken to make sure that the calculation is executed as efficiently as possible. However, since this is more or less given by the chosen geometric distribution, the possibility for runtime reduction is limited.

A far bigger performance benefit can be achieved by identifying candidate grid points efficiently, as many signed distances must be calculated for every candidate point. For the ideal performance case, no candidate points would be discarded in Item 3.(a)i, as only candidate points which are part of the final level set will have been examined. The simplest approach is setting a bounding box for the anticipated result of the advection and using all grid points in this bounding box as candidate points. This is performed in the current implementation, as it already provides reasonable performance and is the simplest solution.

The efficient identification of contribute points presents a very similar problem to picking candidate points. Since contribute points are represented explicitly, standard meshing algorithms could be applied to sort them quickly based on their proximity to the current candidate point. In the current implementation, a linear search is used to identify contribute points, which does impact performance negatively for large structures.

Since the calculation of the signed distance for each candidate point is completely independent, parallelisation is straight forward. However, the choice of suitable candidate points is important for proper load balancing, if course-grained parallelisation is used, i.e. the domain is split into a subdomain for each used thread. A more general and robust method is distributing single candidate points to threads as they become available, thus resulting in fine-grained parallelisation and efficient load

balancing. However, this would require more instances of random access into the initial level set structure, which is comparably slow in the used data structure.

E. Future Improvements

Currently, the identification of candidate points follows a simple bounding box approach, as described in Section III-D. However, a more complex approach which involves reordering the points based on proximity to the old surface could provide significant performance improvements and should be the topic of future work. This way, the new level set values would only be calculated for points a certain distance away from the initial surface, which is likely to contain the final surface.

Contribute points are identified by a simple linear search. Sorting them by proximity to the current candidate point would allow for the quick discarding of all remaining contribute points as soon as the first is identified to not contribute to the level set value. Using a more complex sorting procedure would therefore likely lead to considerable performance benefits and is intended be studied in future work.

Calculation of signed distances for candidate points could be parallelised by distributing single signed distance calculations to threads as they become available. This would result in optimal load balancing and would greatly improve the performance of the parallelised algorithm.

IV. Pinch Off Plasma CVD Process

In order to demonstrate the capabilities of the geometric advection algorithm, a chemical deposition process was emulated using ViennaLS. A pinch-off plasma CVD process [8] was chosen, because its geometric properties can be deduced from scanning transmission electron microscope (STEM) measurements to generate a predictive process emulation model solely reliant on geometric considerations. The simplest approximation for this anisotropic deposition process is based on the view factors of the points in the trench to the opening plane. This approach has already been used to describe deposition processes in trenches and vias [9]. For an infinite trench, an ideal solution exists and is given by [10]

$$F(d) = \frac{1}{2} + \frac{cos(\theta) - \frac{d}{D}}{2\sqrt{1 - 2\frac{d}{D}cos(\theta) + \left(\frac{d}{D}\right)^2}} \quad , \tag{3}$$

where $F(z)$ is the view factor of a point a distance d down the sidewall, which is tapered by the angle θ. This is the view factor with respect to an opening plane of width D. Since the deposition at each point is isotropic a spherical advection kernel was chosen to represent the process. However, since the deposition rate changes with depth down the trench, the radius of the different advection distributions differs, as shown in Fig. 3. Since $F(d)$ only depends on the downward distance along the trench sidewall d, the radius of the spherical advection kernel $R(d)$ also only depends on d with

$$R(d) = R_0 F(d) \quad , \tag{4}$$

where R_0 is the deposition distance at the opening of the trench. For the modelled process R_0 was chosen as one half of the opening width of the trench D, meaning that the timing of the trench pinch-off will be exact during the process. The distance down the slanted sidewall d is not directly available in the simulator, but can be found from the vertical coordinate z of a point on the sidewall:

$$d = \frac{z}{cos(\theta - \pi/2)} \tag{5}$$

The angle of the sidewall slant of the copper lines in the chosen process is $\theta = 95°$ and the pitch of the copper lines is 48nm. The copper line width decreases linearly from a width of 25nm at the top to 18nm at the bottom. The height of copper lines is 54nm [8]. R_0 was chosen to be 12nm, slightly more than half the opening width of 23nm, to ensure a proper closing of the trench.

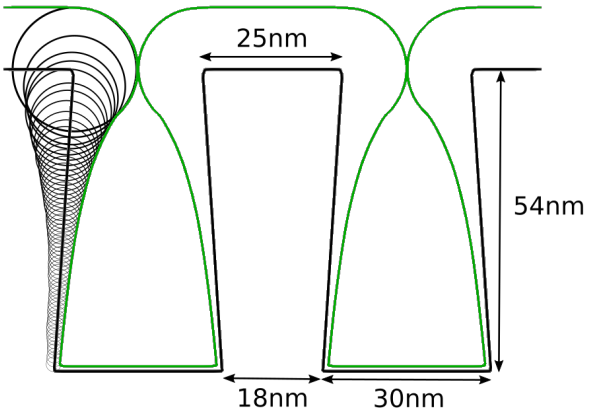

Fig. 3: Schematic showing how the described model creates the resulting geometry (green) using spherical geometric distributions (black circles). The outline of the circles thus creates the final geometry, based on the view factor calculated using Eq. (3).

V. Results

The presented pinch-off plasma CVD process was emulated with high resolution using several advection schemes. The presented algorithm was the only one that could produce the expected results, matching experimental data. The result of a three-dimensional simulation with periodic boundary conditions is shown in Fig. 4

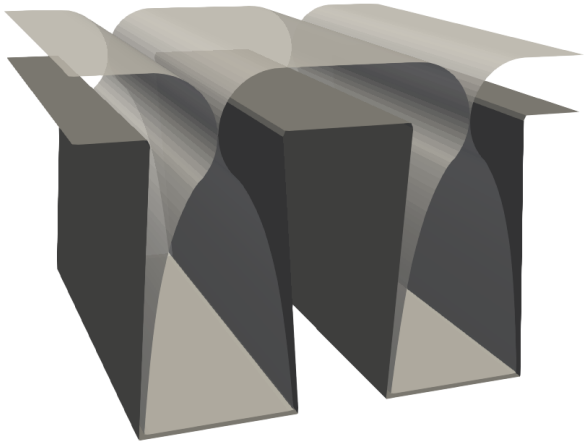

Fig. 4: Result of three-dimensional emulation of the pinch-off plasma CVD model. As expected, the resulting surface is pinched off exactly since the advection is purely geometric.

A comparison of the geometric advection and a sophisticated iterative scheme, the Stencil-Local Lax-Friedrichs (SLLF) scheme [11], is shown in Fig. 5. The exact timing of the closing of the trench is important, as the geometry inside the trench is fixed thereafter. Since the new surface can be calculated directly, the pinch-off follows the predicted time

exactly, because it is produced analytically. As evidenced in Fig. 5, the SLLF scheme is unable to reproduce the pinch-off time, meaning that the air void will not have the desired shape at the end of the simulation. In addition to SLLF, the presented emulation algorithm was compared to several other iterative schemes, as summarized in Table I. As is evident from these results, iterative schemes are unable to predict this pinch-off correctly, as they include numerical errors (i.e., corner flattening discussed earlier), even though exactly the same model is applied. In this case, more complex schemes produce even larger errors, as they include damping terms, slowing the advection.

A runtime comparison with conventional advection schemes is also given in Table I. As can be seen, the geometric advection algorithm provides high accuracy with reasonable performance. Simple advection schemes, such as the Engquist-Osher and Lax-Friedrichs schemes provide smaller runtimes, but suffer from the introduction of numerical errors originating from the numerical derivatives. Furthermore, like all iterative schemes, they are susceptible to the propagation of errors, as discussed previously. More complex schemes, such as the Local-Local-Lax-Friedrichs and the Local-Lax-Friedrichs scheme, produce even larger errors, due to the larger damping terms they include, which makes them more stable in complex geometries, but leads to slower overall front propagation, ergo a larger pinch-off error. Even the complex SLLF scheme, which includes numerous corrections [11], cannot emulate the deposition process correctly. This comparative study further suggests that simply using a higher-order and higher-complexity scheme does not necessarily provide more accurate results. One must always be aware of the geometry which is being simulated and an advection scheme should be chosen accordingly. Here, we note that, for trench CVD and pinch-off, the Lax-Friedrichs scheme is most appropriate for pinch-off simulations; nevertheless, it cannot match the accuracy of the proposed geometric advection scheme.

Another problem of iterative schemes is that they may be accurate for certain applications and geometries, but not all. Therefore, the accuracy they provide is not easily predictable. The great advantage of the proposed geometric advection algorithm is that it always provides algebraic accuracy. This is possible, because it does not rely on the derivatives generated in the level set, which always include some numerical errors. Thus, the proposed advection algorithm is suitable for fast, accurate and predictive high-performance process emulation with predictable accuracy and performance.

Advection scheme	Runtime (s)	Pinch-off error (nm)
Engquist-Osher (1st)	8.5	3.7
Engquist-Osher (2nd)	18.3	2.8
Lax-Friedrichs (1st)	8.5	2.5
Lax-Friedrichs (2nd)	19.0	2.1
Local-Local-Lax-Friedrichs (1st)	8.2	4.0
Local-Local-Lax-Friedrichs (2nd)	18.4	3.0
Local Lax-Friedrichs (1st)	143.3	6.5
Local Lax-Friedrichs (2nd)	171.0	4.1
SLLF (1st) [11]	181.3	3.6
Geometric Advection	**30.6**	**0.0**

TABLE I: Runtimes of different advection schemes for the three-dimensional pinch-off plasma process with 62,662 LS points on the initial surface compared to the presented geometric advection (Fig. 4). The pinch-off error describes, how wide the opening of the trench is at the expected pinch-off time. The approximation order for derivatives is listed in the parenthesis. All simulations are performed on an AMD Ryzen 3950X CPU using 32 threads.

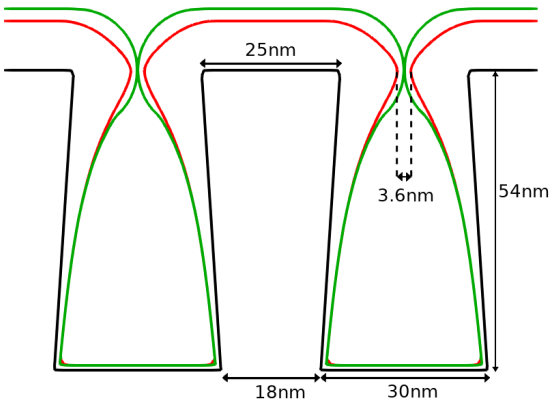

Fig. 5: Slice of a three-dimensional emulation of the pinch-off plasma CVD process inside tapered trenches (black surface). The green surface is the result of the geometric advection, while even the most robust iterative scheme (SLLF [11], red surface) is unable to predict the closing of the trench.

VI. Conclusion

A purely geometric advection scheme was developed and implemented in the level set framework, ViennaLS, allowing for the emulation of an entire deposition process in a single step with higher accuracy than conventional advection schemes. This is due to numerical errors introduced when calculating derivatives in the level set stemming from its limited resolution. Furthermore, conventional schemes must be applied iteratively as there is a limit on how much the values of the level set are permitted to change in a single advection step before the numerical scheme becomes unstable. This leads to the propagation – and possibly the amplification – of errors, such as the flattening of corner geometries.

In order to test the applicability of the proposed algorithm, a pinch-off CVD process was emulated and compared to conventional advection schemes. The proposed advection algorithm was shown to outperform even sophisticated advection schemes in both runtime and accuracy. Compared to the sophisticated SLLF schemes, up to 6 times faster runtimes have been achieved. Although the proposed algorithm was shown to be slower than simple iterative advection schemes, the provided accuracy still warrants the performance penalty of a factor of 2. It could therefore be shown, that the proposed geometric advection algorithm is suitable for efficient and highly accurate emulation of complex semiconductor fabrication processes.

References

[1] O. Ertl, Ph.D. dissertation, TU Wien, 2010.

[2] X. Klemenschits, O. Ertl, P. Manstetten, J. Weinbub, and L. Filipovic. ViennaLS. [Online]. Available: https://github.com/ViennaTools/ViennaLS

[3] S. Osher and J. A. Sethian, *J. Comput. Phys.*, vol. 79, no. 1, pp. 12–49, 1988. doi: 10.1016/0021-9991(88)90002-2

[4] R. Courant, K. Friedrichs, and H. Lewy, *Math. Ann.*, vol. 100, no. 1, pp. 32–74, 1928. doi: 10.1007/BF01448839

[5] J. Sethian and D. Adalsteinsson, *IEEE Trans. Semicond. Manuf.*, vol. 10, no. 1, pp. 167–184, 1997. doi: 10.1109/66.554505

[6] O. Ertl and S. Selberherr, in *Proc. SISPAD*, 2008. pp. 325–328. doi: 10.1109/SISPAD.2008.4648303

[7] X. Klemenschits, S. Selberherr, and L. Filipovic, *Micromachines*, vol. 9, no. 12, p. 631, 2018. doi: 10.3390/mi9120631

[8] S. Nguyen, T. Haigh, K. Cheng, C. Penny, C. Park, J. Li, S. Mehta, T. Yamashita, L. Jiang, and D. Canaperi, *ECS J. Solid State Sc.*, vol. 7, no. 10, pp. P588–P594, 2018. doi: 10.1149/2.0021811jss

[9] P. Manstetten, L. Filipovic, A. Hössinger, J. Weinbub, and S. Selberherr, *Solid-State Electronics*, vol. 128, pp. 141–147, 2017. doi: 10.1016/j.sse.2016.10.029

[10] M. F. Modest, *Radiative Heat Transfer*, 3rd ed. Academic Press, 2013. ISBN 9780123869449

[11] A. Toifl, M. Quell, X. Klemenschits, P. Manstetten, A. Hössinger, S. Selberherr, and J. Weinbub, *IEEE Access*, vol. 8, pp. 115 406–115 422, 2020. doi: 10.1109/ACCESS.2020.3004136