A New Module for Automated Optimization of Process TCAD Model Parameters

Roman Kostal[1*], Tobias Reiter[2], and Lado Filipovic[2]
[1]Institute for Microelectronics, TU Wien, 1040 Vienna, Austria
[2]CDL for Multi-Scale Process Modeling of Semiconductor Devices and Sensors at the Institute for Microelectronics
*Email: kostal@iue.tuwien.ac.at

***Abstract*—A new module for automated optimization of process TCAD model parameters is presented along with example use cases. Features include: novel ways of measuring profile discrepancy, seamless comparison of simulated 3D structures with 2D micrographs of experimental structures, and integration of state-of-the-art open source optimizers and tools for sensitivity analysis. Demonstrated through practical examples, the module enables excellent matching of simulated profiles to experimental micrographs for process calibration and provides valuable insights into parameter sensitivities using global sensitivity analysis, thereby addressing key challenges in TCAD model development.**

***Index Terms*—Deposition, Etching, Process Simulation, Emulation, Optimization, Sensitivity Analysis**

I. Introduction and Motivation

Simulation models for thin-film deposition and etching processes typically rely on a set of user-defined parameters. These parameters can be refined through automated optimization to reduce the discrepancy between simulated and experimentally observed structures. In most cases, experimental data is obtained from two-dimensional (2D) electron micrographs. Because three-dimensional (3D) reconstruction is rarely performed due to its high cost and complexity, model validation is usually restricted to comparisons of 2D cross-sectional surface profiles (SPs). The resulting differences at several critical dimensions (CDs) are then used to define the objective for optimization.

In a recent study [1], we highlighted several limitations of this approach and proposed new global measures of profile discrepancy, implemented in our in-house process simulation framework, ViennaPS [2]. Specifically, these measures improve upon traditional CD-based approaches by quantifying the full profile mismatch, providing a more robust objective for optimization. Building on this work, we now present a unified module for model calibration and evaluation.

II. The Optimization Module

A. Initial Setup

The surface profiles (SPs) of experimental structures are systematically extracted from electron micrographs and represented as polygonal chains (PCs) that precisely capture the geometry of material interfaces. Both fully automated and semi-automated extraction methods have been successfully used for the extraction. Once extracted, the polygonal chains are seamlessly imported into ViennaPS through a standardized interface that preserves geometric accuracy and maintains traceability to the original experimental data. This is achieved by an internal conversion process in which the polygons are transformed into implicit surface representations using the well-established level-set method [3], and subsequently stored in custom data formats. They can then serve as both the target surface (TS), i.e. the result after applying the process of interest, and as the starting surface, e.g. when modeling a deposition process. In case the initial surface has a simple geometry, e.g. when etching into a flat wafer with a regular tapered mask, the extraction of the initial surface can be skipped and built-in functions for geometry creation can be used to set it up instead.

B. Optimization

Evaluating the objective function $f(\mathbf{x})$ at a point $\mathbf{x} = (x_1, \ldots, x_{n_v})$, involves running a simulation and quantifying the discrepancy between the resulting profile and the target profile, ideally by a single number to avoid the added complexity associated with multi-objective optimization. In this study, we make use of two metrics: The area difference metric (AD), which estimates the area where the overlaid profiles do not match, and the narrow band distance metric (NBD), which sums the squares of the distances between the two surfaces on grid points close to the simulated surface. Both methods are visualized on an example structure in Figure 4, while the details of their implementation can be found in [1].

With f defined as above, the gradient ∇f is not readily available. Even the simplest first-order accurate estimate of ∇f, such as forward differences, requires $n_v + 1$ evaluations of f, making this approach prohibitively expensive. Hence, we use derivative-free optimizers from `dlib` [4] and `Nevergrad` [5], which proved to be highly effective. Specifically, we employ `dlib`'s `find_min_global`, and Nevergrad's adaptive meta-optimizer `NGOpt4`. These algorithms blend multiple gradient-free strategies to tackle black-box optimization. The optimization workflow consists of the following steps:

1) Select the model parameters to serve as decision variables and introduce bounds and constraints as needed.
2) Choose an optimizer and define a termination criterion, such as a maximum number of evaluations of f or a runtime limit.
3) Choose a distance metric for evaluating the discrepancy between the simulated and the experimental surface.

4) Execute the optimization loop until the stopping criteria in 2) are met:
   - Run the simulation at a proposed point $\mathbf{x}$.
   - In case the simulation was performed in 3D, slice the resulting structure at specified evaluation planes using the available slicing functions.
   - Evaluate the distance metric.
   - Select a new point based on previous evaluations of the objective function f.

C. Local and Global Sensitivity Analysis

Once a local minimum $f^* = f(\mathbf{x}^*)$ is found, it is important to assess the impact of individual parameters x_i^* and understand the local behavior of f around $\mathbf{x}^*$. This can be done through *local sensitivity analysis* (LSA) by evaluating f and estimating ∇f in the neighborhood of $\mathbf{x}^*$.

However, process TCAD models typically involve a large number of parameters. Allowing n_v of these parameters to become optimization variables, the search space volume V_S grows exponentially with n_v, making optimization increasingly difficult. To address this challenge, we perform global sensitivity analysis (GSA). GSA quantifies how input variables influence output variability across the entire search space. This provides valuable insights into their relative importance. By identifying low-impact variables, these can be fixed or discretized in further investigations, effectively reducing V_S.

To conduct GSA, we integrate the `SALib` Python library [6] into our module. Several popular GSA metrics are available in SALib, from which we chose the *Sobol sensitivity indices* [7], as they are highly regarded and considered a benchmark in the field [8]. These indices are *variance based*, and hence defined within a probabilistic framework, where both the inputs and the output of f are considered random variables. We therefore adopt the probabilistic notation $Y = f(\mathbf{X})$, $\mathbf{X} = (X_1, \ldots, X_{n_v})$ from now on. We also consider the X_i to be uniformly distributed within the valid ranges which are typically chosen during model development. This is common practice in sensitivity analysis applications. The assignment of input distributions is an integral part of the modeling process and does not necessarily imply that the variables have inherent stochastic uncertainty [8].

Given the objective function $f(\mathbf{X})$ with bounds on the variables expressed as $\mathbf{X} \in \Omega \subset \mathbb{R}^{n_v}$, the total variance of f under the above assumptions is then given by:

$$\mathbb{V}[f] = \int_\Omega f^2(\mathbf{X})\, d\mathbf{X} - \left(\int_\Omega f(\mathbf{X})\, d\mathbf{X} \right)^2. \tag{1}$$

The *first-order Sobol index* S_i quantifies the contribution of the variable X_i to the output variance in isolation:

$$S_i = \frac{\mathbb{V}_{X_i}\left[\mathbb{E}_{\mathbf{X}_{\sim i}}[f(\mathbf{X}) \mid X_i]\right]}{\mathbb{V}[f]}, \tag{2}$$

where $\mathbb{E}_{\mathbf{X}_{\sim i}}[\cdot]$ denotes the expectation over all variables except X_i, and $\mathbb{V}_{X_i}[\cdot]$ denotes the variance with respect to X_i.

The *second-order Sobol index* S_{ij} captures the interaction effect between X_i and X_j, beyond their individual contributions:

$$S_{ij} = \frac{\mathbb{V}_{X_i,X_j}\left[\mathbb{E}_{\mathbf{X}_{\sim ij}}[f(\mathbf{X}) \mid X_i, X_j]\right]}{\mathbb{V}[f]} - S_i - S_j. \tag{3}$$

Higher-order Sobol indices can be defined analogously, but are rarely used in practice due to the high computational cost of estimating them.

The *total-order Sobol index* S_i^T measures the contribution of X_i including all interactions with other variables:

$$S_i^T = \frac{\mathbb{E}_{\mathbf{X}_{\sim i}}\left[\mathbb{V}_{X_i}\left[f(\mathbf{X}) \mid \mathbf{X}_{\sim i}\right]\right]}{\mathbb{V}[f]}. \tag{4}$$

Since $f(\mathbf{X})$ is not analytically tractable, the variances and conditional expectations required for computing the indices cannot be evaluated in closed form and instead must be approximated using numerical techniques. We employ *Saltelli's sampling scheme* [9] as available in SALib, which uses quasi-random Sobol sequences in a specific matrix construction to efficiently estimate S_i, S_{ij} and S_i^T. This approach offers improved convergence and reduced estimator variance compared to conventional Monte Carlo integration. Full details of the sampling and estimation procedure can be found in the original reference. Lastly, it is also worth noting that the GSA workflow is almost identical to the optimization workflow described in Section II-B, but with the following minor changes:

- In step 2), instead of choosing the number of evaluations of f directly, one chooses the so-called sample size, N, as appropriate based on n_v. Depending on whether the second-order indices S_{ij} are to be estimated along with the S_i and S_i^T, this then requires $\mathcal{N} = N(2n_v + 2)$ or $\mathcal{N} = N(n_v + 2)$ evaluations of f, respectively.
- In step 4) the loop is executed at predefined points based on the chosen sampling scheme, each visited point and the value of f at that point are stored.

The resulting dataset can be used to estimate the Sobol indices.

III. Examples

A. Optimization of an SF_6/O_2 Etching Model

We applied the workflow described in Section II-B to a classical model of silicon etching in an SF_6/O_2 plasma proposed by Belen et al [10]. The simulations were done in 3D, using a quarter of a hole with reflective boundary conditions to increase computational efficiency. The initial setup and the slicing for profile comparison are shown in Figure 1. The objective function was defined as the sum of the AD and NBD metrics, as described in [1] and illustrated in Figures 1 and 4.

We used `dlib`'s `find_min_global` to minimize this function, varying the following parameters: Ion flux, fluorine (F) flux, oxygen (O) flux, mean ion energy, silicon ion enhanced etching coefficient (IEEC), and passivation ion enhanced etching coefficient. Details on the surface kinetics model are available in the ViennaPS documentation [2].

The experimental data overlaid with optimization results are shown in Figure 2. We see excellent matches for the target

structures. Looking at the convergence graph in Figure 3, we see rapid convergence of the optimizer in the first 100 evaluations of f, followed by a plateau lasting hundreds of evaluations until a further valley with a new local minimum is found.

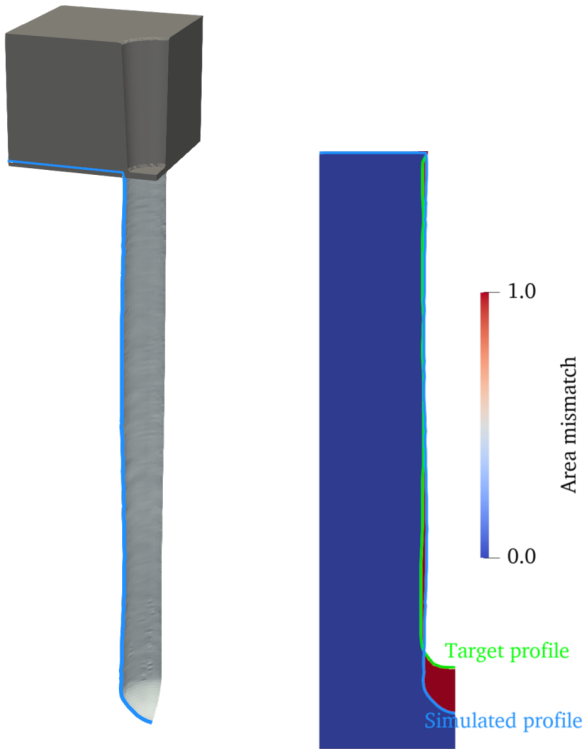

Fig. 1. Left: The setup for silicon (Si) hole etching of the middle hole in Figure 2. Dark gray indicates the mask at the beginning of etching; light gray represents the simulated hole; light blue is the cross-sectional surface profile used for comparison. Right: 2D slice comparison using the AD metric.

B. GSA of a DRIE Emulation Model

Deep reactive-ion etching (DRIE) with SF_6 and C_4F_8 plasmas consists of three alternating steps: (1) isotropic etching with SF_6 plasma, (2) polymer deposition with C_4F_8 plasma, and (3) directional etching with high-bias SF_6 plasma to remove the bottom polymer layer and expose the substrate for another round of isotropic etching (1). This cyclic process preserves vertical sidewalls and enables high-aspect-ratio (HAR) features to be etched.

To emulate this process sequence, we set up a simplified model in which surface segments are propagated by prescribed velocity fields corresponding to each of the steps described above. The model includes four key parameters:

- IED and DED - isotropic etch depth and directional etch depth (Step 1)
- DT - polymer deposition thickness (Step 2)
- PTD - punch-through depth into the substrate (Step 3)

Figure 4 shows the target vs. emulation discrepancy using both the NBD and AD metrics after running the simulation at an arbitrary point $\mathbf{x} \in \Omega$. The sum of NBD and AD was used for the objective function.

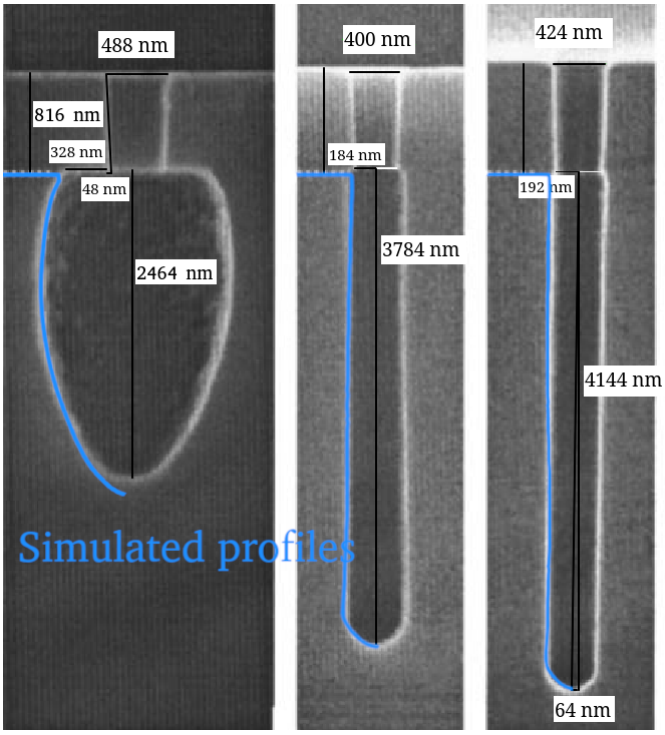

Fig. 2. Si hole etching in an SF_6/O_2 plasma with various fractions of O_2 in the feed gas: $y_{O_2} = 0$, $y_{O_2} = 0.44$, $y_{O_2} = 0.5$. Adapted from [10].

GSA was then performed using Sobol indices to determine the influence of the decision variables. $N = 2048$ samples were chosen and the choice to also estimate the second-order indices resulted in $\mathcal{N} = 20480$ evaluations of f in our case of $n_v = 4$. The total runtime was about 13 hours on a 6-core CPU. The results are shown in Figure 5. They indicate that the IED exhibits the highest total sensitivity, followed by the DED and PTD. Notably, DT has a negligible impact across all sensitivity orders, indicating it can be safely excluded or fixed in future optimization studies. Second-order indices (S2) highlight strong interactions between IED–DED and IED–PTD, suggesting joint tuning is key to capturing model behavior. Overall, the GSA highlights dominant variables and interactions, guiding dimensionality reduction and more efficient parameter exploration in future DRIE modeling.

IV. Discussion

We have presented the capabilities of an open-source software module which addresses a common need in developing process TCAD models. The package is written entirely in Python, is available on GitHub [11] and can be installed with the standard commands. Many features are planned to be added in future versions, such as automated cross-validation methods, support for comparisons of 3D surfaces, faster GSA methods and screening methods, and pre-configured LSA methods.

Acknowledgement

Financial support by the Federal Ministry of Labour and Economy, the National Foundation for Research, Technology

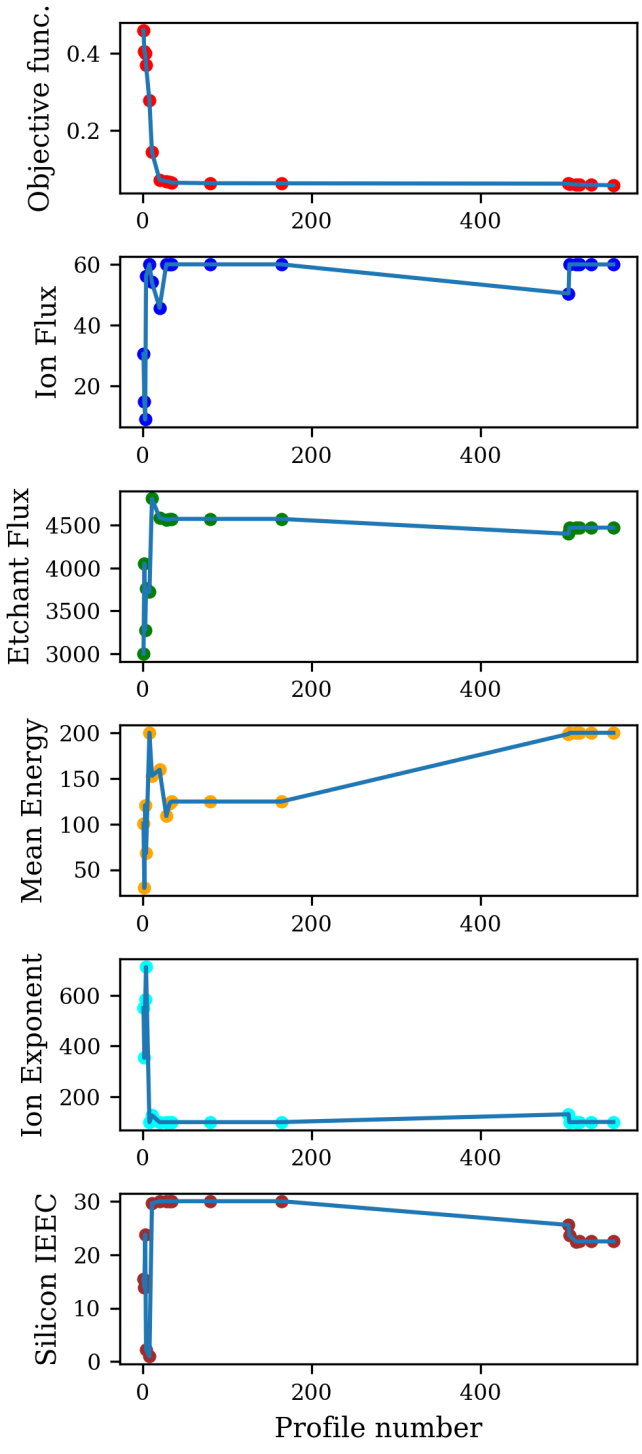

Fig. 3. Convergence plot of the optimization run for the left hole in figure 2. All fluxes are units of $10^{15}\mathrm{cm}^{-2}\mathrm{s}^{-2}$, the mean energy is in eV, the remaining quantities are dimensionless.

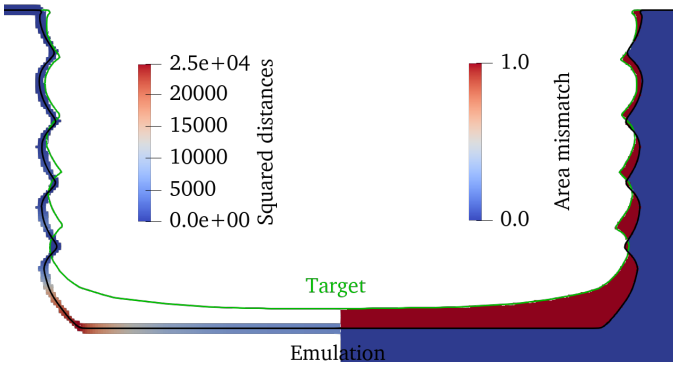

Fig. 4. Setup of III-B and illustration of the f used in both III-A and III-B: NBD metric on the left, the AD metric on the right.

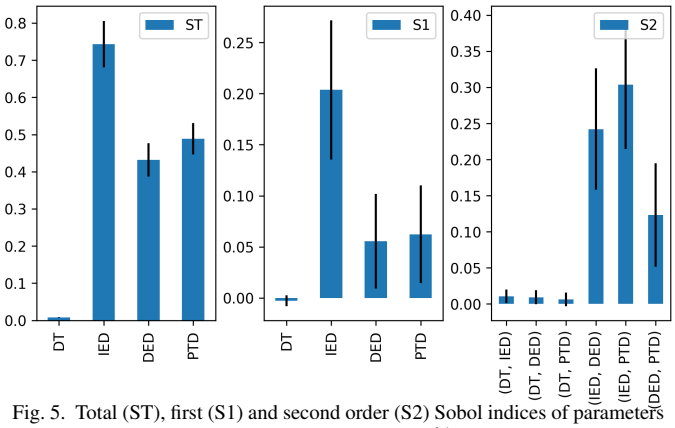

Fig. 5. Total (ST), first (S1) and second order (S2) Sobol indices of parameters of a DRIE emulation model. Black bars show 95% confidence intervals.

and Development and the Christian Doppler Research Association is gratefully acknowledged.

References

[1] R. Kostal *et al.*, "Novel approaches to objective function design for optimizing process TCAD model parameters," in *24th Conference on Insulating Films on Semiconductors*, 2025, accepted.

[2] T. Reiter *et al.*, "ViennaPS," https://github.com/ViennaTools/ViennaPS, 2025.

[3] O. Ertl, "Numerical methods for topography simulation," Doctoral dissertation, Vienna University of Technology, 2010.

[4] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009. [Online]. Available: http://jmlr.org/papers/v10/king09a.html

[5] J. Rapin and O. Teytaud, "Nevergrad - A gradient-free optimization platform," https://github.com/FacebookResearch/Nevergrad, 2018.

[6] J. Herman and W. Usher, "SALib: An open-source python library for sensitivity analysis," *The Journal of Open Source Software*, vol. 2, no. 9, 2017.

[7] I. M. Sobol', "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Mathematics and Computers in Simulation*, vol. 55, no. 1, pp. 271–280, Feb. 2001.

[8] A. Saltelli *et al.*, *Global Sensitivity Analysis. The Primer*. John Wiley & Sons, 2008.

[9] A. Saltelli, "Making best use of model evaluations to compute sensitivity indices," *Computer Physics Communications*, vol. 145, no. 2, pp. 280–297, May 2002.

[10] R. J. Belen *et al.*, "Feature-scale model of Si etching in SF6/O2 plasma and comparison with experiments," *Journal of Vacuum Science & Technology A*, vol. 23, no. 5, Aug. 2005.

[11] R. Kostal, "ViennaFit," https://github.com/ViennaTools/ViennaFit, 2025.