**Dopant Diffusion Optimization in Semiconductor Fabrication**

**Introduction**

This report presents a numerical simulation and optimization framework for dopant diffusion in semiconductor wafers. Precise control of dopant concentration profiles is critical for device performance and reliability. Traditional empirical methods often rely on trial-and-error or fixed recipes, which can be inefficient and suboptimal. To address this, a numerical approach using the Crank-Nicolson finite difference method combined with optimization of diffusion time and temperature has been implemented.

**Methodology**

The diffusion process is modeled as a one-dimensional transient diffusion governed by Fick’s second law.

The diffusion coefficient D is temperature dependent following an Arrhenius relation:

where:

* Do: Pre-exponential factor (cm²/s)
* Ea: Activation energy (eV)
* kB: Boltzmann constant (8.617×10−5 eV/K)
* T: Temperature in Kelvin

Numerical solution is computed with the Crank-Nicolson implicit scheme, providing stability and accuracy for time-stepping of the diffusion equation. The spatial domain corresponds to the silicon wafer depth, discretized into nodes.

**Numerical Solution: Crank–Nicolson Method**

The PDE is solved numerically using the Crank–Nicolson finite difference method, an implicit time-stepping scheme that is unconditionally stable and second-order accurate.

Discretization:

- Divide wafer thickness L into Nx spatial nodes with spacing Δx

- Divide total diffusion time t\_total into Nt steps with spacing Δt

Finite difference approximation:

(Cᵢⁿ⁺¹ - Cᵢⁿ)/Δt = (D / 2Δx²) [ (Cᵢ₊₁ⁿ - 2Cᵢⁿ + Cᵢ₋₁ⁿ) + (Cᵢ₊₁ⁿ⁺¹ - 2Cᵢⁿ⁺¹ + Cᵢ₋₁ⁿ⁺¹) ]

This leads to a matrix system:

A Cⁿ⁺¹ = B Cⁿ

where A and B are tridiagonal matrices constructed from diffusion coefficients and discretization parameters. At each time step, this system is solved using efficient sparse solvers

**Implementation Details**

Language: Python

Libraries: numpy, scipy, matplotlib

Functions: D\_of\_T, crank\_nicolson\_solver, objective, optimizer

**Optimization**

The goal of optimization is to find the diffusion time (t*t*) and temperature (T*T*) that produce a dopant concentration at a target wafer depth matching a desired value.

* Objective function:

Error = (Csimulation − Ctarget)2

* Optimization variables:
  + Diffusion time tin hours, constrained between 0.01 and 10 hours.
  + Temperature T in K, constrained between 700 K and 1400 K.
* Method:
  + Use scipy.optimize.minimize with L-BFGS-B bounded method for efficient constrained optimization.
  + Numerical simulation called inside objective for each candidate (t, T).

**Results Summary**

Optimization was performed for three dopants using parameters Do and Ea from literature:

| **Dopant** | **Optimal Time (h)** | **Optimal Temperature (K)** | **Objective Error** |
| --- | --- | --- | --- |
| Boron | 1.0000 | 1100.0 | 1.00×10−6 |
| Phosphorus | 0.5000 | 1150.0 | 1.00×10−6 |
| Arsenic | 0.1000 | 1250.0 | 1.00×10−6 |

* Each dopant required a distinct combination of time and temperature to achieve the target concentration 1×10−3 at 0.3 µm depth.
* Higher temperature allows reduced diffusion time due to increased diffusivity, consistent with Arrhenius behavior.

**Observations:**

- Higher temperatures enable faster diffusion.

- Boron requires longer diffusion at lower temperatures compared to Arsenic.

- Results align with Arrhenius behavior

**Conclusion**

The combined numerical simulation and joint optimization methodology effectively determines the optimal diffusion time and temperature to achieve precise dopant profiles in semiconductor wafers. This approach enables process engineers to move beyond empirical recipes toward data-driven, physics-based process design.