An Improved Unified Solver for Modeling Defect Dynamics in Materials

A. Shaik[1], D. Brinkman[2], C. Ringhofer[1], D. Vasileska[1]
[1]*Arizona State University, USA*
[2]*San Jose State University, USA*
arshaik@asu.edu

In our previous works, a unified solver was developed and used to explain the copper (Cu) migration in CdTe solar cells [1-3]. In this work we present an improved version of the solver based on an operator splitting scheme, splitting the model equations into the reaction term and drift diffusion term. We employ a finite volume meshing instead of finite difference [1] and finite element meshing [3-4] from our previous solver. The performance improvement is due to our novel method of solving the reactions by a Jacobian method and implicit differencing in time, thus improving stability and accuracy. Although the reaction solver improved the speed and accuracy, we still faced challenges in conservation of charge and mass. We proposed non-linear heuristic correction scheme to avoid the conservation issues. The simulation time scales vary from nano seconds to decades for studying process simulation at elevated temperature to long term stress condition in solar cells. For long term simulation diffusion solver failed to conserve mass and we used method of pseudo inverses to correct them. Finally, we can avoid both issues by not using the operator splitting scheme. This full operator method results in slower simulation speed.

We follow object-oriented programming paradigm of MATLAB to code our numerical engine into classes implementing the different solvers and algorithms. This gives us flexibility to easily incorporate new solver classes and algorithms into our numerical engine. Currently, we support 3 diffusion solvers, 2 reaction solvers and 5 Poisson solvers. In terms of algorithm we support full operator algorithm and operator splitting algorithm.

We validate our solver by comparing the light IV curves simulated through our drift diffusion reaction formalism and Silvaco's atlas commercial solver which implements drift diffusion formalism at thermal equilibrium. The results show that code implementation and our formalism assumptions are correct.

[1] Da Guo et al., J. Phys. D: Appl. Phys., **51**, 153002 (2018).
[2] Da GUO et al., https://nanohub.org/resources/predicts1d. (DOI: 10.21981/D34T6F43J) (2015).
[3] Abdul Shaik et al., https://nanohub.org/resources/predicts2d. (DOI: 10.21981/D32N4ZJ9X) (2016).
[4] Daniel Brinkman et al., J. Appl. Phys. **118**, 035704 (2015).
[5] Silvaco Software Tool., https://www.silvaco.com (2018).

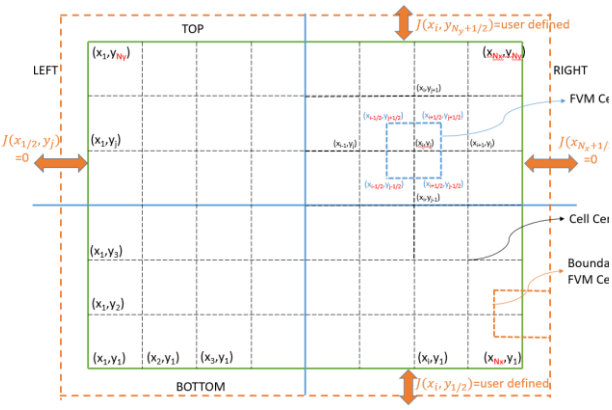

Fig.1: Finite Volume Mesh (FVM) used for discretizing differential equations

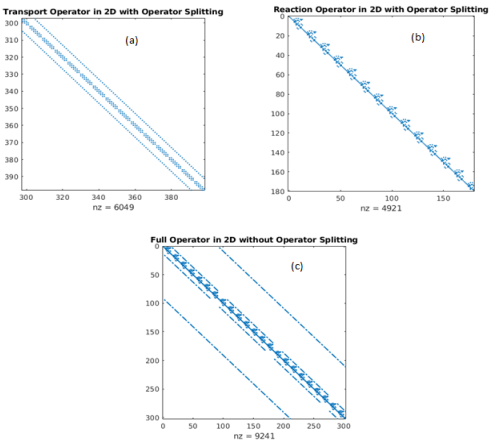

Fig.4: Shape of LU decomposition matrix in operator splitting scheme (a) Transport Operator (b) Reaction Operator and (c) full operator scheme.

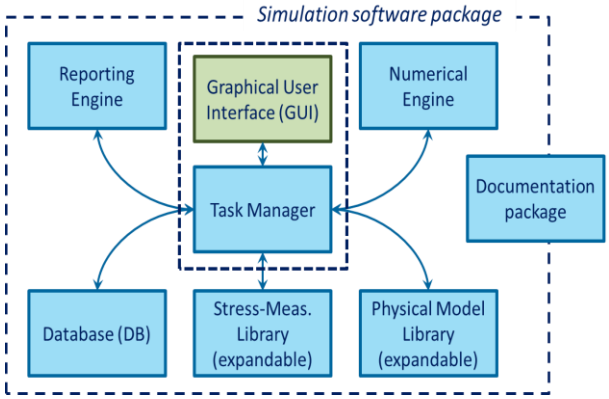

Fig.2: Software modules used in the improved solver..

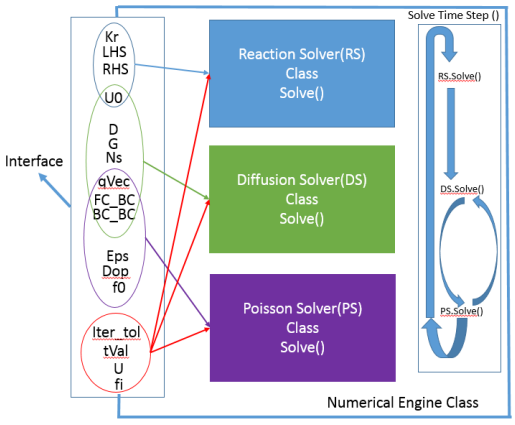

Fig.5 OOP Object organization for implementing numerical engine with reaction solvers, diffusion solver and poisson solvers and algorithms.

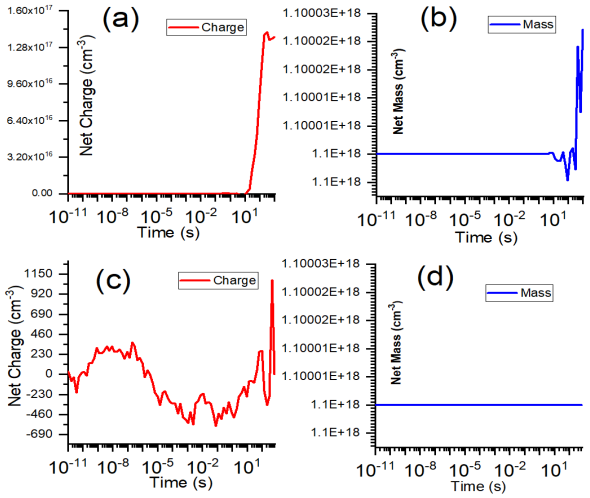

Fig.3: (a) Net charge (b) Net mass without corrections, (c) net charge and (d) net mass with corrections.

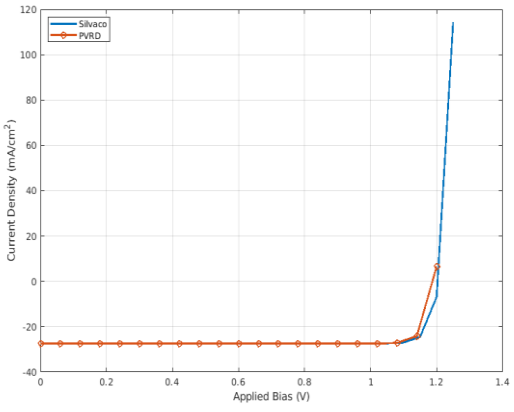

Fig.6: Comparison of light IV curves from our solver to Silvaco atlas solver.