AN ABSTRACT OF THE THESIS OF

Timothy B. Costa for the degree of Master of Science in Mathematics presented on February 27, 2014.

Title: Analysis of Domain Decomposition Methods for the Simulation of Charge Transport in Semiconductor Structures with Heterojunctions

Abstract approved: _____

Malgorzata S. Peszyńska

Charge transport in a semiconductor structure with heterojunction is described by a multiscale partial differential equation model. This model can be used, e.g., for the design of more efficient solar cells.

Phenomena at the heterojunction must be resolved at the angstrom scale while the size of the device is that of microns. The challenge is therefore to account for correct physics and to keep the model computationally tractable. Thus we use an approach introduced by Horio and Yanai in which the physics at the interface is approximated at the device scale, which is handled by tranditional drift diffusion equations, by unusual jump conditions, called thermionic emission equations. In this model the heterojunction region is approximated by an abrupt interface, resulting in a loss of continuity in the primary variables. The thermionic emission equations consist of a nonhomogeneous jump in the electrostatic potential and unusual Robin-like conditions for carrier transport. The data for these jumps is determined from an angstrom scale first principles calculation in the true heterojunction region.

The continuum scale model lends itself well to a domain decomposition approach. In this thesis we present iterative substructuring methods developed for the drift diffusion

system with thermionic emission transmission conditions and analyze the convergence of these algorithms.

Analysis of Domain Decomposition Methods for the Simulation of Charge Transport in Semiconductor Structures with Heterojunctions

by
Timothy B. Costa

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented February 27, 2014
Commencement June 2014

Master of Science thesis of Timothy B. Costa presented on February 27, 2014

APPROVED:

Major Professor, representing Mathematics

Chair of the Department of Mathematics

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Timothy B. Costa, Author

ACKNOWLEDGEMENTS

Academic

I am indebted to Dr. Malgorzata Peszynska for the advice and guidance she provided me as well as for sharing with me her enthusiasm for applied and computational mathematics.

Research in this work was performed under the support of the National Science Foundation, grant 1035513 "SOLAR: Enhanced Photovoltaic Efficiency through Heterojunction Assisted Impact Ionization," (Principal Investigator Stephen Kevan, Co-Principal Investigator Malgorzata Peszynska, Co-Principal Investigator Geraldine Richmond) under the direction of Malgorzata Peszynska.

I would also like to thank my collaborators David Foster and Guenter Schneider from the Physics Department at Oregon State University for their hard work and insights, without which this thesis would want for physical grounding.

Personal

I wish to thank my wife, Jen, and daughter, Emily, for their unwavering support. I also wish to thank my parents, John and Denise Costa, for a lifetime of support and enthusiasm for my endeavors. Additionally, thanks to my brother, Anthony, for many productive conversations on science and applied mathematics.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Analysis of Domain Decomposition Methods for the Simulation of Charge Transport in Semiconductor Structures with Heterojunctions

1. INTRODUCTION

In this study domain decomposition techniques are developed and analyzed for the simulation of a multiscale model for charge transport in semiconductor structures with heterojunctions. This work is motivated by the collaboration of computational mathematicians, physicists, and material scientists who are interested in such structures for the purpose of building more efficient solar cells.

A heterojunction is an interface between distinct semiconductor materials. This interface has a positive width on the angstrom scale. However, the scale of the bulk semiconductor regions is that of microns. Thus we use the model presented in [18] where transport across the heterojunction region is approximated at the device scale by a set of unusual transmission conditions, called the thermionic emission model (TEM). Data for the model is computed by an angstrom scale density functional theory (DFT) calculation. Much of the detail on modeling issues is presented in [12].

Charge transport at the device scale is modeled by the drift diffusion equations. The drift diffusion model is a coupled non-linear system of partial differential equations consisting of a Poisson equation for electrostatic potential and two advection-diffusion equations for conductive electron and hole densities. In a single semiconductor material this model is well-established, however, numerical simulation is riddled by difficulties stemming from its non-linear coupled nature as well as boundary and interior layer behavior.

The heterojunction model is well suited for treatment by domain decomposition

methods. In 1D the problem is amenable to a monolithic approach where the TEM equations are hard coded into the algebraic system resulting from a discretization of the drift diffusion equations in the bulk semiconductor regions. However, this approach may be intractable in higher dimensions where complicated interface geometries may arrise. Using the DDM presented in this work "black box" drift diffusion solvers may be used to simulate semiconductor structures with heterojunction, as the TEM conditions are enforced at the level of the domain decomposition driver. The domain decomposition methods analyzed here were first presented in [7, 12].

The thesis is composed of 6 chapters. In Chapter 2 the model is presented. In Chapter 3 established computational techniques for simulating the model are reviewed. In Chapter 4 novel domain decomposition techniques are presented for the simulation of semiconductor structures with heterojunction. Convergence analysis of these methods is also performed. In Chapter 5 numerical results are presented. Finally in Chapter 6 conclusions and future work are discussed.

2. MATHEMATICAL MODEL FOR SEMICONDUCTOR STRUCTURES WITH HETEROJUNCTION

In this section we describe the model used to simulate charge transport in semiconductor structures with heterojunctions [12].

Let $\Omega \subset \mathbb{R}^N$, $N \in \{1, 2, 3\}$, be an open connected set with a Lipschitz boundary $\partial\Omega$. Let $\Omega_i \subset \Omega$, $i = 1, 2$ be two non-overlapping subsets of Ω s.t. $\overline{\Omega}_1 \cup \overline{\Omega}_2 = \overline{\Omega}$, $\Omega_1 \cap \Omega_1 = \emptyset$, and denote $\Gamma := \overline{\Omega}_1 \cap \overline{\Omega}_2$. We assume Γ is a N-1 dimensional manifold and $\Gamma \cap \partial\Omega = \emptyset$. We adopt the following usual notation: $w_i = w|_{\Omega_i}$, $w_i^\Gamma = \lim_{x \to \Gamma} w_i$, and $[w]_\Gamma = w_2^\Gamma - w_1^\Gamma$ denotes the jump of w. We denote by ν^i the outward normal of Ω_i on Γ, and define $\nu := \nu^1$.

2.1. Drift Diffusion

In the bulk semiconductor domains Ω_i, $i = 1, 2$, the charge transport is described by the drift diffusion system. This system consists of a Poisson equation solved for electrostatic potential ψ and two continuity equations solved for electron and hole densities n and p.

The stationary drift diffusion model is

$$-\nabla \cdot (\epsilon_i \nabla \psi_i) = p - n + N_T, \tag{2.1}$$

$$\nabla \cdot J_n = R(n, p), \tag{2.2}$$

$$\nabla \cdot J_p = -R(n, p). \tag{2.3}$$

Here $N_T(x) = N_D(x) - N_A(x)$ is the given net doping profile including the donor N_D and acceptor doping N_A. $R(n, p)$ is a model for electron and hole generation and recombination, typically a rational function of carrier densities. ϵ_i denotes electrical permittivity.

J_n and J_p are the electron and hole currents defined by

$$J_n = D_n(-n\nabla\psi + \nabla n), \tag{2.4}$$

$$J_p = -D_p(p\nabla\psi + \nabla p). \tag{2.5}$$

Here D_n and D_p are the electron and hole diffusivities, respectively.

For analysis and computation it is convenient to introduce an alternative set of variables. The Slotboom variables u, v are related to n and p by

$$n = \delta_n^2 e^{\psi}\, u, \tag{2.6}$$

$$p = \delta_p^2 e^{-\psi}\, v. \tag{2.7}$$

The scaling parameters δ_n^2 and δ_p^2 depend on the material and doping profile. In Slotboom variables the continuity equations (2.2), (2.3) are self-adjoint, which is advantageous both for analysis as well as for numerical simulation. We pay for this improvement in the Poisson equation (2.1) which is then semi-linear.

Henceforth we will consider the system in Slotboom variables

$$-\nabla\cdot(\epsilon_i\nabla\psi_i) = \delta_p^2 e^{-\psi_i}\, v - \delta_n^2 e^{\psi_i}\, u + N_T := q(\psi_i, p_i, n_i), \tag{2.8}$$

$$-\nabla\cdot J_n = R(\psi_i, u_i, v_i), \tag{2.9}$$

$$-\nabla\cdot J_p = -R(\psi_i, u_i, v_i), \tag{2.10}$$

$$J_n = D_n\delta_n^2 e^{\psi_i}\nabla u_i, \tag{2.11}$$

$$J_p = D_p\delta_p^2 e^{-\psi_i}\nabla v_i. \tag{2.12}$$

2.2. Heterojunction

We are interested in simulating charge transport in a semiconductor structure consisting of distinct semiconductor materials connected by a heterojunction interface. A

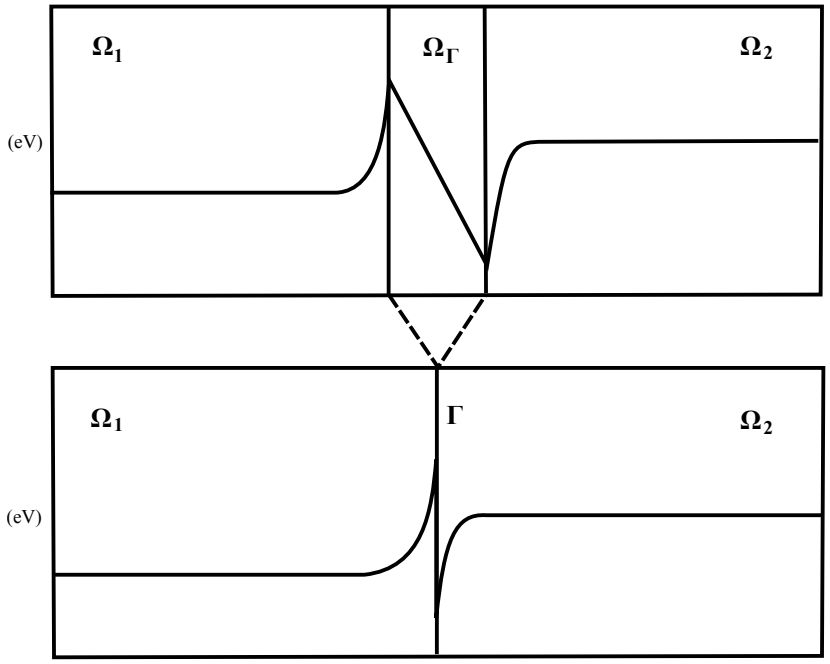

FIGURE 2.1: Top: schematic plot of potential across 1D interface region. Bottom: interface region shrunk to a point resulting in discontinuous potential.

true heterojunction interface is a region of positive width on the angstrom scale. However, a typical device scale for many applications, e.g. solar cells, is that of microns. Thus we cannot simulate transport in the heterojunction region with the drift diffusion system. Instead we adopt the Thermionic Emission Model described in [18] where the interface is treated as an idealized abrupt interface. As Figure 2.1 illustrates, this results in discontinuities in the primary variables.

The Thermionic Emission Model thus consists of transmission conditions for the primary variables across the idealized abrupt interface. These are a jump discontinuity $\psi_{\triangle}$ in the potential and unusual Robin-like conditions relating electron and hole densities across the interface.

$$[\psi]_\Gamma = \psi_\triangle, \tag{2.13}$$

$$\left[\epsilon\frac{\partial\psi}{\partial\nu}\right]_\Gamma = 0, \tag{2.14}$$

$$J_n^\Gamma = a_n^2 n_2^\Gamma - a_1^n n_1^\Gamma, \tag{2.15}$$

$$[J_n]_\Gamma = 0, \tag{2.16}$$

$$J_p^\Gamma = a_1^p p_1^\Gamma - a_2^p p_2^\Gamma, \tag{2.17}$$

$$[J_p]_\Gamma = 0. \tag{2.18}$$

Here the coefficients a_i^n and a_i^p are, up to scaling, mean electron thermal velocities that depend on the tempterature T and the sign of $\psi_\triangle$. The jump $\psi_\triangle$ must be calculated at the angstrom scale in the heterojunction region. Typically this is accomplished via Density Functional Theory. More details on the coefficients can be found in [12].

Note that the homogeneous jumps in the fluxes (2.14), (2.16), (2.18) are an assumption based on observation of the shape of the flux across a heterojunction region. Due to the loss of scale we have no a-priori reason to expect this homogeneity, and thus future models may need to account for a non-homogeneous jump in fluxes.

Finally, the Thermionic Emission Model in Slotboom variables is

$$[\psi]_\Gamma = \psi_\triangle, \tag{2.19}$$

$$\left[\epsilon\frac{\partial\psi}{\partial\nu}\right]_\Gamma = 0, \tag{2.20}$$

$$J_n^\Gamma = a_2^n \left(e^\psi u\right)_2^\Gamma - a_1^n \left(e^\psi u\right)_1^\Gamma, \tag{2.21}$$

$$[J_n]_\Gamma = 0, \tag{2.22}$$

$$J_p^\Gamma = a_1^p \left(e^{-\psi} v\right)_1^\Gamma - a_2^p \left(e^{-\psi} v\right)_2^\Gamma, \tag{2.23}$$

$$[J_p]_\Gamma = 0. \tag{2.24}$$

For each equation in the drift diffusion system, the Thermionic Emission Model provides a pair of transmission conditions. For the potential equation (2.8) we have (2.19) and

(2.20), for the equation for electron transport (2.9) we have (2.21) and (2.22), and for the equation for hole transport (2.10) we have (2.23) and (2.24).

2.3. External Boundary Conditions

The system (2.8)-(2.12), (2.19)-(2.24) is completed by appropriate external boundary conditions.

In the simulations in this work we use Dirichlet boundary conditions for the potential equation (2.8),

$$\psi|_{\partial\Omega} = \psi_D. \tag{2.25}$$

and Robin boundary conditions for the equations for carrier densities, (2.9), (2.10).

To illustrate the derivation of the value ψ_D we restrict to 1 dimension, where $\Omega = (a, b)$, and $\partial\Omega = \{a, b\}$. Then in this case we have

$$\psi_D(a) = \psi_a, \tag{2.26}$$

$$\psi_D(b) = \psi_b. \tag{2.27}$$

To find ψ_a, ψ_b we solve an algebraic problem for the neutral-charge equilibrium values ψ_a^{TE}, ψ_b^{TE} resulting from setting the quasi-Fermi potentials ψ_n and ψ_p to zero everywhere in Ω and dropping the derivatives from (2.8). The quasi-Fermi potentials are an alternative set of variables related to the carrier densities by Maxwell-Boltzmann statistics

$$n = N_C e^{\psi+\chi-\psi_n}, \tag{2.28}$$

$$p = N_V e^{-\psi-\chi+\psi_p-E_g} \tag{2.29}$$

where N_C and N_V are the conductive and valence band density of states, respectively, χ is the electron affinity, and E_g is the bandgap, i.e. the energy required to excite an electron from the highest energy valence band to the lowest energy conductive band.

The resulting system to solve for ψ_a^{TE} reads

$$N_V e^{-\psi_a^{TE}-\chi-E_g} - N_C e^{\psi_a^{TE}+\chi} + N_T|_{x=a} = 0 \tag{2.30}$$

with a similar equation for ψ_b^{TE}. ψ_a and ψ_b are then set by

$$\psi_a = \psi_a^{TE} + V_a, \tag{2.31}$$

$$\psi_b = \psi_b^{TE} + V_b \tag{2.32}$$

where V_a and V_b are physically controlled external voltages.

For (2.9) and (2.10) we assume recombination velocity boundary conditions, which are Robin type. These are specified using individual carrier currents via contact-specific effective recombination velocities. In 1d we denote the effective recombination velocities by $v_{n,a}$, $v_{n,b}$, $v_{p,a}$, $v_{p,b}$. Then the boundary conditions read

$$J_n \cdot \nu|_{x=a} = -v_{n,a}(n - n_0)|_{x=a} \tag{2.33}$$

$$J_p \cdot \nu|_{x=a} = v_{p,a}(p - p_0)|_{x=a} \tag{2.34}$$

$$J_n \cdot \nu|_{x=b} = -v_{n,b}(n - n_0)|_{x=b} \tag{2.35}$$

$$J_p \cdot \nu|_{x=b} = v_{p,b}(p - p_0)|_{x=b} \tag{2.36}$$

where n_0, p_0 are the carrier densities corresponding to the equilibrium values $\psi_n = \psi_p = 0$.

2.4. Well-posedness

Here we review the well-posedness analysis for the drift diffusion system following [26].

2.4.1 Mathematical Background

Before proceeding we recall definitions of Lebesgue $L^p(U)$ spaces and Sobolev spaces $W^{1,p}(U)$ [3]. For $1 \leq p < \infty$, the space $L^p(U)$ is the set of functions v defined on U s.t.

$$\left(\int_U |v|^p \right)^{\frac{1}{p}} < \infty, \tag{2.37}$$

where integration is with respect to the Lebesgue integral. This space is equipped with the norm

$$\|v\|_{L^p} = \left(\int_U |v|^p \right)^{\frac{1}{p}}. \tag{2.38}$$

In the case that $p = 2$, $L^2(U)$ is a Hilbert space when equipped with the inner product

$$(u, v)_{L^2} = \int_U v\, u. \tag{2.39}$$

The space $L^\infty(U)$ is defined as those functions v defined on U s.t. v is bounded almost everywhere on U, with norm given by the essential supremum,

$$\|v\|_{L^\infty} = \inf\{K : |v| \leq K \text{ a.e. on } U\}. \tag{2.40}$$

For $1 \leq p < \infty$ the Sobolev space $W^{1,p}(U)$ is defined as

$$W^{1,p}(U) = \left\{ v \in L^p(U) \ : \ \begin{array}{c} \exists g_1, g_2, \ldots, g_N \in L^p(U) \, s.t. \\ \int_U v \frac{\partial \phi}{\partial x_i} = -\int_U g_i \phi \ \forall \phi \in C_c^\infty(U), \ \forall i = 1, 2, \ldots, N \end{array} \right\}, \tag{2.41}$$

where $C_c^\infty(U)$ denotes the set of infinitely differential functions with compact support. This space is equiped with the norm

$$\|u\|_{W^{1,p}} = \left(\|u\|_{L^p}^p + \sum_{i=1}^N \left\| \frac{\partial u}{\partial x_i} \right\|_{L^p}^p \right)^{\frac{1}{p}}. \tag{2.42}$$

In the case that $p = 2$, we denote $W^{1,p}$ by H^1. H^1 is a Hilbert space with the inner product

$$(u, v)_{H^1} = (u, v)_{L^2} + \sum_{i=1}^N \left(\frac{\partial u}{\partial x_i}, \frac{\partial v}{\partial x_i} \right)_{L^2}. \tag{2.43}$$

The associated H^1 norm, is the same as the $W^{1,2}$ norm.

2.4.2 Single Material Domain

Following [[26], Sec 3.2] we consider the drift diffusion system in one material domain U (bounded in R^N, $N \in \{1,2,3\}$ of class $C^{0,1}$) with no heterojuntion in Slotboom variables with mixed Dirichlet and homogeneous Neumann boundary conditions.

$$-\nabla \cdot (\epsilon \nabla \psi) = q(\psi, u, v), \quad x \in U, \tag{2.44}$$

$$-\nabla \cdot (D_n \delta_n^2 e^{\psi} \nabla u) = R(\psi, u, v), \quad x \in U, \tag{2.45}$$

$$-\nabla \cdot (D_p \delta_p^2 e^{-\psi} \nabla v) = -R(\psi, u, v), \quad x \in U, \tag{2.46}$$

$$\psi|_{\partial U_D} = \psi_D,\ u|_{\partial U_D} = u_D,\ v|_{\partial U_D} = v_D, \tag{2.47}$$

$$\frac{\partial \psi}{\partial \nu}|_{\partial U_N} = \frac{\partial u}{\partial \nu}|_{\partial U_N} = \frac{\partial v}{\partial \nu}|_{\partial U_N} = 0. \tag{2.48}$$

We will be interested in the weak formulation of the system (2.44)-(2.48). We write $\psi = \psi_0 + \psi_D$, $u = u_0 + u_D$ and $v = v_0 + v_D$ and wish to find $(\psi, u, v) \in (H_0^1)^3$ such that

$$\int_U \epsilon \nabla \psi_0 \cdot \nabla \phi = \int_U q(\psi_0 + \psi)\phi - \int_U \epsilon \nabla \psi_D \cdot \nabla \phi, \tag{2.49}$$

$$\int_U D_n \delta_n^2 e^{\psi} \nabla u_0 \cdot \nabla \phi = \int_U R(\psi, u, v)\phi - \int_U D_n \delta_n^2 e^{\psi} \nabla u_D \cdot \nabla \phi, \tag{2.50}$$

$$\int_U D_p \delta_p^2 e^{-\psi} \nabla v_0 \cdot \nabla \phi = \int_U -R(\psi, u, v)\phi - \int_U D_p \delta_p^2 e^{-\psi} \nabla v_D \cdot \nabla \phi, \tag{2.51}$$

for all $\phi \in H_0^1$.

We make the following assumptions:

($A1$) The $N-1$ dimensional Lebesgue measure of ∂U_D is positive.

($A2$) The Dirichlet boundary data satisfies

$$(\psi_D, u_D, v_D) \in (H^1(U))^3, \quad (\psi_D, u_D, v_D)|_{\partial U_D} \in (L^{\infty}(\partial U_D))^3$$

and there is a $K \geq 0$ s.t.

$$e^{-K} \leq \inf_{\partial U_D} u_D, \inf_{\partial U_D} v_D; \quad \sup_{\partial U_D} u_D, \sup_{\partial U_D} v_D \leq e^K.$$

$(A3)$ The recombination-generation rate R satisfies $R = F(x, \psi, u, v)(uv - 1)$, where $F \in C^1$ for all $x \in U$; $F \in L^\infty$, $\nabla_{(\psi,u,v)} F \in L^\infty$, and $F \geq 0$.

$(A4)$ The mobilities D_n, D_p satisfy:

(i) $D_n = D_n(x, \nabla\psi)$, $D_p = D_p(x, \nabla\psi)$, $D_n, D_p : U \times \mathbb{R}^N \to \mathbb{R}$.

(ii) $0 < D_{n,0} \leq D_n \leq D_{n,1}$ for some $D_{n,0}, D_{n,1} \in \mathbb{R}$.

(iii) $0 < D_{p,0} \leq D_p \leq D_{p,1}$ for some $D_{p,0}, D_{p,1} \in \mathbb{R}$.

(iv) $D_n(x, y_1) - D_n(x, y_2)| + |D_p(x, y_1) - D_p(x, y_2)| \leq L|y_1 - y_2|$ for some $L > 0$.

$(A5)$ The doping profile N_T satisfies $N_T \in L^\infty(U)$.

Theorem 2.4.2.1 ([26], p. 35)**.** *Let assumptions* $(A1) - (A5)$ *hold. Then the problem* (2.44)-(2.48) *has a weak solution* $(\psi, u, v) \in (H^1(U) \cap L^\infty(U))^3$*, which satisfies the* L^∞ *estimates,*

$$e^{-K} \leq u \leq e^K \quad a.e.\ in\ U,$$
$$e^{-K} \leq v \leq e^K \quad a.e.\ in\ U.$$

Proof of this theorem has two components. The system is decoupled by an iterative procedure, resulting in a fixed point analysis for the so-called Gummel Map while existence of solutions is demonstrated for each of the component equations independently.

For analysis of the component equations, the reader is refered to [[26], Sec. 3.2]. We note that under physically admittable data it is well known that solutions to the system are not unique. Under more severe restrictions on data solutions can be shown to have H^2 regularity [[26] p. Sec. 3.3].

The decoupling procedure is important for both analysis and computation, and is used in the simulations in this paper. We describe this procedure, the Gummel Map, in the next section.

2.4.3 Gummel Decoupling

One of the difficulties with the drift diffusion system is its coupled nature. The right hand side of each equation depends on the primary variable of the other equations. Additionally, in Slotboom variables the equations for carrier densities have dependence in the differential operator on the electrostatic potential. A classical method of decoupling the equations for analysis as well as so that each may be solved independently is called the Gummel Map [8, 20, 26].

Let $\mathcal{H} := H^1(U) \cap L^\infty(U)$ with norm inherited from H^1. The Gummel Map, $G : \mathcal{H}^2 \to \mathcal{H}^2$, is applied iteratively until a fixed point (u^*, v^*) is reached. The map proceeds by: given $(u^0, v^0) \in \mathcal{H}^2$, for each $k \geq 0$,

1. solve (2.44) with data (u^k, v^k),

$$-\nabla \cdot (\epsilon \nabla \psi^{k+1}) = q(\psi^{k+1}, u^k, v^k) \quad \text{on } U, \tag{2.52}$$

$$\psi^{k+1} = \psi_D \quad \text{on } \partial U_D, \tag{2.53}$$

$$\frac{\partial \psi^{k+1}}{\partial \nu} = 0 \quad \text{on } \partial U_N \tag{2.54}$$

   for $\psi^{k+1} \in \mathcal{H}$.

2. Then solve (2.45) with data (ψ^{k+1}, u^k, v^k) iteration lagging the right hand side semilinearity,

$$-\nabla \cdot (D_n e^{\psi^{k+1}} \nabla u^{k+1}) = R(\psi^{k+1}, u^k, v^k) \quad \text{on } U, \tag{2.55}$$

$$u^{k+1} = u_D \quad \text{on } \partial U_D, \tag{2.56}$$

$$\frac{\partial u^{k+1}}{\partial \nu} = 0 \quad \text{on } \partial U_N \tag{2.57}$$

for $u^{k+1} \in \mathcal{H}$ and solve (2.46) similarly iteration lagging v^k on the right hand side,

$$-\nabla \cdot (D_p e^{-\psi^{k+1}} \nabla v^{k+1}) = R(\psi^{k+1}, u^k, v^k) \quad \text{on } U, \tag{2.58}$$

$$v^{k+1} = v_D \quad \text{on } \partial U_D, \tag{2.59}$$

$$\frac{\partial v^{k+1}}{\partial \nu} = 0 \quad \text{on } \partial U_N \tag{2.60}$$

for $v^{k+1} \in \mathcal{H}$.

A fixed point corresponds to a weak solution $(\psi^*, u^*, v^*) \in \mathcal{H}^3$ of the drift diffusion system.

The Gummel Map can be shown to have a fixed point by the application of the Schauder Fixed Point Theorem.

Theorem 2.4.3.1 (Schauder Fixed Point Theorem)**.** *Let X be a normed vector space, and $K \subset X$ a non-empty, compact, and convex set. Then given any continuous mapping $f : K \to K$ there exists $x \in K$ such that $f(x) = x$.*

In our case $X = \mathcal{H}^2$ and $K \subset X$ is a subset satisfying bounds on u and v guaranteed by the analysis of the component equations of the drift diffusion system. Thus a fixed point is guaranteed by showing that G is continuous and maps a non-empty, convex, compact subset K into itself where K is determined by bounds on solutions to the component equations.

3. NUMERICAL TECHNIQUES

In this section necessary tools for the simulation of the model presented in Section 2 are reviewed.

As Figure 3.1 illustrates, the continuum scale solver used in this work consists of 3 nested loops to handle the coupled nature of the model, the nonlinearity, and the TEM equations. The Gummel Map described in the previous section is the outer most iterative loop and decouples the model. The first interior loop is the domain decomposition algorithm for each decoupled equation. This handles the TEM model and is the subject of the next chapter. Finally, each equation is nonlinear and must be solved iteratively by Newton's method. This chapter describes Newton's method as well as the finite difference discretization for the continuum scale model.

A brief overview of the computational model for the microscopic heterojunction parameter calculation, Density Functional Theory, is given in the Appendix.

3.1. Finite Difference Discretization

In the numerical experiements in Chapter 5 the equations are discretized by centered finite differences. Here a centered finite difference discretization is reviewed for the linear Poisson equation.

$$-u'' = f, \;\; x \in (a, b), \tag{3.1}$$

$$u(a) = \alpha, \quad u(b) = \beta. \tag{3.2}$$

We seek to approximate u on (a, b) by values U_j at grid points x_j, $j = 1, ..., M$, where $a = x_0$ and $b = x_{M+1}$. We will assume a uniform grid, so that $x_{j+1} - x_j = h$, $j = 0, ..., M$.

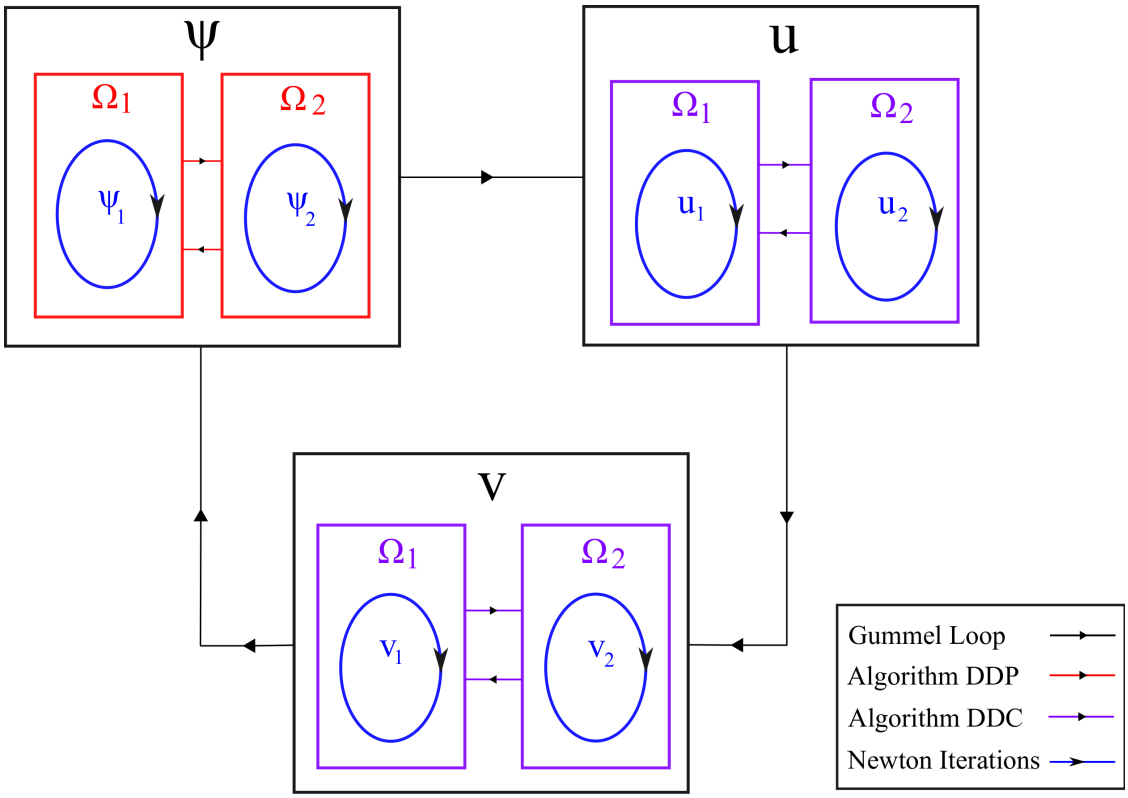

FIGURE 3.1: Illustration of continuum solver structure.

Replace u'' in (3.1) by a centered finite difference approximation,

$$D^2U_j = \frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) \tag{3.3}$$

to obtain a set of algebraic equations

$$-\frac{1}{h^2}(U_{j-1} - 2U_j + U_{j+1}) = f(x_j) \; for \; j = 1, 2, ..., M. \tag{3.4}$$

Incorporating the boundary conditions we obtain a linear system

$$AU = F, \tag{3.5}$$

$$\frac{1}{h^2}\begin{bmatrix} 1 & & & & & & & \\ -1 & 2 & -1 & & & & & \\ & -1 & 2 & -1 & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & -1 & 2 & -1 & & \\ & & & & -1 & 2 & -1 & \\ & & & & & & 1 \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ \vdots \\ U_{M-1} \\ U_M \\ U_{M+1} \end{bmatrix} = \begin{bmatrix} \alpha \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{M-1}) \\ f(x_M) \\ \beta \end{bmatrix} \tag{3.6}$$

The values U_j are an approximation of the true solution u at grid points x_j depending on the width of the mesh h. Thus we expect error between U_j and $u(x_j)$ to shrink as $h \to 0$. In Chapter 5 convergence will be tested in the domain decomposition setting to ensure that the introduction of the domain decomposition algorithm has no effect on the order of convergence of the discretization. For that reason we review the expected order of convergence for the centered finite difference approximation.

We begin by recalling Taylor expansions up to fourth order,

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u^{(3)}(x) + \frac{h^4}{24}u^{(4)} + O(h^5), \tag{3.7}$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u^{(3)}(x) + \frac{h^4}{24}u^{(4)} + O(h^5). \tag{3.8}$$

Then plugging (3.7)-(3.8) into (3.3) we have

$$\begin{aligned} D^2u(x) &= \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} \\ &= \frac{h^2u''(x) + \frac{1}{12}h^4u^{(4)} + O(h^5)}{h^2} \\ &= u''(x) + O(h^2) \end{aligned}$$

where $O(h^2)$ denotes the term whose size is controlled by h^2. Note that this calculation required $u(x) \in C^4$.

This shows that the pointwise error $|D^2U_j - u''(x_j)|$ is controlled by h^2. However,

this does not tell us about how the global error

$$\|E\|_\infty = \|U - u\|_\infty = \max_{1 \le j \le M} |U_j - u(x_j)| \tag{3.9}$$

is controlled by the mesh size h.

To obtain a bound on $\|E\|_\infty$ we compute the local truncation error (LTE), and then use stability to show that the global error can be bounded in terms of the LTE.

Replacing U_j by $u(x_j)$ in (3.4) we define the local truncation error τ_j by

$$\tau_j = \frac{1}{h^2}(u(x_{j-1}) - 2u(x_j) + u(x_{j+1})) + f(x_j) \; for \; j = 1, 2, \ldots, M. \tag{3.10}$$

Then assuming u is sufficiently smooth, we can use the Taylor expansions to obtain

$$\tau_j = \left[u''(x_j) + \frac{1}{12}h^2 u''''(x_j) + O(h^4)\right] + f(x_j). \tag{3.11}$$

Then we note that from the original differential equation, $u''(x_j) + f(x_j) = 0$, thus we have

$$\tau_j = \frac{1}{12}h^2 u''''(x_j) + O(h^4). \tag{3.12}$$

Since u'''' is independent of h, we have $\tau_j = O(h^2)$ as $h \to 0$.

Define $\mathbf{u}$ to be the vector consisting of the true solution evaluated at grid points, so that $\mathbf{u}_j = u(x_j)$. Then let $E = U - \mathbf{u}$, and τ be the vector with components τ_j, the local truncation error at x_j. Next we subtract

$$AU = F + \tau \tag{3.13}$$

from

$$A\mathbf{u} = F \tag{3.14}$$

and consider the equation,

$$A^h E^h = -\tau^h, \tag{3.15}$$

where the superscript h denotes the dependence on h.

Solving (3.15) for E^h we have

$$E^h = -(A^h)^{-1}\tau^h. \tag{3.16}$$

Thus,

$$\|E^h\| \leq \|(A^h)^{-1}\|\|\tau^h\|. \tag{3.17}$$

Then, if there is some constant K s.t.

$$\|(A^h)^{-1}\| \leq K \text{ for sufficiently small } h, \tag{3.18}$$

we have

$$\|E^h\| \leq K\|\tau^h\| \tag{3.19}$$

in whichever norm we choose.

The existence of such a K is referred to as the stability of the method.

In [[23], p. 20] it is shown that A is stable in both the 2-norm and the ∞-norm. Thus we have

$$\|E^h\|_\infty = O(h^2). \tag{3.20}$$

3.2. Newton's Method

In Slotboom variables each component equation of the drift diffusion system is semi-linear. Thus we need a method for solving nonlinear partial differential equations. For this we use Newton's Method [22, 23]. In describing Newton's method we will consider a nonlinear equation

$$G : \mathbb{R}^k \to \mathbb{R}^k, \tag{3.21}$$

$$G(x) = 0, \tag{3.22}$$

e.g. the result of discretizing one of the equations in the drift diffusion system.

Newton's method solves this equation iteratively by starting from an initial guess x^0 and proceeding for each $k \geq 0$,

$$x^{k+1} = x^k - J^{-1}(x^k)G(x^k) \tag{3.23}$$

until a stopping criteria is met, where $J(x)$ is the Jacobian of $G(x)$.

In 1d we can illustrate how Newton's Method works by following the tangent line at $G(x^k)$ from the point $G(x^k)$ to the x-axis. The Newton iterate x^{k+1} is located where the tangent line crosses the x-axis. Figure 3.2 illustrates Newton's Method in one dimension.

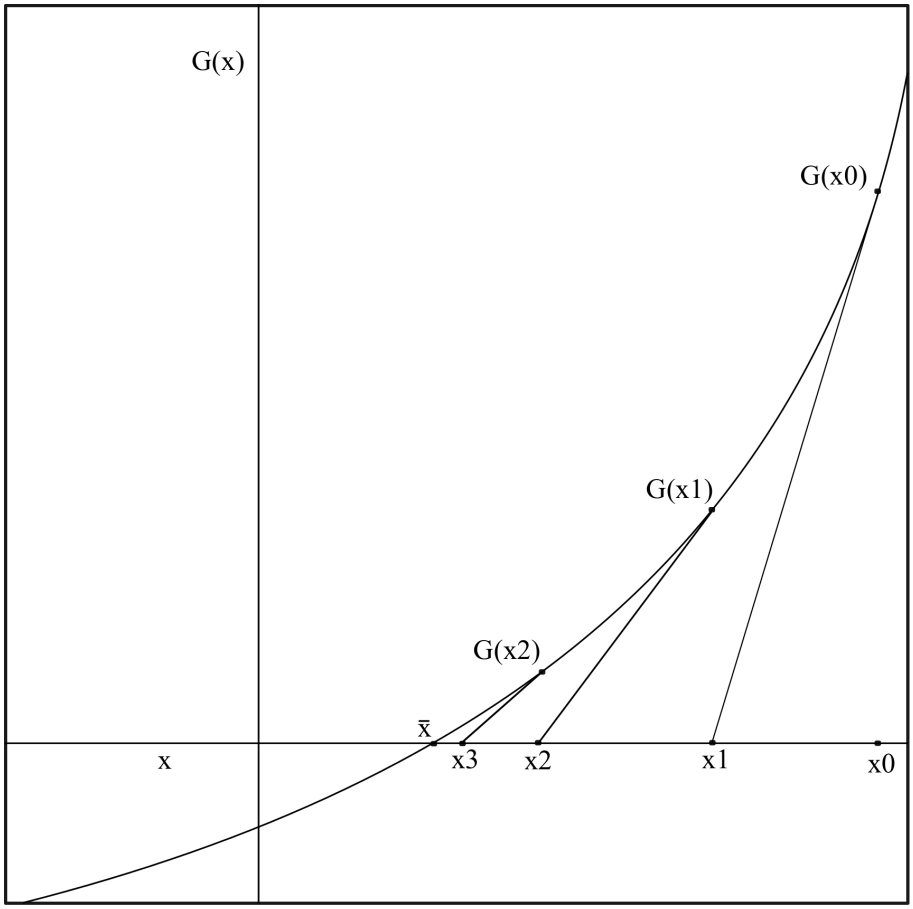

FIGURE 3.2: Illustration of Newton's Method in one dimension. $\overline{x}$ denotes the solution to $G(x) = 0$.

It is well known that Newton's method is not globally convergent, and thus success

depends significantly on the choice of the initial iterate. However, there are conditions under which Newton's method can be guaranteed to converge.

Theorem 3.2.0.2 ([22] p. 71)**.** *Assume the equation $G(x) = 0$ has a solution x^*, that the Jacobian J is Lipschitz continuous and is nonsingular. Then there are constants $K > 0$ and $\delta > 0$ s.t. if $x^k \in B_\delta(x^*)$ then the Newton iterate given by* (3.23) *satisifes*

$$\|x^{k+1} - x^*\| \leq K\|x^k - x^*\|^2. \tag{3.24}$$

The theorem says that there is a neighborhood about the solution in which Newton's method converges quadratically. We note that this neighborhood is, in practice, very difficult to determine.

There is no reason to expect that the Gummel Map described in the previous section will converge quadratically to a solution of the drift diffusion system. Thus the natural question to ask is, if Newton's method converges quadratically, why not solve the coupled drift diffusion system directly with Newton's method? Indeed, in [24, 37] the coupled model is solved directly. However, whereas the range of initial guesses for which Newton's method would converge for the fully coupled system may be frustratingly small and difficult to find, the Gummel Map is much more forgiving. Thus when employing the Gummel Map difficulties associated to initial guesses for Newton's method are restricted to the individual equations.

4. DOMAIN DECOMPOSITION

Domain decomposition methods provide a framework for solving a PDE independently on multiple subdomains. Originally these methods were developed in the interest of parallelization for PDE problems with smooth solutions that were too large to be solved on a single processor. More recently, domain decomposition has developed as a technique for solving problems with discontinuous coefficients or interfaces separating different materials or physical models [4, 5, 9, 25, 31, 32, 33, 43, 44].

First we review the basic concepts of domain decomposition for the linear Poisson equation on Ω following [34]. This setup is then extended to present two algorithms designed for the model for charge transport in semiconductor structures with heterojunctions. Convergence analysis is performed for these algorithms.

4.1. Mathematical Background

The linear Poisson equation has no jumps or discontinuities but this background will serve as a template for the more interesting problems related to semiconductor structures with heterojunctions.

4.1.1 Continuous Formulation

We begin with the linear Poisson equation posed on the whole domain Ω:

$$-\triangle\phi = f, \;\; x \in \Omega, \tag{4.1}$$

$$\phi = 0, \;\; x \in \partial\Omega. \tag{4.2}$$

We divide the domain Ω into two non-overlapping subdomains Ω_i, $i = 1, 2$, (Fig. 4.1 shows examples) and Γ will denote the interface between them.

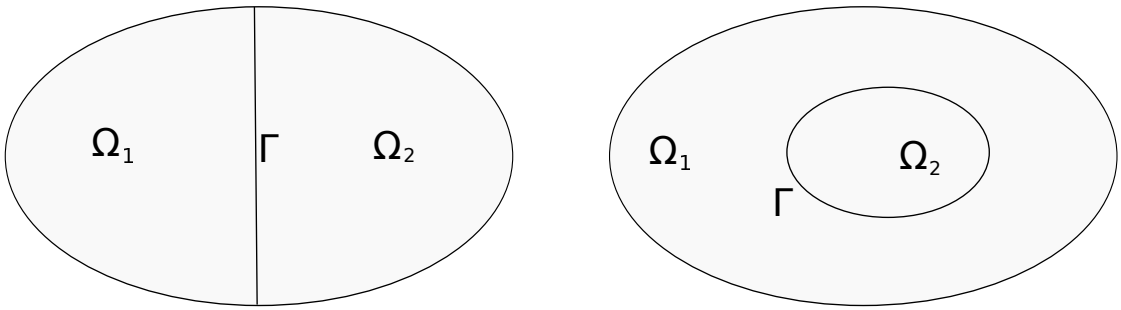

FIGURE 4.1: Examples of dividing Ω in to subdomains.

Next we rewrite (4.1)-(4.2) in a multidomain formulation:

$$-\triangle\phi_1 = f, \quad x \in \Omega_1, \tag{4.3}$$

$$\phi_1 = 0, \quad on \ \partial\Omega_1 \cap \partial\Omega, \tag{4.4}$$

$$\phi_1 = \phi_2 \quad on \ \Gamma, \tag{4.5}$$

$$\frac{\partial\phi_2}{\partial\nu} = \frac{\partial\phi_1}{\partial\nu} \quad on \ \Gamma, \tag{4.6}$$

$$\phi_2 = 0 \quad on \ \partial\Omega_2 \cap \partial\Omega, \tag{4.7}$$

$$-\triangle\phi_2 = f, \quad x \in \Omega_2. \tag{4.8}$$

Equations (4.5)-(4.6) are the transmission conditions for the problem. In a domain decomposition setting we would like to solve the PDE independently on the two subdomains Ω_1 and Ω_2. Clearly if we had a boundary condition for ϕ_i on Γ this would be a straightforward task. Instead, we have the transmission conditions (4.5)-(4.6). The difficulty in domain decomposition lies in resolving these transmission conditions while solving the PDE independently on the two subdomains.

For the class of methods known as iterative substructuring methods this is accomplished by iteratively solving a particular equation posed on the interface. The solution of this equation will provide a boundary condition for the subdomain problems at the interface such that the transmission conditions are satisfied.

Consider the problem

$$-\triangle w_i = f, \quad x \in \Omega_i, \tag{4.9}$$

$$w_i = 0, \quad on \ \partial\Omega_i \cap \partial\Omega, \tag{4.10}$$

$$w_i = \lambda, \quad on \ \partial\Gamma. \tag{4.11}$$

It is clear that for any λ the transmission condition (4.5) is satisfied by w_i. However, there is no reason to expect that (4.6) is satisfied for any particular λ. Clearly, though, for $\lambda = \phi_1|_\Gamma = \phi_2|_\Gamma$ where ϕ solves (4.1)-(4.2), (4.6) is satisfied. Next we describe an equation that will provide a method of finding this correct λ.

For a linear or semilinear problem we can write $w_i = J_i\lambda + P_i f$ where $J_i\lambda$ solves the problem

$$-\triangle v_i = 0, \quad x \in \Omega_i, \tag{4.12}$$

$$v_i = \lambda, \quad on \ \Gamma, \tag{4.13}$$

$$v_i = 0, \quad on \ \partial\Omega_i \cap \partial\Omega, \tag{4.14}$$

and $P_i f$ solves

$$-\triangle v_i = f, \quad x \in \Omega_i, \tag{4.15}$$

$$v_i = 0, \quad on \ \Gamma, \tag{4.16}$$

$$v_i = 0, \quad on \ \partial\Omega_i \cap \partial\Omega. \tag{4.17}$$

This is the standard splitting of the solution to (4.9)-(4.11) into parts responding to the boundary data λ and the forcing term f.

Now define the operators $S\lambda$ and χ by

$$S\lambda := \sum_{i=1}^{2} \frac{\partial J_i \lambda}{\partial \nu^i} \tag{4.18}$$

and

$$\chi := -\sum_{i=1}^{2} \frac{\partial P_i f}{\partial \nu^i} \tag{4.19}$$

and consider the following *Steklov-Poincaré interface equation*

$$S\lambda = \chi, \quad on\ \Gamma. \tag{4.20}$$

If λ satisfies the Steklov-Poincaré interface equation (4.20) then $w_i = \phi_i$ for $i = 1, 2$ [34].

In this paper we consider a class of methods known as iterative substructuring methods. These methods are designed to solve (4.20) iteratively, e.g. by a Richardson scheme. The solution λ provides data for the solution to (4.1)-(4.2) along Γ which guarantees both transmission conditions are satisfied when solving independent subdomain problems.

4.1.2 Algebraic Formulation

In this section we consider the linear Poisson equation (4.1)-(4.2) discretized by finite differences. In this setting domain decomposition methods iteratively solve the Schur-complement system, the discrete counterpart to the Steklov-Poincaré interface equation. Consider the algebraic system resulting from a discretization of (4.1)

$$\mathbf{A}\Phi = \mathbf{F}. \tag{4.21}$$

Denote by $\mathbf{A}_{i,j}$ the parts of $\mathbf{A}$ corresponding to unknowns in domains i and j, and by $\mathbf{A}_{i,\Gamma}$, $\mathbf{A}_{\Gamma,i}$, and $\mathbf{A}_{\Gamma,\Gamma}$ the parts of $\mathbf{A}$ corresponding to unknowns in along the interface in domain i or purely along the interface. Then if we order the unknowns so that $\Phi = (\Phi_1, \Phi_2, \Phi_\Gamma)^T$, where Φ_i denotes unknowns in Ω_i and Φ_Γ denotes interface unknowns, we have

$$\begin{bmatrix} \mathbf{A}_{1,1} & 0 & \mathbf{A}_{1,\Gamma} \\ 0 & \mathbf{A}_{2,2} & A_{2,\Gamma} \\ \mathbf{A}_{\Gamma,1} & \mathbf{A}_{\Gamma,2} & \mathbf{A}_{\Gamma,\Gamma} \end{bmatrix} \quad \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_\Gamma \end{bmatrix}. \tag{4.22}$$

By block elimination we obtain the Schur-complement system

$$\Sigma\Phi_\Gamma = \Theta \tag{4.23}$$

where

$$\Theta := \mathbf{f}_\Gamma - \mathbf{A}_{\Gamma,1}\mathbf{A}_{1,1}^{-1}\mathbf{f}_1 - \mathbf{A}_{\Gamma,2}\mathbf{A}_{2,2}^{-1}\mathbf{f}_2, \tag{4.24}$$

$$\Sigma := \mathbf{A}_{\Gamma,\Gamma} - \mathbf{A}_{\Gamma,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,\Gamma} - \mathbf{A}_{\Gamma,2}\mathbf{A}_{2,2}^{-1}\mathbf{A}_{2,\Gamma}. \tag{4.25}$$

4.1.3 Neumann-Neumann Algorithm

The algorithms developed in the next section are most closely related to the Neumann-Neumann algorithm.

This is a well-known iterative substructuring method that solves independent Dirichlet problems on subdomains while iteratively solving (4.20),(4.23).

Neumann-Neumann Algorithm

Given λ^0, for each $k \geq 0$,

1. For $i = 1, 2$ solve the subdomain problems

$$-\triangle \phi_1^{k+1} = f, \quad x \in \Omega_1, \tag{4.26}$$

$$\phi_1^{k+1} = \lambda, \quad on\ \Gamma, \tag{4.27}$$

$$\phi_1^{k+1} = 0, \quad on\ \partial\Omega_1 \cap \partial\Omega, \tag{4.28}$$

   for ϕ_1^{k+1} and ϕ_2^{k+1}.

2. Then solve in subdomains $i = 1, 2$,

$$-\triangle v_i^{k+1} = 0,\ x \in \Omega_i \tag{4.29}$$

$$v_i^{k+1} = 0 \text{ on } \partial\Omega \cap \partial\Omega_i \tag{4.30}$$

$$\frac{\partial v_i^{k+1}}{\partial \nu} = \left[\frac{\partial \phi}{\partial \nu}\right]_\Gamma \text{ on } \Gamma \tag{4.31}$$

3. Update λ by

$$\lambda^{k+1} = \lambda^k - \theta \left[v^{k+1}\right]_\Gamma. \tag{4.32}$$

4. Continue with (1) unless stopping criterium $\left\| \left[\frac{\partial \phi^{k+1}}{\partial \nu}\right]_\Gamma \right\|$ holds.

As it is shown in [34] the Neumann-Neumann algorithm can be analyzed as a Richardson scheme for the solution of (4.20), and there exists a $\theta_{max} > 0$ s.t. if $\theta \in (0, \theta_{max})$, the Neumann-Neumann algorithm converges to a solution of (4.1)-(4.2).

Also we note that in 1D it is easy to show that step two may be skipped, instead updating λ by

$$\lambda^{k+1} = \lambda^k + \theta \left[\frac{\partial \phi^{k+1}}{\partial \nu} \right]_\Gamma . \tag{4.33}$$

To see this, consider $\Omega = (-1, 1)$, $\Omega_1 = (-1, 0)$, $\Omega_2 = (0, 1)$. Then if v_i solves (4.29)-(4.31), v_i is a linear funtion defined by

$$v_1(x) = \left[\frac{\partial \phi}{\partial \nu} \right]_\Gamma (1 + x), \tag{4.34}$$

$$v_2(x) = \left[\frac{\partial \phi}{\partial \nu} \right]_\Gamma (x - 1). \tag{4.35}$$

So

$$\left[v^{k+1} \right]_\Gamma = v_2^{k+1}(0) - v_1^{k+1}(0) \tag{4.36}$$

$$= -2 \left[\frac{\partial \phi}{\partial \nu} \right]_\Gamma . \tag{4.37}$$

Hence we may avoid step 2 in the Neumann-Neumann algorithm by making this adjustment.

The algorithms in the next two sections designed for the model for semiconductor structures with heterojunctions are extensions of this algorithm. The algorithm for the potential equation allows for discontinuities in the condition (4.5) while the algorithm for the equations for carrier densities allow for a discontinuity whose value in domain 2 depends on the flux of the solution in domain 1.

4.1.4 Richardson Iteration

There are many types of iterative methods for the solution of linear systems of structure similar to (4.20) [22]. The domain decomposition methods in this work can be

analyzed as Richardson schemes. Here we overview this iterative scheme for an equation posed in $\mathbb{R}^k$.

Let $\mathbf{A} : \mathbb{R}^k \to \mathbb{R}^k$, and $b \in \mathbb{R}^k$, so that we consider the problem,

$$\mathbf{A}x = b. \tag{4.38}$$

Write (4.38) as

$$x = (I - \mathbf{A})\lambda + b, \tag{4.39}$$

then define the Richardson iteration by: given x^0 for each $k \geq 1$ iterate

$$x^{k+1} = (I - \mathbf{A})x^k + b. \tag{4.40}$$

A theorem to prove that x^k converges to the solution x of (4.38) is based on the more general scheme

$$x^{k+1} = \mathbf{M}x^k + c, \tag{4.41}$$

where $\mathbf{M}$ is called the iteration matrix.

Theorem 4.1.4.1 ([22] p. 6)**.** *If $\|\boldsymbol{M}\| < 1$ then the iteration* (4.41) *converges to $x = (I - \boldsymbol{M})^{-1}c$ for all initial iterates $x^{(0)}$.*

For (4.38) we have $\mathbf{M} = (I - \mathbf{A})$ and $c = b$ so that if $\|I - \mathbf{A}\| < 1$, the iteration (4.40) converges to the solution $x = (I - I + \mathbf{A})^{-1}b = \mathbf{A}^{-1}b$ of (4.38) for all initial iterates x^0.

Clearly the interface equation (4.20) is not posed in $\mathbb{R}^k$. Given that we would like to analyze the convergence of algorithms designed to solve equations similar to (4.20), this may seem problematic. However, a proof of this theorem does not rely on the finite dimension of $\mathbb{R}^k$. As we will see later in the context of the algorithms for the heterojunction model, the convergence of these methods relies on the contraction property $\|M\| < 1$ as well as the completeness of the vector space in question.

4.1.5 Complexity Comparison

It is well known that the Neumann-Neumann algorithm is mesh independent. That is, as the mesh is refined and the number of computational nodes increases, the number of domain decomposition iterations required to reach a given tolerance remains constant.

Here we analyze the relative computational complexity of solving (4.1)-(4.2) in a single domain versus solving the problem with a mesh indepndent iterative substructuring method like the Neumann-Neumann method.

Let N denote the number of computational nodes in the domain Ω, and for simplicity assume $\frac{N}{2} = N_0$ is the number of nodes in each subdomain.

Suppose as well that the solver for the linear system resulting from a discretization of the problems has $O(N^2)$ complexity.

Then if the domain decomposition algorithm requires K iterations, we have $2KO(N_0^2)$ work done in the domain decomposition setting.

Then using $N = 2N_0$, we note that we have $O((2N_0)^2) = 4O(N_0^2)$ in the single domain case.

Thus we see that the domain decomposition algorithm will require less work when $K \leq 2$.

For anything but the simplest problems, this is unattainable. However, this calculation ignores the possibility of computing the subdomain problems in parallel, which is a considerable advantage for computational time for the domain decomposition setting.

Additionally, in the case of the model for charge transport in semiconductor structures with heterojunctions including the thermionic emission model equations directly in a single domain monolithic solver results in a change of the sparsity pattern of the matrices in the linear systems to be solved, as well as the loss of symmetry. This affects the choice of linear solver for the problem and in many cases affects the efficiency of the solver. This further advocates for the use of domain decomposition, which resolves the

thermionic emission model at a level that does not effect the linear systems being solved.

4.2. DDM for Potential Equation

In the previous section we described the background for iterative substructuring methods including transmission conditions and the Steklov-Poincaré interface equation (4.20). We then introduced the Neumann-Neumann method, a method that can be analyzed as a Richardson scheme for (4.20).

In this section, we extend these techniques to the problem (2.8) with (2.19) and (2.20). The method we develop will be based on the Neumann-Neumann method but will allow for the non-homogeneous jump (2.19).

4.2.1 Continuous Formulation

Here we develop an algorithm to solve (2.8) with (2.19) and (2.20). Homogeneous Dirichlet boundary conditions are assumed at the external boundary for simplicity of exposition. Rather than solving this problem on the whole domain, we would like to solve independent problems in subdomains Ω_1, Ω_2. The iterative substructuring method described here is thus a method for finding a suitable λ defined on the interface Γ so that (2.8) with (2.19) and (2.20) is equivalent to the problem

$$-\nabla \cdot (\epsilon_1 \nabla \psi) = q_1,\ x \in \Omega_1, \quad \psi_1|_\Gamma = \lambda, \tag{4.42}$$

$$-\nabla \cdot (\epsilon_2 \nabla \psi) = q_2,\ x \in \Omega_2, \quad \psi_2|_\Gamma = \lambda + \psi_\triangle. \tag{4.43}$$

For any λ, the system (4.42)-(4.43) satisfies (2.19). However, there is no reason to expect that (2.20) is satisfied by a solution to (4.42)-(4.43). Thus we seek the value λ which guarantees (2.20) is also satisfied.

To develop the method for finding such a λ we recall the definitions of $J_i\lambda$ (4.12)-

(4.14) and $P_i f$ (4.15)-(4.17) and write

$$\psi_1 = J_1\lambda + P_1 q, \quad \psi_2 = J_2\lambda + J_2\psi_\triangle + P_2 q. \tag{4.44}$$

For ψ_1 this is the standard separation of the solution to (4.42) into components responding to the boundary data λ and the forcing term q_1. For ψ_2 we are additionally separating the response to the data $\lambda + \psi_\triangle$ into components responding to λ and $\psi_\triangle$ distinctly.

Then define K and Υ by

$$K\lambda := \epsilon_1 \frac{\partial J_1\lambda}{\partial \nu} - \epsilon_2 \frac{\partial J_2\lambda}{\partial \nu}, \tag{4.45}$$

$$\Upsilon := \epsilon_2 \frac{\partial P_2 q}{\partial \nu} - \epsilon_1 \frac{\partial P_1 q}{\partial \nu} + \epsilon_2 \frac{\partial J_2\psi_\triangle}{\partial \nu}, \tag{4.46}$$

and consider the equation

$$K\lambda = \Upsilon. \tag{4.47}$$

Lemma 4.2.1.1. *Suppose λ satisifes the equation $K\lambda = \Upsilon$. Then a solution of* (4.42)-(4.43) *also solves* (2.8) *with* (2.19) *and* (2.20).

Proof. The proof of this lemma is a straightforward calculation. Assume λ satisfies (4.47) and assume ψ satisfies (4.42)-(4.43) with this λ. Clearly (2.8) is satisfied for $i = 1, 2$. Further, note that for any λ the condition (2.19)

$$[\psi]_\Gamma = \psi_\triangle \tag{4.48}$$

is satisfied by construction. Thus it remains to check (2.20)

$$\left[\epsilon \frac{\partial \psi}{\partial \nu}\right]_\Gamma = 0. \tag{4.49}$$

To see this condition is satisfied we calculate, beginning with (4.47),

$$K\lambda = \Upsilon.$$

Recalling the definitions (4.45)-(4.46) we have

$$\epsilon_1^\Gamma \frac{\partial J_1\lambda}{\partial \nu} - \epsilon_2^\Gamma \frac{\partial J_2\lambda}{\partial \nu} = \epsilon_2^\Gamma \frac{\partial P_2 q}{\partial \nu} - \epsilon_1^\Gamma \frac{\partial P_1 q}{\partial \nu} + \epsilon_2 \frac{\partial J_2\psi_\triangle}{\partial \nu}.$$

Then we group terms by domain,

$$\epsilon_1^\Gamma \left(\frac{\partial J_1 \lambda}{\partial \nu} + \frac{\partial P_1 q}{\partial \nu} \right) \quad = \quad \epsilon_2^\Gamma \left(\frac{\partial J_2 \lambda}{\partial \nu} + \frac{\partial J_2 \psi_\triangle}{\partial \nu} + \frac{\partial P_2 q}{\partial \nu} \right),$$

thus

$$\epsilon_1^\Gamma \frac{\partial (J_1 \lambda + P_1 q)}{\partial \nu} \quad = \quad \epsilon_2^\Gamma \frac{\partial (J_2 \lambda + J_2 \psi_\triangle + P_2 q)}{\partial \nu}.$$

Recalling (4.44) we conclude

$$\epsilon_1^\Gamma \frac{\partial \psi_1}{\partial \nu} \quad = \quad \epsilon_2^\Gamma \frac{\partial \psi_2}{\partial \nu}$$

□

4.2.2 Algebraic Formulation

To construct the algebraic counterpart to (4.47) we consider the centered finite difference discretization of (4.42)-(4.43). After reordering unknowns we have

$$\begin{bmatrix} \mathbf{A}_{1,1} & 0 & \mathbf{A}_{1,\Gamma} & 0 \\ 0 & \mathbf{A}_{2,2} & 0 & \mathbf{A}_{2,\Gamma} \\ \mathbf{A}_{\Gamma,1} & \mathbf{A}_{\Gamma,2} & \mathbf{A}^1_{\Gamma,\Gamma} & \mathbf{A}^2_{\Gamma,\Gamma} \\ 0 & 0 & -\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi^1_\Gamma \\ \Psi^2_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \{\mathbf{Q}_\Gamma\}_\Gamma \\ \Psi_\triangle \end{bmatrix}. \tag{4.50}$$

The first two lines in this system are simply the discretization of the potential equation in the bulk domains 1 and 2. In the third line, we define

$$\{\mathbf{Q}_\Gamma\}_\Gamma := \frac{\mathbf{Q}^1_\Gamma + \mathbf{Q}^2_\Gamma}{2}. \tag{4.51}$$

The last line is the algebraic counterpart to the TEM condition (2.19). Rewriting this system we have four equations

$$\mathbf{A}_{1,1}\Psi_1 + \mathbf{A}_{1,\Gamma}\Psi^1_\Gamma = \mathbf{Q}_1, \tag{4.52}$$

$$\mathbf{A}_{2,2}\Psi_2 + \mathbf{A}_{2,\Gamma}\Psi^2_\Gamma = \mathbf{Q}_2, \tag{4.53}$$

$$\mathbf{A}_{\Gamma,1}\Psi_1 + \mathbf{A}_{\Gamma,2}\Psi_2 + \mathbf{A}^1_{\Gamma,\Gamma}\Psi^1_\Gamma + \mathbf{A}^2_{\Gamma,\Gamma}\Psi^2_\Gamma = \{\mathbf{Q}_\Gamma\}_\Gamma, \tag{4.54}$$

$$\Psi^2_\Gamma - \Psi^1_\Gamma = \Psi_\triangle. \tag{4.55}$$

Substituting (4.55) in (4.53)-(4.54) we have

$$\mathbf{A}_{2,2}\Psi^2 + \mathbf{A}_{2,\Gamma}(\Psi^1_\Gamma + \Psi_\triangle) = \mathbf{Q}_2, \tag{4.56}$$

$$\mathbf{A}_{\Gamma,1}\Psi_1 + \mathbf{A}_{\Gamma,2}\Psi_2 + \mathbf{A}^1_{\Gamma,\Gamma}\Psi^1_\Gamma + \mathbf{A}^2_{\Gamma,\Gamma}(\Psi^1_\Gamma + \Psi_\triangle) = \{\mathbf{Q}_\Gamma\}_\Gamma\,. \tag{4.57}$$

Now using (4.52) with (4.56), (4.57) we solve for Ψ_1, Ψ_2,

$$\Psi_1 = \mathbf{A}^{-1}_{1,1}\mathbf{Q}_1 - \mathbf{A}_{1,1}\mathbf{A}_{1,\Gamma}\Psi^1_\Gamma, \tag{4.58}$$

$$\Psi_2 = \mathbf{A}^{-1}_{2,2}\mathbf{Q}_2 - \mathbf{A}^{-1}_{2,2}\mathbf{A}_{2,\Gamma}\Psi^1_\Gamma - \mathbf{A}^{-1}_{2,2}\Psi_\triangle. \tag{4.59}$$

Substituting (4.58) and (4.59) into (4.57) results in the algebraic counterpart to (4.47)

$$K_h\Psi^1_\Gamma = \Upsilon_h \tag{4.60}$$

where

$$K_h := \mathbf{A}^1_{\Gamma,\Gamma} + \mathbf{A}^2_{\Gamma,\Gamma} - \mathbf{A}_{\Gamma,1}\mathbf{A}^{-1}_{1,1}\mathbf{A}_{1,\Gamma} - \mathbf{A}_{\Gamma,2}\mathbf{A}^{-1}_{2,2}\mathbf{A}_{2,\Gamma} \tag{4.61}$$

and

$$\Upsilon_h := (\mathbf{A}_{\Gamma,1}\mathbf{A}^{-1}_{1,1}\mathbf{Q}_1 + \mathbf{A}_{\Gamma,2}\mathbf{A}_{2,2}\mathbf{Q}_2 + \{\mathbf{Q}_\Gamma\}_\Gamma) + (\mathbf{A}^2_{\Gamma,\Gamma} - \mathbf{A}_{\Gamma,2}\mathbf{A}^{-1}_{2,2})\Psi_\triangle. \tag{4.62}$$

4.2.3 Algorithm

To find an appropriate λ and solve (4.42)-(4.43) we present a method that can be analyzed as a Richardson scheme for (4.47).

Algorithm DDP to solve (2.8) ***with*** (2.19)

Given λ^0, for each $k \geq 0$,

1. Solve (4.42) for ψ_1^{k+1} and (4.43) for ψ_2^{k+1}.

2. Solve the following problem in subdomains $i = 1, 2$,

$$-\nabla \cdot (\epsilon_i \nabla \phi_i^{k+1}) = 0, \; x \in \Omega_i \tag{4.63}$$

$$\phi_i^{k+1} = 0 \text{ on } \partial\Omega \cap \partial\Omega_i \tag{4.64}$$

$$\epsilon_i \frac{\partial \phi_i^{k+1}}{\partial \nu} = \left[\epsilon_i \frac{\partial \psi}{\partial \nu} \right]_\Gamma \text{ on } \Gamma \tag{4.65}$$

3. Update λ by

$$\lambda^{k+1} = \lambda^k - \theta \left[\phi \right]_\Gamma \tag{4.66}$$

4. Continue with (1) unless stopping criterium $\left\| \left[\epsilon \frac{\partial \psi}{\partial \nu} \right]_\Gamma \right\|$ holds.

Similarly to the Neumann-Neumann method, in 1D we may avoid step 2 by adjusting the update of λ to

$$\lambda^{k+1} = \lambda^k + \theta \left[\epsilon \frac{\partial \psi}{\partial \nu} \right]_\Gamma . \tag{4.67}$$

4.2.4 Convergence Analysis

To show that algorithm **DDP** converges we will analyze the weak form of (4.47). Step 3 of the algorithm will be analyzed as a Richardson iteration preconditioned by steps 1 and 2. This will be shown to be a contraction on a complete metric space and the Banach Contraction Theorem will be applied to guarantee convergence.

Let $\gamma_i : H^1(\Omega_i) \to H^{\frac{1}{2}}(\partial\Omega_i)$ be the trace operator, and

$$\begin{aligned} V_i &:= H^1(\Omega_i), && (4.68)\\ V_i^0 &:= \left\{v \in V_i \mid \gamma_i v|_{\partial\Omega\cap\partial\Omega_i} = 0\right\}, && (4.69)\\ V &:= \left\{v \in L^2(\Omega) \;\middle|\; v|_{\Omega_i} \in V_i^0 \text{ for } i = 1,2\right\}, && (4.70)\\ \Lambda &:= H^{\frac{1}{2}}(\Gamma), && (4.71)\\ (u,v)_{\Omega_i} &:= \int_{\Omega_i} u\, v, && (4.72)\\ a_i(u,v) &:= \int_{\Omega_i} \epsilon_i \nabla u \cdot \nabla v, && (4.73) \end{aligned}$$

First we characterize K as an operator from Λ to Λ', where Λ' denotes the space of continuous linear functionals on Λ. Let $\mu \in \Lambda$, multiply by μ, apply the divergence theorem and note that both $\epsilon_i J_i \lambda$ is divergence free in Ω_i and $R_i\mu|_{\partial\Omega\cap\partial\Omega_i} = 0$ for any $\mu \in \Lambda$.

$$\begin{aligned} \langle K\lambda, \mu\rangle &= \int_\Gamma \left(\epsilon_1 \frac{\partial J_1\lambda}{\partial \nu^1} + \epsilon_2 \frac{\partial J_2\lambda}{\partial \nu^2}\right)\mu \\ &= \int_\Gamma \epsilon_1 \frac{\partial J_1\lambda}{\partial \nu^1} R_1\mu + \int_\Gamma \epsilon_2 \frac{\partial J_2\lambda}{\partial \nu^2} R_2\mu \\ &= \int_{\partial\Omega_1} \epsilon_1 \frac{\partial J_1\lambda}{\partial \nu^1} R_1\mu + \int_{\partial\Omega_2} \epsilon_2 \frac{\partial J_2\lambda}{\partial \nu^2} R_2\mu \\ &= \int_{\Omega_1} \epsilon_1 \nabla J_1\lambda \cdot \nabla R_1\mu + \int_{\Omega_2} \epsilon_2 \nabla J_2\lambda \cdot \nabla R_2\mu \\ &= a_1(J_1\lambda, R_1\mu) + a_2(J_2\lambda, R_2\mu) \end{aligned}$$

Since R_i can be any possible extension operator, we take $R_i = J_i$ in this case. Thus we have

$$\langle K\lambda, \mu\rangle = a_1(J_1\lambda, J_1\mu) + a_2(J_2\lambda, J_2\mu). \tag{4.74}$$

The fact that for each $\lambda \in \Lambda$, $\langle K\lambda, \mu\rangle$ is linear as a function of μ follows from the linearity of integration. In the proof of the upcoming Theorem 4.2.4.2 continuity will be shown. Thus as a function of μ, $\langle K\lambda, \mu\rangle \in \Lambda'$ for each $\lambda \in \Lambda$.

Next we proceed similarly for the right hand side of (4.74), recalling the definition of $P_i q$ (4.15)-(4.17) and applying divergence theorem

$$\begin{aligned}
\langle \Upsilon, \mu \rangle &= -\int_\Gamma \epsilon_2 \frac{\partial P_2 q}{\partial \nu^2}\mu - \int_\Gamma \epsilon_1 \frac{\partial P_1 q}{\partial \nu^1}\mu - \int_\Gamma \epsilon_2 \frac{\partial J_2 \psi_\triangle}{\partial \nu^2}\mu \\
&= -\int_\Gamma \epsilon_2 \frac{\partial P_2 q}{\partial \nu^2} R_2\mu - \int_\Gamma \epsilon_1 \frac{\partial P_1 q}{\partial \nu^1} R_1\mu - \int_\Gamma \epsilon_2 \frac{\partial J_2 \psi_\triangle}{\partial \nu^2} R_2\mu \\
&= -\int_{\partial\Omega_2} \epsilon_2 \frac{\partial P_2 q}{\partial \nu^2} R_2\mu - \int_{\partial\Omega_1} \epsilon_2 \frac{\partial P_1 q}{\partial \nu^1} R_1\mu - \int_{\partial\Omega_2} \epsilon_2 \frac{\partial J_2 \psi_\triangle}{\partial \nu^2} R_2\mu \\
&= \int_{\Omega_2} \nabla\cdot(\epsilon_2 \nabla P_2 q) R_2\mu - \int_{\Omega_2} \epsilon_2 \nabla P_2 q \cdot \nabla R_2\mu \\
&\quad + \int_{\Omega_1} \nabla\cdot(\epsilon_1 \nabla P_1 q) R_1\mu - \int_{\Omega_1} \epsilon_1 \nabla P_1 q \cdot \nabla R_1\mu - \int_{\Omega_2} \epsilon_2 \nabla J_2\psi_\triangle \cdot \nabla R_2\mu \\
&= \int_{\Omega_2} q_2 R_2\mu - \int_{\Omega_2} \epsilon_2 \nabla P_2 q \cdot \nabla R_2\mu + \int_{\Omega_1} q_1 R_1\mu - \int_{\Omega_1} \epsilon_1 \nabla P_1 q \cdot \nabla R_1\mu - a_2(J_2\psi_\triangle, R_2\mu) \\
&= \sum_{i=1}^{2} [(q_i, R_i\mu)_{\Omega_i} - a_i(P_i q, R_i\mu)] - a_2(J_2\psi_\triangle, R_2\mu).
\end{aligned}$$

Linearity of $\langle \Upsilon, \mu \rangle$ follows from the linearity of integration. Continuity of $a_i(\cdot,\cdot)$ terms is demonstrated in the proof of Theorem 4.2.4.2. The continuity of the terms $(q_i, R_i\mu)_{\Omega_i}$ remain to be demonstrated. Let $\mu \in \Lambda$,

$$\begin{aligned}
|(q_i, R_i\mu)_{\Omega_i}| &= \left| \int_{\Omega_i} q_i \, R_i\mu \right| \\
&\leq \|q_i\|_{L^2(\Omega_i)} \|R_i\mu\|_{L^2(\Omega_i)} \\
&\leq \|q_i\|_{L^2(\Omega_i)} \|R_i\mu\|_{H^1(\Omega_i)}.
\end{aligned}$$

as long as $q_i \in L^2(\Omega_i)$. Then since $R_i : \Lambda \to V_i^0$ is continuous, there exists $C > 0$ s.t.

$$|(q_i, R_i\mu)_{\Omega_i}| \leq C\|q_i\|_{L^2(\Omega_i)} \|\mu\|_\Lambda.$$

So, $\langle \Upsilon, \mu \rangle \in \Lambda'$.

And now we write the weak form of (4.47) as:

$$find\ \lambda \in \Lambda \ :\ \langle K\lambda, \mu \rangle = \langle \Upsilon, \mu \rangle \quad \forall \mu \in \Lambda. \tag{4.75}$$

To prove convergence of Algorithm **DDP** we use the following lemma, contained in a theorem from [34].

Lemma 4.2.4.1 ([34] p. 120)**.** *Let X be a Hilbert space and $\mathcal{A} : X \to X'$ be an operator. Suppose $\mathcal{A}$ can be split into $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2$, and that $\mathcal{A}_i$ is continuous and coercive with continuity constant C_i and coercivity constant α_i, $i = 1, 2$. That is,*

$$|\langle \mathcal{A}_i x, y\rangle| \leq C\|x\|_X \|y\|_X \quad \forall\, x, y \in X \quad \text{and} \tag{4.76}$$

$$\langle \mathcal{A}_i x, x\rangle \geq \alpha\|x\|_X^2 \quad \forall\, x \in X. \tag{4.77}$$

Define $\mathcal{N} = (\mathcal{A}_1^{-1} + \mathcal{A}_2^{-1})^{-1}$. Note that $\mathcal{N}$ is continuous and coercive since $\mathcal{A}_i$ is continuous and coercive. Let $\alpha_{\mathcal{N}}$ denote the coercivity constant for $\mathcal{N}$ and let $C_{\mathcal{N}}$ denote the continuity constant for $\mathcal{N}$. Further assume $\mathcal{N}$ satisfies the condition that there exists $k^* > 0$ s.t.

$$\langle \mathcal{N}\lambda, \mathcal{N}^{-1}\mathcal{A}\lambda\rangle + \langle \mathcal{A}\lambda, \lambda\rangle \geq k^*\|\lambda\|_X^2 \quad \forall \lambda \in X. \tag{4.78}$$

Then there exists a θ_{max} given by

$$\theta_{max} = \frac{k^*\alpha_{\mathcal{N}}}{C_{\mathcal{N}}\left(\frac{1}{\alpha_1} + \frac{1}{\alpha_2}\right)(C_1 + C_2)^2} \tag{4.79}$$

s.t. for any $0 < \theta < \theta_{max}$, the operator $T_\theta = I - \theta\,\mathcal{N}^{-1}\mathcal{A}$ is a contraction on the space X.

We also note that if $\mathcal{A}_i$ are symmetric, condition (4.78) is equivalent to the coercivity of $\mathcal{A}$ [[34], p. 121].

We will also require the Banach Contraction Mapping Theorem. A proof of this result can be found in any standard real analysis text, e.g. [36].

Theorem 4.2.4.1 (Banach Contraction Mapping Theorem)**.** *Let (X, d) be a complete metric space, and let $f : X \to X$ be a contraction. That is, there exists $0 < K < 1$ such that $d(f(x), f(y)) < K\, d(x, y)$ for all $x, y \in X$. Then there exists a unique fixed point $x_0 \in X$ such that $f(x_0) = x_0$. Further, for any $x \in X$, $\lim_{n\to\infty} f^n(x) = x_0$.*

Theorem 4.2.4.2. *There exists $\theta^{\psi}_{max} > 0$ s.t. for any $0 < \theta < \theta^{\psi}_{max}$, and for any initial guess $\lambda^{(0)}$, Algorithm **DDP** converges to the solution λ of (4.47).*

Proof. A proof of this theorem takes the following steps:

1. First we define an operator splitting for K, and show that Algorithm **DDP** is equivalent to the following preconditioned Richardson scheme for (4.47),

$$(K_1^{-1} + K_2^{-1})^{-1}(\lambda^{k+1} - \lambda^k) = \theta(\Upsilon - K\lambda^k). \tag{4.80}$$

2. Next we show that

$$\mathcal{N}_p := (K_1^{-1} + K_2^{-1})^{-1} \tag{4.81}$$

satisfies the criteria from Lemma 4.2.4.1 on the Hilbert space Λ.

3. Finally we conclude via application of the Banach Contraction Mapping Theorem and Lemma 4.2.4.1 that the algorithm converges.

We will require the Poincaré Inequality.

Theorem 4.2.4.3 (Poincaré Inequality, [3] p. 290)**.** *Suppose that $1 \leq p < \infty$ and U is a bounded open set. Then there exists a constant C such that*

$$\|u\|_{L^p(U)} \leq C\|\nabla u\|_{L^p(U)} \; \forall u \in W_0^{1,p}(U). \tag{4.82}$$

In particular, the expression $\|\nabla u\|_{L^p(U)}$ is a norm on $W_0^{1,p}(U)$, and it is equivalent to the norm $\|u\|_{W^{1,p}}$.

Step 1

We define the operator splitting of K by

$$K_1\lambda := \frac{\partial J_1\lambda}{\partial \nu}, \tag{4.83}$$

$$K_2\lambda := -\frac{\partial J_2\lambda}{\partial \nu}. \tag{4.84}$$

Recall that $\psi_1^{k+1} = J_1\lambda^k + P_1 q$ and $\psi_2^{k+1} = J_2\lambda^k + J_2\psi_\triangle + P_2 q$, therefore,

$$\phi_1^{k+1}|_\Gamma = K_1^{-1}\left(\frac{\partial J_1\lambda}{\partial \nu} + \frac{\partial P_1 q}{\partial \nu} - \frac{\partial J_2\lambda^k + J_2\psi_\triangle}{\partial \nu} - \frac{\partial P_2 q}{\partial \nu}\right) \tag{4.85}$$

$$= -K_1^{-1}(\Upsilon - K\lambda^k). \tag{4.86}$$

Similarly,

$$\phi_2^{k+1}|_\Gamma = K_2^{-1}(\Upsilon - K\lambda^k). \tag{4.87}$$

And so we have,

$$\lambda^{k+1} = \lambda^k + \theta(K_1^{-1} + K_2^{-1})(\Upsilon - K\lambda^k). \tag{4.88}$$

Or equivalently,

$$\mathcal{N}_p(\lambda^{k+1} - \lambda^k) = \theta(\Upsilon - K\lambda^k). \tag{4.89}$$

Step 2

To show that $\mathcal{N}_p$ satisfies the criteria of Lemma 4.2.4.1, we must show that K_i is continuous and coercive on Λ for $i = 1, 2$ and that $\mathcal{N}_p$ satisfies condition (4.78).

To show that K_i is coercive we will require the trace inquality

Proposition 4.2.4.1 (Trace Inequality [34], p. 7)**.** *Let $v_i \in V_i$, then there exists C_i^* s.t.*

$$\|v_i|_\Gamma\|_\Lambda \leq C_i^* \|v_i\|_{H^1(\Omega_i)}. \tag{4.90}$$

In the weak setting we have

$$\langle K_1\lambda, \mu\rangle = a_1(J_1\lambda, J_1\mu), \tag{4.91}$$

$$\langle K_2\lambda, \mu\rangle = a_2(J_2\lambda, J_2\mu). \tag{4.92}$$

Then we calculate,

$$\begin{aligned} \langle K_1\lambda, \lambda\rangle &\geq \epsilon_0 \|\nabla J_1\lambda\|^2_{L^2(\Omega_i)} && (4.93)\\ &\geq \epsilon_0 C \|J_1\lambda\|^2_{H^1(\Omega_i)} && (4.94)\\ &\geq \epsilon_0 C C^* \|J_1\lambda\|^2_\Lambda. && (4.95) \end{aligned}$$

where C is the constant guaranteed by the Poincaré Inquality. Thus K_1 is coercive with $\alpha_1 = \epsilon_0 C C^*$. The calculation for K_2 follows.

Next we calculate

$$\langle K_1\lambda, \mu\rangle \quad \leq \quad \epsilon_1 \|J_1\lambda\|_{H^1(\Omega_1)} \|J_1\mu\|_{H^1(\Omega_1)}. \tag{4.96}$$

From well known estimates for elliptic boundary value problems [[34], p. 9] we know there exists $\beta > 0$ such that

$$\|J_1\lambda\|_{H^1(\Omega_1)} \leq \beta \|\lambda\|_\Lambda. \tag{4.97}$$

Thus K_1 is continuous with constant $C_1 = \epsilon_1\beta$. The calculation for K_2 follows.

Now we simply note that the K_i are clearly symmetric, and thus condition (4.78) is equivalent to the coercivity of K. But the coercivity of K is a direct consequence of the coercivity of the K_i. Thus condition (4.78) is satisfied, and the conditions of Lemma 4.2.4.1 are satisfied.

Step 3

Let θ^ψ_{max} be the constant guaranteed by Lemma 4.2.4.1. Let $0 < \theta < \theta^\psi_{max}$. From Lemma 4.2.4.1 we know that the operator

$$T^\psi_\theta := I - \theta \mathcal{N}_p^{-1} K \tag{4.98}$$

is a contraction on Λ. Notice,

$$\mathcal{N}_p(\lambda^{k+1} - \lambda^k) = \theta(\Upsilon - K\lambda^k)$$

then solving for λ^{k+1},

$$\lambda^{k+1} = \lambda^k - \theta \mathcal{N}_p^{-1}(K\lambda^k - \Upsilon),$$

and applying (4.98),

$$\lambda^{k+1} = T^\psi_\theta \lambda^k + \theta \mathcal{N}_p^{-1}\Upsilon.$$

Then since $\theta\mathcal{N}_p^{-1}\Upsilon$ does not depend on λ^k, T^ψ_θ a contraction implies

$$G^\psi_\theta \lambda = T^\psi_\theta \lambda + \theta \mathcal{N}_p^{-1}\Upsilon \tag{4.99}$$

is a contraction on the Hilbert space Λ.

Now applying the Banach Contraction Mapping Principle we know that Algorithm **DDP** converges to a fixed point, λ_0 of the map G_θ^ψ. Then we have

$$G_\theta^\psi \lambda_0 = \lambda_0,$$

then recalling (4.99)

$$T_\theta^\psi \lambda_0 + \theta \mathcal{N}_p^{-1} \Upsilon = \lambda_0,$$

then recalling (4.98)

$$\lambda_0 - \theta \mathcal{N}_p^{-1}(K\lambda_0 - \Upsilon) = \lambda_0,$$

then isolating the two free λ^0 terms,

$$\lambda_0 - \lambda_0 = \theta \mathcal{N}_p^{-1}(K\lambda_0 - \Upsilon),$$

so,

$$0 = K\lambda_0 - \Upsilon.$$

□

Estimating θ_{max}^ψ directly from material parameters is a subject of ongoing research. The coercivity and continuity constants depend on known material parameters, and using optimal values for θ has a dramatic influence on the efficiency of this method [12].

4.3. DDM for Carrier Transport

In this section we extend the techniques developed in Section 4.1. to the problem for electron and hole densities. The method developed here is again related to the Neumann-Neumann method, but is designed to handle the unusual Robin-like transmission condition (2.21) or (2.23), rather than the homogenous jump in the primary variable from the background on the linear Poisson equation.

4.3.1 Continuous Formulation

Here we develop an algorithm to solve (2.9) with (2.21) and (2.22) or (2.10) with (2.23) and (2.24). To this end we note that each problem for carrier transport in Slotboom variables can be written in the following form, $i = 1, 2$,

$$-\nabla \cdot (k_i \nabla u_i) = f_i, \quad x \in \Omega_i, \tag{4.100}$$

$$k_1^\Gamma \frac{\partial u_1}{\partial \nu^1} = a_2 u_2^\Gamma - a_1 u_1^\Gamma, \tag{4.101}$$

$$k_1^\Gamma \frac{\partial u_1}{\partial \nu^1} = k_2^\Gamma \frac{\partial u_2}{\partial \nu^1}, \tag{4.102}$$

$$u|_{\partial\Omega \cap \partial\Omega_i} = 0. \tag{4.103}$$

We note that homogeneous Dirichlet boundary conditions (4.103) are assumed at the external boundary for simplicity of exposition.

Again k is data s.t. $0 < k_{min} < k < k_{max}$ on Ω for some constants k_{min}, k_{max}. We also assume $f \in L^2(\Omega_i)$ for $i = 1, 2$ and a_i are positive and bounded.

Rather than solving the problem on the entire domain, we would like to solve independent problems in subdomains Ω_1, Ω_2. The iterative substructuring method described here is thus a method for finding a suitable value λ defined on the interface Γ so that (4.100)-(4.103) is equivalent to the problem

$$-\nabla \cdot (k_1 \nabla u_1) = f_1, \ x \in \Omega_1, \quad u_1|_\Gamma = \lambda, \tag{4.104}$$

$$-\nabla \cdot (k_2 \nabla u_2) = f_2, \ x \in \Omega_2, \quad u_2|_\Gamma = \frac{a_1}{a_2}\lambda + \frac{1}{a_2} k_1^\Gamma \frac{\partial u_1}{\partial \nu^1}. \tag{4.105}$$

For any λ the solution of the system (4.104)-(4.105) satisifes (4.101). However, there is no reason to expect that (4.102) is satisfied by a solution to (4.104)-(4.105). Thus we seek the value λ which guarantees (4.102) is also satisfied.

To develop a method for finding such a λ, we write

$$u_1 = H_1\lambda + P_1 f, \quad u_2 = H_2\lambda + H_2\left(\frac{1}{a_1} k_1^\Gamma \left(\frac{\partial H_1 \lambda}{\partial \nu^1} + \frac{\partial P_1 f}{\partial \nu^1}\right)\right) + P_2 f, \tag{4.106}$$

where u is the solution of (4.104)-(4.105). Here we recall the definition of $P_i f$ (4.15)-(4.17), and define $H_i\lambda$ as the solution of the problems

$$-\nabla \cdot (k_i \nabla v_i) = 0, \quad x \in \Omega_i, \tag{4.107}$$

$$v_1 = \lambda, \quad on\ \Gamma \tag{4.108}$$

$$v_2 = \frac{a_1}{a_2}\lambda, \quad on\ \Gamma \tag{4.109}$$

$$v_i = 0 \quad on\ \partial\Omega_i \cap \partial\Omega. \tag{4.110}$$

For u_1 this is the standard separation of the solution to (4.104) into components responding to the boundary data λ and the forcing term f_1. For u_2 we additionally separate the response to the data $\frac{a_1}{a_2}\lambda + \frac{1}{a_2}k_1^\Gamma \frac{\partial u_1}{\partial \nu^1}$ into components responding to $\frac{a_1}{a_2}\lambda$ and $\frac{1}{a_2}k_1^\Gamma \frac{\partial u_1}{\partial \nu^1}$ distinctly. The multiplicative factor $\frac{a_1}{a_2}$ is included in the definition of $H_2\lambda$ on the boundary data to simplify calculations.

Then define Ξ and Σ

$$\Xi\lambda := k_1^\Gamma \frac{\partial H_1\lambda}{\partial \nu} - k_2^\Gamma \frac{\partial H_2\lambda}{\partial \nu} - k_2^\Gamma \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right)}{\partial \nu}, \tag{4.111}$$

$$\Sigma := k_2^\Gamma \frac{\partial P_2 f}{\partial \nu} - k_1^\Gamma \frac{\partial P_1 f}{\partial \nu} + k_2^\Gamma \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right)}{\partial \nu} \tag{4.112}$$

$$\tag{4.113}$$

and we consider the equation

$$\Xi\lambda = \Sigma. \tag{4.114}$$

Lemma 4.3.1.1. *Suppose λ satisfies the equation $\Xi\lambda = \Sigma$. Then a solution to* (4.104)-(4.105) *also solves* (4.100)-(4.103).

Proof. The proof of this lemma is a simple calculation. Assume λ satisfies (4.114). Assume u satisfies (4.104)-(4.105) with this λ. Clearly (4.100) is satisfied for $i = 1, 2$. Further, note that for any λ the condition (4.101) is satisfied by construction. So, the only issue

to check is the condition (4.102). So we calculate, beginning with (4.114).

$$\Xi\lambda \;=\; \Sigma.$$

Recalling (4.111)-(4.112) we have

$$k_1^\Gamma \frac{\partial H_1 \lambda}{\partial \nu} - k_2^\Gamma \frac{\partial H_2 \lambda}{\partial \nu} - k_2^\Gamma \frac{\partial H_2 \left(\frac{1}{a_1} \frac{\partial H_1 \lambda}{\partial \nu} \right)}{\partial \nu} \;=\; k_2^\Gamma \frac{\partial P_2 f}{\partial \nu} - k_1^\Gamma \frac{\partial P_1 f}{\partial \nu} + k_2^\Gamma \frac{\partial H_2 \left(\frac{1}{a_1} \frac{\partial P_1 f}{\partial \nu} \right)}{\partial \nu}.$$

Then we group terms by domain,

$$k_1^\Gamma \left(\frac{\partial H_1 \lambda}{\partial \nu} + \frac{\partial P_1 f}{\partial \nu} \right) \;=\; k_2^\Gamma \left(\frac{\partial H_2 \left(\lambda + \frac{1}{a_1} \left(\frac{\partial H_1 \lambda}{\partial \nu} + \frac{\partial P_1 f}{\partial \nu} \right) \right)}{\partial \nu} + \frac{\partial P_2 f}{\partial \nu} \right),$$

Recalling (4.106) we conclude

$$k_1^\Gamma \frac{\partial u_1}{\partial \nu^1} \;=\; k_2^\Gamma \frac{\partial u_2}{\partial \nu^1}.$$

□

4.3.2 Algebraic Formulation

To construct the algebraic counterpart to (4.114) we consider a finite difference discretization of (4.104)-(4.105). After reordering unknowns, we have

$$\begin{bmatrix} \mathbf{A}_{1,1} & 0 & \mathbf{A}_{1,\Gamma} & 0 \\ 0 & \mathbf{A}_{2,2} & 0 & \mathbf{A}_{2,\Gamma} \\ \mathbf{A}_{\Gamma,1} & \mathbf{A}_{\Gamma,2} & \mathbf{A}^1_{\Gamma,\Gamma} & \mathbf{A}^2_{\Gamma,\Gamma} \\ \mathbf{A}^{flux}_{1,\Gamma} & 0 & a_1\mathbf{I} & -a_2\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \\ \mathbf{U}^1_\Gamma \\ \mathbf{U}^2_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \{\mathbf{F}_\Gamma\}_\Gamma \\ 0 \end{bmatrix}. \tag{4.115}$$

The first two lines of this system are simply the discretization of the the equations in the bulk domains Ω_1 and Ω_2. In the third line we define

$$\{\mathbf{F}_\Gamma\}_\Gamma := \frac{\mathbf{F}^1_\Gamma + \mathbf{F}^2_\Gamma}{2}. \tag{4.116}$$

The last line is the algebraic counterpart of the TEM equation (2.21) or (2.23), with $\mathbf{A}^{flux}_{1,\Gamma}$ describing the discrete flux of the solution $\mathbf{U}_1$ as it approaches Γ. Rewriting this system

we have four equations

$$\mathbf{A}_{1,1}\mathbf{U}_1 + \mathbf{A}_{1,\Gamma}\mathbf{U}_\Gamma^1 = \mathbf{F}_1, \tag{4.117}$$

$$\mathbf{A}_{2,2}\mathbf{U}_2 + \mathbf{A}_{2,\Gamma}\mathbf{U}_\Gamma^2 = \mathbf{F}_2, \tag{4.118}$$

$$\mathbf{A}_{\Gamma,1}\mathbf{U}_1 + \mathbf{A}_{\Gamma,2}\mathbf{U}_2 + \mathbf{A}_{\Gamma,\Gamma}^1\mathbf{U}_\Gamma^1 + \mathbf{A}_{\Gamma,\Gamma}^2\mathbf{U}_\Gamma^2 = \{\mathbf{F}_\Gamma\}_\Gamma\,, \tag{4.119}$$

$$\mathbf{A}_{1,\Gamma}^{flux}\mathbf{U}_1 + a_1\mathbf{I}\mathbf{U}_\Gamma^1 - a_2\mathbf{I}\mathbf{U}_\Gamma^2 = 0. \tag{4.120}$$

Substituting (4.120) into (4.118) and (4.119) we have

$$\mathbf{A}_{2,2}\mathbf{U}_2 + \mathbf{A}_{2,\Gamma}\left(\frac{1}{a_2}\mathbf{A}_{1,\Gamma}^{flux}\mathbf{U}_1 + \frac{a_1}{a_2}\mathbf{U}_\Gamma^1\right) = \mathbf{F}_2, \tag{4.121}$$

$$\mathbf{A}_{\Gamma,1}\mathbf{U}_1 + \mathbf{A}_{\Gamma,2}\mathbf{U}_2 + \mathbf{A}_{\Gamma,\Gamma}^1\mathbf{U}_\Gamma^1 + \mathbf{A}_{\Gamma,\Gamma}^2\left(\frac{1}{a_2}\mathbf{A}_{1,\Gamma}^{flux}\mathbf{U}_1 + \frac{a_1}{a_2}\mathbf{U}_\Gamma^1\right) = \{\mathbf{F}_\Gamma\}_\Gamma\,. \tag{4.122}$$

Now using (4.117) and (4.121), (4.122) to solve for Ψ_1, Ψ_2 we have

$$\mathbf{U}_1 = \mathbf{A}_{1,1}^{-1}\mathbf{F}_1 - \mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,\Gamma}\mathbf{U}_\Gamma^1, \tag{4.123}$$

$$\mathbf{U}_2 = \mathbf{A}_{2,2}^{-1}\mathbf{F}_2 - \frac{1}{a_2}\mathbf{A}_{2,2}^{-1}\mathbf{A}_{2,\Gamma}\mathbf{A}_{1,\Gamma}^{flux}\mathbf{A}_{1,1}^{-1}\mathbf{F}_1 - \mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,\Gamma}\mathbf{U}_\Gamma^1 + \frac{a_1}{a_2}\mathbf{A}_{2,2}^{-1}\mathbf{A}_{2,\Gamma}\mathbf{U}_\Gamma^1. \tag{4.124}$$

Finally substituting (4.123) and (4.124) into (4.122) gives

$$\Xi_h\mathbf{U}_\Gamma^1 = \Sigma_h, \tag{4.125}$$

where

$$\begin{aligned}\Xi_h := \mathbf{A}_{\Gamma,1}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,\Gamma} - \mathbf{A}_{\Gamma,2}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,\Gamma} + \tfrac{a_1}{a_2}\mathbf{A}_{\Gamma,2}\mathbf{A}_{2,2}^{-1}\mathbf{A}_{2,\Gamma} + \mathbf{A}_{\Gamma,\Gamma}^1,\\ -\tfrac{1}{a_2}\mathbf{A}_{\Gamma,\Gamma}^2\mathbf{A}_{1,\Gamma}^{flux}\mathbf{A}_{1,1}^{-1}\mathbf{A}_{1,\Gamma} + \tfrac{a_1}{a_2}\mathbf{A}_{\Gamma,\Gamma}^2,\end{aligned} \tag{4.126}$$

and

$$\begin{aligned}\Sigma_h := \{\mathbf{F}_\Gamma\}_\Gamma - \mathbf{A}_{\Gamma,1}\mathbf{A}_{1,1}^{-1}\mathbf{F}_1 - \mathbf{A}_{\Gamma,2}\mathbf{A}_{2,2}^{-1}\mathbf{F}_2 + \tfrac{1}{a_2}\mathbf{A}_{\Gamma,2}\mathbf{A}_{2,2}^{-1}\mathbf{A}_{2,\Gamma}\mathbf{A}_{1,\Gamma}^{flux}\mathbf{A}_{1,1}^{-1}\mathbf{F}_1\\ -\tfrac{1}{a_2}\mathbf{A}_{\Gamma,\Gamma}^2\mathbf{A}_{1,\Gamma}^{flux}\mathbf{A}_{1,1}^{-1}\mathbf{F}_1.\end{aligned} \tag{4.127}$$

4.3.3 Algorithm

To find an appropriate λ and solve (4.104)-(4.105) we consider a method that can be analyzed as a Richardson scheme for (4.114).

Algorithm DDC to solve (4.100)-(4.103)

Given λ^0, for each $k \geq 0$,

1. Solve (4.104) for u_1^{k+1} and then solve (4.105) for u_2^{k+1}.

2. Solve the following problem in subdomains $i = 1, 2$,

$$-\nabla \cdot (k \nabla \phi_i^{k+1}) = 0, \ x \in \Omega_i \tag{4.128}$$

$$\phi_i^{k+1} = 0 \text{ on } \partial\Omega \cap \partial\Omega_i \tag{4.129}$$

$$k_i \frac{\partial \phi_i^{k+1}}{\partial \nu} = \left[k \frac{\partial \psi}{\partial \nu} \right]_\Gamma \text{ on } \Gamma \tag{4.130}$$

3. Update λ by

$$\lambda^{k+1} = \lambda^k - \theta_n \left[\phi^{k+1} \right]_\Gamma \tag{4.131}$$

4. Continue with (1) unless stopping criterium $\left\| \left[k \frac{\partial u}{\partial \nu^1} \right]_\Gamma \right\|$ holds.

Similarly to the Neumann-Neumann method and Algorithm **DDP** we may avoid step 2 in 1D by adjusting

$$\lambda^{k+1} = \lambda^k + \theta \left[\frac{\partial k u^{k+1}}{\partial \nu} \right]_\Gamma . \tag{4.132}$$

4.3.4 Convergence Analysis

To show that Algorithm **DDC** converges we will analyze the weak form of (4.114). This analysis is very similar to that of Algorithm **DDP**. This is not surprising since both methods are designed to be extensions of the Neumann-Neumann algorithm, and thus have similar structures. However, the calculations do differ due to the difference in the definitions of the interface equations, (4.47) and (4.114).

In this analysis the Richardson scheme in step 3 of the algorithm will be shown to

be a contraction on a complete metric space and the Banach Contraction Theorem will again be applied to guarantee convergence.

We recall the definitions (4.68)-(4.73) and let R_i and R_2^* denote any possible continuous extension operators from Λ to V_i^0. Also we define the bilinear form

$$b_i(u,v) := \int_{\Omega_i} k_i \nabla u \cdot \nabla v. \tag{4.133}$$

First we characterize Ξ as an operator from Λ to Λ'. Let $\mu \in \Lambda$, multiply by μ, apply the divergence theorem, and note both that $k_i\, H_i\lambda$ is divergence free in Ω_i and $R_i\mu|_{\partial\Omega\cap\partial\Omega_i} = 0$ for any $\mu \in \Lambda$.

$$\begin{aligned}
\langle \Xi\lambda, \mu\rangle &= \int_\Gamma \left(k_1 \frac{\partial H_1\lambda}{\partial \nu^1} + k_2 \frac{\partial H_2\lambda}{\partial \nu^2} + k_2 \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial\nu}\right)}{\partial\nu} \right) \mu \\
&= \int_\Gamma k_1 \frac{\partial H_1\lambda}{\partial \nu^1} R_1\mu + \int_\Gamma k_2 \frac{\partial H_2\lambda}{\partial \nu^2} R_2\mu + \int_\Gamma k_2 \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial\nu}\right)}{\partial\nu} R_2^*\mu \\
&= \int_{\partial\Omega_1} k_1 \frac{\partial H_1\lambda}{\partial \nu^1} R_1\mu + \int_{\partial\Omega_2} k_2 \frac{\partial H_2\lambda}{\partial \nu^2} R_2\mu + \int_{\partial\Omega_2} k_2 \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial\nu}\right)}{\partial\nu} R_2^* \\
&= \int_{\Omega_1} k_1 \nabla H_1\lambda \cdot \nabla R_1\mu + \int_{\Omega_2} k_2 \nabla H_2\lambda \cdot \nabla R_2\mu + \int_{\Omega_2} k_2 \nabla H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial\nu}\right) \cdot \nabla R_2^*\mu \\
&= \sum_{i=1}^{2} b_i(H_i\lambda,\, R_i\mu) + b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial\nu}\right), R_2^*, \mu\right)
\end{aligned}$$

Since R_i and R_2^* can be any possible continuous extension operators, we will take $R_i = H_i$ and $R_2^*\mu = H_2\left(\frac{1}{a_1}\frac{\partial H_1\mu}{\partial\nu}\right)$ in this case. These choices ensure symmetry of the operators, which is useful in this analysis. Then we have

$$\langle \Xi\lambda, \mu\rangle = \sum_{i=1}^{2} b_i(H_i\lambda, H_i\mu) + b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial\nu}\right), H_2\left(\frac{1}{a_1}\frac{\partial H_1\mu}{\partial\nu}\right)\right). \tag{4.134}$$

The fact that for each $\lambda \in \Lambda$, $\langle \Xi\lambda, \mu\rangle$ is linear as a function of μ follows from the linearity of integration. In the proof of the upcoming Theorem 4.3.4.1 it will be shown that Ξ is continuous. Thus for each $\lambda \in \Lambda$, $\langle \Xi\lambda, \mu\rangle \in \Lambda'$.

Next we proceed similarly for the right hand side of (4.114), recalling the definition of $P_i f$ and applying divergence theorem:

$$
\begin{aligned}
\langle \Sigma,\, \mu \rangle &= -\int_\Gamma k_2 \frac{\partial P_2 f}{\partial \nu^2}\mu - \int_\Gamma k_1 \frac{\partial P_1 f}{\partial \nu^2}\mu - \int_\Gamma k_2 \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right)}{\partial \nu^2}\mu \\
&= -\int_\Gamma k_2 \frac{\partial P_2 f}{\partial \nu^2} R_2\mu - \int_\Gamma k_1 \frac{\partial P_1 f}{\partial \nu^2} R_1\mu - \int_\Gamma k_2 \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right)}{\partial \nu^2} R_2^*\mu \\
&= -\int_{\partial\Omega_2} k_2 \frac{\partial P_2 f}{\partial \nu^2} R_2\mu - \int_{\partial\Omega_1} k_1 \frac{\partial P_1 f}{\partial \nu^1} R_1\mu - \int_{\partial\Omega_2} k_2 \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right)}{\partial \nu^2} R_2^*\mu \\
&= \int_{\Omega_2} \nabla\cdot(k_2\nabla P_2 f) R_2\mu - \int_{\Omega_2} k_2 \nabla P_2 f\cdot\nabla R_2\mu \\
&\qquad + \int_{\Omega_1} \nabla\cdot(k_1\nabla P_1 f) R_1\mu - \int_{\Omega_1} k_1\nabla P_1 f\cdot\nabla R_1\mu - \int_{\Omega_2} k_2 \nabla H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right)\cdot\nabla R_2^*\mu \\
&= \int_{\Omega_2} f_2 R_2\mu - \int_{\Omega_2} k_2\nabla P_2 f\cdot\nabla R_2\mu + \int_{\Omega_1} f_1 R_1\mu \\
&\qquad - \int_{\Omega_1} k_1\nabla P_1 f\cdot\nabla R_1\mu - b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right), R_2^*\mu\right) \\
&= \sum_{i=1}^{2}[(f_i, R_i\mu)_{\Omega_i} - b_i(P_i f, R_i\mu)] - b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial P_1 f}{\partial \nu}\right), R_2^*\mu\right).
\end{aligned}
$$

Linearity of $\langle \Sigma, \mu\rangle$ follows from linearity of integration. Continuity of the $b_i(\cdot,\cdot)$ terms is demonstrated in the proof of Theorem 4.3.4.1. In the analysis of Algorithm **DDP** we saw that as long as $f_i \in L^2(\Omega_i)$, the terms $(f_i, R_i\mu)_{\Omega_i}$ are continuous. Thus $\langle \Sigma, \mu\rangle \in \Lambda'$.

Now we can write the weak form of (4.114) as:

$$find\ \lambda \in \Lambda\ :\ \langle \Xi\lambda, \mu\rangle = \langle \Sigma, \mu\rangle \quad \forall\, \mu\ \in\ \Lambda. \tag{4.135}$$

Theorem 4.3.4.1. *There exists $\theta^u_{max} > 0$ such that for any $0 < \theta < \theta^u_{max}$, and for any initial guess $\lambda^{(0)}$, Algorithm **DDC** converges to the solution to* (4.114).

Proof. Similarly to the proof for Algorithm **DDP**, the proof of this theorem takes the following steps:

1. First we define an operator splitting for Σ, and show that Algorithm **DDC** is equivalent to the following preconditioned Richardson scheme for (4.114),

$$(\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}(\lambda^{k+1} - \lambda^k) = \theta(\Sigma\lambda - \Xi). \tag{4.136}$$

2. Next we show that

$$\mathcal{N}_c := (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \tag{4.137}$$

satisfies the criteria from Lemma 4.2.4.1 on the Hilbert space Λ.

3. Finally we conclude via application of the Banach Contraction Mapping Theorem that the algorithm converges.

Step 1

We define the operator splitting of Ξ by

$$\Xi_1\lambda := k_1^\Gamma \frac{\partial H_1\lambda}{\partial \nu^1} \tag{4.138}$$

$$\Xi_2\lambda := k_2^\Gamma \frac{\partial H_2\lambda}{\partial \nu^2} + k_2^\Gamma \frac{\partial H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right)}{\partial \nu^2}. \tag{4.139}$$

Recall that $u_1^{k+1} = H_1\lambda^k + P_1 f$ and $u_2^{k+1} = H_2\lambda^k + H_2\left(\frac{1}{a_1}k_1^\Gamma\left(\frac{\partial H_1\lambda^k}{\partial \nu} + \frac{\partial P_1 f}{\partial \nu}\right)\right) + P_2 f$ therefore,

$$\begin{aligned}
\phi_1^{k+1}|_\Gamma &= H_1^{-1}\left(\frac{\partial H_1\lambda}{\partial \nu} + \frac{\partial P_1 f}{\partial \nu}\right. && (4.140)\\
&\quad \left. - \frac{\partial\left(H_2\lambda^k + H_2\left(\frac{1}{a_1}k_1^\Gamma\left(\frac{\partial H_1\lambda^k}{\partial \nu} + \frac{\partial P_1 f}{\partial \nu}\right)\right)\right)}{\partial \nu} - \frac{\partial P_2 f}{\partial \nu}\right) && (4.141)\\
&= -H_1^{-1}(\Sigma - \Xi\lambda^k) && (4.142)
\end{aligned}$$

Similarly,

$$\phi_2^{k+1}|_\Gamma = H_2^{-1}(\Sigma - \Xi\lambda^k). \tag{4.143}$$

And so we have,

$$\lambda^{k+1} = \lambda^k + \theta(H_1^{-1} + H_2^{-1})(\Sigma - \Xi\lambda^k). \tag{4.144}$$

Or equivalently,

$$\mathcal{N}_c(\lambda^{k+1} - \lambda^k) = \theta(\Sigma - \Xi\lambda^k). \tag{4.145}$$

Step 2

To show that $\mathcal{N}_c$ satisfies the criteria of Lemma 4.2.4.1, we must show that Ξ_i is continuous and coercive on Λ for $i = 1, 2$ and that $\mathcal{N}_c$ satisfies condition (4.78).

To show that Ξ_i is coercive we will again use the trace inquality. In the weak setting we have

$$\langle \Xi_1 \lambda, \mu \rangle = b_1(H_1\lambda, H_1\mu), \tag{4.146}$$

$$\langle \Xi_2 \lambda, \mu \rangle = b_2(H_2\lambda, H_2\mu) + b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right), H_2\left(\frac{1}{a_1}\frac{\partial H_1\mu}{\partial \nu}\right)\right). \tag{4.147}$$

Then we calculate,

$$\begin{aligned} \langle \Xi_1 \lambda, \lambda \rangle &\geq k_0 \|\nabla H_1\lambda\|^2_{L^2(\Omega_1)} && (4.148)\\ &\geq k_0 C \|H_1\lambda\|^2_{H^1(\Omega_i)} && (4.149)\\ &\geq k_0 C C^* \|H_1\lambda\|^2_{\Lambda} && (4.150) \end{aligned}$$

where C is the constant guaranteed by the Poincaré Inquality. Thus Ξ_1 is coercive with $\alpha_1 = k_0 C C^*$.

To see that Ξ_2 is also coercive, note that

$$b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right), H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right)\right) \geq 0. \tag{4.151}$$

So,

$$\begin{aligned} \langle \Xi_2 \lambda, \lambda \rangle &= b_2(H_2\lambda, H_2\lambda) + b_2\left(H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right), H_2\left(\frac{1}{a_1}\frac{\partial H_1\lambda}{\partial \nu}\right)\right) && (4.152)\\ &\geq b_2(H_2\lambda, H_2\lambda). && (4.153) \end{aligned}$$

The calculation to show that $b_2(H_2\lambda, H_2\lambda)$ is coercive is identical to the calculation for Ξ_1. Thus we conclude that Ξ_2 is coercive on Λ.

Next we calculate,

$$\langle \Xi_1 \lambda, \mu \rangle \leq k_1 \|H_1\lambda\|_{H^1(\Omega_1)} \|H_1\mu\|_{H^1(\Omega_1)} \tag{4.154}$$

From the well known estimates for elliptic boundary value problems [[34], p. 9] we know there exists $\beta > 0$ such that

$$\|H_1\lambda\|_{H^1(\Omega_1)} \leq \beta\|\lambda\|_\Lambda. \tag{4.155}$$

Thus Ξ_1 is continuous with constant $C_1 = \epsilon_1\beta$. Again the calculation for Ξ_2 follows.

Now we simply note that the Ξ_i are clearly symmetric, and thus condition (4.78) is equivalent to the coercivity of Ξ. But the coercivity of Ξ is a direct consequence of the coercivity of the Ξ_i. Thus condition (4.78) is satisfied, and the conditions of Lemma 4.2.4.1 are satisfied.

Step 3

Let θ^u_{max} be the constant guaranteed by Lemma 4.2.4.1. Let $0 < \theta < \theta^u_{max}$. From Lemma 4.2.4.1 we know that the operator

$$T^u_\theta := I - \theta\mathcal{N}_c^{-1}\Xi \tag{4.156}$$

is a contraction on Λ. Notice,

$$\mathcal{N}_c(\lambda^{k+1} - \lambda^k) = \theta(\Sigma - \Xi\lambda^k),$$

then solving for λ^{k+1},

$$\lambda^{k+1} = \lambda^k - \theta\mathcal{N}_c^{-1}(\Xi\lambda^k - \Sigma),$$

and applying (4.156),

$$\lambda^{k+1} = T^u_\theta\lambda^k + \theta\mathcal{N}_c^{-1}\Sigma.$$

Then since $\theta\mathcal{N}_c^{-1}\Sigma$ does not depend on λ^k, T^u_θ a contraction implies

$$G^u_\theta\lambda = T^u_\theta\lambda + \theta\mathcal{N}_c^{-1}\Sigma \tag{4.157}$$

is a contraction on the Hilbert space Λ.

Now applying the Banach Contraction Mapping Principle we know that Algorithm **DDC** converges to a fixed point, λ_0 of the map G^u_θ. Then we have

$$G^u_\theta \lambda_0 = \lambda_0,$$

then recalling (4.157),

$$T^u_\theta \lambda_0 + \theta \mathcal{N}_c^{-1} \Sigma = \lambda_0,$$

and recalling (4.156),

$$\lambda_0 - \theta \mathcal{N}_c^{-1} (\Xi \lambda_0 - \Sigma) = \lambda_0,$$

then isolating the two free λ_0 terms

$$\lambda_0 - \lambda_0 = \theta \mathcal{N}_c^{-1} (\Xi \lambda_0 - \Sigma).$$

Hence

$$0 = \Xi \lambda_0 - \Sigma$$

□

Calculating θ^u_{max} directly from material parameters is a subject of ongoing research. The coercivity and continuity constants depend on known material parameters, and just as with θ^ψ_{max}, using optimal values for θ again has a dramatic influence on the efficiency of this method.

4.3.5 Alternative Algorithm

A more natural algorithm for the simulation of the equations for carrier densities is one in which the subdomain solves involve a Neumann boundary condition at the interface and iterates to satisfy (4.101). Here we present an algorithm with this structure that has had promising performance on test problems.

In this setup we seek to find a value λ defined on the interface Γ so that (4.100)-(4.103) is equivalent to the problem

$$-\nabla \cdot (k_1 \nabla u_1) = f_1,\ x \in \Omega_1, \quad k_1 \frac{\partial u_1}{\partial \nu} = \lambda, \tag{4.158}$$

$$-\nabla \cdot (k_2 \nabla u_2) = f_2,\ x \in \Omega_2, \quad k_2 \frac{\partial u_2}{\partial \nu} = \lambda. \tag{4.159}$$

To develop the method for finding λ, we define $u_i = N_i\lambda + M_i f$ where

$$-\nabla \cdot (k_i \nabla N_i \lambda) = 0,\ x \in \Omega_i, \quad k_i \frac{\partial N_i \lambda}{\partial \nu}|_\Gamma = \lambda, \tag{4.160}$$

$$-\nabla \cdot (k_i \nabla M_i f) = f_i,\ x \in \Omega_i, \quad k_i \frac{\partial M_i f}{\partial \nu}|_\Gamma = 0. \tag{4.161}$$

Then define L and Y

$$L\lambda := \left(k_1^\Gamma \frac{\partial N_1 \lambda}{\partial \nu} + a_1 N_1 \lambda - a_2 N_2 \lambda \right) |_\Gamma, \tag{4.162}$$

$$Y := \left(a_2 M_2 f - a_1 M_1 f - k_1 \frac{\partial M_1 f}{\partial \nu} \right) |_\Gamma. \tag{4.163}$$

We then consider an algorithm that can be analyzed as a Richardson scheme for the equation $L\lambda = Y$,

Algorithm DDN to solve (4.100)-(4.103)

Given λ^0, for each $k \geq 0$,

1. Solve (4.158) for u_1^{k+1} and (4.159) for u_2^{k+1}.
2. Update λ by
$$\lambda^{k+1} = \lambda^k - \theta \left(k_1 \frac{\partial u_1^{k+1}}{\partial \nu} - a_2 u_2^{k+1} + a_1 u_1^{k+1} \right) |_\Gamma \tag{4.164}$$
3. Continue with (1) unless stopping criterium $\left\| \left(k_1 \frac{\partial u_1^{k+1}}{\partial \nu} - a_2 u_2^{k+1} + a_1 u_1^{k+1} \right) |_\Gamma \right\|$ holds.

Convergence analysis and implementation for a true semiconductor structure is a subject of ongoing research and so neither will not be presented in this work. The mixed boundary value problems on the subdomains in this algorithm with true heterojunction parameters are very ill-conditioned. Solving these subdomain problems accurately is a matter of ongoing work.

5. NUMERICAL RESULTS

In this section we present results from numerical simulations. First, the domain decomposition algorithms developed in this work are demonstrated on model problems for the potential equation and the equations for carrier densities. Convergence studies are performed and various performance aspects are discussed. These model problems are useful for testing but do not have the diffulties associated with a true heterojunction semiconductor such as coupled nonlinearity, widely discontinuous coefficients, and layer behavior. These problems were designed to have solutions with similar qualitative behavior to solutions to a true heterojunction problem.

Full semiconductor structures with heterojunctions are simulated in the section following the model problems. More details of these simulations can be found in [7, 12].

5.1. Model Problems

5.1.1 Potential Equation

Here we present numerical results from the simulation of the following model problem,

$$-(\epsilon\psi_1')' = f_1(x,\psi_1), \quad x \in (-1,0), \tag{5.1}$$

$$-(\epsilon\psi_2')' = f_2(x,\psi_2), \quad x \in (0,1), \tag{5.2}$$

$$[\psi]_0 = \psi_\triangle, \tag{5.3}$$

$$\left[\epsilon\psi'\right]_0 = 0. \tag{5.4}$$

This problem has the same form as the potential equation in the model for semiconductor structures with heterojunctions.

We discretize (5.1)-(5.2) by centered finite differences and solve (5.1)-(5.4) using

Algorithm **DDP**, while the subdomain problems are solved by Newton's method.

The DDM algorithms developed in this paper add a third iterative loop to the full simulation of heterojunction problems. With this in mind we would like to know how this effects the computational complexity of the full solver. Many iterative substructuring algorithms are known to be mesh independent, including the Neumann-Neumann method, and so we check the number of DDM iterations performed at difference mesh sizes. Like with the Neumann-Neumann method we observe mesh independence for Algorithm **DDP**, thus we do not increase the computational complexity of the problem by using the domain decomposition algorithm.

Table 5.1 presents the observed order of convergence and the number of DDM iterations at each mesh size for this problem. As expected by the finite difference discretization, we see order two convergence.

In these simulations we set

$$f_1(x,\psi) = f_2(x,\psi) = -\cos(\psi)\left(\frac{1}{x^2+1} - (\sin(\psi)+2)\left(\frac{2x}{(x^2+1)^2}\right)\right), \quad (5.5)$$

$$\epsilon_1 = \epsilon_2 = 1, \quad \psi_\triangle = -0.1. \quad (5.6)$$

Figure 5.1 presents the solution of this problem. A good value for θ_ψ was determined experimentally to be 0.248.

N	h	l2 error	Observed Order	DD iterations
200	1e-02	1.77626e-06	-	6
400	5e-03	4.52010e-07	1.9744	6
600	3.33e-03	2.02026e-07	1.9861	6
800	2.5e-03	1.13961e-07	1.9902	6

TABLE 5.1: Observed order of convergence and DDM iterations for Algorithm **DDP**.

The efficiency of Algorithm **DDP** is strongly dependent on the choice of the relaxation parameter θ_ψ. Table 5.2 shows the number of DDM iterations for varied values of

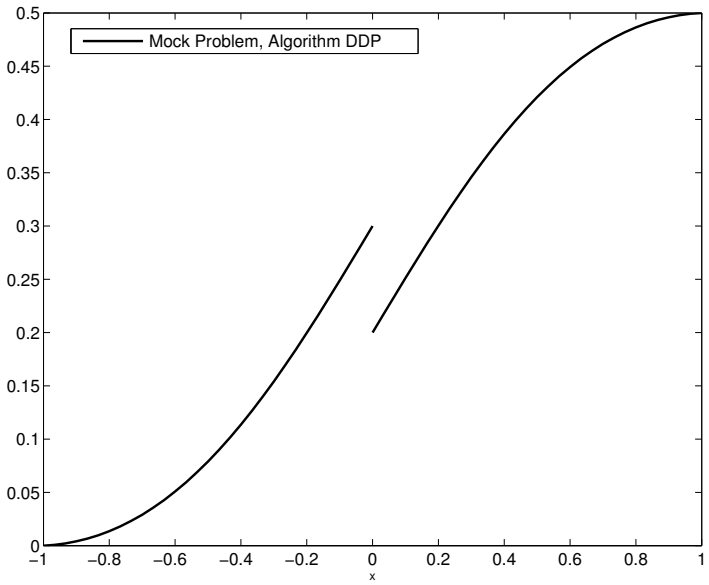

FIGURE 5.1: Solution to the model problem for the potential equation with Algorithm **DDP**.

θ_ψ. As the table shows, if the value is made too small, the method converges slowly, while if the value is too large the method does not converge.

θ_ψ	Algorithm **DDP** iterations
0.1	45
0.2	15
0.248	6
0.9	diverged

TABLE 5.2: DDM iterations for various θ_ψ for **DDP**.

5.1.2 Equations for Carrier Densities

Here we present numerical results from the simulation of the following model problem,

$$-(ku')' = r_1(x,u), \quad x \in (-1,0), \tag{5.7}$$

$$-(ku')' = r_2(x,u), \quad x \in (0,1), \tag{5.8}$$

$$ku' = a_2u_2(0) - a_1u_1(0), \tag{5.9}$$

$$\left[ku'\right]_0 = 0. \tag{5.10}$$

This problem has the same form as each of the equations for carrier densities in the model for semiconductor structures with heterojunctions.

We discretize (5.7)-(5.8) by centered finite differences and solve (5.7)-(5.10) using Algorithm **DDC**, while subdomain problems are solved by Newton's method.

As we did with Algorithm **DDP**, we check the number of DDM iterations at various mesh sizes for Algorithm **DDC** to ensure computational complexity is not increased by the use of the algorithm. Again we observe mesh independence.

Table 5.3 presents observed order of convergence and DDM iterations for this problem. As expected by the centered finite difference discretization, the order observed is 2. Also in our simulations Algorithm **DDC** is mesh independent. Figure 5.2 presents the solution of this problem. In this case we set

$$r_1(x,u) = r_2(x,u) = -\cos(u)\left(\frac{1}{x^2+1} - (\sin(u)+2)\left(\frac{2x}{(x^2+1)^2}\right)\right), \tag{5.11}$$

$$k_1 = k_2 = a_1 = a_2 = 1. \tag{5.12}$$

A good value for θ_u in this case was determined experimentally to be 0.47.

The efficiency of Algorithm **DDC** is strongly dependent on the choice of the relaxation parameter θ_u. Table 5.4 shows the number of DDM iterations for varied values of θ_u. As the table shows, if the value is made too small, the method converges slowly, while if the value is too large the method does not converge.

N	h	l2 error	Observed Order	DD iterations
80	0.0125	4.69017e-06	-	6
160	0.00625	1.26278e-06	1.893	6
320	0.00313	3.27393e-07	1.9475	6
640	0.00156	8.38261e-08	1.9656	6

TABLE 5.3: Observed order of convergence and DDM iterations for Algorithm **DDC**.

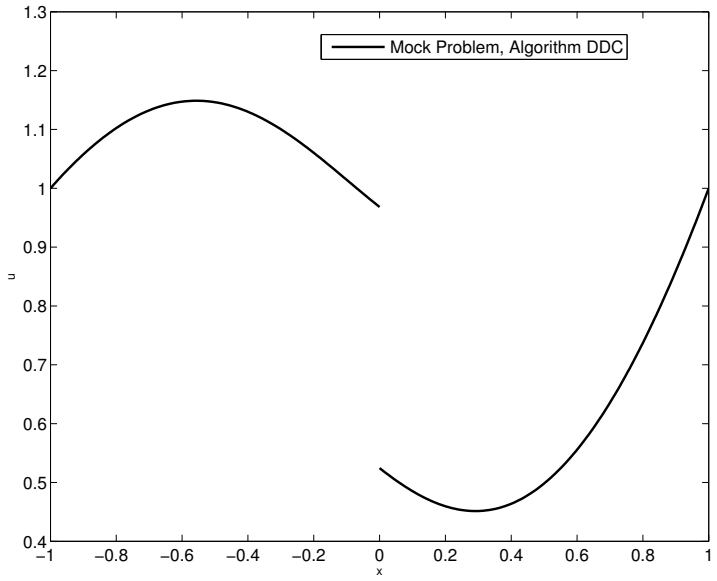

FIGURE 5.2: Solution to the model problem for carrier densities with Algorithm **DDC**.

θ_u	Algorithm **DDC** iterations
0.1	54
0.25	18
0.47	6
0.9	diverged

TABLE 5.4: DDM iterations for various θ_u for **DDC**.

5.2. Semiconductor Structures with Heterojunctions

Here we present results from the simulation of two semiconductor structures with heterojunctions using the algorithms **DDP** and **DDC**. In these simulations the full solver

is used for the continuum scale. The equations are discretized with the finite difference Scharfetter-Gummel discretization [26]. This is a centered finite difference approximation that is designed to handle the layer behavior typical in a semiconductor device simulation. The system is solved with the Gummel map as the outer loop, inside of which Algorithm **DDP** solves the potential equation and Alrogithm **DDC** solves each of the equations for carrier density. All subdomains problems are solved by Newton's method.

Table 5.5 provides data for these two structures. In the simulations, Structure 1 consists of Material L1 in the left domain and Material R1 in the right domain. Structure 2 consists of Copper-Phosphorus-Silicon (CPS) in the left domain and Silicon (Si) in the right domain.

property	L1	R1	CPS	Si
permittivity ϵ	10.0	10.0	15.1 [13]	11.9 [41]
electron affinity χ (eV)	5	5	4.05	4.05 [41]
band gap E_g (eV)	1.0	0.5	1.4 [13]	1.12 [41]
eff. electron density of states $\tilde{N}_C$ (cm^{-3})	5×10^{18}	5×10^{18}	3×10^{19}	2.8×10^{19}
eff. hole density of states $\tilde{N}_V$ (cm^{-3})	5×10^{18}	5×10^{18}	1.2×10^{18}	1×10^{19}
dopant charge density $\tilde{N}_T$ (cm^{-3})	1×10^{16}	-1×10^{15}	-6×10^{17} [13]	1×10^{15}
electron diffusion constant $\tilde{D}_n$ (cm^2/s)	2.0	2.0	2.6 [19]	37.6 [41]
hole diffusion constant $\tilde{D}_p$ (cm^2/s)	1.0	1.0	0.5	12.9 [41]
constant photogeneration density G ($\mathrm{cm}^{-3}/\mathrm{s}$)	1×10^{17}	1×10^{20}	1×10^{21}	1×10^{18}
direct recombination constant R_{dc} (cm^3/s)	1×10^{-10}	1×10^{-10}	1×10^{-10}	1×10^{-15}
jump in potential $\Delta\psi$ (eV)	-0.15		-0.01	

TABLE 5.5: Material and Structure Parameters

Figures 5.3 and 5.4 show potentials and carrier densities, respectively, for both of these structures. In Tables 5.6 and 5.7 DDM iterations are provided for each component equation at each Gummel iteration. As these show, the DDM algorithms remain essentially mesh independent when simulating a true semiconductor structure with a heterojunction.

Table 5.7 also shows how dramatically different the acceleration parameter θ for Algorithm **DDC** can be with differing parameters. A good value for the equation for

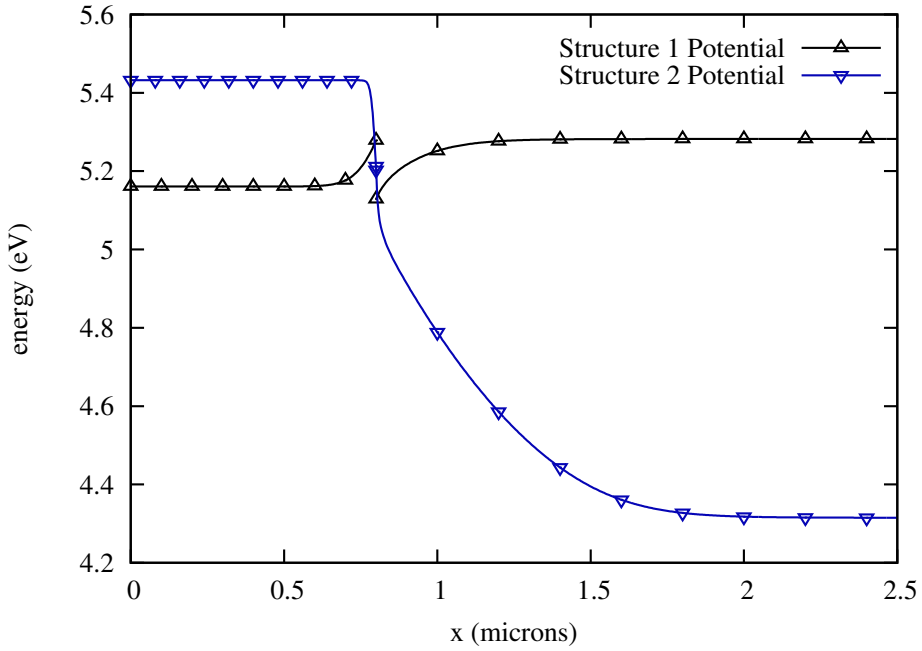

FIGURE 5.3: Simulation of Structures 1 and 2 Potentials

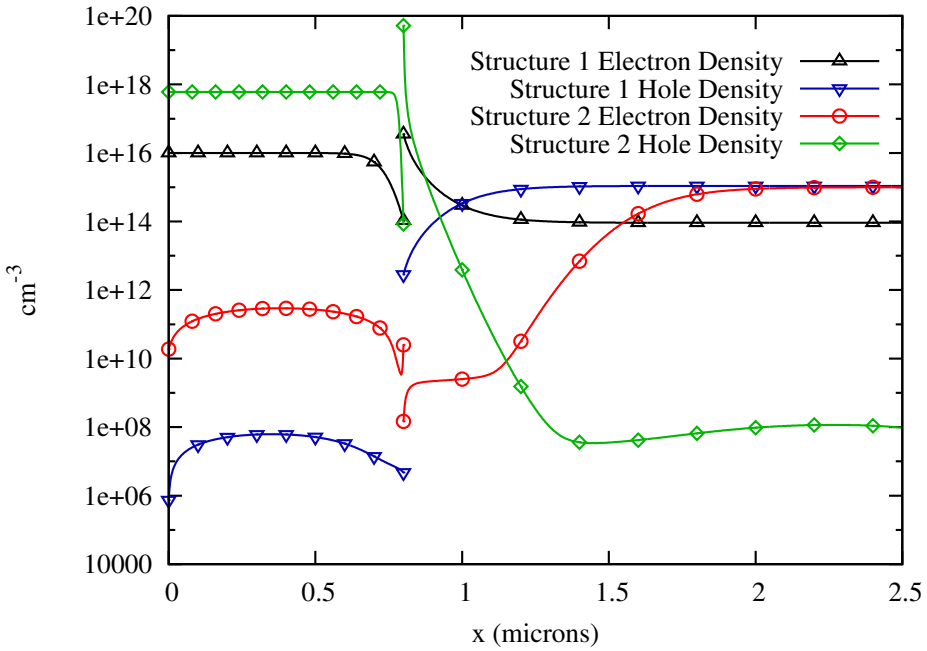

FIGURE 5.4: Simulation of Structures 1 and 2 Carrier Densities

	DDP ψ, $\theta_\psi = 0.0025$			
N	GI 1	GI 2	GI 3	GI 4
201	1	14	10	4
401	2	12	9	5
601	3	11	12	3

TABLE 5.6: Number of iterations at each Gummel Iteration (GI) and parameter θ_ψ for Structure 1 and Algorithm **DDP**.

	DDC u, $\theta_n = 2.5$				DDC v, $\theta_p = 180$			
N	GI 1	GI 2	GI 3	GI 4	GI 1	GI 2	GI 3	GI 4
201	6	2	1	1	5	3	1	1
401	5	2	1	1	8	4	1	1
601	3	2	1	1	8	4	1	1

TABLE 5.7: Number of iterations at each Gummel Iteration (GI) and parameters θ_n, θ_p for Structure 1 and algorithm **DDC**.

electron density for Structure 1 was 2.5, while a good value for the equation for hole density was 180. In fact, when these values are swapped, iterations for electron density did not converge, while the run for hole density was stopped at 2000 iterations without having converged.

This hightlights a key feature of the acceleration parameter. A good choice in each case is $\theta_{max} - \epsilon$ for very small ϵ. With any value larger than θ_{max} the algorithms does not converge, while with values much smaller than θ_{max}, the algorithm converges slowly.

6. CONCLUSIONS

In this work a model for charge transport in semiconductor structures with heterojunctions was presented. Analysis of the drift diffusion system was presented for a single semiconductor material and numerical techniques necessary for simulation of the model were presented.

The heterojunction model consists of a coupled system of partial differential equations posed in subdomains connected by unusual transmission conditions involving a jump discontinuity in electrostatic potential and Robin-like internal boundary conditions at the interface for the carrier densities. This setup lends itself naturally to a domain decomposition approach. We proposed novel iterative substructuring methods and convergence analysis was performed.

Finally we presented numerical results. First the domain decomposition algorithms were tested on model problems with the same structure as the component equations of the heterojunction model for which they were designed. In this setting convergence rates and mesh independence were demonstrated. Second, we simulated realistic semiconductor structures with heterojunctions.

In future work we would like to finish analysis and implementation of the more natural alternative algorithm for the carrier densities.

Given the importance of the choice of acceleration parameter θ for each algorithm, we also intend to implement methods for calculating optimal values directly from material parameters.

Additionally we will extend these methods to the time dependent model for charge transport in semiconductor structures with heterojunctions.

Finally, we are working to use the domain decomposition framework developed in this work as a tool for extending the traditional well-posedness analysis for the drift

diffusion system in one semiconductor domain to the model for charge transport in a structure with a heterojunction.

BIBLIOGRAPHY

1. R. E. Bank, D. J. Rose, and W. Fichtner. Numerical methods for semiconductor device simulation. *SIAM J. Sci. Statist. Comput.*, 1983.

2. C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In H. Brezis and J. L. Lions, editors, *Nonlinear partial differential equations and their applications*. Longman Scientific & Technical, UK, 1994.

3. H. Brezis. *Functional analysis, Sobolev spaces, and partial differential equations*. Springer, 2010.

4. Yanzhao Cao, Max Gunzburger, Xiaoming He, and Xiaoming Wang. Robin-Robin domain decomposition methods for the steady-state Stokes-Darcy system with the Beavers–Joseph interface condition. *Numerische Mathematik*, 117(4):601–629, 2011.

5. K. S. Chang and D. Y. Kwak. Discontinuous bubble scheme for elliptic problems with jumps in the solution. *Comput. Methods Appl. Mech. Engrg.*, 2011.

6. Z. Chen and J. Zou. Finite element methods and their convergence for elliptic and parabolic interface problems. *Numerische Mathematik*, 1998.

7. T. Costa, D. Foster, and M. Peszynska. Domain decomposition for heterojunction problems in semiconductors. *(Submitted)*, 2013.

8. C. de Falco, J. W. Jerome, and R. Sacco. Quantum-corrected drift-diffusion models: solution fixed point map and finite element approximation. *J. Comput. Phys.*, 2009.

9. Marco Discacciati, Alfio Quarteroni, and Alberto Valli. Robin-Robin domain decomposition methods for the Stokes-Darcy coupling. *SIAM Journal on Numerical Analysis*, 45(3):1246–1268, 2007.

10. Eberhard Engel and Reiner M Dreizler. *Density Functional Theory: An Advanced Course*. Springer, 2011.

11. Carlos Fiolhais, Fernando Nogueira, and Miguel AL Marques. *A primer in density functional theory*, volume 620. Springer, 2003.

12. D. Foster, T. Costa, M. Peszynska, and G. Schneider. Multiscale modeling of solar cells with interface phenomena. *Journal of Coupled Systems and Multiscale Dynamics*, 2013.

13. D. H. Foster, F. L. Barras, J. M. Vielma, and G. Schneider. Defect physics and electronic properties of cu3pse4 from first principles. *Physical Review B*, 2013.

14. Vivette Girault and Béatrice Rivière. DG approximation of coupled Navier-Stokes and Darcy equations by Beaver-Joseph-Saffman interface condition. *SIAM J. Numer. Anal.*, 47(3):2052–2089, 2009.

15. Vivette Girault, Béatrice Rivière, and Mary F. Wheeler. A discontinuous Galerkin method with nonoverlapping domain decomposition for the Stokes and Navier-Stokes problems. *Math. Comp.*, 74(249):53–84 (electronic), 2005.

16. R. Glowinski and M. F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Periaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 144–172. SIAM, Philadelphia, 1988.

17. Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Physical review*, 1964.

18. K. Horio and H. Yanai. Numerical modeling of heterojunctions including the thermionic emission mechanism at the heterojunction interface. *IEEE Trans. Electron Devices*, 1990.

19. Vorranutch Itthibenchapong, Robert S. Kokenyesi, Andrew J. Ritenour, Lev N. Zakharov, Shannon W. Boettcher, John F. Wager, and Douglas A. Keszler. Earth-abundant cu-based chalcogenide semiconductors as photovoltaic absorbers. *Journal of Materials Chemistry C*, 1(4):657–662, December 2012.

20. J. W. Jerome. *Analysis of charge transport.* Springer-Verlag, 1996.

21. C. Johnson. *Numerical solutions of partial differential equations by the finite element method.* Dover Publications, 2009.

22. C. T. Kelly. *Iterative methods for linear and nonlinear equations.* Society for Industrial and Applied Mathematics, 1995.

23. R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations.* Society for Industrial and Applied Mathematics, 2007.

24. Paul T Lin, John N Shadid, Marzio Sala, Raymond S Tuminaro, Gary L Hennigan, and Robert J Hoekstra. Performance of a parallel algebraic multilevel preconditioner for stabilized finite element semiconductor device modeling. *Journal of Computational Physics*, 228(17):6250–6267, 2009.

25. Q. Lu, M. Peszynska, and M. F. Wheeler. A parallel multi-block black-oil model in multi-model implementation. *SPE Journal*, pages 278–297, 2002.

26. P. A. Markowich. *The stationary semiconductor equations.* Springer-Verlag Wien, 1986.

27. P.A. Markowich, C. A. Ringhoffer, and C. Schmeiser. *Semiconductor equations*. Springer-Verlag Vienna, 1990.

28. V. Martin, J. Jaffré, and J. E. Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 2005.

29. F. Morales and R. E. Showalter. Interface approximation of darcy flow in a narrow channel. *Math. Methods Appl. Sci*, 2012.

30. Ali Saada N. Frih, Jean E. Roberts. Modeling fractures as interfaces: A model for Forchheimer fractures. *Computational Geosciences*, 2, 2008.

31. M. Peszynska. Multiphysics coupling of three-phase and two-phase models of flow in porous media. *Analysis and Simulation of Multifield Problems*, 2003.

32. M. Peszynska, Q. Lu, and M. F. Wheeler. Multiphysics coupling of codes. *Computational Methods in Water Resources*, pages 175–182, 2000.

33. Małgorzata Peszyńska, Mary F. Wheeler, and Ivan Yotov. Mortar upscaling for multiphase flow in porous media. *Comput. Geosci.*, 6(1):73–100, 2002.

34. A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford Science Publications, 1999.

35. B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations*. Society for Industrial and Applied Mathematics, 2008.

36. Halsey Lawrence Royden and Patrick Fitzpatrick. *Real analysis*, volume 4. Prentice Hall New York, 1988.

37. Konrad Sakowski, Leszek Marcinkowski, Stanislaw Krukowski, Szymon Grzanka, and Elzbieta Litwin-Staszewska. Simulation of trap-assisted tunneling effect on characteristics of gallium nitride diodes. *Journal of Applied Physics*, 111(12):123115–123115, 2012.

38. S. Selberherr. *Analysis and simulation of semiconductor devices*. Springer-Verlag, 1984.

39. R. E. Showalter. *Monotone operators in Banach space and nonlinear partial differential equations*, volume 49. AMS Bookstore, 1997.

40. R. E. Showalter. *Hilbert space methods in partial differential equations*. Dover Publications, 2010.

41. Simon Sze and Kwok Ng. *Physics of Semiconductor Devices*. Wiley-Interscience, 2006.

42. L. N. Trefethen and D. Bau III. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, 1997.

43. M. F. Wheeler, T. Arbogast, S. Bryant, J. Eaton, Q. Lu, M. Peszyńska, and I. Yotov. A parallel multiblock/multidomain approach to reservoir simulation. In *Fifteenth SPE Symposium on Reservoir Simulation, Houston, Texas*, pages 51–62. Society of Petroleum Engineers, 1999. SPE 51884.

44. Y. Zhang, M. Peszynska, and S. Yim. Coupling of viscous and potential flow models with free surface for near and far field wave propagation. *International Journal of Numerical Analysis and Modeling*, 4(3):256–282, 2013.

APPENDIX

A Density Functional Theory

Here we review the computational model for the microscopic calculation of heterojunction parameters, Density Functional Theory.

Heterojunction parameters are determined by the quantum mechanics of electrons. First principles methods solve a 3d quantum electron system having discrete atoms with a characteristic spacing of 2-4 angstroms. DFT is a widely used, low cost first principles method which solves the zero temperature, zero current ground state of a system [10, 11]. The local pseudopotential calculated by DFT is continuous at an interface, and can be used with known material properties to obtain the change in the continuous electrostatic potential ψ occuring very close to a heterojunction. The transport, scattering, and recombination/generation processes involved in an active semiconductor device are not suitable for handling by first principles calculation methods, and thus must remain in the realm of macroscopic models.

The fundamental equation describing quantum behavior is the Schrödinger equation. However, the problem of an interacting N electron system remains computationally intractable. Density Functional Theory, [10, 11], provides an efficient method of determining material properties from first principles by shifting focus from wave functions to

electron density, $n(\boldsymbol{r})$. This is accomplished by application of the theory of Hohenberg and Kohn to the formulation of a minimization problem in electron density equivalent to the solution of the Shrödinger equation for the ground state.

We consider here the standard Hamiltonian, $\hat{H} = \hat{T} + \hat{V}_{ee} + \hat{V}_{\text{ext}}$, of N interacting electrons, ignoring spin for brevity. $\hat{T}$ is the kinetic energy operator, $\hat{V}_{ee}$ the Coulomb interaction between electrons, and $\hat{V}_{\text{ext}}$ the interaction of electrons with an external potential, $v_{\text{ext}}(\boldsymbol{r})$.

$$\hat{T} = \sum_{i=1}^{N} -\frac{\hbar^2}{2m}\nabla_i^2, \quad \hat{V}_{ee} = \sum_{\substack{i,j=1 \\ i<j}}^{N} \frac{e^2}{|\boldsymbol{r}_i - \boldsymbol{r}_j|}, \quad \hat{V}_{\text{ext}} = \int v_{\text{ext}}(\boldsymbol{r})\, n(\boldsymbol{r})\, d^3r$$

where $\hbar = \frac{h}{2\pi}$ with h the Planck constant, m is the electron mass, $-e = -|e|$ is the charge of the electron, and $\hat{\boldsymbol{r}}_j$ is the quantum mechanical position operator for electron j.

The solutions of the stationary Schrödinger equation

$$\hat{H}|\Psi_n\rangle = E_n|\Psi_n\rangle \tag{A.1}$$

are the many-electron wavefunctions $|\Psi_n\rangle$, with energy E_n. For the ground state $|\Psi_0\rangle$ with energy E_0, (A.1) is equivalent to the minimization problem

$$E = \min_{\Psi} \langle \Psi | \hat{H} | \Psi \rangle \tag{A.2}$$

In [17], Hohenberg and Kohn establish the existence of a density variational principle for the ground state,

$$E = \min_{n} \left\{ F[n] + \int v_{\text{ext}}(\boldsymbol{r})\, n(\boldsymbol{r})\, d^3r \right\}, \quad F[n] = \min_{\Psi \to n} \langle \Psi | \hat{T} + \hat{V}_{ee} | \Psi \rangle \tag{A.3}$$

The solution to (A.3) is computationally a considerable improvement over the solution of the Schrödinger equation; in (A.3), density is a function in $\mathbb{R}^3$, while for (A.1), Ψ is posed in $\mathbb{C}^{3N}$. However, the functional $F[n]$ is unknown and DFT requires approximate solutions for $F[n]$. The Kohn-Sham equations provide a framework for the solution of the minimization problem (A.3) as well as a basis for approximating density functionals.

For a system of N non-interacting electrons we may find $F[n]$ by considering single electron Shrödinger equations,

$$\left(-\frac{\hbar}{2m}\nabla^2 + v_{\text{es}}(\boldsymbol{r}) + v_{\text{xc}}([n];\boldsymbol{r})\right)\psi_k(\boldsymbol{r}) = \epsilon_k\psi_k(\boldsymbol{r}) \tag{A.4}$$

where ψ_k, ϵ_k, denote the eigenstate and energy of a single particle, and $v_{\text{es}}(\boldsymbol{r})$ the electrostatic potential. The exchange-correlation potential $v_{\text{xc}}([n];\boldsymbol{r})$ is the functional derivative of the exchange correlation energy $E_{\text{xc}}[n]$ with respect to the electron density. $E_{\text{xc}}[n]$ is the remainder of the functional $F[n]$, after the kinetic energy of the N electrons, $T_{\text{ni}}[n]$, and the Hartree energy $U[n]$, have been subtracted:

$$E_{\text{xc}}[n] = F[n] - T_{\text{ni}}[n] - U[n] \tag{A.5}$$

A solution to the Kohn-Sham equations (A.4)-(A.5) can be found iteratively for a suitable choice of approximation of the exchange correlation energy.

B Code

B1 Algorithm DDP

Below find a matlab implementation of Algorithm **DDP** for a linear problem. The setup here is easily extended to a nonlinear problem by employing nonlinear solvers in place of the calls to bpsolve2.m.

```
% Solves
% -k_1 u_1''(x) = f_1(x) on [a,c]
% -k_2 u_2''(x) = f_2(x) on [c,b]
% u_1(a) = alpha, u_2(b) = beta
% [u]_c = Jd, [k u']_c = 0
% with algorithm DDP


%% Setup
```

```
% User edits funddp.m with specifics of the problem to be solved.
funddp;

% Solver parameters:
% If vectors: g1, g2 are grids on domains 1,2 resp.
% If scalars, g1, g2 are the number of grid points on domains 1,2 resp.
% theta = relaxation coefficient
% count = # of D-D iterations
g1 = 30;
g2 = 30;
count = 0;
theta = 1/4;
p = @(x) 0*x;

% grid2 generates grid from g1, g2.
[h1,sx1,x1] = grid2(g1,a,c);
[h2,sx2,x2] = grid2(g2,c,b);

% Evaluate coefficient functions along grids x1, x2.
k1 = k1fun(x1);
k2 = k2fun(x2);

% Initialization for solutions U1, U2 on domains 1,2 respectively.
U1 = zeros(sx1,1);
U2 = zeros(sx2,1);

% Assemble linear systems.
[A1,F1] = assembly4(x1,sx1,k1,h1,bc1,1,rhs1,1,1);
[A2,F2] = assembly4(x2,sx2,k2,h2,1,bc2,rhs2,1,1);
[A3,F3] = assembly4(x1,sx1,k1,h1,bc1,2,p,1,1);
[A4,F4] = assembly4(x2,sx2,k2,h2,2,bc2,p,1,1);

%% Solve
```

```
while abs((3/(2*h1(sx1-1)))*U1(length(x1)) - (2/h1(sx1-2))*U1(length(x1)-1) ...
      + (1/(2*h1(sx1-3)))*U1(length(x1)-2) + (3/(2*h2(1)))*U2(1) - (2/h2(2))*U2(2) ...
                                  + (1/(2*h2(3)))*U2(3)) > 10^(-10) || count == 0


    count = count + 1


    % [U] = bpsolve2(A,F,sx,a,b) solves the linear system AU=F with length(U) = sx, and boundary
    %       data a and b.
    [U1] = bpsolve2(A1,F1,sx1,alpha,lambda);
    [U2] = bpsolve2(A2,F2,sx2,lambda - Jd,beta);
    [P1] = bpsolve2(A3,F3,sx1,0,((3/(2*h1(sx1-1)))*U1(sx1) - (2/h1(sx1-2))*U1(sx1-1) ...
             + (1/(2*h1(sx1-2)))*U1(sx1-2)) - ((2/h2(2))*U2(2) - (3/(2*h2(1)))*U2(1) ...
                                                        - (1/(2*h2(3)))*U2(3)));
    [P2] = bpsolve2(A4,F4,sx2,(3/(2*h1(sx1-1)))*U1(sx1) - (2/h1(sx1-2))*U1(sx1-1) ...
                                  + (1/(2*h1(sx1-3))*U1(sx1-2)) - ((2/h2(2))*U2(2) ...
                                  - (3/(2*h2(1)))*U2(1) - (1/(2*h2(3)))*U2(3)),0);


    lambda = lambda - theta*(P1(sx1) - P2(1));
end
```

B2 Algorithm DDC

Next find a matlab implementation of Algorithm **DDC** for a linear problem. The setup here is easily extended to a nonlinear problem by employing nonlinear solvers in place of the calls to bpsolve2.m.

```
% Solves
% - (k1 u1')' = f1 on (a,c)
% - (k2 u2')' = f2 on (c,b)
% u1(a) = alpha, u2(b) = beta
% k1 u1'(c) = d2 u2(c) - d1 u1(c)
% k2 u2'(c) = k1 u1'(c)
% with algorithm DDC


%% Setup
```

```
% User edits funddc.m with specifics of the problem to be solved.
funddc;

% Solver parameters:
% If vectors: g1, g2 are grids on domains 1,2 resp.
% If scalars, g1, g2 are the number of grid points on domains 1,2 resp.
% theta = relaxation coefficient for dd algorithm
% count = # of dd iterations on the interface problem
ddcount = 0;
theta = .125;

% grid2 generates grid from g1, g2. h1,2 is vector of distances between
% grid points on domains 1,2 resp. sx1,2 are length of grids 1,2 resp.
% x1,x2 are grids on domains 1,2 resp.
g1 = 30;
g2 = 30;
[h1,sx1,x1] = grid2(g1,a,c);
[h2,sx2,x2] = grid2(g2,c,b);

% evaluate coefficient functions along grids x1, x2
k1 = k1fun(x1);
k2 = k2fun(x2);

% Initialization for solutions U1, U2 on domains 1,2 respectively.
U1 = zeros(sx1,1);
U2 = zeros(sx2,1);

% assembly.m assembles linear system
[A1,F1] = assembly4(x1,sx1,k1,h1,bc1,1,rhs1,1,1);
[A2,F2] = assembly4(x2,sx2,k2,h2,1,bc2,rhs2,1,1);

% DDP loop residual
ddres = 0;
```

```
%% Solve
while ddres > 10^(-10) || ddcount == 0
    ddcount = ddcount + 1

    % [U] = bpsolve2(A,F,sx,a,b) solves the linear system AU=F with length(U) = sx,
    %       and boundary data a and b.
    [U1] = bpsolve2(A1,F1,sx1,alpha,lambda / d1);
    U1p = k1(sx1)*((3/(2*h1(sx1-1)))*U1(sx1) - ...
          (2/h1(sx1-2))*U1(sx1-1) + (1/(2*h1(sx1-3)))*U1(sx1-2));
    [U2] = bpsolve2(A2,F2,sx2,(lambda + U1p) / d2,beta);
    U2p = k2(1)*((-3/(2*h2(1)))*U2(1) + ...
          (2/h2(2))*U2(2) - (1/(2*h2(3)))*U2(3));

    lambda = lambda - theta * (U1p - U2p);

    ddres = abs(U1p - U2p)
end
```

B3 Other Code

Here find the addtional files needed to run the code for Algorithms **DDP**, **DDC**. These include grid2.m, which constucts a grid; assembly4.m, which constructs the linear system to be solved for a centered finite difference discretization of a variable coefficient Poissoin equation; bpsolve2.m, which solves the subdomain problems; as well as funddp.m and funddc.m which provide problem definitions for the solvers.

```
% grid2.m
function [h,sx,x] = grid2(g,a,b)
% Inputs: a = left endpoint, b = right endpoint.
% If g is a vector, producus vector h of distances between nodes of x.
% If g is scalar, produces uniform grid from a to b with mesh width g.
%
% Outputs: h = vector of distances between nodes.
% sx = # of nodes in grid x.
```

```
% x = grid.
    if length(g) == 1
        hs = (b-a)/g;
        x = a:hs:b;
        sx = length(x);
        h = hs*ones(sx-1,1);
    else
        x = g;
        sx = length(g);
        h = zeros(sx-1,1);

        for p = 1:sx - 1
            h(p) = x(p+1) - x(p);
        end
    end
end
```

```
% assembly4.m
%
function [A,F] = assembly4(x,sx,k,h,bc1,bc2,rhs,bv1,bv2)
%
% assembly4.m assembles the linear system Ax = F for a centered finite
% difference discretization of the problem
% -(k u')' = f with Dirichlet, Neumann, or mixed boundary conditions.
%
% Inputs: x = grid
% sx = # nodes in grid
% k = vector of function k evaluated at grid points
% h = vector distances between nodes
% bc1 = 1 for Dirichlet, 2 for Neumann at left endpoint
% bc2 = 1 for Dirichlet, 2 for Neumann at right endpoint
% rhs = function handle for forcing term
```

```
% bv1 = boundary value for left endpoint
% bv2 = boundary value for right endpoint
%
% Outputs: A = centered FD matrix
% F = force vector
    e = ones(sx,1);
    A = spdiags([e e e], -1:1, sx, sx);

    for p = 1:sx - 1
        kt(p) = 0.5 * (k(p) + k(p+1));
    end

    for p = 2:sx-1
        A(p,p-1) = -kt(p-1) / h(p-1); %#ok<SPRIX>
        A(p,p) = kt(p-1)/ h(p-1) + kt(p) / h(p); %#ok<SPRIX>
        A(p,p+1) = -kt(p) / h(p); %#ok<SPRIX>
    end

    if bc1 == 1
        A(1,2) = 0;
        A(1,1) = 1;
    elseif bc1 == 2
        A(1,1) = (-3 * k(1)) / (2*h(1));
        A(1,2) = (2 * k(1)) / h(2);
        A(1,3) = (-1 * k(1)) / (2*h(3));
    end

    if bc2 == 1
        A(sx,sx-1) = 0;
        A(sx,sx) = 1;
    elseif bc2 == 2
        A(sx,sx-2) = k(sx) / (2 * h(sx-3));
        A(sx,sx-1) = (-2 * k(sx)) / h(sx-2);
```

```
        A(sx,sx) = (3 * k(sx)) / (2 * h(sx-1));
    end

    F = zeros(sx,1);
    F(1) = bv1;
    F(2) = bv2;

    for p = 2:sx-1
        F(p) = ((h(p) + h(p-1)) / 2) * rhs(x(p));
    end
end
```

```
% bpsolve2.m
function [U] = bpsolve2(A,F,sx,bv1,bv2)
%
% Solves the problem Ax = F, coming from a discretization
% of the problem -(k u')' = f,  with boundary data bv1, bv2.
% sx = grid size.
% bv1 = boundary data at left end point.
% bv2 = boundary data at right end point.
    F(1) = bv1;
    F(sx) = bv2;

    U = A\F;
end
```

```
% This script defines the parameters of the problem to be solved by ddp.m

% [a,b] is the domain, c is the interface
a = -1;
b = 1;
c = 0;
```

```
% Jd is the size of the jump discontinuity in function value at c,
% specifically Jd = U1(c) - U2(c)
Jd = 1;

% lambda0 = initial guess for condition at C for problem on
% Omega1
lambda = 1/2;

% -(k1 u')' = rhs1 on [a,c]
k1fun = @(x) 0*x + 1;
rhs1 = @(x) (2*x)/((x.^2+1).^2);

% -(k2 u')' = rhs2 on [c,b]
k2fun = @(x) 0*x + 1;
rhs2 = @(x) (2*x)/((x.^2+1).^2);

% alpha is the value of the boundary condition at a
% bc1 = 1 for Dirichlet, bc1 = 2 for Neumann
% beta is the value of the boundary condition at b
% bc2 = 1 for Dirichlet, bc2 = 2 for Neumann
alpha = 0;
bc1 = 1;
beta = .5;
bc2 = 1;
```

```
% This script defines the parameters of problem for ddc.m

% [a,b] is the domain, c is the interface
a = 0;
b = 1;
c = 0.5;
```

```
% lambda0 = initial guess for condition at C for problem on
% Omega1
lambda = 0;

% -k1u''(x) = rhs1 on [a,c]
k1fun = @(x) 0*x + 1;
rhs1 = @(x) (2*x)/((x.^2+1).^2);

% -k2u''(x) = rhs2 on [c,b]
k2fun = @(x) 0*x + 1;
rhs2 = @(x) (2*x)/((x.^2+1).^2);

% alpha is the value of the boundary condition at a
% bc1 = 1 for Dirichlet, bc1 = 2 for Neumann, bc1 = 3 for Robin
% beta is the value of the boundary condition at b
% bc2 = 1 for Dirichlet, bc2 = 2 for Neumann, bc2 = 3 for Robin
alpha = 0;
bc1 = 1;
beta = 1;
bc2 = 1;

% u1' = d2 u2 - d1 u1 at c
d2 = 1;
d1 = 1;
```