

DYNAMIC MODELS FOR METAL FLOWS

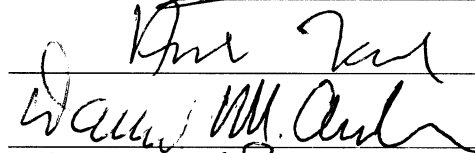
by

Marc Anthony Angulo
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Mathematics

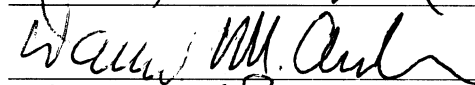
Committee:



Dr. Padmanabhan Seshaiyer, Thesis Director



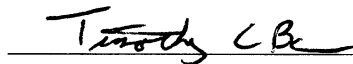
Dr. Harbir Lamba, Committee Member



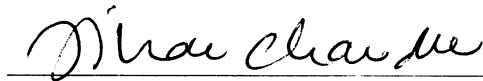
Dr. Daniel Anderson, Committee Member



Dr. Stephen H. Saperstone, Department Chairperson



Dr. Timothy Born, Associate
Dean for Academic and Student
Affairs, College of Science



Dr. Vikas Chandhoke, Dean,
College of Science

Date: 22 July 2011

Summer Semester 2011
George Mason University
Fairfax, VA

Dynamic Models for Metal Flows

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Marc Anthony Angulo
Bachelor of Science, Mathematics
University of Texas at Austin, 2006

Director: Dr. Padmanabhan Seshaiyer, Associate Professor
Department of Mathematical Sciences

Summer Semester 2011
George Mason University
Fairfax, VA

Copyright: 2011 by Marc Anthony Angulo
All Rights Reserved

Acknowledgements

I would like to extend my greatest gratitude to my advisor, Dr. Seshaiyer, for the countless hours devoted to the progression of my graduate studies, culminating with this thesis. His mentoring definitely shaped and cultivated my mathematical maturity. I also want to thank Dr. Lamba and Dr. Anderson for their time serving on my committee and providing much valuable approaches, insights, and critiques to the subject at hand. Dr. Lamba's insight into economic theory has instilled a deep passion for further research in the subject. Dr. Anderson's meticulous review and questioning of my thesis immensely strengthened the work. I am indebted to my colleagues at the U.S. Geological Survey, with special thanks to John Papp, for providing a sounding board for the theory of this work.

I owe much to my family and friends. The gift of their support is without measure: Mom, Kim, and Monique for their continuous words of encouragement; Jenny for her love, patience, and understanding; Estela and the "Furias" for their warmth and friendship; and Dorian, Christian, Nicholas, Lindsey, and Andy for always reminding me of the beauty and creativity of a child's mind.

Table of Contents

| | Page |
|---|------|
| List of Table..... | vi |
| List of Figures..... | vii |
| Abstract..... | ix |
| | |
| 1 Introduction | 1 |
| 2 Ordinary Differential Equation Models for Metal Flows | 4 |
| 2.1 PUE Model | 6 |
| 2.2 PUEP Model..... | 10 |
| 2.3 PUEP Model with Inflow Domains..... | 13 |
| 2.4 PUEP Model Incorporating Trade and Stocks | 17 |
| 2.5 PUEP Model with a Reformulation of the Virgin Metal Domain..... | 19 |
| 2.6 Model Modifications | 22 |
| 2.6.1 Ordinary Differential Equation PUE Model with Economic Equilibrium..... | 23 |
| 2.6.2 Ordinary Differential Equation PUEP Model with Hicksian Business Cycles | 26 |
| 2.6.3 Time Scale Analysis for the ODE PUEP Model | 27 |
| 2.7 Well-Posedness for the Dynamic Metal Flow Models..... | 32 |
| 2.6.1 Existence and Uniqueness | 33 |
| 2.6.2 Stability | 36 |
| 2.8 Sustainability Thresholds | 40 |
| 2.9 Parameter Optimization for the Dynamic Metal Flow Models | 43 |
| 2.9.1 Parameter Optimization for an Initial Value Problem..... | 44 |
| 2.9.2 Parameter Optimization for a System of Differential Equations | 47 |
| 2.9.3 Parameter Optimization for the ODE PUE Model | 49 |
| 2.9.4 Parameter Optimization for the ODE PUEP Model..... | 51 |
| 2.9.5 Parameter Optimization for the ODE PUEP with Inflow Domains Model..... | 53 |
| 2.9.6 Parameter Optimization for the ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 54 |
| 2.9.7 Parameter Optimization for the Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 56 |
| 3 Partial Differential Equation Models for Metal Flows | 58 |
| 3.1 PUE Models | 58 |
| 3.2 PUEP Models | 61 |

| | | |
|-------|---|-----|
| 3.3 | PUEP Models with Inflow Domains | 63 |
| 3.4 | PUEP Models Incorporating Trade and Stocks | 67 |
| 3.5 | PUEP Models with a Reformulation of the Virgin Metal Domain | 68 |
| 4 | Stochastic Differential Equation Models for Metal Flows | 71 |
| 4.1 | PUE Model | 72 |
| 4.2 | PUEP Model..... | 74 |
| 4.3 | PUEP Model with Inflow Domains..... | 76 |
| 4.4 | PUEP Model Incorporating Trade and Stocks | 79 |
| 4.5 | PUEP Model with a Reformulation of the Virgin Metal Domain..... | 81 |
| 5 | Numerical Analysis and Graphical User Interface | 85 |
| 5.1 | Numerical Methods | 85 |
| 5.1.1 | Numerical Methods for the Ordinary Differential Equation Models | 86 |
| 5.1.2 | Numerical Methods for the Partial Differential Equation Models | 93 |
| 5.1.3 | Numerical Methods for the Stochastic Differential Equation Models | 105 |
| 5.2 | Graphical User Interface | 112 |
| 6 | Conclusion and Future Work | 117 |
| A | Material Flow Analysis and Metal Flows in Industrial Ecology | 119 |
| A.1 | Short History of Material Flow Analysis | 119 |
| A.2 | Metal Flows and Metal Flow Models..... | 120 |
| B | Mathematical Modeling in Epidemiology | 124 |
| B.1 | A Short History of Mathematical Modeling in Epidemiology | 125 |
| B.2 | Selected Models | 126 |
| B.2.1 | Standard SIR Model | 127 |
| B.2.2 | SIRS Models: Immunity Temporarily Conferred | 131 |
| B.2.3 | Spatial Diffusion of Infectious Disease..... | 133 |
| C | Source Codes and Selected Outputs | 135 |
| C.1 | Stability for the ODE PUEP Model | 135 |
| C.2 | Stability for the ODE PUEP Model with Inflow Domains | 137 |
| C.3 | Stability for the ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 140 |
| C.4 | Stability for the Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 144 |
| C.5 | Parameter Optimization for the ODE PUE Model | 145 |
| C.6 | Euler's Method for the Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 149 |
| C.7 | BTCS Method for the Modified PDE PUEP Model with Inflow Domains, Stock and Trade Considerations and Price Cross Diffusion | 156 |
| C.8 | Euler-Maruyama Method for the SDE PUE Model | 169 |
| C.9 | Euler-Maruyama Method for the SDE PUEP Model..... | 173 |
| C.10 | Euler-Maruyama Method for the SDE PUEP Model with Inflow Domains..... | 178 |
| | Bibliography | 184 |

List of Tables

| Table | Page |
|--|------|
| 4.1 State Changes and Transition Probabilities for the SDE PUE Model | 73 |
| 4.2 State Changes and Transition Probabilities for the SDE PUEP Model | 75 |
| 4.3 State Changes and Transition Probabilities for the SDE PUEP Model with Inflow Domains | 77 |
| 4.4 State Changes and Transition Probabilities for the SDE PUEP Model with Inflow Domains and Stock and Trade Considerations | 79 |
| 4.5 State Changes and Transition Probabilities for the Modified SDE PUEP Model with Inflow Domains and Stock and Trade Considerations | 82 |

List of Figures

| Figure | Page |
|--|------|
| 2.1 The PUE System for Metal Flows | 7 |
| 2.2 The PUEP System for Metal Flows | 12 |
| 2.3 The PUEP with Inflow Domains System for Metal Flows..... | 15 |
| 2.4 The PUEP System with Inflow Domains and Stock and Trade Considerations ... | 20 |
| 2.5 Economic Equilibrium for Linear Supply and Demand Curves | 24 |
| 2.6 Two Domain PUEP System for Metal Flows | 29 |
| 2.7 Model Comparison for Aluminum in Infrastructure..... | 30 |
| 2.8 Model Comparison for Aluminum in Automobiles..... | 30 |
| 2.9 Model Comparison for Aluminum in Packaging..... | 31 |
| 2.10 Model Comparison for Aluminum in Infrastructure, with high loss rates..... | 31 |
| 5.1 Model Selection Screen | 114 |
| 5.2 Simulation for the ODE PUE Model | 114 |
| 5.3 Simulation for the Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations | 115 |
| 5.4 Simulation for the PDE PUEP Model..... | 116 |
| 5.5 Simulation for the SDE PUEP Model with Inflow Domains | 116 |
| A.1 Aluminum Metal Flow Model in 2000 | 122 |
| A.2 Metal Flows System..... | 123 |
| B.1 Standard SIR Models for Epidemics..... | 128 |
| C.1 Comparison of two solutions to the ODE PUE Model | 153 |
| C.2 Comparison of two solutions to the ODE PUEP Model..... | 153 |
| C.3 Comparison of two solutions to the ODE PUEP Model with Inflow Domains .. | 154 |
| C.4 Comparison of two solutions to the ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 154 |
| C.5 Comparison of two solutions to the ODE PUEP Model with Inflow Domains and Stock and Trade Considerations..... | 155 |
| C.6 Comparison of two solutions to the PDE PUE Model..... | 164 |
| C.7 Comparison of two solutions to the PDE PUE Model with Price Cross Diffusion | 164 |
| C.8 Comparison of two solutions to the PDE PUEP Model | 165 |
| C.9 Comparison of two solutions to the PDE PUEP Model with Price Cross Diffusion | 165 |
| C.10 Comparison of two solutions to the PDE PUEP Model with Inflow Domains.... | 166 |
| C.11 Comparison of two solutions to the PDE PUEP Model with Inflow Domains and Price Cross Diffusion | 166 |

| | |
|--|-----|
| C.12 Comparison of two solutions to the PDE PUEP Model with Inflow Domains and Trade and Stock Considerations..... | 167 |
| C.13 Comparison of two solutions to the PDE PUEP Model with Inflow Domains, Trade and Stock Considerations and Price Cross Diffusion..... | 167 |
| C.14 Comparison of two solutions to the Modified PDE PUEP Model with Inflow Domains and Trade and Stock Considerations | 168 |
| C.15 Comparison of two solutions to the Modified PDE PUEP Model with Inflow Domains, Trade and Stock Considerations and Price Cross Diffusion | 168 |
| C.16 Comparison of three trajectories for the SDE PUE Model | 172 |
| C.17 Comparison of three trajectories for the SDE PUEP Model..... | 177 |
| C.18 Comparison of three trajectories for the SDE PUEP Model with Inflow Domains | 183 |

Abstract

DYNAMIC MODELS FOR METAL FLOWS

Marc Anthony Angulo, MS

George Mason University, 2011

Thesis Director: Dr. Padmanabhan Seshaiyer

Metal flow analysis, a discipline in industrial ecology, describes the manner in which metals propagate throughout a coupled ecological and industrial system. The current models in industrial ecology are static representations that provide analysts with only a qualitative understanding of the dynamics at play. As such, many industry metrics are difficult to ascertain, such as recyclability and loss rates, future metal demand, and sustainability thresholds. The discipline can be expanded for its current state to a more quantitative framework where reliance on mathematical modeling is central. The intent of this thesis is to provide a fundamental framework for the mathematical modeling of metal flows.

A seamless transition is made from the static models to mathematical representations by incorporating dependence (independent or random variables) to the existing models and studying the rate at which metal flows throughout a system. Due to the composition of the static metal flow models (disjoint domains and adherence to

conservation of mass), compartmental modeling that has been used for over a century in epidemiology to study the spread of infectious diseases is utilized. Deterministic models consisting of ordinary differential equations are constructed and analyzed. Parameter optimization techniques using the Levenberg-Marquardt algorithm and numerical schemes for the acquisition of approximation solutions to the models are constructed and programmed in the Matlab language. In attempts to provide users with the ability to interface with the models—analyze and solve the models based on various parameters—standalone graphical user interfaces written in Matlab are constructed. To provide insight into the modification and improvement of the models, economic principles are introduced into the ordinary differential equation models, a pricing independent variable is included into the models producing systems of reaction-diffusion equations, and economic fluctuations and volatility are modeled using stochastic differential equations.

Chapter 1: Introduction

Material flow analysis (MFA) is defined as a systematic assessment of the flows and accumulation of materials throughout a defined system. MFA assesses the impact human activities, such as mining, processing, and producing goods, have on the natural environment and demonstrates when these activities are sustainable. The underlining principle of MFA is conservation of mass (mass balancing)—all inflows and outflows of material into and out of the system must be accounted—establishing the concept that no material can be created within the system’s domains. MFA has been used over the past few decades as a decision-making tool in environmental, resource, and waste management.

In industrial ecology, metal flow analysis, commonly referred to as “metal flows”, describes the flow of metals throughout an interconnected industrial and ecological system. The current metal flow models are static representations consisting of system descriptions coupled with empirical data (see Appendix A for a history and description of MFA and metal flows). Organizations that utilize these models may find it difficult to provide critical assessments of various metal metrics (such as recyclability rates, future metal demand, and sustainability threshold) through the use of the static models since the models do not provide the dynamics intrinsic to the system. Metal flow modeling can be improved from its current state as a qualitative tool for the analysis of metal flows, to a

more quantitative framework, where mathematical modeling is central. Analysis can be applied to the resulting mathematical models to obtain more accurate metal flow assessments and predictions. This thesis develops mathematical modeling techniques that can be applied to metal flows.

Due to the nature of the current static metal flow models, which consist of disjoint homogenous domain with adherence to conservation of mass, compartmental modeling that has been used for a century in epidemiology to describe the spread of infectious diseases provides a valuable framework for dynamic metal flow modeling. As such, a description of the mathematical models in epidemiology is provided in Appendix B. Epidemic thresholds, which are assumed to be analogous to metal sustainability thresholds in metal flows, are also highlighted in that section.

The development of dynamic metal flow models consisting of systems of ordinary differential equations (ODE), partial differential equations (PDE), and stochastic differential equations (SDE) will be the focus of Chapters 2 through 4. The modeling process will follow a sequential formulation: a metal flow system is defined; assumptions are made to describe the inflows and outflow of the system; differential equations are introduced to model the flows; and complexity is added to the metal flow system to generate subsequent models. The PDE and SDE models are built upon the ODE models by incorporating an independent variable for metal price (Chapter 3) or defining random variables (Chapter 4). Note that this thesis places an emphasis on the ODE models as opposed to the PDE or SDE counterparts. As such, research on the ODE models is

exhausted, whereas analysis and some applications for the PDE and SDE models are left as future work.

Existence, uniqueness, and stability of solutions, sustainability thresholds, and parameter optimizations for the ODE models are presented in Chapter 2. Parameter optimization for the ODE models will follow a Levenberg-Marquardt (LM) scheme that incorporates finite difference methods into the algorithm. Source code for the implementation of LM is presented in Appendix C.

The intent of Section 2.6 is to provide the reader with insight for the modification of the models based on various economic and mathematical interpretations. Since the inflow and outflow assumptions for the models are analogous to the transfer of an infectious disease within a population (perhaps a problematic approach for the analysis of economic systems), basic economic principles are incorporated into two of the models. Since scale analysis can play an important role in the simplification of mathematical equations, time-scale modifications to an ODE model is also presented in this section.

Analytical solutions to the models are difficult to obtain; therefore, reliance on numerical approximations for the analytical solutions will be paramount. Chapter 5 introduces three numerical schemes for the acquisition of numerical solutions to the models—the Euler, backward in time-central in space (BTCS), and Euler-Maruyama methods. Matlab scripts and functions are written to implement these methods (Appendix C). An overview of the standalone graphical user interfaces written in the Matlab language to provide users with interactive simulations are also presented in this chapter.

Chapter 2: Ordinary Differential Equation Models for Metal Flows

The static metal flow models established in industrial ecology coupled with U.S. Geological Survey data on metals (Appendix A) provide pertinent information on the past dynamic of the metals industry. From this information, past and present recyclability rates, metal availability, and industry efficiencies—valuable metrics for the understanding and improvement of the industry—can be ascertained. Static models and past system data, however, cannot provide a deterministic or probabilistic representation of the industry since they do not establish the governing dynamics underpinning the system, although correlations and dependencies can be understood through the application of statistical methods. Thus, accurate industry predictions, such as future metal availability, consumption, and recyclability rate, cannot be acquired through the use of these tools. The construction of dynamical models that describes the governing dynamics of the metal flow system can bridge this gap. A mathematically well-posed dynamical model for metal flows can, in theory, provide highly accurate predictions for a multitude of future conditions, such as efficiencies, consumption, metal sustainability, recyclability rates, etc. Therefore, it is advantageous for the development of such models.

This argument is not to diminish the importance or usefulness of the information provided by the USGS or the established static models. First and foremost, any dynamical model of metal flows should be build upon and adhere to the established static

models. The static models provide guidelines on the flow structure, sources and types of materials in the system, and system interactions. USGS data will provide a means for the optimization (Section 2.9) and testing of the dynamical models. In other words, any dynamical model of metal flows must be compliant on the a posteriori knowledge established by the static models and metal flow data sets.

This thesis establishes 23 dynamical metal flow models, each having varying purposes and layers of complexity. Three types of models are established, referred to as the *PUE Models*, *PUEP Models*, and *PUEP Models with Inflow Domains*. The *PUE Models* describe a quantity of metal that travels throughout a closed metal flow system. These models are established to offer intuition on the dynamics at work—supply, demand, and pricing—and provide a foundation for more elaborate modeling. These models, however, are not created to make predictions for or explain the nature of a particular metal industry. The *PUEP Models* describe a metal flow system that is not closed—the inclusion of metal losses and ore consumption—and contains the recycling of metals. These models are a simplistic representation of a metal flow industry and can be used to model metal industries on a global scale or a domestic industry not influence by foreign trade. The *PUEP Models with Inflow Domains* are elaborate models that describe intricate metal industries. Trade, ore consumption and production, recycle collection, and new scrap usage are incorporated into these models.

Due to the intrinsic nature of the metal flow system, compartmental modeling is utilized to develop the dynamical models. As seen in *Figure A.2*, metal quantities can be assigned to demarcated locations or domains, such as production manufacture, use, or

end-of-life. The data that is collected on the metal industry also establishes these distinctions. Conservation of mass is inherent to this framework: all metal must be accounted for throughout the life of that metal—metal cannot be created nor destroyed within the system’s domains. The dynamic metal flow models will assume that metal must reside in a specified disjoint domain (compartment) and conservation of mass must be adhered to during the transfer of metal from one domain to another. Since epidemiology has utilized compartment modeling for over a century, the mathematical formulations of epidemiological models may provide a useful guide.

2.1 PUE Model

To start with simplified models of metal flows, we will study the dynamics of a system containing three homogeneous disjoint domains: *Production*, *Use*, and *End-of-Life*. It is assumed that the domains illustrated in *Figure A.2* will be confined to these three classes. Although this minimalistic level of abstraction will not provide for a good representation of real-world observations for the flow of metals, these models will provide basic insights, notions, and intuition for more refined models.

The Production domain contains metal used in the construction of products, such as metal from ores (virgin material) or recyclates (new and old scrap). In the minerals industry, a distinction can be made as to the location in which ores resides; that is, Resource, Reserve Base, Reserve, etc. Each designation will have a quantity of ore associated to it, be it demonstrated or inferred, and will vary based on the source of information. Solely for the sake of simplicity at this stage, ore contained in the Production

domain will not adhere to specific resource or reserve definition, but will be viewed as a theoretical entity having a specific initial amount, referred to as the initial condition or initial density. In other words, the *PUE Model* will describe a closed system for the flow of metal, meaning it will demonstrate the dynamics of a certain amount of ore and recycle throughout the useable lifetime of that metal. Reserve and recycle distinctions will be incorporated at a later stage.

The Use domain contains metal residing in the consumer market, in the form of metal products. Consumer metal products, such as cars, building material, or electronics, will reside in the Use domain for a specific amount of time, dictated by the useable life cycle of the product or a consumer discard or replacement of the product. Once the metal is discarded or no longer in use, the metal will enter the End-of-Life domain. Conservation of mass is intrinsic to this formulation: metal cannot be created or destroyed in any domain and all metal must be accounted for throughout the life cycle. This system is illustrated in *Figure 2.1*.

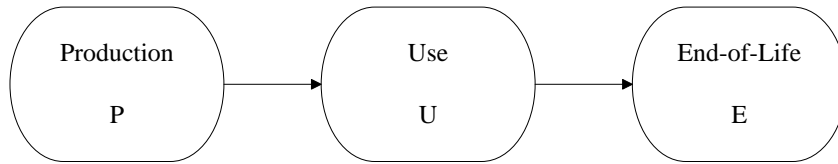


Figure 2.1: The PUE System for Metal Flows

Let each domain be dependent only on a temporal variable t , with $t \geq 0$. The following assumptions are made:

- i. The governing dynamics of metal flows is intrinsically linked to economic supply and demand. The density of the Production domain can provide a measure for supply and the density of the Use domain can be a measure of demand. The rate of change of the Production domain, that is the amount of ore and recyclates being consumed to produce products, is dictated by some function of supply and demand. Assume that this rate of change is proportional to the product of the supply and demand measures: that is, $-\mu PU$ where the constant μ will be referred to as the *supply-demand coefficient*. Likewise, the rate of change of the Use domain is equal to μPU . (This may appear to be a gross oversimplification of supply and demand for metal commodities. In fact, there are numerous determinants for supply and demand and each commodity possesses a different set of determinants that should be considered on an individual basis. Future refinement of this work should start with the enhancement of the supply-demand component).
- ii. All metal products have a finite life and will be discarded at some time t . Assume that the product lifetime is distributed exponentially; thus, the mean product lifetime is defined as $\int_0^\infty \exp(-et) dt = 1/e$, where \exp is the exponential function and constant $e > 0$. This implies that the rate of removal of metal from U is proportional to the amount of metal in U ; that is $-eU$ for constant $e > 0$. Thus, the rate at which metal enters E from U equals eU . The coefficient e represents the lifetime rate for metal products with the average product lifetime equaling $1/e$. In this context, e is referred to as the *end-of-life rate*. A particular metal is used in multiple products, such

as in automobiles, construction material, and electronics. As such, each product will have a different e assigned to it; likewise, each metal will have a different e .

As an illustration of these assumptions, take the production, consumption, and end-of-life of a particular metal used in the car industry as a case study. As a car is no longer operable, the metal from that car leaves the Use domain and enters the End-of-Life domain. The consumer will replace the old car with a new one, transferring metal from the Production domain into the Use domain. Depending on economic conditions at some time t , the consumer may decide to purchase multiple/fewer replacement cars or a larger/smaller vehicle requiring more/less metal. The supply-demand coefficient μ is used to capture that information. To mimic a real-world scenario, μ should be a function of time or a random variable. For simplicity at this stage, μ is assumed to be a constant and complexity is introduced at a later stage.

Defining the differentiable functions $P(t)$, $U(t)$, and $E(t)$ with respect to time $t > 0$ to describe the amount of metal in each domain at time t and combining the assumptions, the following system of ordinary differential equations is obtained (*ODE PUE Model*):

$$\begin{aligned}
\frac{dP(t)}{dt} &= -\mu P(t)U(t) & P(0) &= P_0, & P(t) &> 0 \\
\frac{dU(t)}{dt} &= \mu P(t)U(t) - eU(t) & U(0) &= U_0, & U(t) &> 0 \\
\frac{dE(t)}{dt} &= eU(t) & E(0) &= E_0, & E(t) &\geq 0
\end{aligned} \tag{2.1}$$

2.2 PUEP Model

The model discussed in the previous section described a closed metal flow system in which no metal was lost during production processes or new metal introduced in the form of virgin metal or recycle. The model was beneficial in two ways: it explains the dynamics of a discrete quantity of metal in the commodity economy that was based solely on supply-demand and product lifetime rates; and it provided a foundation for more elaborate modeling. The intent of this section is to expound on the model by incorporating dynamics inherent to metal flows: metal recycling, metal loss, and ore production. The model developed will be referred to as the *PUEP Model*.

In each domain of metal flows, a certain amount of metal can be lost. In the production of metal products, metal is lost in the form of shavings, trimmings, or cuttings. Metal can be lost from the Use and End-of-Life domains if metal products residing in these domains are discarded in such a manner that future recycling of the products cannot take place, as by discarding into a landfill. In *Figure A.2*, this metal lost is classified as *residues*. As a refinement to the *PUE Model*, we can introduce a parameter that captures this information. Let $l_P P(t)$, $l_U U(t)$, and $l_E E(t)$ be the quantity of metal lost at some time t from the Production, Use, and End-of-Life domains, respectively, where $l_P \in [0,1]$, $l_U \in [0,1]$, and $l_E \in [0,1]$. The fraction l_D will be referred to as the *metal lost rates* for domain D . These functions are analogous to death rates observed in epidemiological models.

An important consideration of metal flows is the recycling of metal. Theoretically, metal can be recycled infinitely many times, forming a cost effective and

reliable source of metal for the industry. As a metal product is discarded and enters the End-of-Life domain, it can reenter the Production domain if the metal is recycled. It follows that the rate of removal of metal from the End-of-Life domain is proportional to the amount of metal in that domain: i.e. $-rE(t)$, where r will be called the *recyclability rate*.

With the metal loss and recyclability rates defined for the domains, the End-of-Life domain takes on a different meaning from the one prescribed in the *PUE Model*. In the *PUE Model*, the End-of-Life domain is populated with all discarded metal. The End-of-Life domain in the *PUEP Model* can be viewed as a sort of “metals repository”. The domain contains discarded metal that can, at a future time, reenter the system if recycled. This domain is analogous to the removal domain in the SIRS model, in which individuals are temporarily immune to the infectious disease. This domain definition is advantageous in the construction of efficiency criteria. In order for a metal flow system to better utilize recycled metal, the collection of discarded metal, indicated by the End-of-Life domain’s density, must be maximized.

A “birth rate” term should to be added to the Production compartment to mimic the entry of metal through the mining of ore. In endemic models, the birth rate is in terms of the total population of the system, defined as $N = S(t_0) + I(t_0) + R(t_0)$ at time t_0 . In industrial ecology, the rate of entry of ore into the Production domain is governed by the available natural resources of the metal ore. At each time step, a certain fraction of the ore reserve can be economically extracted and will enter the Production domain. This availability can be modeled based on a coefficient, say a . Therefore we obtain the *ore*

rate term $aV(t)$, where $V(t)$ is a function of time representing the amount of ore reserve and a will be referred to as the *ore availability rate*. Later, the assumption will be made that V is a separate domain defined as a function of independent variables. *Figure 2.2* illustrates the assumptions discussed.

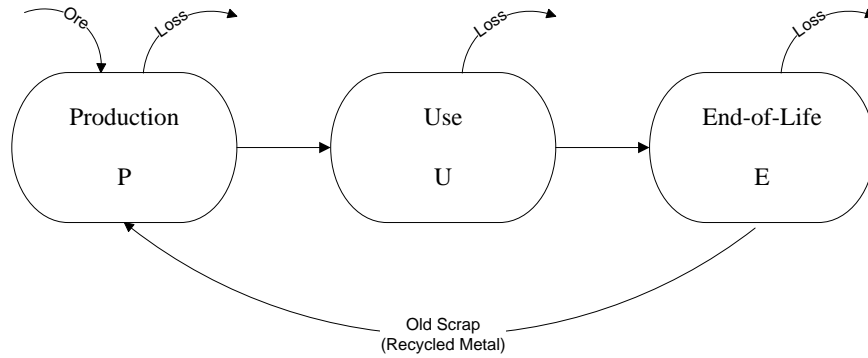


Figure 2.2: The PUEP System for Metal Flows

At this juncture, it is important to make a distinction between the two metal supplies present in the model: ore supply and refined metal supply. The ore supply is indicated by the function V , whereas the refined metal supply is indicated by the density of P . Since our model is governed by supply and demand, as in many economic systems, we must specify which supply is governing the system. For this work, the assumption is made that the refined metal supply drives the supply-demand dynamic, although the argument can be made to the contrary. A refinement of this thesis could consider reformulating the supply-demand dynamic based on ore supply.

Let us assume that the metal flow domains are functions with respect to time $t \geq 0$. Incorporating our metal loss, recyclability, and ore terms into the *PUE Model* (2.1), the *ODE PUEP Model* is obtained:

$$\begin{aligned}
\frac{dP(t)}{dt} &= aV(t) - \mu P(t)U(t) + rE(t) - l_P P(t) & P(0) &= P_0 \\
\frac{dU(t)}{dt} &= \mu P(t)U(t) - eU(t) - l_U U(t) & U(0) &= U_0 \\
\frac{dE(t)}{dt} &= eU(t) - rE(t) - l_E E(t) & E(0) &= E_0
\end{aligned} \tag{2.2}$$

2.3 PUEP Model with Inflow Domains

The *PUEP Model* introduced recycling, metal loss, and ore in order to establish a better representation for the flow of metal. A distinction can be made as to the type of recycled material entering the Production domain, which has major implications for the metal industry. The virgin metal entering the Production domain in the *PUEP Models* was described as a function of time, but in fact should be model as a separate domain since it changes with respect to time and pricing. It is also beneficial to model the inflow of virgin metal as a separate domain in attempts to construct sustainability thresholds. The purpose of this section is to incorporate *Inflow Domains* into the models to better represent metal flows.

The metal industry makes a distinction between the types of recycled metal. In this work, the two types of recycled material that will be explored are old scrap and new scrap. Old scrap is recycled metal that is obtained from discarded products. New scrap is

recycled material that is obtained from metal discarded during the manufacturing of products, in the form of trimmings, shavings, and cuttings. There is a major economic and industry difference between these two types of scrap. Old scrap is typically more energy, labor, and monetarily intensive to extract metal in comparison to new scrap. The extraction of metal from discarded products relies mainly on the products composition and recycling technologies. New scrap exists in a purer metal or alloy state and can be introduced back into the manufacture process more readily.

To capture this metal distinction, an incorporation of the inflow domains—Old Scrap *O* and New Scrap *N*—into the model is made. Metal composing the Old Scrap domain comes from recycled metal that exits the End-of-Life domain. This metal can be viewed as the collected scrap for the purposes of recycling and is referred to in industrial ecology as the *Scrap Supply Availability*. With the Old Scrap domain defined in this manner, the End-of-Life domain takes on a new meaning. The End-of-Life domain contains metal from products that reached their useful lifetime and is available for recycling. In industrial ecology, this domain is sometimes referred to as the *Recyclate Supply Availability*. The New Scrap domain contains metal flowing out of the Production domain and flowing back into Production.

The inflow of ore into the Production domain will be modeled using the *Virgin Metal* domain, which consists of metal ore used in the production of products. In this section, the Virgin Metal domain will consist of ore in metal reserves. However, a reformulation of this domain will be needed in order to mimic data sets provided by the

USGS, which is accomplished in Section 2.5. *Figure 2.3* illustrates the inclusion of these inflow domains.

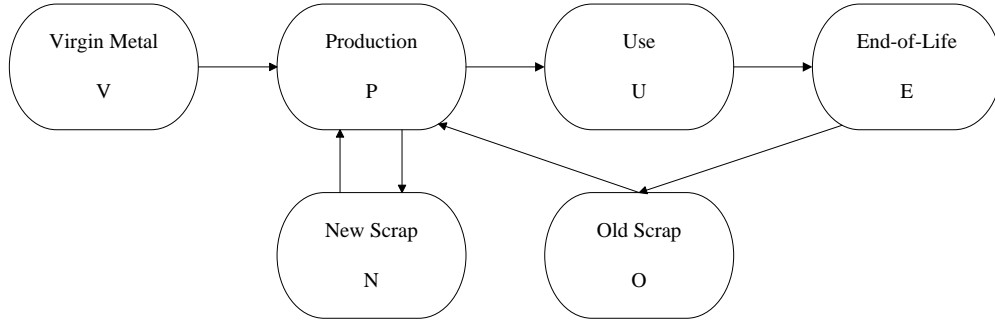


Figure 2.3: The PUEP with Inflow Domains System for Metal Flows

As seen in the previous model, the quantity of metal lost in a domain D can be model using a “death rate” term $l_D D(t)$, where $l_D \in [0,1]$. The metal loss in the Virgin Metal, Old Scrap, and New Scrap domains due to metal manufacturing or processing can be model using the *loss rate* terms $l_V V(t)$, $l_N N(t)$, and $l_O O(t)$, respectively, where $l_V \in [0,1]$, $l_N \in [0,1]$, and $l_O \in [0,1]$. The fraction l_D will be referred to as the *metal lost rates* for domain D .

The rate of metal removal from the Virgin Metal domain is dictated by the availability of ore (“Reserves”) that will feed the Production domain. Assuming that a certain fraction a of the Reserve is available to supply the Production domain, we can model the outflow of metal from the Virgin Metal domain into the Production domain by the term $aV(t)$.

The inflow of metal into the Production domain entering from the New Scrap and Old Scrap domains can be model with the terms $r_N N(t)$ and $r_O O(t)$, respectively. The outflow of metal from the Production domain entering the New Scrap domain can be captured by the term $r_P P(t)$. The fractions r_N , r_O , and r_P are referred to as the *recyclability rates* of old scrap, new scrap, and production. In practice, the metal industry will observe $r_P \ll r_O \ll r_N$.

Assume that the amount of metal in each domain is described by differentiable functions with respect to the temporal variable $t \geq 0$. Introducing the inflow domains into the *ODE PUEP Model* (2.2), the *ODE PUEP Model with Inflow Domains* is obtained:

$$\begin{aligned}
\frac{dV(t)}{dt} &= -aV(t) - l_V V(t) \\
\frac{dP(t)}{dt} &= aV(t) - \mu P(t)U(t) + r_O O(t) + r_N N(t) - r_P P(t) - l_P P(t) \\
\frac{dN(t)}{dt} &= r_P P(t) - r_N N(t) - l_N N(t) \\
\frac{dU(t)}{dt} &= \mu P(t)U(t) - eU(t) - l_U U(t) \\
\frac{dE(t)}{dt} &= eU(t) - r_E E(t) - l_E E(t) \\
\frac{dO(t)}{dt} &= r_E E(t) - r_O O(t) - l_O O(t)
\end{aligned} \tag{2.3}$$

$$P(0) = P_0, \quad U(0) = U_0, \quad E(0) = E_0$$

$$V(0) = V_0, \quad N(0) = N_0, \quad O(0) = O_0$$

2.4 PUEP Model with Trade and Stock Considerations

When studying the dynamics of metal flows occurring inside the boundaries of a country, *trade* (imports and exports of metals) and *stocks* (government and industry metal stockpiles) provide a major source of inflow and outflow of metals. For example, the U.S. cesium market is 100% reliant on imports to feed domestic consumption, since no cesium ore is mined in the United States [3], which implies that a cesium compartmental model must have Imports, Old Scrap, and New Scrap as inflows into the Production domain. The purpose of this section is to establish a model with trade and stock considerations. The model produced can be used to study the flow of metals for a particular county.

Importation and stock releases can be seen as metal inflows into the metal flow system. Metal is imported in three forms: ore, metal products, and old scrap. Ore that is mined in another country can be imported for processing or refinement, entering the Production domain. Metal products that are imported, such as cars, electronics, or building material, will enter the Use domain. Recyclable material collected in another country can be imported, which enters the Old Scrap domain. The total amount of metal imported can be expressed by a function of time $I(t)$. Let $i_P \in [0,1]$, $i_U \in [0,1]$, and $i_o \in [0,1]$ be the fractions of the total metal imported that enters the Production, Use, and Old Scrap domains, respectively, implying $i_P I(t) + i_U I(t) + i_o I(t) = I(t)$. The term i_D is the *import rate* for domain D . Metal stocks referred to stockpile releases that enter the Production domain. The quantity of metal being released from stockpiles can be modeled with the function of time $S(t)$ (see *Figure 2.4*).

Exports of metal can be viewed as a form of metal loss from the system. Similar to metal imports, metal exports are in the form of ore, finished and semi-manufactured metal products, and old scrap. Unlike the import terms used above, the export terms should be modeled based on the amount of metal residing in the domain in which the metal is being exported. The quantity of ore, metal products, and old scrap exports are limited by the amount of metal residing in the Virgin Metal, Production, Use, and End-of-Life domains. For example, ore exports cannot exceed the amount of metal in the Virgin Metal domain. A certain fraction x_V of the Virgin Metal domain will be exported in the form of ore. The fraction x_P of semi-fabricated metal product will leave the Production domain. The Use domain will export $x_U \in [0,1]$ of its metal in the form of finished metal products. The Old Scrap domain will export $x_O \in [0,1]$ of its metal. The fraction x_D will be referred to as the *export rate* for the D -domain.

Letting the differentiable functions $V(t)$, $P(t)$, $N(t)$, $U(t)$, $E(t)$, and $O(t)$ with respect to time $t \geq 0$ describe the amount of metal in each domain and incorporating trade and stock terms into the *ODE PUEP Model with Inflow Domains* (2.3), the *ODE PUEP Model with Inflow Domains and Stock and Trade Considerations* is obtained:

$$\begin{aligned}
\frac{dV(t)}{dt} &= -aV(t) - l_V V(t) - x_V V(t) \\
\frac{dP(t)}{dt} &= aV(t) - \mu P(t)U(t) + r_O O(t) + r_N N(t) - r_P P(t) - l_P P(t) - x_P P(t) \\
&\quad + i_P I(t) + S(t) \\
\frac{dN(t)}{dt} &= r_P P(t) - r_N N(t) - l_N N(t)
\end{aligned} \tag{2.4}$$

$$\frac{dU(t)}{dt} = \mu P(t)U(t) - eU(t) - l_U U(t) - x_U U(t) + i_U I(t)$$

$$\frac{dE(t)}{dt} = eU(t) - r_E E(t) - l_E E(t)$$

$$\frac{dO(t)}{dt} = r_E E(t) - r_O O(t) - l_O O(t) - x_O O(t) + i_O I(t)$$

$$P(0) = P_0, \quad U(0) = U_0, \quad E(0) = E_0$$

$$V(0) = V_0, \quad N(0) = N_0, \quad O(0) = O_0$$

2.5 PUEP Model with a Reformulation of the Virgin Metal Domain

In the previous inflow domain model, the Virgin Metal domain was defined to include Reserves of metal ore. In order to have a representation similar to the USGS materials flow, a distinction should be made between Reserves and ore production (also referred to as “production of primary metal”). In the process, two different entities are obtained: a Virgin Metal domain and a Reserve function.

In industrial ecology, a metal Reserve refers to the mineral resource that can be economically extracted at a given time t . Ore production refers to the extraction of minerals for the Reserve. The new Virgin Metal domain will consist of metals in ore production. Since the amount of metal residing in the Reserve domain changes with respect to time $t > 0$, the Reserve domain is modeled as the continuous function $R(t)$. The Virgin Metal changes with respect to the availability of ore, which is a fraction a of the metal Reserve $R(t)$, establishing the Virgin Metal rate of change as $aR(t)$ and the

removal rate of the Reserve as $-aR(t)$. The coefficient a will be referred to as the *ore availability* coefficient.

The supply-demand dynamic instituted in the previous model remains the same, since we are assuming that *refined metal supply* drives the dynamic. The inflow of metal entering the Production domain from the Virgin Metal domain is proportional to the amount of metal in the Virgin Metal domain. This term is given by $fV(t)$, where $f \in [0,1]$ will be referred to as the *ore processing rate*. Likewise, the rate at which metal leaving the Virgin Metal domain is equal to $-fV(t)$. Figure 2.4 represents the metal flow dynamic.

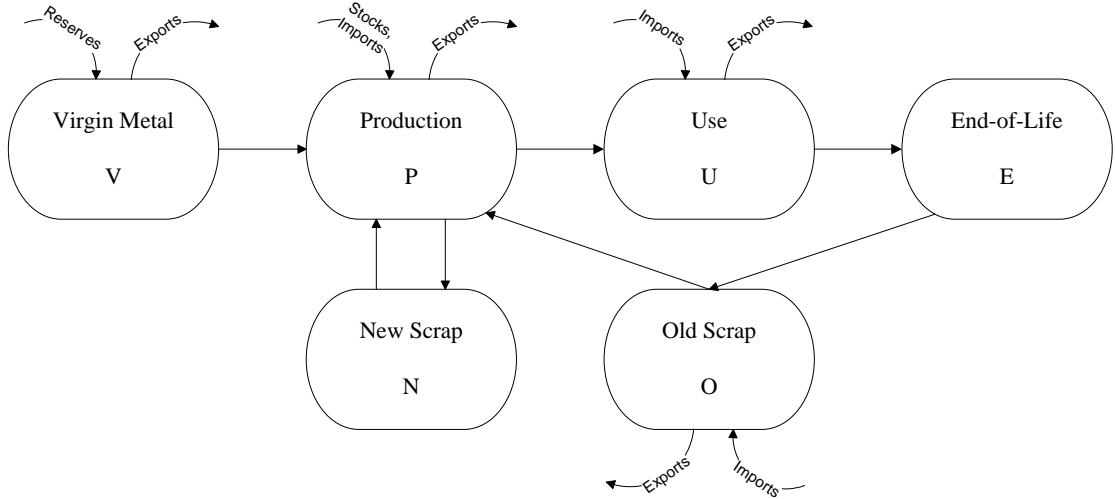


Figure 2.4: The PUEP System with Inflow Domains and Stock and Trade Considerations

Assume that the differentiable functions $V(t)$, $P(t)$, $N(t)$, $U(t)$, $E(t)$, and $O(t)$ with respect to t describe the amount of metal in the metal flow domains. If we modify

system (2.4) to capture the reformulation of the Virgin Metal domain, the following system of ordinary differential equations will be obtained (*Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations*):

$$\begin{aligned}
\frac{dV(t)}{dt} &= aR(t) - fV(t) - l_V V(t) - x_V V(t) \\
\frac{dP(t)}{dt} &= fV(t) - \mu P(t)U(t) + r_O O(t) + r_N N(t) - r_P P(t) - l_P P(t) - x_P P(t) \\
&\quad + i_P I(t) + S(t) \\
\frac{dN(t)}{dt} &= r_P P(t) - r_N N(t) - l_N N(t) \\
\frac{dU(t)}{dt} &= \mu P(t)U(t) - eU(t) - l_U U(t) - x_U U(t) + i_U I(t) \\
\frac{dE(t)}{dt} &= eU(t) - r_E E(t) - l_E E(t) \\
\frac{dO(t)}{dt} &= r_E E(t) - r_O O(t) - l_O O(t) - x_O O(t) + i_O I(t)
\end{aligned} \tag{2.5}$$

$$\begin{aligned}
P(0) &= P_0, \quad U(0) = U_0, \quad E(0) = E_0 \\
V(0) &= V_0, \quad N(0) = N_0, \quad O(0) = O_0
\end{aligned}$$

The metal flow models developed throughout this chapter were formulated in a systematic manner by specifying the metal flow domains and defining the system of ordinary differential equations that describe the flow of metals throughout the system, with complexity being added to the previous model. This formulation style lends well in the construction phase of mathematical modeling. Many of the models, however, can be generalized, meaning one model can produce other models through the manipulation of the coefficients.

The ordinary differential equation models constructed in Sections 2.1 and 2.2 can be generalized by the *ODE PUEP Model* of Section 2.2. To obtain the *ODE PUE Model*, the coefficients a , r , l_p , l_U , and l_E should be set to 0. Likewise, the models constructed in Section 2.3 through 2.5 can be obtained from the *Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations* by setting relevant coefficient equal to 0.

2.6 Model Modifications

The metal flow models discussed throughout this chapter were constructed based on numerous assumptions, many of which were analogous to ideas appearing in population dynamics. An economist studying this work can make the argument that the logic behind these assumptions does not conform to realistic economic governance. Although this argument is legitimate, the major tenant of this thesis is not to develop standalone economic models for metal flows, but merely to provide a foundation for more sophisticated and elaborate modeling. In other words, the ideas and techniques discussed throughout can be used as tools by experts in industrial ecology for advanced, multifaceted models. As such, it will be the purpose of this section to demonstrate the manner in which modifications can be made to the metal flow models that would better adhere to economic principles. The first section will incorporate the principle of economic equilibrium into the *ODE PUE Model*. In the second section, Hicksian business cycle theory is incorporated into the *ODE PUEP Model*. Since the time scales associated with the Production domain is very small in comparison to the scales associated with the

Use domain, the third section will introduce simplifications to the *ODE PUEP Model* using time-scale analysis.

2.6.1 Ordinary Differential Equation PUE Model with Economic Equilibrium

Recall that the *ODE PUE* and *ODE PUEP Models* describe the flow of metal for systems consisting of 3 disjoint, homogenous domains (see *Figures 2.1* and *2.2*). The assumption was made that the flow of metal from the Production domain into the Use domain at time t was proportional to the product of the density of the Production domain and the density of the Use domain for time t : $\mu P(t)U(t)$. This assumption stemmed from the notion that supply (measured by the Production density) and demand (measured by the Use density) governed the transfer of metal. This argument, however, is analogous to the governing dynamic of the SIR model in epidemic compartmental modeling and might not provide a true economic representation. To alleviate this concern, economic principles can be incorporated into the system. In this section, we will assume that the transfer of metal from the Production domain into the Use domain is dictated by the demand of metal products by the consumer market. Thus, we will set out to find a demand function, $D(t)$, that will replace the supply/demand term $\mu P(t)U(t)$.

The principle of economic equilibrium for supply and demand, as discussed in [17], will be used to determine the demand function. Let s represent the price of metal, $D(s)$ represents the demand for metal products at price s , and $P(s)$ represents the supply of metal products at price s . The following assumptions are established:

$D(m_1) = n_1$: Metal consumers will demand all available metal (n_1) priced at $s = m_1$

$D(m_2) = 0$: Metal consumers will not demand metal priced at $s = m_2$

$P(m_3) = 0$: Metal suppliers will not supply metal priced at $s = m_3$

$P(m_4) = n_2$: Metal suppliers will supply all available metal (n_2) priced at $s = m_4$

Note that $m_1 < m_2$, $m_3 < m_4$, $n_1 > 0$, and $n_2 > 0$. Assume that the economic system is governed by a linear system:

$$\begin{aligned} D(s) &= \frac{-n_1}{m_2 - m_1} s + \frac{m_1 n_1}{m_2 - m_1} + n_1 \\ P(s) &= \frac{n_2}{m_4 - m_3} s - \frac{m_3 n_2}{m_4 - m_3} \end{aligned} \tag{2.6}$$

Figure 2.5 depicts the supply-demand curves for this economic system.

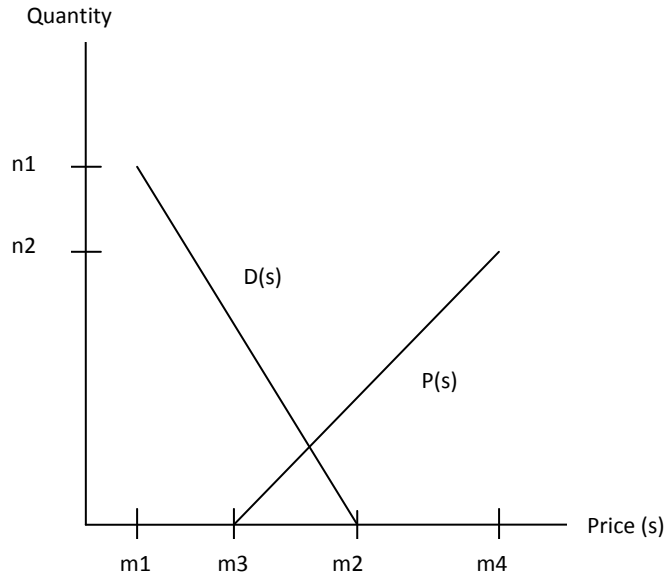


Figure 2.5: Economic Equilibrium for Linear Supply and Demand Curves

As in the *ODE PUE Model* of Section 2.1, the assumption is made that the Production domain comprises all metal that will be used in the construction of metal products, which consists of metal ore, recycled metal, and refined metal products. The assumption is made that the amount of refined metal products for time t , measured by $\mu P(t)$ with $\mu \in [0,1]$, represents the supply of the system that is at economic equilibrium—supply and demand are equal. Let $s_e(t)$ represent the equilibrium price at time t . From (2.6), we will obtain

$$\begin{aligned}\mu P(t) &= \frac{n_2}{m_4 - m_3} s_e(t) - \frac{m_3 n_2}{m_4 - m_3} \\ \Rightarrow s_e(t) &= \frac{m_4 - m_3}{n_2} \mu P(t) + m_3\end{aligned}\quad (2.7)$$

Therefore, demand at equilibrium is defined as

$$D(t) = \frac{-n_1}{m_2 - m_1} \left(\frac{m_4 - m_3}{n_2} \mu P(t) + m_3 \right) + \frac{m_1 n_1}{m_2 - m_1} + n_1 \quad (2.8)$$

Incorporating the demand function (2.8) into the *ODE PUE Model* (2.1), the *ODE PUE Model with Economic Equilibrium* is obtained:

$$\begin{aligned}\frac{dP(t)}{dt} &= \frac{n_1}{m_2 - m_1} \left(\frac{m_4 - m_3}{n_2} \mu P(t) + m_3 \right) - \frac{m_1 n_1}{m_2 - m_1} - n_1 \\ \frac{dU(t)}{dt} &= \frac{-n_1}{m_2 - m_1} \left(\frac{m_4 - m_3}{n_2} \mu P(t) + m_3 \right) + \frac{m_1 n_1}{m_2 - m_1} + n_1 - eU(t) \\ \frac{dE(t)}{dt} &= eU(t)\end{aligned}\quad (2.9)$$

where $m_1 < m_2$, $m_3 < m_4$, $m_3 \leq m_2$, $n_1 > 0$, $n_2 > 0$, $P(0) = P_0$, $U(0) = U_0$, $E(0) = E_0$, $P(t) > 0$, $U(t) > 0$, and $E(t) \geq 0$.

In order for the system to be well define, that is if the domains remain non-negative for all time t , the following inequality must hold:

$$\frac{-n_1}{m_2 - m_1} \left(\frac{m_4 - m_3}{n_2} \mu P(t) + m_3 \right) + \frac{m_1 n_1}{m_2 - m_1} + n_1 \leq P(t) \quad (2.10)$$

implying that the following condition must hold:

$$(m_2 - m_3) \left(\mu \frac{m_4 - m_3}{n_2} + \frac{m_2 - m_1}{n_1} \right)^{-1} \leq P(t) \quad (2.11)$$

2.6.2 Ordinary Differential Equation PUEP Model with Hicksian Business Cycles

The *ODE PUEP Model* of Section 2.2 contained three metal inflows that can be assumed to be governed by different types of demand: the inclusion of metal into the Production domain in the form of ore metal indicated by the term $aV(t)$; the inclusion of metal into the Use domain from the Production domain dictated by $\mu P(t)U(t)$; and the inclusion of metal into the Production domain from the End-of-Life domain indicated by $rE(t)$. To illustrate the manner to which these terms can be modeled using economic principles, a Hicksian business cycle formulation, as discussed in [33], is developed for these terms, thus establishing a discrete-time system for the flow of metals.

It is assumed that the demand dynamics of the *ODE PUEP Model* can be measured by the consumption of metal. As in a Hicksian formulation, we will assume that metal consumption C at time t is proportional to income Y at time $t - 1$: $C_t = cY_{t-1}$. Since we are dealing with three different economic markets (mineral commodity, consumer, and metal scrap markets), we must make a distinction between the types of consumptions and incomes. Consumption of ore metal by the Production domain (metal

product manufacturers) at time t , indicated by C_t^P , is assumed to be proportional to the income of the Production domain Y_{t-1}^P at time $t - 1$: $C_t^P = c^P Y_{t-1}^P$. Likewise, the consumption of metal products by the Use domain (consumer market) C_t^U will equal $c^U Y_{t-1}^U$ and the consumption of scrap by the Production domain is defined as $C_t^E = c^E Y_{t-1}^P$. If we assume that the coefficients change over time, then we have consumption equations with non-constant coefficients: $C_t^P = c_t^P Y_{t-1}^P$, $C_t^U = c_t^U Y_{t-1}^U$, and $C_t^E = c_t^E Y_{t-1}^P$.

If we incorporate the consumption equations into the *ODE PUEP Model* by establishing the rate of change of the domains as difference equations (since the consumptions are defined as discrete time maps), we will obtain the *ODE PUEP Model with Hicksian Business Cycles*:

$$\begin{aligned}
P_t &= P_{t-1} + \Delta t (c_t^P Y_{t-1}^P - c_t^U Y_{t-1}^U + c_t^E Y_{t-1}^P - l_P P_{t-1}) \\
U_t &= U_{t-1} + \Delta t (c_t^U Y_{t-1}^U - e U_{t-1} - l_P U_{t-1}) \\
E_t &= E_{t-1} + \Delta t (e U_{t-1} - c_t^E Y_{t-1}^P - l_E E_{t-1})
\end{aligned} \tag{2.12}$$

2.6.3 Time Scale Analysis for the ODE PUEP Model

Scale analysis can be used to simplify equations where the temporal or spatial scales for various terms are different by orders of magnitude [8]. For a specific metal in metal flow analysis, the temporal scales between the Production and the Use domains can differ by orders of magnitude. For example, the lifetime rate of aluminum in infrastructure, which is given by $\frac{1}{e}$ in the *ODE PUEP Model*, is of the order of 100 years, whereas the product manufacture time, given by $\frac{1}{\mu}$, is of the order of weeks. A simplified

version of the *ODE PUEP Model* could involve eliminating the Production domain by including it into the Use domain. This section will present a time-scale modification for the *ODE PUEP Model* and discuss the benefits and limitations for this approach.

In order to eliminate the product manufacture component of the *ODE PUEP Model*, the assumption will be made that the metal flow system consists of two discrete homogeneous domain—Use and End-of-Life—as depicted in *Figure 2.6*. The Use domain will consist of processed and refined ore, metal in the manufacturing of consumer products, and products in the consumer market. The Production domain that was constructed for the *ODE PUEP Model* will reside in the Use domain. As such, the transfer of metal between the Production and Use domains will be eliminated. The End-of-Life domain consists of metal that has been discarded after it reaches its useable lifetime. Metal losses will be endured at the product manufacturing phase, given by $l_p(aV(t))$, in the consumer market, $l_U U(t)$, and in discarded products, $l_E E(t)$. Ore material will enter the Use domain, given by the term $aV(t)$. Old scrap will reenter the Use domain from the End-of-Life domain, which is captured by the term $rE(t)$. Collecting these terms, the *Two Domain ODE PUEP Model* (2.13) will be obtained.

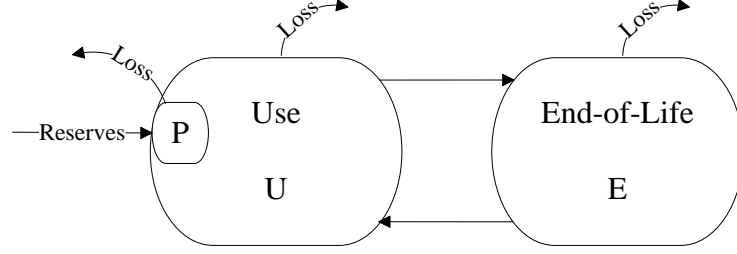


Figure 2.6: Two Domain PUEP System for Metal Flows

$$\begin{aligned}
 \frac{dU(t)}{dt} &= aV(t)(1 - l_p) - eU(t) - l_U U(t) + rE(t) & U(0) &= U_0 \\
 \frac{dE(t)}{dt} &= eU(t) - rE(t) - l_E E(t) & E(0) &= E_0
 \end{aligned} \tag{2.13}$$

The *Two Domain ODE PUEP Model* (2.13) approximates the *ODE PUEP Model* rather closely for metals systems with small loss rates, small product manufacture rate, and long product lifetimes (see *Figures 2.7* and *2.8*). The simplified model, however, does not accurately approximate the *ODE PUEP Model* where loss rates are high or the product manufacture and product lifetime rates are similar (see *Figure 2.9* and *2.10*).

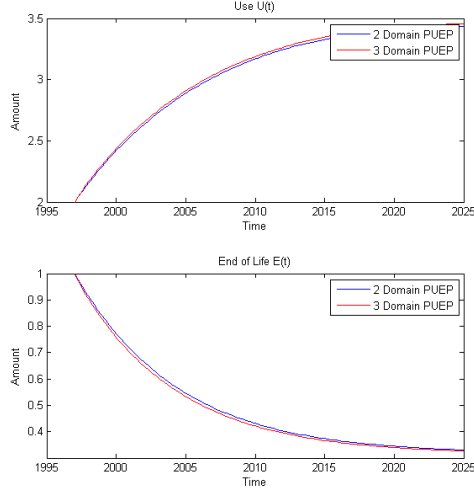


Figure 2.7: Model Comparison for Aluminum in Infrastructure: *Two Domain ODE PUEP Model* vs. the *ODE PUEP Model*. The following parameters were considered: $P(0) = 1, U(0) = 1, E(0) = 1, V(0) = 10, l_p = l_U = l_E = .01, \mu = 0.5, a = 0.01, r = 0.1, e = 0.01$. The choice of parameters may describe aluminum metal used in infrastructure.

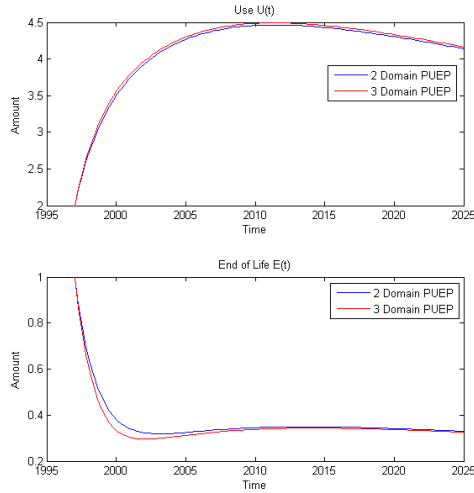


Figure 2.8: Model Comparison for Aluminum in Automobiles: *Two Domain ODE PUEP Model* vs. the *ODE PUEP Model*. The following parameters were considered: $P(0) = 1, U(0) = 1, E(0) = 1, V(0) = 10, l_p = l_U = l_E = .01, \mu = 0.5, a = 0.04, r = 0.5, e = 0.04$. The choice of parameters may describe aluminum metal used in automobiles.

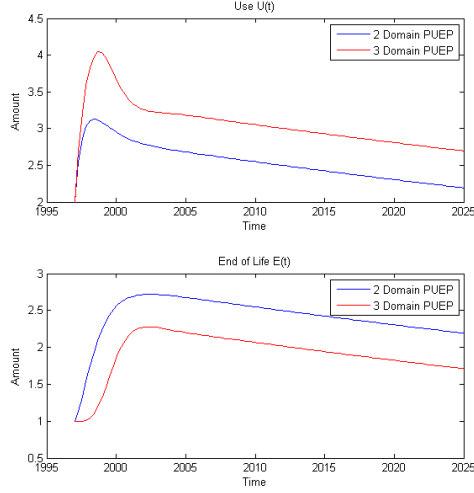


Figure 2.9: Model Comparison for Aluminum in Packaging: *Two Domain ODE PUEP Model* vs. the *ODE PUEP Model*. The following parameters were considered: $P(0) = 1, U(0) = 1, E(0) = 1, V(0) = 10, l_p = l_U = l_E = 0.01, \mu = 0.5, a = 0.25, r = 0.5, e = 1$. The choice of parameters may describe aluminum metal used in packaging.

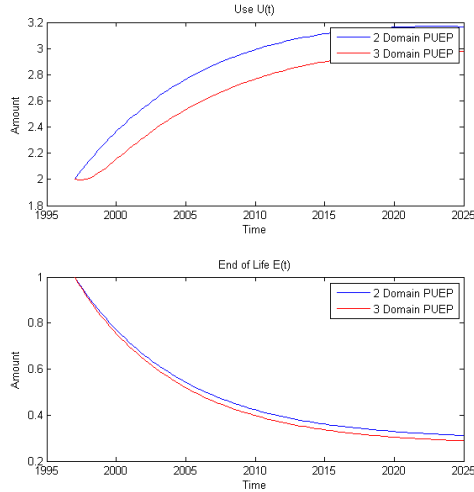


Figure 2.10: Model Comparison for Aluminum in Infrastructure, with high loss rates: *Two Domain ODE PUEP Model* vs. the *ODE PUEP Model*. The following parameters were considered: $P(0) = 1, U(0) = 1, E(0) = 1, V(0) = 10, l_p = 0.2, l_U = l_E = 0.01, \mu = 0.5, a = 0.01, r = 0.1, e = 0.01$. The choice of parameters may describe aluminum metal used in infrastructure, with large loss rates.

The time scale simplification is beneficial when the entire dynamic of the metal flow system is required, as opposed to the dynamics of individual domains. The simplification is advantageous for analysis (it is easier to analyze a smaller system of equations) and in computation (solving smaller systems reduces computational costs).

The simplification may hinder a few elements inherent to the original models. First, the dynamics for particular domains will be hard to acquire since domains are contained in other domains. For example, the dynamics of the Production domain as defined in the *ODE PUEP Model* will be impossible to obtain using the *Two Domain ODE PUEP Model* since the interaction between the Production and Use domains was eliminated. Second, sustainability thresholds may be hard to obtain depending of the simplification. Take, for example, a simplification of the inflow domain models of Sections 2.3-2.5 by eliminating the Virgin Metal domain. Calculating sustainability by setting $\frac{dV(t)}{dt} = 0$ will be impossible.

2.7 Well-Posedness for the Dynamic Metal Flow Models

Existence, uniqueness, and stability are fundamental elements in the theoretical study of differential equations [14] and are the properties required for a mathematical model of a physical phenomenon to be considered “well-posed” [19]. As such, the models developed in this chapter must demonstrate these properties in order for the models to be considered reliable mathematical representation of metal flows. In the first section, existence and uniqueness of a solution to the models is demonstrated through the use of an existence and uniqueness theorem discussed in [2] and proven in [11]. The

second section demonstrates the criteria required to guarantee stable solutions for the models.

2.7.1 Existence and Uniqueness

There are several existence and uniqueness theorems for initial value problems which can be extended to systems of ordinary differential equations [14]. Theorem 2.1 can be used to prove existence and uniqueness for the nonlinear metal flow models.

Theorem 2.1 Existence and Uniqueness of a Solution for a System of Ordinary

Differential Equations: *Consider the following system of ordinary differential equations*

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t))$$

where

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{pmatrix}$$

and

$$\mathbf{f}(t) = \begin{pmatrix} f_1(k, t, x_1, x_2, \dots, x_N) \\ f_2(k, t, x_1, x_2, \dots, x_N) \\ \vdots \\ f_M(k, t, x_1, x_2, \dots, x_N) \end{pmatrix}$$

Assume that \mathbf{f} and its first partial derivatives with respect to \mathbf{x} are continuous on an open set U . Then for any real number t_0 and real vector \mathbf{x}_0 there is an open interval containing t_0 , on which there exists a solution satisfying the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$, and the solution is unique.

In the development of the metal flow models in Section 2.1-2.5, the functions $V(t)$, $P(t)$, $N(t)$, $U(t)$, $E(t)$, and $O(t)$ were assumed to be differentiable function with respect to time t on the open set $(0, \infty)$, implying continuity. We must demonstrate that the first partial derivatives of \mathbf{f} with respect to $\mathbf{x} = (V, P, N, U, E, O)^T$ are also continuous in $(0, \infty)$ in order to satisfy Theorem 2.1, thus proving existence and uniqueness of a solution for the models.

Since system (2.1) can be obtained from system (2.2) by setting the coefficients a , r , l_P , l_U , and l_E equal to 0, only the partial derivatives of \mathbf{f} with respect to \mathbf{x} of system (2.2) are analyzed. Likewise, only the partial derivatives of (2.5) with respect to \mathbf{x} are analyzed since systems (2.3) and (2.4) can be obtained from (2.5).

For the *ODE PUEP Model* in Section 2.2, $\mathbf{f}(t) = \begin{pmatrix} aV - \mu PU + rE - l_P P \\ \mu PU - eU - l_U U \\ eU - rE - l_E E \end{pmatrix}$ where

a , μ , r , l_P , l_U , and l_E are constants and V , P , U , and E are function with respect to time, implying that

$$\begin{aligned} \frac{\partial \mathbf{f}(t)}{\partial V} &= \begin{pmatrix} a \\ 0 \\ 0 \end{pmatrix}, & \frac{\partial \mathbf{f}(t)}{\partial P} &= \begin{pmatrix} -\mu U - l_P \\ \mu U \\ 0 \end{pmatrix}, \\ \frac{\partial \mathbf{f}(t)}{\partial U} &= \begin{pmatrix} -\mu P \\ \mu P - e - l_U \\ e \end{pmatrix}, & \frac{\partial \mathbf{f}(t)}{\partial E} &= \begin{pmatrix} r \\ 0 \\ -r - l_E \end{pmatrix} \end{aligned} \quad (2.14)$$

Since P and U are continuous function and a , μ , e , r , l_P , l_U , and l_E are constants, then $\frac{\partial \mathbf{f}(t)}{\partial \mathbf{x}}$ is continuous on $(0, \infty)$ by (2.14), thus implying that there exists a unique solution for the three-domain ordinary differential equation models.

For the *Modified ODE PUEP Model with Inflow Domains and Stock and Trade*

Considerations, \mathbf{f} is defined as

$$\mathbf{f}(t) = \begin{pmatrix} aR - fV - l_V V - x_V V \\ fV - \mu PU + r_O O + r_N N - r_P P - l_P P - x_P P + i_P I + S \\ r_P P - r_N N - l_N N \\ \mu PU - eU - l_U U - x_U U + i_U I \\ eU - r_E E - l_E E \\ r_E E - r_O O - l_O O - x_O O + i_O I \end{pmatrix} \quad (2.15)$$

which follows that

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial V} &= \begin{pmatrix} -f - l_V - x_V \\ f \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, & \frac{\partial \mathbf{f}}{\partial P} &= \begin{pmatrix} 0 \\ -\mu U - r_P - l_P - x_P \\ r_P \\ \mu U \\ 0 \\ 0 \end{pmatrix}, \\ \frac{\partial \mathbf{f}}{\partial N} &= \begin{pmatrix} 0 \\ r_N \\ -r_N - l_N \\ 0 \\ 0 \\ 0 \end{pmatrix}, & \frac{\partial \mathbf{f}}{\partial U} &= \begin{pmatrix} 0 \\ -\mu P \\ 0 \\ \mu P - e - l_U - x_U \\ e \\ 0 \end{pmatrix}, \\ \frac{\partial \mathbf{f}}{\partial E} &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -r_E - l_E \\ r_E \end{pmatrix}, & \frac{\partial \mathbf{f}}{\partial O} &= \begin{pmatrix} 0 \\ r_O \\ 0 \\ 0 \\ 0 \\ -r_O - l_O - x_O \end{pmatrix} \end{aligned} \quad (2.16)$$

Since V, P, N, U, E , and O are defined as continuous function and $f, a, \mu, e, r_N, r_P, r_E, x_V,$

$x_P, x_U, x_O, l_V, l_P, l_N, l_U, l_E$, and l_O are constants, then $\frac{\partial \mathbf{f}(t)}{\partial x}$ is continuous on $(0, \infty)$ by (2.16),

thus implying the existence of a unique solution for the six-domain ordinary differential equation models.

Equations (2.14) and (2.16) demonstrates that in order for a unique solution to exist for the dynamic ordinary differential equation metal flow models, the functions representing the amount of metal in each metal domain must be continuous, with the system's coefficients defined as constants (note, existence and uniqueness will still hold if the coefficients were defined as continuous functions).

2.7.2 Stability

The stability for a system of ordinary differential equations can be analyzed by observing the behavior when perturbing the fixed points (equilibria or steady states) of the system. The eigenvalues of the Jacobian of the system evaluated at the fixed points can demonstrate this behavior, thus establishing stability criteria for the model [2]. In other words, the stability of the systems can be analyzed by taking a local linear approximation in the neighborhood of an isolated fixed point. This procedure is referred as an eigenvalue stability method [28]. If the eigenvalues do not have real parts equal to 0, then the fixed points are called hyperbolic and the eigenvalues will describe the stability of the system at the fixed point. If the real part of any of the eigenvalues is 0, then higher order terms must be taken to determine the stability. If the real part of each of the eigenvalues is strictly negative, then the system is asymptotically stable at the fixed point. If the real part of any of the eigenvalues is nonnegative, then the system is considered unstable at that particular fixed point.

In this section, a stability criterion is determined for systems (2.2) through (2.5). For systems (2.2) to (2.3), the fixed points and Jacobians for systems are explicitly

determined and Matlab scripts are written to determine the eigenvalues of the Jacobian evaluated at the fixed points. Explicit forms of the fixed points for system (2.4) and (2.5) are difficult to obtain; therefore, Matlab programs were written to find the fixed points and describe the stability at those points.

The fixed points for the *ODE PUE Model* (2.1) can be found by solving the following system of equations

$$\begin{aligned} -\mu P(t)U(t) &= 0 \\ \mu P(t)U(t) - eU(t) &= 0 \\ eU(t) &= 0 \end{aligned} \tag{2.17}$$

Thus, the fixed points are $(P^*, U^*, E^*) = \left(\frac{e}{\mu}, 0, E\right)$ where $E > 0$. Since we have the condition $U > 0$, there are no fixed points in the defined domain.

By setting $\frac{dP(t)}{dt} = 0$, $\frac{dU(t)}{dt} = 0$, and $\frac{dE(t)}{dt} = 0$, the fixed points of the *ODE PUEP Model* are obtained:

$$\begin{aligned} (P^*, U^*, E^*) &= \left(\frac{e}{\mu} + \frac{l_U}{\mu}, \left(aV - \frac{l_P}{\mu}(e + l_U)\right)\left(e + l_U - \frac{re}{r + l_E}\right)^{-1}, \right. \\ &\quad \left. \frac{e}{r + l_E} \left(aV - \frac{l_P}{\mu}(e + l_U)\right)\left(e + l_U - \frac{re}{r + l_E}\right)^{-1}\right) \end{aligned} \tag{2.18}$$

Note that $P^* > 0$, $U^* > 0$, and $E^* \geq 0$ since the system is only defined for positive metal quantities. The Jacobian for (2.2) evaluated at the fixed points is

$$\mathbf{J}(P^*, U^*, E^*) = \begin{pmatrix} \frac{\partial \dot{P}}{\partial P} & \frac{\partial \dot{P}}{\partial U} & \frac{\partial \dot{P}}{\partial E} \\ \frac{\partial \dot{U}}{\partial P} & \frac{\partial \dot{U}}{\partial U} & \frac{\partial \dot{U}}{\partial E} \\ \frac{\partial \dot{E}}{\partial P} & \frac{\partial \dot{E}}{\partial U} & \frac{\partial \dot{E}}{\partial E} \end{pmatrix} = \begin{pmatrix} -\mu U^* - l_P & -\mu P^* & r \\ \mu U^* & \mu P^* - e - l_U & 0 \\ 0 & e & -r - l_E \end{pmatrix} \quad (2.19)$$

The eigenvalues λ for $\mathbf{J}(P^*, U^*, E^*)$ can be found by solving

$$\det(\mathbf{J}(P^*, U^*, E^*) - \lambda I) = 0 \quad (2.20)$$

Thus, the eigenvalues are the roots of

$$\begin{aligned} & (-\mu U^* - \lambda)(\mu P^* - e - l_U - \lambda)(-r - l_E - \lambda) - (-\mu P^*)(\mu U^*)(-r - l_E - \lambda) = \\ & \left(-\mu \left(aV - \frac{l_P}{\mu}(e + l_U) \right) \left(e + l_U - \frac{re}{r + l_E} \right)^{-1} - \lambda \right) (-\lambda)(-r - l_E \\ & - \lambda)(\mu e r) \left(\left(aV - \frac{l_P}{\mu}(e + l_U) \right) \left(e + l_U - \frac{re}{r + l_E} \right)^{-1} \right) \\ & + (e - l_U) \left(\mu \left(aV - \frac{l_P}{\mu}(e + l_U) \right) \left(e + l_U - \frac{re}{r + l_E} \right)^{-1} \right) (-r - l_E \\ & - \lambda) = 0 \end{aligned} \quad (2.21)$$

As discussed above, the real part of the eigenvalues will describe the stability of the system at the fixed point. In order to evaluate the stability of the fixed points for this system, a Matlab script was written that allows the user to input various system parameters and a description of the stability is provided for the fixed points (see Appendix C.1). For example, the fixed point $(P^*, U^*, E^*) = (2.33, 0.03, 0.10)$ with parameters $(V, a, e, \mu, r, l_P, l_U, l_E) = (2, 0.25, 0.75, 0.75, 0.25, 0.20, 1, 0)$ is unstable since the eigenvalues are $\lambda_1 = 0.007 + 0.2324i$, $\lambda_2 = 0.007 - 0.2324i$, and $\lambda_3 = -0.289$, which have a positive real part.

The fixed points $(V^*, P^*, N^*, U^*, E^*, O^*)$ for the *ODE PUEP Model with Inflow*

Domains are given by

$$\begin{aligned}
 V^* &= 0, & P^* &= \frac{e + l_U}{\mu}, & N^* &= \frac{r_p}{\mu} \frac{e + l_U}{r_N + l_N}, \\
 U^* &= \left(-\frac{e + l_U}{\mu} (r_p + l_p) + r_N \frac{r_p}{\mu} \frac{e + l_U}{r_N + l_N} \right) \left(e + l_U - \frac{r_O r_E e}{r_E + l_E} (r_O + l_O) \right)^{-1}, \\
 E^* &= \frac{e}{r_E + l_E} \left(-\frac{e + l_U}{\mu} (r_p + l_p) + r_N \frac{r_p}{\mu} \frac{e + l_U}{r_N + l_N} \right) \left(e + l_U - \frac{r_O r_E e}{r_E + l_E} (r_O + l_O) \right)^{-1}, \\
 O^* &= \frac{r_E}{r_O + l_O} \frac{e}{r_E + l_E} \left(-\frac{e + l_U}{\mu} (r_p + l_p) \right. \\
 &\quad \left. + r_N \frac{r_p}{\mu} \frac{e + l_U}{r_N + l_N} \right) \left(e + l_U - \frac{r_O r_E e}{r_E + l_E} (r_O + l_O) \right)^{-1}
 \end{aligned} \tag{2.22}$$

The Jacobian for the system is

$$\begin{aligned}
 &\mathbf{J}(V^*, P^*, N^*, U^*, E^*, O^*) \\
 &= \begin{pmatrix} -a - l_V & 0 & 0 & 0 & 0 & 0 \\ a & -\mu U^* - r_p - l_p & r_N & -\mu P^* & 0 & r_O \\ 0 & r_p & -r_N - l_N & 0 & 0 & 0 \\ 0 & \mu U^* & 0 & \mu P^* - e - l_U & 0 & 0 \\ 0 & 0 & 0 & e & -r_E - l_E & 0 \\ 0 & 0 & 0 & 0 & r_E & -r_O - l_O \end{pmatrix} \tag{2.23}
 \end{aligned}$$

A Matlab function was written to allow a user to enter various parameters and obtain a description of the stability at the fixed point (see Appendix C.2).

To describe the stability of (2.4) and (2.5), analysis can be performed on the fixed points of the system; however, the explicit form of the fixed points will be hard to obtain since the models are nonlinear. As such, a Matlab program was written to describe the

stability of fixed points, computed numerically, for user-defined parameters (see Appendix C.3 and C.4).

2.8 Sustainability Thresholds

A condition that is important for the study of metal flows is the condition under which sustainability will occur. Metal sustainability has various meanings and connotations dependent on the context, but for the purposes of this thesis, metal sustainability will be defined as the condition to which the use of virgin metal is eliminated from the system. Mathematically, this definition is straightforward to acquire: set the rate of change of the “ore” domain equal to 0. Throughout this work, this condition is defined as the “sustainability threshold”. More input is required to ascertain sustainability thresholds given by other definitions, such as the following: (1) a system that will not deplete the metal resource; or (2) using metals at a level that will preserve present and future metal needs. To an extent, these two thresholds are subjective—the amount of resources and present and future metal needs are not explicitly expressed in the model and will require the examiner’s judgment. Substituting “usage” for “need” in definition (2) can provide a method for the acquisition of the threshold: solving $U(0) = U(T)$ where $U(0)$ represents current metal usage and $U(T)$ represents future metal usage. However, since the nonlinear metal flow models are difficult to solve analytically, this threshold will be difficult to compute, but could provide a valuable assessment.

In this section, sustainability thresholds for the *ODE PUE Model*, *ODE PUEP Model with Inflow Domains*, and *ODE PUEP Model with Inflow Domains and Stock and*

Trade Considerations will be established. Other model's thresholds can be acquired through the extension of these examples.

To find the sustainability threshold for (2.1), the first equation needs to be reformulated to make the distinction between ore and recycle metals. Note that the inflow models establish these domains explicitly; as a result, making the thresholds more readily accessible. As previously mentioned, the Production domain is comprised of ore and recycle metal. Let $\psi \in [0,1]$ be the fraction of the Production domain that comprises ore, and then $1 - \psi$ the fraction of the Production domain that comes from recycle material. Define the differentiable functions $V(t) = \psi P(t)$ and $R(t) = (1 - \psi)P(t)$, then

$$P(t) = \psi P(t) + (1 - \psi)P(t) = V(t) + R(t) \quad (2.24)$$

Since $V(t)$ and $R(t)$ are differentiable, we have from the first equation of (2.1)

$$\begin{aligned} \frac{dP(t)}{dt} &= \frac{dV(t)}{dt} + \frac{dR(t)}{dt} = -\mu(V(t) + R(t))U(t) = -\mu V(t)U(t) - \mu R(t)U(t) \\ \Rightarrow \frac{dV(t)}{dt} &= -\mu V(t)U(t) - \mu R(t)U(t) - \frac{dR(t)}{dt} \end{aligned} \quad (2.25)$$

Setting $V' = 0$, we will obtain

$$\frac{dR(t)}{dt} + \mu R(t)U(t) + \mu V(t)U(t) = 0 \quad (2.26)$$

The solution to the differential equation is

$$R(t) + \mu \exp^{-\mu \int U(t)dt} \int \exp^{\mu \int U(t)dt} V(t)U(t)dt = 0 \quad (2.27)$$

Notice that threshold (2.27) only has trivial solutions: $R(t) = \mu = 0$ or $R(t) = V(t) = 0$, which is consistent with this model. The *ODE PUE Model* describes how a

specified quantity of metal flows through a system. In order to obtain an equilibrium in which no virgin metal is consumed, the system must have no demand for it and recycle metal is depleted, given by $\mu = 0$ and $R(t) = 0$, or virgin metal and recycle metal must be depleted, given by $V(t) = 0$ and $R(t) = 0$. To describe this dynamic qualitatively, sustainability (as define narrowly above) can only be achieved within this model under two circumstances: (i) metal is not available to be consumed or (ii) there is no demand for metal and recycle metal is no longer available. This threshold analysis demonstrates that in a closed economic system which contains only a finite amount of metal with no continuous inflow of recycled metal to supply a demand, the depletion of metal resources is inevitable.

With a Virgin Metal domain established in the *ODE PUEP Model with Inflow Domains*, a sustainability threshold is readily attained by setting the rate of change of the Virgin Metal equal to 0; thus, implying that no ore metal will be used to sustain the system. From the first 2 equations of (2.3), we will obtain

$$\frac{dV(t)}{dt} = -\frac{dP(t)}{dt} - \mu P(t)U(t) - r_p P(t) - l_p P(t) + r_o O(t) + r_N N(t) - l_v V(t) \quad (2.28)$$

Setting $\frac{dV(t)}{dt} = 0$, we will obtain

$$\frac{dP(t)}{dt} + P(t)(\mu U(t) + r_p + l_p) + (-r_o O(t) - r_N N(t) + l_v V(t)) = 0 \quad (2.29)$$

with a solution

$$P(t) = \exp^{-\int(\mu U(t)+r_p+l_p)dt} \int \exp^{\int(\mu U(t)+r_p+l_p)dt} (r_o O(t) + r_N N(t) - l_v V(t)) dt \quad (2.30)$$

If (2.32) holds, then the sustainability threshold is met.

A sustainability threshold for (2.4) can be ascertained by setting the rate of change of the Virgin Metal domain equation to 0. In doing so, the conditions under which no metal ore will be needed to sustain the metal flow system will be established. From the first two equations of (2.4), we will obtain

$$\begin{aligned} \frac{dV(t)}{dt} = & -\frac{dP(t)}{dt} + P(t)(-\mu U(t) - r_p - l_p - x_p) \\ & + (r_O O(t) + r_N N(t) + i_P I(t) + S(t) - l_V V(t) - x_V V(t)) \end{aligned} \quad (2.31)$$

Setting $\frac{dV(t)}{dt} = 0$, we will obtain

$$\begin{aligned} \frac{dP(t)}{dt} + P(t)(\mu U(t) + r_p + l_p + x_p) \\ + (-r_O O(t) - r_N N(t) - i_P I(t) - S(t) + l_V V(t) + x_V V(t)) \\ = 0 \end{aligned} \quad (2.32)$$

with a solution

$$\begin{aligned} P(t) = \exp^{-\int(\mu U(t) + r_p + l_p + x_p)dt} \int \exp^{\int(\mu U(t) + r_p + l_p + x_p)dt} (r_O O(t) + r_N N(t) \\ + i_P I(t) + S(t) - l_V V(t) - x_V V(t)) dt \end{aligned} \quad (2.33)$$

If (2.35) holds, then a sustainability threshold for the system is met.

2.9 Parameter Optimization for the Dynamic Metal Flow Models

The models presented in this chapter created a mathematical framework for metal flows that were dependent on multiple parameters, such as supply and demand, end-of-life, recyclability, and metal loss rates. These parameters are reliant on the type of metal industry under study, and should be constructed on a case-by-case basis. Given empirical

data on metal flows for a particular industry, the best parameters for a model that best represents the empirical data can be computed by using parameter optimization techniques. The assumption will be made that a particular metal flow model best represents the dynamics of an industry and data will be complied for the models' domains. In order to obtain the optimal parameters \mathbf{k} for the model, we need to find \mathbf{k} that minimizes a cost function $S(\mathbf{k}) = \|\mathbf{y}(\mathbf{k}) - \mathbf{y}^*\|$, where \mathbf{y}^* is the empirical data and $\mathbf{y}(\mathbf{k})$ is the analytical solution of the model. For the ordinary differential equation models, the Levenberg-Marquardt algorithm (LM) is applied.

LM is an iterative method that finds a local minimum of a non-linear, multivariate function and is used to solve least-squares problems. LM is a combination of the steepest decent and Gauss-Newton's methods. LM uses a trust-region over the line search strategy of Gauss-Newton in order to solve problems with rank-deficient or nearly rank-deficient Jacobian [9].

The first two sections will establish the modifications needed to implement the LM algorithm for a system of ordinary differential equations where their analytical solutions are unknown. The proceeding sections will establish the Jacobians and error vectors need in the LM algorithm for the 5 ordinary differential equation metal flow models. Implementation of the parameter optimization method for the *ODE PUE Model* written in the Matlab language is provided in Appendix C.5.

2.9.1 Parameter Optimization for an Initial Value Problem

Consider the following initial value problem:

$$\begin{aligned}\frac{dy(t)}{dt} &= f(\mathbf{k}, t, y) \\ y(t) &= y_0\end{aligned}\tag{2.34}$$

The best parameter $\mathbf{k} = (k_1, k_2, \dots, k_N)$ such that $y(\mathbf{k}, t)$ that satisfies (2.34) is close to an empirical data set \mathbf{y}^* is sought. In order to acquire \mathbf{k} , LM is implemented and detailed in the following algorithm.

Algorithm 2.1 Levenberg-Marquardt Method for an Initial Value Problem

Step 1: Let \mathbf{k}_0 be an initial guess for the sought parameter. If the analytical solution

$y(\mathbf{k}_0, t)$ is “close” to the empirical data set \mathbf{y}^ , that is if*

$$S(\mathbf{k}_0) = \left(\sum_{i=1}^N (y_i(\mathbf{k}_0, t) - y_i^*)^2 \right)^{\frac{1}{2}} = \|\mathbf{y}(\mathbf{k}_0) - \mathbf{y}^*\| \leq \text{tolerance} \tag{2.35}$$

then \mathbf{k}_0 is assumed to be the optimal parameter. S is referred to as the cost function

and $\|\cdot\|$ is the L^2 -norm.

Step 2: If $S(\mathbf{k}_0) > \text{tolerance}$, an adjustment to \mathbf{k}_0 is made. Let $\mathbf{k}_1 = \mathbf{k}_0 + \delta_1$ be the new

adjusted parameter. For a small $\|\delta_1\|$, a Taylor series expansion applied to the

analytical solution $y(\mathbf{k}_0 + \delta_1, t)$ yields

$$y(\mathbf{k}_0 + \delta_1, t) \approx y(\mathbf{k}_0, t) + \mathbf{J} \delta_1 \tag{2.36}$$

where \mathbf{J} is the Jacobian matrix $\frac{\partial y(\mathbf{k}_0, t)}{\partial \mathbf{k}_0}$. The cost function then becomes

$$S(\mathbf{k}_1) = \|\mathbf{y}(\mathbf{k}_0 + \delta_1) - \mathbf{y}^*\| = \|\mathbf{J} \delta_1 + \mathbf{y}(\mathbf{k}_0, t) - \mathbf{y}^*\| = \|\mathbf{J} \delta_1 - \epsilon\| \tag{2.37}$$

Step 3: Find δ_1 that minimizes $S(k_1)$. δ_1 is thus the solution to the linear least squares problem: the minimum is obtained when $\mathbf{J} \delta_1 - \epsilon$ is orthogonal to the column space of \mathbf{J} , which implies that $\mathbf{J}^T (\mathbf{J} \delta_1 - \epsilon) = 0$. The δ_1 that meets this criteria is therefore the solution to the normal equations

$$\mathbf{J}^T \mathbf{J} \delta_1 = \mathbf{J}^T \epsilon \quad (2.38)$$

Step 4: The LM algorithm solves a slight variation of (2.38)—the augmented normal equations

$$\mathbf{N} \delta_1 = \mathbf{J}^T \epsilon \quad (2.39)$$

where $\mathbf{N} = \mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})$, with λ referred to as the damping term.

Step 5: If the updated \mathbf{k}_1 leads to a reduction in error ϵ , then the update is accepted, the damping term λ is reduced and proceed to Step 1. Otherwise, the damping term is increased and proceed to Step 4.

In order to implement Algorithm 2.1, a finite difference approximation for the analytical solution to (2.34) is used:

$$\begin{aligned} y_{i+1} &= y_i + \Delta t f(k_0, t_i, y_i) \quad \text{for } i = 0, \dots, N-1 \\ \Rightarrow \frac{\partial y_{i+1}}{\partial k_0} &= \frac{\partial y_i}{\partial k_0} + \Delta t \frac{\partial}{\partial k_0} f(k_0, t_i, y_i) \\ \Rightarrow \frac{\partial y_{i+1}}{\partial k_0} &= \Delta t \frac{\partial}{\partial k_0} f(k_0, t_i, y_i) + \Delta t \frac{\partial}{\partial k_0} f(k_0, t_{i-1}, y_{i-1}) + \dots \\ &\quad + \Delta t \frac{\partial}{\partial k_0} f(k_0, t_0, y_0) = \Delta t \sum_{j=0}^i \frac{\partial}{\partial k_0} f(k_0, t_j, y_j) \end{aligned} \quad (2.40)$$

which is the recursive relationship $\frac{\partial y_{i+1}}{\partial k_0} = \Delta t \frac{\partial}{\partial k_0} f(k_0, t_i, y_i)$ for $i = 0, \dots, N$. The

Jacobian is therefore $J = \Delta t \frac{\partial f(k_0, t, y)}{\partial k_0}$, with each row computed recursively.

2.9.2 Parameter Optimization for a System of Differential Equations

Consider the following system of ordinary differential equations

$$\frac{d\mathbf{F}}{dt} = \begin{pmatrix} \frac{dx(t)}{dt} \\ \frac{dy(t)}{dt} \\ \frac{dz(t)}{dt} \end{pmatrix} = \begin{pmatrix} f(\mathbf{k}, t, x, y, z) \\ g(\mathbf{k}, t, x, y, z) \\ h(\mathbf{k}, t, x, y, z) \end{pmatrix} \quad \text{with} \quad \begin{matrix} x(0) = x_0 \\ y(0) = y_0 \\ z(0) = z_0 \end{matrix} \quad (2.41)$$

We want to find the best parameter $\mathbf{k} = (k_1, k_2, \dots, k_N)$ such that $F(\mathbf{k}, t)$ satisfying (2.41)

is close to an empirical data set \mathbf{F}^* . Algorithm 2.1 can be extended for system (2.41),

which is detailed in the following algorithm.

Algorithm 2.2 Levenberg-Marquardt Method for a System of Ordinary Differential Equations

Step 1: Let \mathbf{k}_0 be an initial guess for the sought parameter. If the analytical solution

$\mathbf{F}(\mathbf{k}_0, t)$ is “close” to the empirical data set \mathbf{F}^ , that is if we have*

$$S(\mathbf{k}_0) = \left(\sum_{i=1}^N \|\mathbf{F}(\mathbf{k}_0) - \mathbf{F}^*\|^2 \right)^{\frac{1}{2}} \leq \text{tolerance} \quad (2.42)$$

then \mathbf{k}_0 is assumed to be the optimal parameter. S is referred to as the cost function

and $\|\cdot\|$ is the L^2 -norm.

Step 2: If $S(\mathbf{k}_0) > \text{tolerance}$, we will need to adjust \mathbf{k}_0 . Let $\mathbf{k}_1 = \mathbf{k}_0 + \boldsymbol{\delta}_1$ be the new adjusted parameter. For a small $\|\boldsymbol{\delta}_1\|$, applying a Taylor series expansion to the analytical solution $\mathbf{F}(\mathbf{k}_0 + \boldsymbol{\delta}_1, t)$ yields

$$\mathbf{F}(\mathbf{k}_0 + \boldsymbol{\delta}_1, t) \approx \mathbf{F}(\mathbf{k}_0, t) + \mathbf{J} \boldsymbol{\delta}_1 \quad (2.43)$$

where \mathbf{J} is the Jacobian matrix $\frac{\partial \mathbf{F}(\mathbf{k}_0, t)}{\partial \mathbf{k}_0}$. The cost function then becomes

$$S(\mathbf{k}_1) = \|\mathbf{F}(\mathbf{k}_0 + \boldsymbol{\delta}_1) - \mathbf{F}^*\| = \|\mathbf{J} \boldsymbol{\delta}_1 + \mathbf{F}(\mathbf{k}_0, t) - \mathbf{F}^*\| = \|\mathbf{J} \boldsymbol{\delta}_1 - \epsilon\| \quad (2.44)$$

Step 3: Find $\boldsymbol{\delta}_1$ that minimizes $S(\mathbf{k}_1)$. $\boldsymbol{\delta}_1$ is thus the solution to the linear least squares problem: the minimum is obtained when $\mathbf{J} \boldsymbol{\delta}_1 - \epsilon$ is orthogonal to the column space of \mathbf{J} , which implies that $\mathbf{J}^T (\mathbf{J} \boldsymbol{\delta}_1 - \epsilon) = 0$. The $\boldsymbol{\delta}_1$ that meets this criteria is therefore the solution to the normal equations

$$\mathbf{J}^T \mathbf{J} \boldsymbol{\delta}_1 = \mathbf{J}^T \epsilon \quad (2.45)$$

A finite difference approximation for the analytical solution to (2.41) will be used:

$$\begin{aligned} x_{i+1} &= x_i + \Delta t f(\mathbf{k}_0, t_i, x_i, y_i, z_i) = x_i + \Delta t f_i \\ y_{i+1} &= y_i + \Delta t g(\mathbf{k}_0, t_i, x_i, y_i, z_i) = y_i + \Delta t g_i \\ z_{i+1} &= z_i + \Delta t h(\mathbf{k}_0, t_i, x_i, y_i, z_i) = z_i + \Delta t h_i \end{aligned} \quad (2.46)$$

It follows that

$$\epsilon = \begin{pmatrix} x_i + \Delta t f_i - \mathbf{x}_{i+1}^* \\ y_i + \Delta t g_i - \mathbf{y}_{i+1}^* \\ z_i + \Delta t h_i - \mathbf{z}_{i+1}^* \end{pmatrix} \quad (2.47)$$

As illustrated in (2.40), the Jacobian evaluated for \mathbf{k}_0 is computed recursively by:

$$\mathbf{J} = \begin{pmatrix} \Delta t \frac{\partial f_i(\mathbf{k}_0, t, y)}{\partial k_1} & \Delta t \frac{\partial f_i(\mathbf{k}_0, t, y)}{\partial k_2} & \dots & \Delta t \frac{\partial f_i(\mathbf{k}_0, t, y)}{\partial k_N} \\ \Delta t \frac{\partial g_i(\mathbf{k}_0, t, y)}{\partial k_1} & \Delta t \frac{\partial g_i(\mathbf{k}_0, t, y)}{\partial k_2} & \dots & \Delta t \frac{\partial g_i(\mathbf{k}_0, t, y)}{\partial k_N} \\ \Delta t \frac{\partial h_i(\mathbf{k}_0, t, y)}{\partial k_1} & \Delta t \frac{\partial h_i(\mathbf{k}_0, t, y)}{\partial k_2} & \dots & \Delta t \frac{\partial h_i(\mathbf{k}_0, t, y)}{\partial k_N} \end{pmatrix} \quad (2.48)$$

Step 4: The LM algorithm solves the augmented normal equations

$$\mathbf{N} \boldsymbol{\delta}_1 = \mathbf{J}^T \boldsymbol{\epsilon} \quad (2.49)$$

where $\mathbf{N} = \mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})$ with λ referred to as the damping term.

Step 5: If the updated \mathbf{k}_1 leads to a reduction in error ϵ , then the update is accepted, the damping term λ is reduced and proceed to Step 1. Otherwise, the damping term is increased and proceed to Step 4.

2.9.3 Parameter Optimization for the ODE PUE Model

The parameters for the *ODE PUE Model* (μ, e) can be optimized using Algorithm

2.2. Recall that μ is the supply/demand rate and e is the end-of-life rate for the model.

The optimization problem becomes: *find the optimal (μ, e) such that (P, U, E) which satisfies the ODE PUE Model is close to the empirical data set $(\mathbf{P}^*, \mathbf{U}^*, \mathbf{E}^*)$.*

In order to construct the Jacobian matrix and the epsilon vectors need in the LM algorithm, the analytical solution to the model will be approximated using a first order finite difference scheme:

$$\begin{aligned}
P_{i+1} &= P_i - \Delta t (\mu P_i U_i) \\
U_{i+1} &= U_i + \Delta t (\mu P_i U_i - e U_i) \text{ for } i = 0, \dots, N-1 \\
E_{i+1} &= E_i + \Delta t (e U_i)
\end{aligned} \tag{2.50}$$

By (2.48), the Jacobian for the system evaluated for a defined (μ, e) is

$$\begin{aligned}
\mathbf{J} &= \begin{pmatrix} \Delta t \frac{\partial}{\partial \mu} (-\mu P_i U_i) & \Delta t \frac{\partial}{\partial e} (-\mu P_i U_i) \\ \Delta t \frac{\partial}{\partial \mu} (\mu P_i U_i - e U_i) & \Delta t \frac{\partial}{\partial e} (\mu P_i U_i - e U_i) \\ \Delta t \frac{\partial}{\partial \mu} (e U_i) & \Delta t \frac{\partial}{\partial e} (e U_i) \end{pmatrix} = \begin{pmatrix} -\Delta t (P_i U_i) & 0 \\ \Delta t (P_i U_i) & \Delta t (-U_i) \\ 0 & \Delta t (U_i) \end{pmatrix} \\
&\text{for } i = 0, \dots, N-1 \\
\Rightarrow \mathbf{J} &= \Delta t \begin{pmatrix} -P_0 U_0 & 0 \\ -P_1 U_1 & 0 \\ \vdots & \vdots \\ -P_{N-1} U_{N-1} & 0 \\ P_0 U_0 & -U_0 \\ P_1 U_1 & -U_1 \\ \vdots & \vdots \\ P_{N-1} U_{N-1} & -U_{N-1} \\ 0 & U_0 \\ 0 & U_1 \\ \vdots & \vdots \\ 0 & U_{N-1} \end{pmatrix} \tag{2.51}
\end{aligned}$$

By (2.47), the epsilon vector for the system is

$$\epsilon = \begin{pmatrix} P_i - \Delta t (\mu P_i U_i) - \mathbf{P}_{i+1}^* \\ U_i + \Delta t (\mu P_i U_i - e U_i) - \mathbf{U}_{i+1}^* \\ E_i + \Delta t (e U_i) - \mathbf{E}_{i+1}^* \end{pmatrix}$$

$$= \begin{pmatrix} P_0 - \Delta t (\mu P_0 U_0) - \mathbf{P}_1^* \\ P_1 - \Delta t (\mu P_1 U_1) - \mathbf{P}_2^* \\ \vdots \\ P_{N-1} - \Delta t (\mu P_{N-1} U_{N-1}) - \mathbf{P}_N^* \\ U_0 + \Delta t (\mu P_0 U_0 - e U_0) - \mathbf{U}_1^* \\ U_1 + \Delta t (\mu P_1 U_1 - e U_1) - \mathbf{U}_2^* \\ \vdots \\ U_{N-1} + \Delta t (\mu P_{N-1} U_{N-1} - e U_{N-1}) - \mathbf{U}_N^* \\ E_0 + \Delta t (e U_0) - \mathbf{E}_1^* \\ E_1 + \Delta t (e U_1) - \mathbf{E}_2^* \\ \vdots \\ E_{N-1} + \Delta t (e U_{N-1}) - \mathbf{E}_N^* \end{pmatrix} \quad (2.52)$$

Now that the Jacobian matrix and ϵ -vector are constructed for the system, the LM

Algorithm 2.2 can be implemented. See Appendix C.5 for the source code.

2.9.4 Parameter Optimization for the ODE PUEP Model

Recall the *ODE PUEP Model* discussed in Section 2.2. The parameters $(a, \mu, r, e, l_P, l_U, l_E)$ that produces an analytical solution that are close to an empirical data set $(\mathbf{P}^*, \mathbf{U}^*, \mathbf{E}^*)$ are sought. Recall that the parameters are

a —Ore availability rate

μ —Supply/demand rate

r —Recyclability rate

e —End-of-life rate

l_P, l_U, l_E —Loss rate of the Production, Use, and End-of-Life domains

The analytical solution (P, U, E) for the *ODE PUEP Model* is approximated using the finite difference scheme:

$$P_{i+1} = P_i + \Delta t (a V_i - \mu P_i U_i + r E_i - l_P P_i) = P_i + \Delta t f_i \quad (2.53)$$

$$U_{i+1} = U_i + \Delta t (\mu P_i U_i - e U_i - l_U U_i) = U_i + \Delta t g_i \quad \text{for } i = 0, \dots, N-1$$

$$E_{i+1} = E_i + \Delta t (e U_i - r E_i - l_E E_i) = E_i + \Delta t h_i$$

It follows by (2.48) that the Jacobian for the system evaluated for a particular parameter

$\mathbf{k} = (a, \mu, r, e, l_P, l_U, l_E)$ is

$$\begin{aligned} \mathbf{J} &= \Delta t \begin{pmatrix} \frac{\partial f_i(\mathbf{k})}{\partial a} & \frac{\partial f_i(\mathbf{k})}{\partial \mu} & \frac{\partial f_i(\mathbf{k})}{\partial r} & \frac{\partial f_i(\mathbf{k})}{\partial e} & \frac{\partial f_i(\mathbf{k})}{\partial l_P} & \frac{\partial f_i(\mathbf{k})}{\partial l_U} & \frac{\partial f_i(\mathbf{k})}{\partial l_E} \\ \frac{\partial g_i(\mathbf{k})}{\partial a} & \frac{\partial g_i(\mathbf{k})}{\partial \mu} & \frac{\partial g_i(\mathbf{k})}{\partial r} & \frac{\partial g_i(\mathbf{k})}{\partial e} & \frac{\partial g_i(\mathbf{k})}{\partial l_P} & \frac{\partial g_i(\mathbf{k})}{\partial l_U} & \frac{\partial g_i(\mathbf{k})}{\partial l_E} \\ \frac{\partial h_i(\mathbf{k})}{\partial a} & \frac{\partial h_i(\mathbf{k})}{\partial \mu} & \frac{\partial h_i(\mathbf{k})}{\partial r} & \frac{\partial h_i(\mathbf{k})}{\partial e} & \frac{\partial h_i(\mathbf{k})}{\partial l_P} & \frac{\partial h_i(\mathbf{k})}{\partial l_U} & \frac{\partial h_i(\mathbf{k})}{\partial l_E} \end{pmatrix} \\ &= \Delta t \begin{pmatrix} V_i & -P_i U_i & E_i & 0 & -P_i & 0 & 0 \\ 0 & P_i U_i & 0 & -U_i & 0 & -U_i & 0 \\ 0 & 0 & -E_i & U_i & 0 & 0 & -E_i \end{pmatrix} \quad \text{for } i = 0, \dots, N-1 \\ &\Rightarrow \mathbf{J} = \Delta t \begin{pmatrix} V_0 & -P_0 U_0 & E_0 & 0 & -P_0 & 0 & 0 \\ V_1 & -P_1 U_1 & E_1 & 0 & -P_1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ V_{N-1} & -P_{N-1} U_{N-1} & E_{N-1} & 0 & -P_{N-1} & 0 & 0 \\ 0 & P_0 U_0 & 0 & -U_0 & 0 & -U_0 & 0 \\ 0 & P_1 U_1 & 0 & -U_1 & 0 & -U_1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & P_{N-1} U_{N-1} & 0 & -U_{N-1} & 0 & -U_{N-1} & 0 \\ 0 & 0 & -E_0 & U_0 & 0 & 0 & -E_0 \\ 0 & 0 & -E_1 & U_1 & 0 & 0 & -E_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -E_{N-1} & U_{N-1} & 0 & 0 & -E_{N-1} \end{pmatrix} \quad (2.54) \end{aligned}$$

By (2.47), the distance error is

$$\epsilon = \begin{pmatrix} P_i + \Delta t (a V_i - \mu P_i U_i + r E_i - l_P P_i) - \mathbf{P}_{i+1}^* \\ U_i + \Delta t (\mu P_i U_i - e U_i - l_U U_i) - \mathbf{U}_{i+1}^* \\ E_i + \Delta t (e U_i - r E_i - l_E E_i) - \mathbf{E}_{i+1}^* \end{pmatrix}$$

$$= \begin{pmatrix} P_0 + \Delta t (a V_0 - \mu P_0 U_0 + r E_0 - l_P P_0) - \mathbf{P}_1^* \\ P_1 + \Delta t (a V_1 - \mu P_1 U_1 + r E_1 - l_P P_1) - \mathbf{P}_2^* \\ \vdots \\ P_{N-1} + \Delta t (a V_{N-1} - \mu P_{N-1} U_{N-1} + r E_{N-1} - l_P P_{N-1}) - \mathbf{P}_N^* \\ U_0 + \Delta t (\mu P_0 U_0 - e U_0 - l_U U_0) - \mathbf{U}_1^* \\ U_1 + \Delta t (\mu P_1 U_1 - e U_1 - l_U U_1) - \mathbf{U}_2^* \\ \vdots \\ U_{N-1} + \Delta t (\mu P_{N-1} U_{N-1} - e U_{N-1} - l_U U_{N-1}) - \mathbf{U}_N^* \\ E_0 + \Delta t (e U_0 - r E_0 - l_E E_0) - \mathbf{E}_1^* \\ E_1 + \Delta t (e U_1 - r E_1 - l_E E_1) - \mathbf{E}_2^* \\ \vdots \\ E_{N-1} + \Delta t (e U_{N-1} - r E_{N-1} - l_E E_{N-1}) - \mathbf{E}_N^* \end{pmatrix} \quad (2.55)$$

Now that the Jacobian matrix and ϵ -vector are constructed for the system, the LM Algorithm 2.2 can be implemented.

2.9.5 Parameter Optimization for the ODE PUEP with Inflow Domains Model

Recall the *ODE PUEP Model with Inflow Domains* discussed in Section 2.3.

Given empirical data of a metal industry, the optimal parameters

$\mathbf{k} = (a, \mu, r_P, r_O, r_N, r_E, e, l_V, l_P, l_N, l_U, l_E, l_O)$ such that the analytical solution

$(V(\mathbf{k}), P(\mathbf{k}), N(\mathbf{k}), U(\mathbf{k}), E(\mathbf{k}), O(\mathbf{k}))$ to the model is close to an empirical data set are

sought. As in the previous sections, the analytical solution to the model will be

approximated with a finite difference scheme, which will be used to construct the

Jacobian matrix and distance error of the system. With the Jacobian matrix and error

vector in hand, the LM algorithm is implemented to compute the optimal parameters

based on the empirical data set.

The finite difference scheme for $i = 0, \dots, N - 1$ for the model is as follows:

$$V_{i+1} = V_i + \Delta t(-l_V V_i - a V_i) \quad (2.56)$$

$$P_{i+1} = P_i + \Delta t (-l_P P_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i + a V_i)$$

$$N_{i+1} = N_i + \Delta t (-l_N N_i + r_P P_i - r_N N_i)$$

$$U_{i+1} = U_i + \Delta t (-l_U U_i + \mu P_i U_i - e U_i)$$

$$E_{i+1} = E_i + \Delta t (-l_E E_i + e U_i - r_E E_i)$$

$$O_{i+1} = O_i + \Delta t (-l_O O_i + r_E E_i - r_O O_i)$$

The Jacobian matrix for the system evaluated for parameter

$\mathbf{k} = (a, \mu, r_P, r_O, r_N, r_E, e, l_V, l_P, l_N, l_U, l_E, l_O)$ is

$\mathbf{J} =$

$$\Delta t \begin{pmatrix} -V_i & 0 & 0 & 0 & 0 & 0 & 0 & -V_i & 0 & 0 & 0 & 0 & 0 \\ V_i & -P_i U_i & -P_i & O_i & N_i & 0 & 0 & 0 & -P_i & 0 & 0 & 0 & 0 \\ 0 & 0 & P_i & 0 & -N_i & 0 & 0 & 0 & 0 & -N_i & 0 & 0 & 0 \\ 0 & P_i U_i & 0 & 0 & 0 & 0 & -U_i & 0 & 0 & 0 & -U_i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -E_i & U_i & 0 & 0 & 0 & 0 & -E_i & 0 \\ 0 & 0 & 0 & -O_i & 0 & E_i & 0 & 0 & 0 & 0 & 0 & 0 & -O_i \end{pmatrix} \quad (2.57)$$

By (2.47), the distance error is

$$\epsilon = \begin{pmatrix} V_i + \Delta t (-l_V V_i - a V_i) - \mathbf{V}_{i+1}^* \\ P_i + \Delta t (-l_P P_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i + a V_i) - \mathbf{P}_{i+1}^* \\ N_i + \Delta t (-l_N N_i + r_P P_i - r_N N_i) - \mathbf{N}_{i+1}^* \\ U_i + \Delta t (-l_U U_i + \mu P_i U_i - e U_i) - \mathbf{U}_{i+1}^* \\ E_i + \Delta t (-l_E E_i + e U_i - r_E E_i) - \mathbf{E}_{i+1}^* \\ O_i + \Delta t (-l_O O_i + r_E E_i - r_O O_i) - \mathbf{O}_{i+1}^* \end{pmatrix} \quad (2.58)$$

2.9.6 Parameter Optimization for the ODE PUEP Model with Inflow Domains and

Stock and Trade Considerations

Recall the *ODE PUEP Model with Inflow Domains and Stock and Trade*

Considerations constructed in Section 2.4. The parameters that can be optimize are

$\mathbf{k} = (a, \mu, e, r_P, r_O, r_N, r_E, l_V, l_P, l_N, l_U, l_E, l_O, x_V, x_P, x_U, x_O, i_P, i_U, i_O)$. Depending of the

manner to which imports and exports are defined in the model, the import and export rates can be considered to be known information and eliminated from the optimization problem. At this point, we will assume that the import and export rates are unknown and need to be optimized. The Jacobian matrix and the distance error are computed by using the finite difference approximation for the analytical solution of the model.

The finite difference scheme for $i = 0, \dots, N - 1$ is given by

$$\begin{aligned}
V_{i+1} &= V_i + \Delta t(-l_V V_i - a V_i - x_V V_i) \\
P_{i+1} &= P_i + \Delta t(-l_P P_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i + a V_i - x_P P_i + i_P I_i + S_i) \\
N_{i+1} &= N_i + \Delta t(-l_N N_i + r_P P_i - r_N N_i) \\
U_{i+1} &= U_i + \Delta t(-l_U U_i + \mu P_i U_i - e U_i - x_U U_i + i_U I_i) \\
E_{i+1} &= E_i + \Delta t(-l_E E_i + e U_i - r_E E_i) \\
O_{i+1} &= O_i + \Delta t(-l_O O_i + r_E E_i - r_O O_i - x_O O_i + i_O I_i)
\end{aligned} \tag{2.59}$$

The Jacobian matrix for the system evaluated for parameter \mathbf{k} is computed by:

$$\mathbf{J} = \Delta t(\mathbf{A} \quad \mathbf{R} \quad \mathbf{L} \quad \mathbf{X} \quad \mathbf{I}) \text{ where}$$

$$\mathbf{A} = \begin{pmatrix} -V_i & 0 & 0 \\ V_i & -P_i U_i & -P_i \\ 0 & 0 & P_i \\ 0 & P_i U_i & -U_i \\ 0 & 0 & U_i \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -P_i & O_i & N_i & 0 \\ P_i & 0 & -N_i & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -E_i \\ 0 & -O_i & 0 & E_i \end{pmatrix} \tag{2.60}$$

$$\mathbf{L} = \begin{pmatrix} -V_i & 0 & 0 & 0 & 0 & 0 \\ 0 & -P_i & 0 & 0 & 0 & 0 \\ 0 & 0 & -N_i & 0 & 0 & 0 \\ 0 & 0 & 0 & -U_i & 0 & 0 \\ 0 & 0 & 0 & 0 & -E_i & 0 \\ 0 & 0 & 0 & 0 & 0 & -O_i \end{pmatrix},$$

$$\mathbf{X} = \begin{pmatrix} -V_i & 0 & 0 & 0 \\ 0 & -P_i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -U_i & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -O_i \end{pmatrix}, \mathbf{I} = \begin{pmatrix} 0 & 0 & 0 \\ I_i & 0 & 0 \\ 0 & 0 & 0 \\ 0 & I_i & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I_i \end{pmatrix}$$

The distance error is computed by:

$$\epsilon = \begin{pmatrix} V_i + \Delta t(-l_V V_i - aV_i - x_V V_i) - \mathbf{V}_{i+1}^* \\ P_i + \Delta t \left(-l_P P_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i + \dots \right. \\ \quad \left. + a V_i - x_P P_i + i_P I_i + S_i \right) - \mathbf{P}_{i+1}^* \\ N_i + \Delta t(-l_N N_i + r_P P_i - r_N N_i) - \mathbf{N}_{i+1}^* \\ U_i + \Delta t(-l_U U_i + \mu P_i U_i - eU_i - x_U U_i + i_U I_i) - \mathbf{U}_{i+1}^* \\ E_i + \Delta t(-l_E E_i + eU_i - r_E E_i) - \mathbf{E}_{i+1}^* \\ O_i + \Delta t(-l_O O_i + r_E E_i - r_O O_i - x_O O_i + i_O I_i) - \mathbf{O}_{i+1}^* \end{pmatrix} \quad (2.61)$$

With the Jacobian matrix and epsilon vector computed, the LM algorithm to obtain the optimal parameters can be applied.

2.9.7 Parameter Optimization for the Modified ODE PUEP Model with Inflow

Domains and Stock and Trade Considerations

The *Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations*, as discussed in Section 2.5, reformulated the Virgin Metal domain in order to make a distinction between Reserves and production of primary metal. In order to accomplish this, the dynamics of the Virgin Metal and Production domains changed slightly with respect to the *ODE PUEP Model with Inflow Domains and Stock and Trade Considerations*, with the other 4 domain remaining unchanged. The inclusion and reformulation of terms in the first 2 equation of the model will likewise change the Jacobian matrix and distance error. The parameters that need to optimized for this model are $\mathbf{k} = (a, f, \mu, e, r_P, r_O, r_N, r_E, l_V, l_P, l_N, l_U, l_E, l_O, x_V, x_P, x_U, x_O, i_P, i_U, i_O)$. As stated in

Section 2.9.6, if the trade coefficients are assumed to be known, then they can be eliminated from the optimization problem. The finite difference scheme for this model will remain the same as in (2.59), with changes only to the first 2 equations:

$$\begin{aligned} V_{i+1} &= V_i + \Delta t(-l_V V_i + aR_i - f V_i - x_V V_i) \\ P_{i+1} &= P_i + \Delta t(-l_P P_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i + f V_i - x_P P_i + i_P I_i + S_i) \end{aligned} \quad (2.62)$$

The Jacobian matrix for the system evaluated for parameter \mathbf{k} is computed by:

$$\mathbf{J} = \Delta t(\mathbf{A} \quad \mathbf{R} \quad \mathbf{L} \quad \mathbf{X} \quad \mathbf{I}) \text{ where}$$

$$\mathbf{A} = \begin{pmatrix} -R_i & -V_i & 0 & 0 \\ 0 & V_i & -P_i U_i & -P_i \\ 0 & 0 & 0 & P_i \\ 0 & 0 & P_i U_i & -U_i \\ 0 & 0 & 0 & U_i \\ 0 & 0 & 0 & 0 \end{pmatrix}, \text{ with } \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{I} \text{ the same as in (2.60)} \quad (2.63)$$

The distance error for this model will be the same as in (5.61), with changes only to the first two equations:

$$\epsilon = \begin{pmatrix} V_{i+1} = V_i + \Delta t(-l_V V_i + aR_i - f V_i - x_V V_i) - \mathbf{V}_{i+1}^* \\ P_i + \Delta t \left(-l_P P_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i + \dots \right. \\ \quad \left. + f V_i - x_P P_i + i_P I_i + S_i \right) - \mathbf{P}_{i+1}^* \\ N_i + \Delta t(-l_N N_i + r_P P_i - r_N N_i) - \mathbf{N}_{i+1}^* \\ U_i + \Delta t(-l_U U_i + \mu P_i U_i - e U_i - x_U U_i + i_U I_i) - \mathbf{U}_{i+1}^* \\ E_i + \Delta t(-l_E E_i + e U_i - r_E E_i) - \mathbf{E}_{i+1}^* \\ O_i + \Delta t(-l_O O_i + r_E E_i - r_O O_i - x_O O_i + i_O I_i) - \mathbf{O}_{i+1}^* \end{pmatrix} \quad (2.64)$$

With the Jacobian matrix and error distance computed for this model, the LM algorithm can be implemented to find the optimal parameters that produce an analytical solution that is close to empirical data.

Chapter 3: Partial Differential Equation Models for Metal Flows

The metal flow models constructed in Chapter 2 assumed that the metal domains changed only with respect to time. In any economic system, pricing plays a pivotal role in the overall dynamics and should be incorporated into the modeling of such systems. In this chapter, pricing will be considered to be a continuous independent variable, represented by s , and incorporated into the five models outlined in Chapter 2. Due to the assumptions made about the influence of pricing on the domains—pricing changes the domains in a diffusive manner—the models generated will consist of systems of reaction-diffusion equations, with and without cross-diffusion. The numerical schemes for the approximate solutions to the models will be presented in Chapter 5, with source code for the numerical schemes provided in Appendix C.

Modification to the models generated in this chapter may incorporate other deformations of the domains based on price, such as the inclusion of volatility, shocks, or fluctuations. Note that this thesis treats the partial differential equation models as an extension to the primary work produced in Chapter 2. To this end, the analysis of the partial differential equation models is left as future work.

3.1 PUE Models

Let the three domains define in Section 2.1 be dependent variables with respect to time t and price s , for $t \geq 0$ and $s \geq 0$. The assumption is made that the domains change with respect to price in a diffusive manner due to the nature of the mineral and metal industries. At a time step, the density of the Production domain for a particular metal is derived from certain minerals, which have associated pricings. At a future time step, these minerals are refined into multiple downstream chemicals or substances, with a particular price associated to each. For instance, strontium metal is derived mainly from the strontium mineral celestite [4]. Celestite is refined into a multitude for strontium products, each having a certain associated price. Thus, at a particular time the majority of metal residing in the Production domain has a narrow range of prices, but at a later time, the price range spreads.

The same behavior is indicative to the Use and End-of-Life domains. A particular group of consumer products entering the Use domain will have a pricing structure. As time elapses, the products will develop varying prices due to their treatment. This dynamic, a mass spreading down the concentration gradient, can be modeled well with the linear diffusion operator $\delta_{ii} \frac{\partial^2}{\partial s^2}$. Incorporating diffusion terms into system (2.1), the following system of reaction-diffusion equations (*PDE PUE Model*) is obtained:

$$\begin{aligned} \frac{\partial P(t, s)}{\partial t} &= -\mu P(t, s)U(t, s) + \delta_{PP} \frac{\partial^2 P(t, s)}{\partial s^2} & P(t, s) > 0 \\ \frac{\partial U(t, s)}{\partial t} &= \mu P(t, s)U(t, s) - eU(t, s) + \delta_{UU} \frac{\partial^2 U(t, s)}{\partial s^2} & U(t, s) > 0 \end{aligned} \quad (3.1)$$

$$\frac{\partial E(t, s)}{\partial t} = eU(t, s) + \delta_{EE} \frac{\partial^2 E(t, s)}{\partial s^2} \quad E(t, s) > 0$$

with boundary and initial conditions

$$P(t, 0) = f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0$$

$$U(t, 0) = g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0$$

$$E(t, 0) = h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0$$

The boundary conditions need to be considered on an individual metal basis. The coefficients δ_{ii} are referred to as the *price diffusivity coefficients*. Since the Production domain consists of minerals being refined into metals diffusing the metal domains rapidly, the diffusion coefficient δ_{PP} will be relatively large. The Use and End-of-Life domains will not see such large quantity diffusion; therefore, the diffusion coefficients δ_{UU} and δ_{EE} will not be quite as large.

The argument can be made that the pricing of metal in a particular domain will affect the movement of metal in another. This effect is apparent in metal pricing determined by competitive commodity exchanges and influences the collection of recycled metal. As pricing of metal in the Production domain changes due to economic conditions (such as changes in mineral availability, transportation costs, demand, or commodity exchanges), the valuation of the metal in the Use and End-of-Life domains will change accordingly, changing the likelihood the metal will stay in those domain. For example, as the price of a precious metal increases or decreases due to an economic condition, the value of jewelry composed of the precious metal (in use or discarded) will likewise increase or decrease, changing the likelihood the jewelry will stay in use or stay

discarded. Also, the metal in the End-of-Life domain can be affected by the price of metal in the Use domain. This behavior can be model based on cross-diffusion in the direction of the affecting domain: $-\delta_{ij} \frac{\partial^2}{\partial s^2}$, where domain j cross diffuses in the direction of domain i . If these cross-diffusion terms are incorporated into system (3.1), the *PDE PUE Model with Price Cross-Diffusion* is obtained:

$$\begin{aligned}
\frac{\partial P(t, s)}{\partial t} &= -\mu P(t, s)U(t, s) + \delta_{PP} \frac{\partial^2 P(t, s)}{\partial s^2} \\
\frac{\partial U(t, s)}{\partial t} &= \mu P(t, s)U(t, s) - eU(t, s) + \delta_{UU} \frac{\partial^2 U(t, s)}{\partial s^2} - \delta_{PU} \frac{\partial^2 P(t, s)}{\partial s^2} \\
\frac{\partial E(t, s)}{\partial t} &= eU(t, s) + \delta_{EE} \frac{\partial^2 E(t, s)}{\partial s^2} - \delta_{PE} \frac{\partial^2 P(t, s)}{\partial s^2} - \delta_{UE} \frac{\partial^2 U(t, s)}{\partial s^2} \quad (3.2) \\
P(t, 0) &= f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0 \\
U(t, 0) &= g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0 \\
E(t, 0) &= h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0
\end{aligned}$$

3.2 PUEP Models

Recall the *ODE PUEP Model* discussed in Section 2.2. If the assumption is made that the domains change with respect to price in a diffusive manner, linear diffusion operators $\delta_{ii} \frac{\partial^2}{\partial s^2}$ can be incorporated into the model. In doing so, the *PDE PUEP Model* will be obtained (the subscript notation for the partial derivative $u_t = \frac{\partial u(t, s)}{\partial t}$ and $u_{ss} = \frac{\partial^2 u(t, s)}{\partial s^2}$ will be used throughout the remainder of this work)

$$\begin{aligned}
P_t &= aV - \mu PU + rE - l_P P + \delta_{PP} P_{SS} \\
U_t &= \mu PU - eU - l_U U + \delta_{UU} U_{SS} \\
E_t &= eU - rE - l_E E + \delta_{EE} E_{SS} \\
P(t, 0) &= f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0 \\
U(t, 0) &= g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0 \\
E(t, 0) &= h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0
\end{aligned} \tag{3.3}$$

As in the *PDE PUE Models*, the boundary and initial condition needs to be considered on an individual metal basis, since each metal will have different conditions associated with it.

The interplay of pricing between domains is a crucial consideration in the dynamics of mineral and metal economics. As in Section 3.1, the assumption will be made that the pricing scheme in the Production domain affects the rate of change in the Use and End-of-Life domains and the pricing in the Use domain affect End-of-Life rate of change. Since recycled material is entering the Production domain, the pricing of that material will play a role (perhaps only a small one) in the rate of change of metal in the Production domain. There exists a greater incentive to sell recycle, meaning the metal will flow out of the End-of-Life domain into the Production domain at a faster rate, if the recycle price is relatively high, affecting the Production domain's rate of change.

Introducing cross-diffusion terms into system (3.3) in order to capture this inter-domain pricing dynamic, the *PDE PUEP Model with Price Cross-Diffusion* is obtained:

$$P_t = aV - \mu PU + rE - l_P P + \delta_{PP} P_{SS} - \delta_{EP} E_{SS} \tag{3.4}$$

$$\begin{aligned}
U_t &= \mu PU - eU - l_U U + \delta_{UU} U_{ss} - \delta_{PU} P_{ss} \\
E_t &= eU - rE - l_E E + \delta_{EE} E_{ss} - \delta_{PE} P_{ss} - \delta_{UE} U_{ss} \\
P(t, 0) &= f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0 \\
U(t, 0) &= g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0 \\
E(t, 0) &= h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0
\end{aligned}$$

3.3 PUEP Models with Inflow Domains

The price diffusion occurring in the Inflow Domains, in practice, should not be as prominent as in the Production, Use, and End-of-Life domains since Inflow Domain metal is consumed quicker and will not undergo the same transformations—processing into downstream products with various price ranges. Thus, the Inflow Domain's price diffusivity coefficients δ_{VV} , δ_{NN} , and δ_{OO} , will approach 0. Incorporating price diffusion into the *ODE PUEP with Inflow Domains Model*, the following system of reaction-diffusion equations will be obtained (*PDE PUEP Model with Inflow Domains*):

$$\begin{aligned}
V_t &= -aV - l_V V + \delta_{VV} V_{ss} \\
P_t &= aV - \mu PU + r_O O + r_N N - r_P P - l_P P + \delta_{PP} P_{ss} \\
N_t &= r_P P - r_N N - l_N N + \delta_{NN} N_{ss} \\
U_t &= \mu PU - eU - l_U U + \delta_{UU} U_{ss} \\
E_t &= eU - r_E E - l_E E + \delta_{EE} E_{ss} \\
O_t &= r_E E - r_O O - l_O O + \delta_{OO} O_{ss} \\
P(t, 0) &= f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0
\end{aligned} \tag{3.5}$$

$$U(t, 0) = g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0$$

$$E(t, 0) = h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0$$

$$V(t, 0) = V_1(t) \quad V(t, S) = V_2(t) \quad V(0, s) = V_0$$

$$N(t, 0) = N_1(t) \quad N(t, S) = N_2(t) \quad N(0, s) = N_0$$

$$O(t, 0) = O_1(t) \quad O(t, S) = O_2(t) \quad O(0, s) = O_0$$

Note that the Inflow domains are affected by the pricing of the other domains, but the Inflow domains' pricing can also affect the rate of change of the other domains. The price of ore and metal used in manufacturing products, represented by the metal residing in the Virgin Metal and Production domains, are intrinsically linked. As the price of ore increases or decreases, that price structure is carried into the Production domain, the main consumer of ore. With a change in product pricing (pricing in the Production domain), the amount of metal leaving the Production domain will be accordingly affected. Also, as the pricing of manufacturing metal changes due to the commodities market, for example, the valuation of the ore will also change accordingly, thus changing the likelihood that more or less ore will be consumed. This linkage of price can be modeled with the price cross-diffusion terms $-\delta_{PV} \frac{\partial^2 P(t,s)}{\partial s^2}$ and $-\delta_{VP} \frac{\partial^2 V(t,s)}{\partial s^2}$.

The pricing scheme of the Old Scrap is also intrinsically linked to the pricing in the Production domain. Since metal in the Old Scrap domain is consumed by the Production domain to produce products, the pricing of the scrap will influence the pricing of the metal products, which in turn will affect the rate at which metal enters and leaves the Production domain. For instance, metal producers will tend to buy more old scrap if

the associated costs decrease, which will transfer metal out of the Old Scrap and into the Production domain at a fast rate. The opposite situation also holds. Metal producers will be more reliant on virgin metal than old scrap if the pricing of old scrap increase, which will limit the rate at which metal is transferred out of the Old Scrap domain and into the Production domain. The cross-diffusion towards the Old Scrap domain by the Production domain can be modeled using the term $-\delta_{OP} \frac{\partial^2 O(t,s)}{\partial s^2}$, and the cross-diffusion towards the Production pricing by the Old Scrap can be modeled with the term $-\delta_{PO} \frac{\partial^2 P(t,s)}{\partial s^2}$.

The pricing influence between New Scrap and Production is limited, but should be considered. Since New Scrap is produced by the discarding of metal during the manufacturing of products, the pricing in the New Scrap domain is linked to the pricing in the Production domain. However, the metal in the New Scrap domain should be assumed to have a short lifetime since new scrap tends to reenter the Production domain rather quickly; therefore, the change in new scrap pricing cause by Production price changes should be minimal. The pricing scheme of the New Scrap domain creates limited influence on other domains, so cross-diffusion towards the New Scrap domain is not introduced into the rate of change of the other domains. The price cross-diffusion towards the pricing scheme of the Production domain by the New Scrap domain can be modeled using the term $-\delta_{PN} \frac{\partial^2 P(t,s)}{\partial s^2}$.

The valuation of metal residing in the End-of-Life domain influences the pricing of metal in the Old Scrap domain. The reasoning behind this influence is similar to that of the Production-New Scrap interaction. Since metal entering the Old Scrap domain comes

from the End-of-Life domain, old scrap pricing is linked to the *recyclate supply* pricing. Since the pricing in the End-of-Life domain changes depending upon many factors, the old scrap pricing will correspondingly change. With a change in old scrap pricing, the rate at which old scrap is consumed by the Production domain will change; thus, the rate at which Old Scrap changes over time will depend on the pricing of End-of-Life metal. However, since the Old Scrap domain is considered to model *recyclate supply availability*, the time that metal resides in the Old Scrap domain is considered to be short, resulting in a small price cross-diffusion. The Old Scrap domain price cross-diffusion towards the End-of-Life domain can be modeled using the term $-\delta_{EO} \frac{\partial^2 E(t,s)}{\partial s^2}$.

Assume that the pricing influences between the Production, Use, and End-of-Life domains are the same as described in system (3.5). Including the cross-diffusion terms into system (3.5), the *PDE PUEP Model with Inflow Domains and Price Cross-Diffusion* will be obtained:

$$\begin{aligned}
V_t &= -aV - l_V V + \delta_{VV} V_{ss} - \delta_{PV} P_{ss} \\
P_t &= aV - \mu PU + r_O O + r_N N - r_P P - l_P P + \delta_{PP} P_{ss} - \delta_{VP} V_{ss} - \delta_{OP} O_{ss} \\
N_t &= r_P P - r_N N - l_N N + \delta_{NN} N_{ss} - \delta_{PN} P_{ss} \\
U_t &= \mu PU - eU - l_U U + \delta_{UU} U_{ss} - \delta_{PU} P_{ss} \\
E_t &= eU - r_E E - l_E E + \delta_{EE} E_{ss} - \delta_{PE} P_{ss} - \delta_{UE} U_{ss} \\
O_t &= r_E E - r_O O - l_O O + \delta_{OO} O_{ss} - \delta_{PO} P_{ss} - \delta_{EO} E_{ss}
\end{aligned}
\tag{3.6}$$

$$\begin{aligned}
P(t, 0) &= f_1(t), & P(t, S) &= f_2(t), & P(0, s) &= P_0 \\
U(t, 0) &= g_1(t), & U(t, S) &= g_2(t), & U(0, s) &= U_0
\end{aligned}$$

$$E(t, 0) = h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0$$

$$V(t, 0) = V_1(t) \quad V(t, S) = V_2(t) \quad V(0, s) = V_0$$

$$N(t, 0) = N_1(t) \quad N(t, S) = N_2(t) \quad N(0, s) = N_0$$

$$O(t, 0) = O_1(t) \quad O(t, S) = O_2(t) \quad O(0, s) = O_0$$

3.4 PUEP Models with Inflow Domains and Stock and Trade Considerations

If stock and trade considerations are incorporated into system (3.5), the *PDE*

PUEP Model with Inflow Domains and Stock and Trade Considerations will be obtained:

$$V_t = -aV - l_V V - x_V V + \delta_{VV} V_{ss}$$

$$P_t = aV - \mu PU + r_O O + r_N N - r_P P - l_P P - x_P P + i_P I + S + \delta_{PP} P_{ss}$$

$$N_t = r_P P - r_N N - l_N N + \delta_{NN} N_{ss}$$

$$U_t = \mu PU - eU - l_U U - x_U U + i_U I + \delta_{UU} U_{ss}$$

$$E_t = eU - r_E E - l_E E + \delta_{EE} E_{ss}$$

$$O_t = r_E E - r_O O - l_O O - x_O O + i_O I + \delta_{OO} O_{ss}$$

(3.7)

$$P(t, 0) = f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0$$

$$U(t, 0) = g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0$$

$$E(t, 0) = h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0$$

$$V(t, 0) = V_1(t) \quad V(t, S) = V_2(t) \quad V(0, s) = V_0$$

$$N(t, 0) = N_1(t) \quad N(t, S) = N_2(t) \quad N(0, s) = N_0$$

$$O(t, 0) = O_1(t) \quad O(t, S) = O_2(t) \quad O(0, s) = O_0$$

Price cross-diffusion terms could also be introduced in system (3.7), producing the following system of reaction-diffusion equations (*PDE PUEP Model with Inflow Domains, Price Cross-Diffusion, and Stock and Trade Considerations*)

$$\begin{aligned}
V_t &= -aV - l_V V - x_V V + \delta_{VV} V_{ss} - \delta_{PV} P_{ss} \\
P_t &= aV - \mu P U + r_O O + r_N N - r_p P - l_P P - x_P P + i_P I + S + \delta_{PP} P_{ss} \\
&\quad - \delta_{VP} V_{ss} - \delta_{OP} O_{ss} \\
N_t &= r_p P - r_N N - l_N N + \delta_{NN} N_{ss} - \delta_{PN} P_{ss} \\
U_t &= \mu P U - eU - l_U U - x_U U + i_U I + \delta_{UU} U_{ss} - \delta_{PU} P_{ss} \\
E_t &= eU - r_E E - l_E E + \delta_{EE} E_{ss} - \delta_{PE} P_{ss} - \delta_{UE} U_{ss} \\
O_t &= r_E E - r_O O - l_O O - x_O O + i_O I + \delta_{OO} O_{ss} - \delta_{PO} P_{ss} - \delta_{EO} E_{ss} \tag{3.8}
\end{aligned}$$

$$\begin{aligned}
P(t, 0) &= f_1(t), & P(t, S) &= f_2(t), & P(0, s) &= P_0 \\
U(t, 0) &= g_1(t), & U(t, S) &= g_2(t), & U(0, s) &= U_0 \\
E(t, 0) &= h_1(t), & E(t, S) &= h_2(t), & E(0, s) &= E_0 \\
V(t, 0) &= V_1(t) & V(t, S) &= V_2(t) & V(0, s) &= V_0 \\
N(t, 0) &= N_1(t) & N(t, S) &= N_2(t) & N(0, s) &= N_0 \\
O(t, 0) &= O_1(t) & O(t, S) &= O_2(t) & O(0, s) &= O_0
\end{aligned}$$

3.5 PUEP Model with a Reformulation of the Virgin Metal Domain

The *Modified PDE PUEP Model with Inflow Domains and Stock and Trade Considerations* will be obtained if price diffusion is introduced into system (2.5):

$$V_t = aR - fV - l_V V - x_V V + \delta_{VV} V_{ss} \tag{3.9}$$

$$P_t = fV - \mu PU + r_O O + r_N N - r_P P - l_P P - x_P P + i_P I + S + \delta_{PP} P_{ss}$$

$$N_t = r_P P - r_N N - l_N N + \delta_{NN} N_{ss}$$

$$U_t = \mu PU - eU - l_U U - x_U U + i_U I + \delta_{UU} U_{ss}$$

$$E_t = eU - r_E E - l_E E + \delta_{EE} E_{ss}$$

$$O_t = r_E E - r_O O - l_O O - x_O O + i_O I + \delta_{OO} O_{ss}$$

$$P(t, 0) = f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0$$

$$U(t, 0) = g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0$$

$$E(t, 0) = h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0$$

$$V(t, 0) = V_1(t) \quad V(t, S) = V_2(t) \quad V(0, s) = V_0$$

$$N(t, 0) = N_1(t) \quad N(t, S) = N_2(t) \quad N(0, s) = N_0$$

$$O(t, 0) = O_1(t) \quad O(t, S) = O_2(t) \quad O(0, s) = O_0$$

Note that the only difference between systems (3.7) and (3.9) is in the first two equations.

As discussed in previous sections, pricing in a domain affects the dynamics of other domains. If we introduce cross-diffusion into system (3.9), the *Modified PDE PUEP Model with Inflow Domains, Price Cross-Diffusion, and Stock and Trade Considerations* will be obtained:

$$V_t = aR - fV - l_V V - x_V V + \delta_{VV} V_{ss} - \delta_{PV} P_{ss}$$

$$P_t = fV - \mu PU + r_O O + r_N N - r_P P - l_P P - x_P P + i_P I + S + \delta_{PP} P_{ss}$$

$$- \delta_{VP} V_{ss} - \delta_{OP} O_{ss} \tag{3.10}$$

$$N_t = r_P P - r_N N - l_N N + \delta_{NN} N_{ss} - \delta_{PN} P_{ss}$$

$$U_t = \mu PU - eU - l_U U - x_U U + i_U I + \delta_{UU} U_{ss} - \delta_{PU} P_{ss}$$

$$E_t = eU - r_E E - l_E E + \delta_{EE} E_{ss} - \delta_{PE} P_{ss} - \delta_{UE} U_{ss}$$

$$O_t = r_E E - r_O O - l_O O - x_O O + i_O I + \delta_{OO} O_{ss} - \delta_{PO} P_{ss} - \delta_{EO} E_{ss}$$

$$P(t, 0) = f_1(t), \quad P(t, S) = f_2(t), \quad P(0, s) = P_0$$

$$U(t, 0) = g_1(t), \quad U(t, S) = g_2(t), \quad U(0, s) = U_0$$

$$E(t, 0) = h_1(t), \quad E(t, S) = h_2(t), \quad E(0, s) = E_0$$

$$V(t, 0) = V_1(t) \quad V(t, S) = V_2(t) \quad V(0, s) = V_0$$

$$N(t, 0) = N_1(t) \quad N(t, S) = N_2(t) \quad N(0, s) = N_0$$

$$O(t, 0) = O_1(t) \quad O(t, S) = O_2(t) \quad O(0, s) = O_0$$

Chapter 4: Stochastic Differential Equation Models for Metal Flows

The deterministic models in the previous two chapter assumed that the domains were differentiable function with respect to time and price; however, the mineral and commodity markets do not behave in this manner. In fact, fluctuations and volatility are commonly associated with metal flows. In order to construct models reflective of this phenomenon, probabilistic models should be explored. In this chapter, the modeling method used to construct probabilistic models will be introduced, along with the stochastic differential equation models. The modeling method will follow the technique discussed in [1] and involves the following 3 steps:

- Step 1. Develop a discrete stochastic model that represents the system under study which experiences random influences. This will allow us to determine the transition probabilities for some small Δt .
- Step 2. The expectation vector and covariance matrix for the transition probabilities are computed.
- Step 3. Develop stochastic differential equations using the expectation vector and covariance matrix by inferring similarities in the discrete and continuous forward Kolmogorov equations.

4.1 PUE Model

Recall the *ODE PUE Model* discussed in Section 2.1. Assume that the metal flow model experiences random influences and that $P(t)$, $U(t)$, and $E(t)$ are discrete random variables representing the amount of metal residing in each domain at some time t .

Assume that $P(t)$ can change by -1 or 0 in some small Δt , which represents a metal unit, such as a metric ton or kilogram. This captures the notion of a closed system discussed in the deterministic *PUE Models*—the Production domain can either transfer metal into the Use domain or no metal transfer occurs. The Use domain can receive metal from the Production domain, lose metal to the End-of-Life domain, or not transfer any metal. Therefore, $U(t)$ can change by -1, 0, or 1. Likewise, the End-of-Life domain can change by 0 or 1.

Assume that the Markov property holds—a future state is determined solely on the present state and irrespective of past states—and the same dynamics of the *ODE PUE Model* are adhered. Let $P(t) = P$, $U(t) = U$ and $E(t) = E$ represent the amount of metal in each domain at time t . For the following state changes

$$\Delta \mathbf{S} = [\Delta S^{(1)}, \Delta S^{(2)}, \Delta S^{(3)}]$$

where

$$\Delta S^{(i)} = [P(\Delta t + t) - P(t), U(\Delta t + t) - U(t), E(\Delta t + t) - E(t)]^T$$

the transition probabilities detailed in *Table 4.1* are obtained.

| Change | Transition Probability |
|---------------------------------|------------------------------|
| $\Delta S^{(1)} = [-1, 1, 0]^T$ | $p_1 = dPU\Delta t$ |
| $\Delta S^{(2)} = [0, -1, 1]^T$ | $p_2 = eU\Delta t$ |
| $\Delta S^{(3)} = [0, 0, 0]^T$ | $p_3 = 1 - \sum_{i=1}^2 p_i$ |

Table 4.1: The state changes and transition probabilities for the *SDE PUE Model*

For example, p_1 is the probability that P transfers 1 unit to U and p_2 is the probability that U transfers 1 unit into E . Notice that the probabilities are proportional to Δt ; thus, Δt must be chosen small enough to allow for $p_i \in [0,1]$ for $i = 1,2$, and 3. The expectation vector for this system is given by

$$\mu(t, P, U, E) = \frac{1}{\Delta t} \sum_{i=1}^3 p_i \Delta S^{(i)} = \begin{pmatrix} -dPU \\ dPU + eU \\ -eU \end{pmatrix}$$

with the covariance matrix

$$\mathbf{C}(t, P, U, E) = \frac{1}{\Delta t} \sum_{i=1}^3 p_i (\Delta S^{(i)} (\Delta S^{(i)})^T) = \begin{pmatrix} dPU & -dPU & 0 \\ -dPU & dPU + eU & -eU \\ 0 & -eU & eU \end{pmatrix}$$

Let $p(t, x_1, x_2, x_3)$ be the probability that $P(t) = x_1, U(t) = x_2$ and $E(t) = x_3$ at time t . This probability distribution solves the discrete Fokker-Planck equation (Kolmogorov forward equation)

$$\begin{aligned} \frac{\partial p(t, x_1, x_2, x_3)}{\partial t} = & - \sum_{i=1}^3 \frac{\partial}{\partial x_i} [\mu_i(t, x_1, x_2, x_3) p(t, x_1, x_2, x_3)] \\ & + \frac{1}{2} \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 \frac{\partial^2}{\partial x_i \partial x_j} [b_{i,k}(t, x_1, x_2, x_3) b_{j,k}(t, x_1, x_2, x_3) p(t, x_1, x_2, x_3)] \end{aligned}$$

where μ_i is the i^{th} -term of the expectation vector μ , and $b_{i,j}$ is the i,j -term of the square root of the covariance matrix. As discussed in [1], the probability distribution also solve the system of stochastic differential equations

$$\frac{d\mathbf{S}(t)}{dt} = \mu(t, x_1, x_2, x_3) + \mathbf{B}(t, x_1, x_2, x_3) \frac{d\mathbf{W}}{dt}$$

where $\mathbf{S}(t) = [P(t), U(t), E(t)]^T$, $\mathbf{B} = \sqrt{\mathbf{C}(t, P, U, E)}$ and $\mathbf{W} = [W_1, W_2, W_3]^T$ are independent Wiener processes. For emphasis, we have the *SDE PUE Model*

$$\begin{aligned} \frac{dP(t)}{dt} &= -dP(t)U(t) + B_{11} \frac{dW_1}{dt} + B_{12} \frac{dW_2}{dt} + B_{13} \frac{dW_3}{dt} \\ \frac{dU(t)}{dt} &= dP(t)U(t) - eU(t) + B_{21} \frac{dW_1}{dt} + B_{22} \frac{dW_2}{dt} + B_{23} \frac{dW_3}{dt} \\ \frac{dE(t)}{dt} &= eU(t) + B_{31} \frac{dW_1}{dt} + B_{32} \frac{dW_2}{dt} + B_{33} \frac{dW_3}{dt} \end{aligned} \quad (4.1)$$

where $\mathbf{B} = \sqrt{\mathbf{C}(t, P, U, E)}$ and

$$\mathbf{C}(t, P, U, E) = \begin{pmatrix} dPU & -dPU & 0 \\ -dPU & dPU + eU & -eU \\ 0 & -eU & eU \end{pmatrix}$$

W_i are three independent Wiener processes. Note that \mathbf{C} is not positive definite; therefore, a unique, real positive definite square root covariance matrix is not guaranteed. The model can be approximated by perturbing the diagonal slightly, creating a positive definite covariance matrix.

4.2 PUEP Model

In this section, we will apply the method discussed in Section 4.1 to develop a probabilistic *PUEP Model* analogous to the time dependent *ODE PUEP Model*. Assume

that the system experience random influences and $P(t)$, $U(t)$, and $E(t)$ are discrete random variables representing the amount of metal residing in each domain at time t . Let ΔS be a change in the domains for a small Δt : that is $\Delta S = [\Delta S^{(1)}, \Delta S^{(2)}, \Delta S^{(3)}]$, where $\Delta S^{(i)} = [P(\Delta t + t) - P(t), U(\Delta t + t) - U(t), E(\Delta t + t) - E(t)]^T$. For the following changes, we will have the corresponding transition probability

| Change | Transition Probability |
|---------------------------------|------------------------------|
| $\Delta S^{(1)} = [1, 0, 0]^T$ | $p_1 = aV\Delta t$ |
| $\Delta S^{(2)} = [-1, 1, 0]^T$ | $p_2 = dPU\Delta t$ |
| $\Delta S^{(3)} = [1, 0, -1]^T$ | $p_3 = rE\Delta t$ |
| $\Delta S^{(4)} = [-1, 0, 0]^T$ | $p_4 = l_P P\Delta t$ |
| $\Delta S^{(5)} = [0, -1, 1]^T$ | $p_5 = eU\Delta t$ |
| $\Delta S^{(6)} = [0, -1, 0]^T$ | $p_6 = l_U U\Delta t$ |
| $\Delta S^{(7)} = [0, 0, -1]^T$ | $p_7 = l_E E\Delta t$ |
| $\Delta S^{(8)} = [0, 0, 0]^T$ | $p_8 = 1 - \sum_{i=1}^7 p_i$ |

Table 4.2: State changes and transition probabilities for the *SDE PUEP Model*

For example, p_1 denotes the probability that 1 unit enters P from the ore reserve and p_2 is the probability that 1 unit leave P and enters U . By computing the expectation vector and covariance matrix for the system as discussed in the previous section, the *SDE PUEP Model* will be obtained

$$\frac{dP(t)}{dt} = aV(t) - dP(t)U(t) + rE(t) - l_P P(t) + \sum_{i=1}^3 B_{1i} \frac{dW_i}{dt} \quad (4.2)$$

$$\frac{dU(t)}{dt} = dP(t)U(t) - eU(t) - l_U U(t) + \sum_{i=1}^3 B_{2i} \frac{dW_i}{dt}$$

$$\frac{dE(t)}{dt} = eU(t) - rE(t) - l_E E(t) + \sum_{i=1}^3 B_{3i} \frac{dW_i}{dt}$$

where $\mathbf{B} = \sqrt{\mathbf{C}}$ and

$$\mathbf{C} = \begin{pmatrix} aV + dPU + rE + l_P P & -dPU & -rE \\ -dPU & dPU + eU + l_U U & -eU \\ -rE & -eU & rE + eU + l_E E \end{pmatrix}$$

Notice that the covariance matrix is positive definite. This implies that a unique real positive definite square root covariance matrix exists.

4.3 PUEP Model with Inflow Domains

Assume that the domains in the *PUEP Model* of Section 2.3 experiences random influences and $V(t)$, $P(t)$, $N(t)$, $U(t)$, $E(t)$, and $O(t)$ are discrete random variables representing the amount of metal in the domains at time t . Assume the same dynamics are at play as in the *ODE PUEP Model with Inflow Domains*: inflows, outflows, loss rates, etc. As in Section 4.1, the transition probabilities for the discrete case are computed (Table 4.3), with the assumption that the Markov property holds, to arrive at a system of stochastic differential equations

| Change | Transition Probability |
|---|------------------------------------|
| $\Delta S^{(1)} = [-1, 1, 0, 0, 0, 0]^T$ | $p_1 = aV\Delta t$ |
| $\Delta S^{(2)} = [-1, 0, 0, 0, 0, 0]^T$ | $p_2 = l_V V\Delta t$ |
| $\Delta S^{(3)} = [0, -1, 0, 1, 0, 0]^T$ | $p_3 = dPU\Delta t$ |
| $\Delta S^{(4)} = [0, 1, 0, 0, 0, -1]^T$ | $p_4 = r_O O\Delta t$ |
| $\Delta S^{(5)} = [0, 1, -1, 0, 0, 0]^T$ | $p_5 = r_N N\Delta t$ |
| $\Delta S^{(6)} = [0, -1, 1, 0, 0, 0]^T$ | $p_6 = r_P P\Delta t$ |
| $\Delta S^{(7)} = [0, -1, 0, 0, 0, 0]^T$ | $p_7 = l_P P\Delta t$ |
| $\Delta S^{(8)} = [0, 0, -1, 0, 0, 0]^T$ | $p_8 = l_N N\Delta t$ |
| $\Delta S^{(9)} = [0, 0, 0, -1, 1, 0]^T$ | $p_9 = eU\Delta t$ |
| $\Delta S^{(10)} = [0, 0, 0, -1, 0, 0]^T$ | $p_{10} = l_U U\Delta t$ |
| $\Delta S^{(11)} = [0, 0, 0, 0, -1, 1]^T$ | $p_{11} = r_E E\Delta t$ |
| $\Delta S^{(12)} = [0, 0, 0, 0, -1, 0]^T$ | $p_{12} = l_E E\Delta t$ |
| $\Delta S^{(13)} = [0, 0, 0, 0, 0, -1]^T$ | $p_{13} = l_O O\Delta t$ |
| $\Delta S^{(14)} = [0, 0, 0, 0, 0, 0]^T$ | $p_{14} = 1 - \sum_{i=1}^{13} p_i$ |

Table 4.3: State changes and transition probabilities for the *SDE PUEP Model with Inflow Domains*

Computing the expectation vector and covariance matrix for the probability distribution, the *SDE PUEP Model with Inflow Domains* is obtained:

$$\begin{aligned}
\frac{dV(t)}{dt} &= -aV(t) - l_V V(t) + \sum_{i=1}^6 B_{1i} \frac{dW_i}{dt} \\
\frac{dP(t)}{dt} &= aV(t) - dP(t)U(t) + r_O O(t) + r_N N(t) - r_P P(t) - l_P P(t) \\
&\quad + \sum_{i=1}^6 B_{2i} \frac{dW_i}{dt}
\end{aligned} \tag{4.3}$$

$$\frac{dN(t)}{dt} = r_p P(t) - r_N N(t) - l_N N(t) + \sum_{i=1}^6 B_{3i} \frac{dW_i}{dt}$$

$$\frac{dU(t)}{dt} = dP(t)U(t) - eU(t) - l_U U(t) + \sum_{i=1}^6 B_{4i} \frac{dW_i}{dt}$$

$$\frac{dE(t)}{dt} = eU(t) - r_E E(t) - l_E E(t) + \sum_{i=1}^6 B_{5i} \frac{dW_i}{dt}$$

$$\frac{dO(t)}{dt} = r_E E(t) - r_O O(t) - l_O O(t) + \sum_{i=1}^6 B_{6i} \frac{dW_i}{dt}$$

where $\mathbf{B} = \sqrt{\mathbf{C}}$, $\mathbf{C} = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6]$ and

$$C_1 = \begin{pmatrix} aV + l_V V \\ -aV \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_2 = \begin{pmatrix} -aV \\ aV + dPU + r_O O + r_N N + r_p P + l_P P \\ -r_N N - r_p P \\ -dPU \\ 0 \\ -r_O O \end{pmatrix},$$

$$C_3 = \begin{pmatrix} 0 \\ -r_N N - r_p P \\ r_N N + r_p P + l_N N \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_4 = \begin{pmatrix} 0 \\ -dPU \\ 0 \\ dPU + eU + l_U U \\ -eU \\ 0 \end{pmatrix},$$

$$C_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -eU \\ eU + r_E E + l_E E \\ -r_E E \end{pmatrix}, \quad C_6 = \begin{pmatrix} 0 \\ -r_O O \\ 0 \\ 0 \\ -r_E E \\ r_O O + r_E E + l_O O \end{pmatrix}$$

The covariance matrix \mathbf{C} is positive definite, therefore a unique real positive definite square root covariance matrix exists.

4.4 PUEP Model with Inflow Domains and Stock and Trade Considerations

As in the previous stochastic models, $V(t)$, $P(t)$, $N(t)$, $U(t)$, $E(t)$, and $O(t)$ are assumed to be discrete random variables representing the amount of metal for time t . By computing the transition probabilities (Table 4.4) and computing the expectation vector and covariance matrix for the probability distribution, we will arrive at a system of stochastic differential equations.

| Change | Transition Probability |
|---|------------------------------------|
| $\Delta S^{(1)} = [-1, 1, 0, 0, 0, 0]^T$ | $p_1 = aV\Delta t$ |
| $\Delta S^{(2)} = [-1, 0, 0, 0, 0, 0]^T$ | $p_2 = l_V V\Delta t$ |
| $\Delta S^{(3)} = [-1, 0, 0, 0, 0, 0]^T$ | $p_3 = x_V V\Delta t$ |
| $\Delta S^{(4)} = [0, -1, 0, 1, 0, 0]^T$ | $p_4 = dPU\Delta t$ |
| $\Delta S^{(5)} = [0, 1, 0, 0, 0, -1]^T$ | $p_5 = r_O O\Delta t$ |
| $\Delta S^{(6)} = [0, 1, -1, 0, 0, 0]^T$ | $p_6 = r_N N\Delta t$ |
| $\Delta S^{(7)} = [0, -1, 1, 0, 0, 0]^T$ | $p_7 = r_P P\Delta t$ |
| $\Delta S^{(8)} = [0, -1, 0, 0, 0, 0]^T$ | $p_8 = l_P P\Delta t$ |
| $\Delta S^{(9)} = [0, -1, 0, 0, 0, 0]^T$ | $p_9 = x_P P\Delta t$ |
| $\Delta S^{(10)} = [0, 1, 0, 0, 0, 0]^T$ | $p_{10} = i_P P\Delta t$ |
| $\Delta S^{(11)} = [0, 1, 0, 0, 0, 0]^T$ | $p_{11} = S\Delta t$ |
| $\Delta S^{(12)} = [0, 0, -1, 0, 0, 0]^T$ | $p_{12} = l_N N\Delta t$ |
| $\Delta S^{(13)} = [0, 0, 0, -1, 1, 0]^T$ | $p_{13} = eU\Delta t$ |
| $\Delta S^{(14)} = [0, 0, 0, -1, 0, 0]^T$ | $p_{14} = l_U U\Delta t$ |
| $\Delta S^{(15)} = [0, 0, 0, -1, 0, 0]^T$ | $p_{15} = x_U U\Delta t$ |
| $\Delta S^{(16)} = [0, 0, 0, 1, 0, 0]^T$ | $p_{16} = i_U I\Delta t$ |
| $\Delta S^{(17)} = [0, 0, 0, 0, -1, 1]^T$ | $p_{17} = r_E E\Delta t$ |
| $\Delta S^{(18)} = [0, 0, 0, 0, -1, 0]^T$ | $p_{18} = l_E E\Delta t$ |
| $\Delta S^{(19)} = [0, 0, 0, 0, 0, -1]^T$ | $p_{19} = l_O O\Delta t$ |
| $\Delta S^{(20)} = [0, 0, 0, 0, 0, -1]^T$ | $p_{20} = x_O O\Delta t$ |
| $\Delta S^{(21)} = [0, 0, 0, 0, 0, 1]^T$ | $p_{21} = i_O I\Delta t$ |
| $\Delta S^{(22)} = [0, 0, 0, 0, 0, 0]^T$ | $p_{22} = 1 - \sum_{i=1}^{21} p_i$ |

Table 4.4: State changes and transition probabilities for the *SDE PUEP Model with Inflow Domains and Stock and Trade Considerations*

Thus, we will obtain the *Time Dependent SDE PUEP Model with Inflow Domains and Stock and Trade Considerations*

$$\begin{aligned}
\frac{dV(t)}{dt} &= -aV(t) - l_V V(t) - x_V V(t) + \sum_{i=1}^6 B_{1i} \frac{dW_i}{dt} \\
\frac{dP(t)}{dt} &= aV(t) - dP(t)U(t) + r_O O(t) + r_N N(t) - r_P P(t) - l_P P(t) - x_P P(t) \\
&\quad + i_P I(t) + S(t) + \sum_{i=1}^6 B_{2i} \frac{dW_i}{dt} \\
\frac{dN(t)}{dt} &= r_P P(t) - r_N N(t) - l_N N(t) + \sum_{i=1}^6 B_{3i} \frac{dW_i}{dt} \\
\frac{dU(t)}{dt} &= dP(t)U(t) - eU(t) - l_U U(t) - x_U U(t) + i_U I(t) + \sum_{i=1}^6 B_{4i} \frac{dW_i}{dt} \\
\frac{dE(t)}{dt} &= eU(t) - r_E E(t) - l_E E(t) + \sum_{i=1}^6 B_{5i} \frac{dW_i}{dt} \\
\frac{dO(t)}{dt} &= r_E E(t) - r_O O(t) - l_O O(t) - x_O O(t) + i_O I(t) + \sum_{i=1}^6 B_{6i} \frac{dW_i}{dt}
\end{aligned} \tag{4.4}$$

where $\mathbf{B} = \sqrt{\mathbf{C}}$, $\mathbf{C} = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6]$ and

$$C_1 = \begin{pmatrix} aV + l_V V + x_V V \\ -aV \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{aligned}
C_2 &= \begin{pmatrix} -aV \\ aV + dPU + r_O O + r_N N + r_P P + l_P P + x_P P + i_P P + S \\ -r_N N - r_P P \\ -dPU \\ 0 \\ -r_O O \end{pmatrix}, \\
C_3 &= \begin{pmatrix} 0 \\ -r_N N - r_P P \\ r_N N + r_P P + l_N N \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_4 = \begin{pmatrix} 0 \\ -dPU \\ 0 \\ dPU + eU + l_U U + x_U U + i_U I \\ -eU \\ 0 \end{pmatrix}, \\
C_5 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -eU \\ eU + r_E E + l_E E \\ -r_E E \end{pmatrix}, \quad C_6 = \begin{pmatrix} 0 \\ -r_O O \\ 0 \\ 0 \\ -r_E E \\ r_O O + r_E E + l_O O + x_O O + i_O I \end{pmatrix}
\end{aligned}$$

Note that the covariance matrix C is positive definite, which implies the existence of a unique real positive definite square root covariance matrix.

4.5 PUEP Model with a Reformulation of the Virgin Metal Domain

The *Modified Time Dependent SDE PUEP Model with Inflow Domains and Stock and Trade Considerations* will be the same as the un-modified version, with the inclusion of the ore inflow term (aR) and the ore processing term (fV). Thus, the transition probabilities, expectation vector, and covariance matrix will be similar to those in Section 4.4.

| Change | Transition Probability |
|---|------------------------------------|
| $\Delta S^{(1)} = [1, 0, 0, 0, 0, 0]^T$ | $p_1 = aR\Delta t$ |
| $\Delta S^{(2)} = [-1, 1, 0, 0, 0, 0]^T$ | $p_2 = fV\Delta t$ |
| $\Delta S^{(3)} = [-1, 0, 0, 0, 0, 0]^T$ | $p_3 = l_V V\Delta t$ |
| $\Delta S^{(4)} = [-1, 0, 0, 0, 0, 0]^T$ | $p_4 = x_V V\Delta t$ |
| $\Delta S^{(5)} = [0, -1, 0, 1, 0, 0]^T$ | $p_5 = dPU\Delta t$ |
| $\Delta S^{(6)} = [0, 1, 0, 0, 0, -1]^T$ | $p_6 = r_O O\Delta t$ |
| $\Delta S^{(7)} = [0, 1, -1, 0, 0, 0]^T$ | $p_7 = r_N N\Delta t$ |
| $\Delta S^{(8)} = [0, -1, 1, 0, 0, 0]^T$ | $p_8 = r_P P\Delta t$ |
| $\Delta S^{(9)} = [0, -1, 0, 0, 0, 0]^T$ | $p_9 = l_P P\Delta t$ |
| $\Delta S^{(10)} = [0, -1, 0, 0, 0, 0]^T$ | $p_{10} = x_P P\Delta t$ |
| $\Delta S^{(11)} = [0, 1, 0, 0, 0, 0]^T$ | $p_{11} = i_P P\Delta t$ |
| $\Delta S^{(12)} = [0, 1, 0, 0, 0, 0]^T$ | $p_{12} = S\Delta t$ |
| $\Delta S^{(13)} = [0, 0, -1, 0, 0, 0]^T$ | $p_{13} = l_N N\Delta t$ |
| $\Delta S^{(14)} = [0, 0, 0, -1, 1, 0]^T$ | $p_{14} = eU\Delta t$ |
| $\Delta S^{(15)} = [0, 0, 0, -1, 0, 0]^T$ | $p_{15} = l_U U\Delta t$ |
| $\Delta S^{(16)} = [0, 0, 0, -1, 0, 0]^T$ | $p_{16} = x_U U\Delta t$ |
| $\Delta S^{(17)} = [0, 0, 0, 1, 0, 0]^T$ | $p_{17} = i_U I\Delta t$ |
| $\Delta S^{(18)} = [0, 0, 0, 0, -1, 1]^T$ | $p_{18} = r_E E\Delta t$ |
| $\Delta S^{(19)} = [0, 0, 0, 0, -1, 0]^T$ | $p_{19} = l_E E\Delta t$ |
| $\Delta S^{(20)} = [0, 0, 0, 0, 0, -1]^T$ | $p_{20} = l_O O\Delta t$ |
| $\Delta S^{(21)} = [0, 0, 0, 0, 0, -1]^T$ | $p_{21} = x_O O\Delta t$ |
| $\Delta S^{(22)} = [0, 0, 0, 0, 0, 1]^T$ | $p_{22} = i_O I\Delta t$ |
| $\Delta S^{(23)} = [0, 0, 0, 0, 0, 0]^T$ | $p_{23} = 1 - \sum_{i=1}^{22} p_i$ |

Table 4.5: State changes and transition probabilities for the *Modified SDE PUEP Model with Inflow Domains and Stock and Trade Considerations*

The *Modified Time Dependent SDE PUEP Model with Inflow Domains and Stock and Trade Considerations* is given by

$$\frac{dV(t)}{dt} = aR(t) - fV(t) - l_V V(t) - x_V V(t) + \sum_{i=1}^6 B_{1i} \frac{dW_i}{dt} \quad (4.5)$$

$$\frac{dP(t)}{dt} = fV(t) - dP(t)U(t) + r_O O(t) + r_N N(t) - r_P P(t) - l_P P(t) - x_P P(t)$$

$$+ i_P I(t) + S(t) + \sum_{i=1}^6 B_{2i} \frac{dW_i}{dt}$$

$$\frac{dN(t)}{dt} = r_P P(t) - r_N N(t) - l_N N(t) + \sum_{i=1}^6 B_{3i} \frac{dW_i}{dt}$$

$$\frac{dU(t)}{dt} = dP(t)U(t) - eU(t) - l_U U(t) - x_U U(t) + i_U I(t) + \sum_{i=1}^6 B_{4i} \frac{dW_i}{dt}$$

$$\frac{dE(t)}{dt} = eU(t) - r_E E(t) - l_E E(t) + \sum_{i=1}^6 B_{5i} \frac{dW_i}{dt}$$

$$\frac{dO(t)}{dt} = r_E E(t) - r_O O(t) - l_O O(t) - x_O O(t) + i_O I(t) + \sum_{i=1}^6 B_{6i} \frac{dW_i}{dt}$$

where $\mathbf{B} = \sqrt{\mathbf{C}}$, $\mathbf{C} = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6]$ and

$$C_1 = \begin{pmatrix} aR + fV + l_V V + x_V V \\ -fV \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} -fV \\ fV + dPU + r_O O + r_N N + r_P P + l_P P + x_P P + i_P P + S \\ -r_N N - r_P P \\ -dPU \\ 0 \\ -r_O O \end{pmatrix},$$

$$\begin{aligned}
C_3 &= \begin{pmatrix} 0 \\ -r_N N - r_p P \\ r_N N + r_p P + l_N N \\ 0 \\ 0 \\ 0 \end{pmatrix}, & C_4 &= \begin{pmatrix} 0 \\ -dPU \\ 0 \\ dPU + eU + l_U U + x_U U + i_U I \\ -eU \\ 0 \end{pmatrix}, \\
C_5 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -eU \\ eU + r_E E + l_E E \\ -r_E E \end{pmatrix}, & C_6 &= \begin{pmatrix} 0 \\ -r_O O \\ 0 \\ 0 \\ -r_E E \\ r_O O + r_E E + l_O O + x_O O + i_O I \end{pmatrix}
\end{aligned}$$

The covariance matrix C is positive definite, which implies that of a unique real positive definite square root covariance matrix exists.

Chapter 5: Numerical Analysis and Graphical User Interface

The metal flow models developed throughout this thesis provided a mathematical representation for the flow of metal through a defined system. In order to fully understand and make predictions using these models, solutions to the models need to be established. Since the models consist of nonlinear systems of ordinary, partial, or stochastic differential, analytical solutions will be hard to acquire; therefore, reliance is placed on numerical methods for the computation of approximate solutions. Since scientific visualization plays an important role in understanding data and information, standalone graphic user interfaces (GUIs) in Matlab were constructed to simulate and visualize the approximate solutions. In Section 5.1, the numerical methods used to compute the solutions for the models are presented. Section 5.2 describes the GUI's constructed to visualize the solutions.

5.1 Numerical Methods

There are various methods to compute the numerical solutions to systems of ordinary, partial, and stochastic differential equations [18]. In this chapter, first order finite difference schemes will be used. In Section 5.1.1, the Euler method, an explicit finite difference scheme, will be used to solve the models consisting of ordinary differential equations. In Section 5.1.2, an implicit finite difference scheme, the backward

in time-central in space method (BTCS), will be constructed to solve the models consisting of partial differential equations. Section 5.1.3 uses the Euler-Maruyama method to compute the numerical solution to the models consisting of stochastic differential equations.

5.1.1 Numerical Methods for the Ordinary Differential Equation Models

The ordinary differential equation models developed in Chapter 4 are initial value problems of the form

$$\frac{d\mathbf{F}(t)}{dt} = \begin{pmatrix} \frac{dx(t)}{dt} \\ \frac{dy(t)}{dt} \\ \vdots \\ \frac{dz(t)}{dt} \end{pmatrix} = \begin{pmatrix} f_1(k, t, x, y, \dots, z) \\ f_2(k, t, x, y, \dots, z) \\ \vdots \\ f_M(k, t, x, y, \dots, z) \end{pmatrix}$$

with initial conditions

$$\begin{pmatrix} x(0) = x_0 \\ y(0) = y_0 \\ \vdots \\ z(0) = z_0 \end{pmatrix}$$

A first order finite difference scheme (Euler's method) will be used to approximate the analytical solution for the models. The time domain $[0, T]$ is partitioned as $t_i = i\Delta t$ for $i = 0, \dots, N - 1$ where $\Delta t = \frac{T}{N}$. Let $\mathbf{F}_i \approx \mathbf{F}(t_i)$. The first order difference approximation will be of the form

$$\begin{aligned}
x_{i+1} &= x_i + \Delta t f_1(k, t_i, x_i, y_i, \dots, z_i), & x(0) &= x_0 \\
y_{i+1} &= y_i + \Delta t f_2(k, t_i, x_i, y_i, \dots, z_i), & y(0) &= y_0 \\
&\vdots \\
z_{i+1} &= z_i + \Delta t f_M(k, t_i, x_i, y_i, \dots, z_i), & z(0) &= z_0
\end{aligned} \tag{5.1}$$

The stability of (5.1) is established by observing the behavior a small perturbation $\delta > 0$ added to (x_i, y_i, \dots, z_i) will affect (5.1) over time [38]. In other words, if the finite difference scheme (5.1) is stable, then a small change in the initial condition will not change the solution greatly. Substituting $(x_i + \delta x_i, y_i + \delta y_i, \dots, z_i + \delta z_i)$ into (5.1), we will obtain

$$\begin{aligned}
x_{i+1} + \delta x_{i+1} &= x_i + \delta x_i + \Delta t f_1(k, t_i, x_i + \delta x_i, y_i + \delta y_i, \dots, z_i + \delta z_i) \\
y_{i+1} + \delta y_{i+1} &= y_i + \delta y_i + \Delta t f_2(k, t_i, x_i + \delta x_i, y_i + \delta y_i, \dots, z_i + \delta z_i) \\
&\vdots \\
z_{i+1} + \delta z_{i+1} &= z_i + \delta z_i + \Delta t f_M(k, t_i, x_i + \delta x_i, y_i + \delta y_i, \dots, z_i + \delta z_i)
\end{aligned} \tag{5.2}$$

Apply Taylor series expansion to the f_i terms, we will obtain

$$\begin{aligned}
x_{i+1} + \delta x_{i+1} &= x_i + \delta x_i \\
&+ \Delta t \left(\delta x_i \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \delta y_i \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right. \\
&\quad \left. + \dots + \delta z_i \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right)
\end{aligned} \tag{5.3}$$

$$y_{i+1} + \delta y_{i+1} = y_i + \delta y_i$$

$$+ \Delta t \left(\delta x_i \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \delta y_i \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right. \\ \left. + \dots + \delta z_i \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right) \\ \vdots$$

$$z_{i+1} + \delta z_{i+1} = z_i + \delta z_i$$

$$+ \Delta t \left(\delta x_i \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \delta y_i \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right. \\ \left. + \dots + \delta z_i \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right)$$

Since (x_{i+1}, x_i) , (y_{i+1}, y_i) , \dots , and (z_{i+1}, z_i) are linked by the Euler relationship (5.1),

(5.3) will reduce to

$$\delta x_{i+1} = \delta x_i + \delta \Delta t \left(x_i \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + y_i \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \dots \right. \\ \left. + z_i \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right) \quad (5.4)$$

$$\begin{aligned}
\delta y_{i+1} &= \delta y_i + \delta \Delta t \left(x_i \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} + y_i \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} + \dots \right. \\
&\quad \left. + z_i \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} \right) \\
&\quad \vdots \\
\delta z_{i+1} &= \delta z_i + \delta \Delta t \left(x_i \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} + y_i \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} + \dots \right. \\
&\quad \left. + z_i \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} \right)
\end{aligned}$$

Letting $g_1 x_i = x_{i+1}$, $g_2 y_i = y_{i+1}$, and $g_M z_i = z_{i+1}$, where g_k are called growth factor, we will obtain

$$\begin{aligned}
g_1 &= 1 + \Delta t \left(\frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} + \frac{y_i}{x_i} \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} + \dots \right. \\
&\quad \left. + \frac{z_i}{x_i} \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{y=y_i \\ \vdots \\ z=z_i}} \right)
\end{aligned} \tag{5.5}$$

$$\begin{aligned}
g_2 &= 1 + \Delta t \left(\frac{x_i}{y_i} \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} + \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} + \dots \right. \\
&\quad \left. + \frac{z_i}{y_i} \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} \right) \\
&\quad \vdots \\
g_M &= 1 + \Delta t \left(\frac{x_i}{z_i} \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} + \frac{y_i}{z_i} \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} + \dots \right. \\
&\quad \left. + \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} \right)
\end{aligned}$$

For the scheme to be stable, the growth factors must be in $[-1, 1]$. Thus, we will obtain the following conditions to guarantee stability

$$\begin{aligned}
&\Delta t \left(\frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} + \frac{y_i}{x_i} \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} + \dots \right. \\
&\quad \left. + \frac{z_i}{x_i} \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ z=z_i}} \right) \in [-2, 0]
\end{aligned} \tag{5.6}$$

$$\begin{aligned}
& \Delta t \left(\frac{x_i}{y_i} \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \dots \right. \\
& \quad \left. + \frac{z_i}{y_i} \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right) \in [-2, 0] \\
& \quad \vdots \\
& \Delta t \left(\frac{x_i}{z_i} \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \frac{y_i}{z_i} \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \dots \right. \\
& \quad \left. + \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \right) \in [-2, 0]
\end{aligned}$$

Conditions (5.6) illustrate that the numerical scheme (5.1) will be unstable if

$$T(f) = \frac{\partial f_1(k, t, x, y, \dots, z)}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \frac{\partial f_2(k, t, x, y, \dots, z)}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} + \dots + \frac{\partial f_M(k, t, x, y, \dots, z)}{\partial z} \Big|_{\substack{x=x_i \\ y=y_i \\ \vdots \\ z=z_i}} \text{ is positive}$$

(note that (x_i, y_i, \dots, z_i) are defined as positive since they describe positive amounts of metals). If $T(f)$ is negative, then Δt can be choose to stabilize the algorithm. Also note

that the stability of the scheme is dependent on the size of $\left(\frac{x_i}{y_i}, \dots, \frac{x_i}{z_i}\right)$, $\left(\frac{y_i}{x_i}, \dots, \frac{y_i}{z_i}\right)$, and

$\left(\frac{z_i}{x_i}, \dots, \frac{z_i}{y_i}\right)$, implying that the ratio of quantities of metal in each domain at each time step

must be considered.

In this section, a list of the first order finite difference approximations for the ordinary differential equation metal flow models is provided. In Appendix C, Matlab

scripts for these schemes are presented with solution outputs. Only two schemes are presented since the methods can be generalized.

Finite Difference Method for the ODE PUE Model and the ODE PUEP Model

The finite difference scheme for the *ODE PUE Model* can be obtained by setting the coefficients a , r , l_P , l_U , and l_E of (5.7) equal to 0. The source code for the method and selected outputs are given in Appendix C.6.

$$\begin{aligned}
 P_0 &= P(0) & P_{i+1} &= P_i + \Delta t (aV_i - \mu P_i U_i + rE_i - l_P P_i) \\
 U_0 &= U(0) & U_{i+1} &= U_i + \Delta t (\mu P_i U_i - eU_i - l_U U_i) \\
 E_0 &= E(0) & E_{i+1} &= E_i + \Delta t (eU_i - rE_i - l_E E_i)
 \end{aligned} \tag{5.7}$$

Finite Difference Method for the ODE PUEP Models with Inflow Domains

The finite difference method for the inflow domain models can be obtained from (5.8) by setting relevant coefficients equal to 0. The source code and selected outputs are presented in Appendix C.7.

$$\begin{aligned}
 V_0 &= V(0) & U_0 &= U(0) \\
 P_0 &= P(0) & E_0 &= E(0) \\
 N_0 &= N(0) & O_0 &= O(0) \\
 V_{i+1} &= V_i + \Delta t (aR_i - fV_i - l_V V_i - x_V V_i) \\
 P_{i+1} &= P_i + \Delta t (fV_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i - l_P P_i - x_P P_i + i_P I_i + S_i) \\
 N_{i+1} &= N_i + \Delta t (r_P P_i - r_N N_i - l_N N_i)
 \end{aligned} \tag{5.8}$$

$$U_{i+1} = U_i + \Delta t (\mu P_i U_i - e U_i - l_U U_i - x_U U_i + i_U I_i)$$

$$E_{i+1} = E_i + \Delta t (e U_i - r_E E_i - l_E E_i)$$

$$O_{i+1} = O_i + \Delta t (r_E E_i - r_O O_i - l_O O_i - x_O O_i + i_O I_i)$$

5.1.2 Numerical Methods for the Partial Differential Equation Models

The partial differential equation models developed in Chapter 3 are systems of reaction-diffusion equations with and without price cross-diffusion. The systems without cross-diffusion are of the form

$$\frac{\partial \mathbf{F}(t)}{\partial t} = \begin{pmatrix} \frac{\partial x(t)}{\partial t} \\ \frac{\partial y(t)}{\partial t} \\ \vdots \\ \frac{\partial z(t)}{\partial t} \end{pmatrix} = \begin{pmatrix} f_1(k, t, x, y, \dots, z) \\ f_2(k, t, x, y, \dots, z) \\ \vdots \\ f_M(k, t, x, y, \dots, z) \end{pmatrix} + \begin{pmatrix} \delta_{xx} \frac{\partial x(t)}{\partial s} \\ \delta_{yy} \frac{\partial y(t)}{\partial s} \\ \vdots \\ \delta_{zz} \frac{\partial z(t)}{\partial s} \end{pmatrix}$$

and the systems with cross-diffusion are of the form

$$\frac{\partial \mathbf{F}(t)}{\partial t} = \begin{pmatrix} f_1(k, t, x, y, \dots, z) \\ f_2(k, t, x, y, \dots, z) \\ \vdots \\ f_M(k, t, x, y, \dots, z) \end{pmatrix} + \begin{pmatrix} \delta_{xx} \frac{\partial x(t)}{\partial s} \\ \delta_{yy} \frac{\partial y(t)}{\partial s} \\ \vdots \\ \delta_{zz} \frac{\partial z(t)}{\partial s} \end{pmatrix} - \begin{pmatrix} \delta_{yx} \frac{\partial y(t)}{\partial s} \\ \delta_{xy} \frac{\partial x(t)}{\partial s} \\ \vdots \\ \delta_{xz} \frac{\partial x(t)}{\partial s} \end{pmatrix} - \dots - \begin{pmatrix} \delta_{zx} \frac{\partial x(t)}{\partial s} \\ \delta_{zy} \frac{\partial y(t)}{\partial s} \\ \vdots \\ \delta_{yz} \frac{\partial y(t)}{\partial s} \end{pmatrix}$$

The numerical solutions to these models will be computed using the implicit finite difference scheme BTCS [12]. The time domain $[0, T]$ will be partitioned as $t_j = j\Delta t$ for $j = 0, \dots, N - 1$ where $\Delta t = \frac{T}{N}$. The price domain $[0, S]$ will be partitioned as $s_i = i\Delta s$ for $i = 0, \dots, M - 1$ where $\Delta s = \frac{S}{M}$. The BTCS scheme approximates the partial derivatives as follows:

$$\frac{\partial D}{\partial t} \approx \frac{D_i^{j+1} - D_i^j}{\Delta t} \quad (5.9)$$

and

$$\frac{\partial^2 D}{\partial s^2} \approx \frac{D_{i+1}^{j+1} - 2D_i^{j+1} + D_{i-1}^{j+1}}{\Delta s^2} \quad (5.10)$$

Applying (5.9), (5.10), and including Dirichlet boundary conditions to the partial differential equation models, the numerical solution D_i^j for each domain D will be obtained at each time step by solving the system of equations

$$D^{j+1} = \mathbf{A}^{-1} \mathbf{x}$$

where

$$D^{j+1} = \begin{pmatrix} D_1^{j+1} \\ D_2^{j+1} \\ \vdots \\ D_{N-1}^{j+1} \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} b & a & 0 & \dots & 0 \\ a & b & a & \dots & 0 \\ 0 & a & b & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & b \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1^j \\ x_2^j \\ \vdots \\ x_{N-1}^j \end{pmatrix}$$

In this section, the BTCS schemes will be presented for the partial differential equation models. A generalization for the three- and six-domain models is made. In Appendix C, Matlab scripts for these schemes are presented along with solution outputs.

Three-Domain Models

The four *PDE PUE* and *PUEP Models* of Sections 3.1 and 3.2 are generalized into one scheme. All four models will have the same matrix coefficients a and b , but will have various λ_i^j and γ_i^j terms. At each time step, 3 systems need to be solved.

System 1: The solution to system (5.11) is an approximate solution to $P(t)$ at time $j + 1$.

$$P^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda_1^j P_1^j + \gamma_1^j \\ \lambda_2^j P_2^j + \gamma_2^j \\ \vdots \\ \lambda_{N-1}^j P_{N-1}^j + \gamma_{N-1}^j \end{pmatrix} \quad (5.11)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{PP}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{PP}}{(\Delta S)^2} \quad (5.12)$$

PUE Model:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j, \quad \gamma_i^j = 0 \quad (5.13)$$

PUE Model with Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j, \quad \gamma_i^j = 0 \quad (5.14)$$

PUEP Model:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j + l_P, \quad \gamma_i^j = -r E_i^j - \alpha V_i^j \quad (5.15)$$

PUEP Model with Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j + l_P, \quad \gamma_i^j = \frac{\delta_{EP}}{(\Delta S)^2} (E_{i+1}^{j+1} - 2E_i^{j+1} + E_{i-1}^{j+1}) - \alpha V_i^j - r E_i^j \quad (5.16)$$

System 2: The solution to system (5.17) is an approximate solution to $U(t)$ at time $j + 1$.

$$U^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda_1^j U_1^{j+1} + \gamma_1^{j+1} \\ \lambda_2^j U_2^{j+1} + \gamma_2^{j+1} \\ \vdots \\ \lambda_{N-1}^j U_{N-1}^{j+1} + \gamma_{N-1}^{j+1} \end{pmatrix} \quad (5.17)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{UU}}{(\Delta s)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{UU}}{(\Delta s)^2} \quad (5.18)$$

PUE Model:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e, \quad \gamma_i^j = 0 \quad (5.19)$$

PUE Model with Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e, \quad \gamma_i^j = \frac{\delta_{UU}}{(\Delta s)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.20)$$

PUEP Model:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e + l_U, \quad \gamma_i^j = 0 \quad (5.21)$$

PUEP Model with Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e + l_U, \quad \gamma_i^j = \frac{\delta_{PU}}{(\Delta s)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.22)$$

System 3: The solution to system (5.23) is an approximate solution to $E(t)$ at time $j + 1$.

$$E^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda E_1^j + \gamma_1^{j+1} \\ \lambda E_2^j + \gamma_2^{j+1} \\ \vdots \\ \lambda E_{N-1}^j + \gamma_{N-1}^{j+1} \end{pmatrix} \quad (5.23)$$

Matrix coefficients for all models:

$$a = -\frac{\Delta t \delta_{EE}}{(\Delta s)^2}, \quad b = 1 + \frac{2 \Delta t \delta_{EE}}{(\Delta s)^2} \quad (5.24)$$

PUE Model:

$$\lambda_i^j = 1, \quad \gamma_i^j = \Delta t e U_i^j \quad (5.25)$$

PUE Model with Price Cross-Diffusion:

$$\begin{aligned}\lambda_i^j &= 1, & \gamma_i^j &= \Delta t e U_i^j - \frac{\Delta t \delta_{PE}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2 P_i^{j+1} + P_{i-1}^{j+1}) \\ & & & - \frac{\Delta t \delta_{UE}}{(\Delta S)^2} (U_{i+1}^{j+1} - 2 U_i^{j+1} + U_{i-1}^{j+1})\end{aligned}\tag{5.26}$$

PUEP Model:

$$\lambda_i^j = 1 - \Delta t (l_E + r), \quad \gamma_i^j = \Delta t e U_i^j\tag{5.27}$$

PUEP Model with Price Cross-Diffusion:

$$\begin{aligned}\lambda_i^j &= 1 - \Delta t (l_E + r), \\ \gamma_i^j &= \Delta t \left(e U_i^j - \frac{\delta_{PE}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2 P_i^{j+1} + P_{i-1}^{j+1}) - \frac{\delta_{UE}}{(\Delta S)^2} (U_{i+1}^{j+1} - 2 U_i^{j+1} + U_{i-1}^{j+1}) \right)\end{aligned}\tag{5.28}$$

Six-Domain Models

The six *PDE Inflow Domain Models* of Sections 3.3 through 3.5 are generalized into one scheme. All six models will have the same matrix coefficients a and b , but will have various λ_i^j and γ_i^j terms. At each time step, 6 systems need to be solved.

System 1: The solution to the system (5.29) is an approximate solution to $V(t)$ at time $j + 1$.

$$V^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda_1^j V_1^j + \gamma_1^j \\ \lambda_2^j V_2^j + \gamma_2^j \\ \vdots \\ \lambda_{M-1}^j V_{M-1}^j + \gamma_{M-1}^j \end{pmatrix}\tag{5.29}$$

Matrix coefficients for all models:

$$a = \frac{\delta_{VV}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{VV}}{(\Delta S)^2} \quad (5.30)$$

PUEP Model with Inflow Domains:

$$\lambda_i^j = -\frac{1}{\Delta t} + l_V + \alpha, \quad \gamma_i^j = 0 \quad (5.31)$$

PUEP Model with Inflow Domains and Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} + l_V + \alpha, \quad \gamma_i^j = \frac{\delta_{PP}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.32)$$

PUEP Model with Inflow Domains and Stock and Trade Considerations:

$$\lambda_i^j = -\frac{1}{\Delta t} + l_V + \alpha, \quad \gamma_i^j = 0 \quad (5.33)$$

PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} + l_V + \alpha, \quad \gamma_i^j = \frac{\delta_{PP}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.34)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations:

$$\lambda_i^j = -\frac{1}{\Delta t} + l_V + f, \quad \gamma_i^j = \alpha R_i^j \quad (5.35)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} + l_V + f, \quad \gamma_i^j = \frac{\delta_{PV}}{(\Delta S)^2} (P_{i+1}^j - 2P_i^j + P_{i-1}^j) - \alpha R_i^j \quad (5.36)$$

System 2: The solution to system (5.37) is an approximate solution to $P(t)$ at time $j + 1$.

$$P^{j+1} = A^{-1} \begin{pmatrix} \lambda_1^j P_1^j + \gamma_1^j \\ \lambda_2^j P_2^j + \gamma_2^j \\ \vdots \\ \lambda_{M-1}^j P_{M-1}^j + \gamma_{M-1}^j \end{pmatrix} \quad (5.37)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{PP}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{PP}}{(\Delta S)^2} \quad (5.38)$$

PUEP Model with Inflow Domains:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j + r_p + l_p, \quad \gamma_i^j = -r_O O_i^j - r_N N_i^j - \alpha V_i^j \quad (5.39)$$

PUEP Model with Inflow Domains and Price Cross-Diffusion:

$$\begin{aligned} \lambda_i^j &= -\frac{1}{\Delta t} + \mu U_i^j + r_p + l_p, \\ \gamma_i^j &= \frac{\delta_{VP}}{(\Delta S)^2} (V_{i+1}^{j+1} - 2V_i^{j+1} + V_{i-1}^{j+1}) + \frac{\delta_{OP}}{(\Delta S)^2} (O_{i+1}^{j+1} - 2O_i^{j+1} + O_{i-1}^{j+1}) - r_O O_i^j \\ &\quad - r_N N_i^j - \alpha V_i^j \end{aligned} \quad (5.40)$$

PUEP Model with Inflow Domains and Stock and Trade Considerations:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j + r_p + l_p + x_p, \quad \gamma_i^j = -r_O O_i^j - r_N N_i^j - \alpha V_i^j - i_P I^j - S^j \quad (5.41)$$

PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\begin{aligned} \lambda_i^j &= -\frac{1}{\Delta t} + \mu U_i^j + r_p + l_p + x_p, \\ \gamma_i^j &= \frac{\delta_{VP}}{(\Delta S)^2} (V_{i+1}^{j+1} - 2V_i^{j+1} + V_{i-1}^{j+1}) + \frac{\delta_{OP}}{(\Delta S)^2} (O_{i+1}^{j+1} - 2O_i^{j+1} + O_{i-1}^{j+1}) - r_O O_i^j \\ &\quad - r_N N_i^j - \alpha V_i^j - i_P I^j - S^j \end{aligned} \quad (5.42)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations:

$$\lambda_i^j = -\frac{1}{\Delta t} + \mu U_i^j + r_p + l_p + x_p, \quad \gamma_i^j = -r_o O_i^j - r_N N_i^j - f V_i^j - i_p I^j - S^j \quad (5.43)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\begin{aligned} \lambda_i^j &= -\frac{1}{\Delta t} + \mu U_i^j + r_p + l_p + x_p, \\ \gamma_i^j &= \frac{\delta_{VP}}{(\Delta S)^2} (V_{i+1}^{j+1} - 2V_i^{j+1} + V_{i-1}^{j+1}) + \frac{\delta_{OP}}{(\Delta S)^2} (O_{i+1}^{j+1} - 2O_i^{j+1} + O_{i-1}^{j+1}) - r_o O_i^j \\ &\quad - r_N N_i^j - f V_i^j - i_p I^j - S^j \end{aligned} \quad (5.44)$$

System 3: The solution to system (5.45) is an approximate solution to $N(t)$ at time $j + 1$.

$$N^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda N_1^j + \gamma_1^j \\ \lambda N_2^j + \gamma_2^j \\ \vdots \\ \lambda N_{M-1}^j + \gamma_{M-1}^j \end{pmatrix} \quad (5.45)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{NN}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{NN}}{(\Delta S)^2} \quad (5.46)$$

PUEP Model with Inflow Domains:

$$\lambda = -\frac{1}{\Delta t} + l_N + r_N, \quad \gamma_i^j = -r_p P_i^j \quad (5.47)$$

PUEP Model with Inflow Domains and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_N + r_N, \quad \gamma_i^j = -r_p P_i^j + \frac{\delta_{PN}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.48)$$

PUEP Model with Inflow Domains and Stock and Trade Considerations:

$$\lambda = -\frac{1}{\Delta t} + l_N + r_N, \quad \gamma_i^j = -r_p P_i^j \quad (5.49)$$

PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-

Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_N + r_N, \quad \gamma_i^j = -r_p P_i^j + \frac{\delta_{PN}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.50)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations:

$$\lambda = -\frac{1}{\Delta t} + l_N + r_N, \quad \gamma_i^j = -r_p P_i^j \quad (5.51)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price

Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_N + r_N, \quad \gamma_i^j = -r_p P_i^j + \frac{\delta_{PN}}{(\Delta S)^2} (P_{i+1}^j - 2P_i^j + P_{i-1}^j) \quad (5.52)$$

System 4: The solution to system (5.53) is an approximate solution for $U(t)$ at time $j + 1$.

$$U^{j+1} = A^{-1} \begin{pmatrix} \lambda_1^j U_1^j + \gamma_1^j \\ \lambda_2^j U_2^j + \gamma_2^j \\ \vdots \\ \lambda_{M-1}^j U_{M-1}^j + \gamma_{M-1}^j \end{pmatrix} \quad (5.53)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{UU}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{UU}}{(\Delta S)^2} \quad (5.54)$$

PUEP Model with Inflow Domains:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e + l_U, \quad \gamma_i^j = 0 \quad (5.55)$$

PUEP Model with Inflow Domains and Price Cross-Diffusion:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e + l_U, \quad \gamma_i^j = \frac{\delta_{PU}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) \quad (5.56)$$

PUEP Model with Inflow Domains and Stock and Trade Considerations:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e + l_U + x_U, \quad \gamma^j = -i_U I^j \quad (5.57)$$

PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\begin{aligned} \lambda_i^j &= -\frac{1}{\Delta t} - \mu P_i^j + e + l_U + x_U, \\ \gamma_i^j &= \frac{\delta_{PU}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) - i_U I^j \end{aligned} \quad (5.58)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations:

$$\lambda_i^j = -\frac{1}{\Delta t} - \mu P_i^j + e + l_U + x_U, \quad \gamma^j = -i_U I^j \quad (5.59)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\begin{aligned} \lambda_i^j &= -\frac{1}{\Delta t} - \mu P_i^j + e + l_U + x_U, \\ \gamma_i^j &= \frac{\delta_{PU}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) - i_U I^j \end{aligned} \quad (5.60)$$

System 5: The solution to system (5.61) is an approximate solution for $E(t)$ at time $j + 1$.

$$E^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda E_1^j + \gamma_1^j \\ \lambda E_2^j + \gamma_2^j \\ \vdots \\ \lambda E_{M-1}^j + \gamma_{M-1}^j \end{pmatrix} \quad (5.61)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{EE}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2\delta_{EE}}{(\Delta S)^2} \quad (5.62)$$

PUEP Model with Inflow Domains:

$$\lambda = -\frac{1}{\Delta t} + l_E + r, \quad \gamma_i^j = -eU_i^j \quad (5.63)$$

PUEP Model with Inflow Domains and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_E + r, \quad (5.64)$$

$$\gamma_i^j = -eU_i^j + \frac{\delta_{PE}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) + \frac{\delta_{UE}}{(\Delta S)^2} (U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1})$$

PUEP Model with Inflow Domains and Stock and Trade Considerations:

$$\lambda = -\frac{1}{\Delta t} + l_E + r, \quad \gamma_i^j = -eU_i^j \quad (5.65)$$

PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_E + r, \quad (5.66)$$

$$\gamma_i^j = -eU_i^j + \frac{\delta_{PE}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) + \frac{\delta_{UE}}{(\Delta S)^2} (U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1})$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations:

$$\lambda = -\frac{1}{\Delta t} + l_E + r, \quad \gamma_i^j = -eU_i^j \quad (5.67)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_E + r, \quad (5.68)$$

$$\gamma_i^j = -eU_i^j + \frac{\delta_{PE}}{(\Delta S)^2}(P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) + \frac{\delta_{UE}}{(\Delta S)^2}(U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1})$$

System 6: The solution to system (5.69) is an approximate solution for $O(t)$ at time $j + 1$.

$$O^{j+1} = \mathbf{A}^{-1} \begin{pmatrix} \lambda O_1^j + \gamma_1^j \\ \lambda O_2^j + \gamma_2^j \\ \vdots \\ \lambda O_{M-1}^j + \gamma_{M-1}^j \end{pmatrix} \quad (5.69)$$

Matrix coefficients for all models:

$$a = \frac{\delta_{OO}}{(\Delta S)^2}, \quad b = -\frac{1}{\Delta t} - \frac{2 \delta_{OO}}{(\Delta S)^2} \quad (5.70)$$

PUEP Model with Inflow Domains:

$$\lambda = -\frac{1}{\Delta t} + l_O + r_O, \quad \gamma_i^j = -r_E E_i^j \quad (5.71)$$

PUEP Model with Inflow Domains and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_O + r_O, \quad (5.72)$$

$$\gamma_i^j = -r_E E_i^j + \frac{\delta_{PO}}{(\Delta S)^2}(P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) + \frac{\delta_{EO}}{(\Delta S)^2}(E_{i+1}^{j+1} - 2E_i^{j+1} + E_{i-1}^{j+1})$$

PUEP Model with Inflow Domains and Stock and Trade Considerations:

$$\lambda = -\frac{1}{\Delta t} + l_O + r_O + x_O, \quad \gamma_i^j = -r_E E_i^j - i_O I^j \quad (5.73)$$

PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_O + r_O + x_O, \quad (5.74)$$

$$\gamma_i^j = -r_E E_i^j + \frac{\delta_{PO}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) + \frac{\delta_{EO}}{(\Delta S)^2} (E_{i+1}^{j+1} - 2E_i^{j+1} + E_{i-1}^{j+1}) - i_O I^j$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations:

$$\lambda = -\frac{1}{\Delta t} + l_O + r_O + x_O, \quad \gamma_i^j = -r_E E_i^j - i_O I^j \quad (5.75)$$

Modified PUEP Model with Inflow Domains and Stock, Trade Considerations, and Price Cross-Diffusion:

$$\lambda = -\frac{1}{\Delta t} + l_O + r_O + x_O, \quad \gamma_i^j = -r_E E_i^j + \frac{\delta_{PO}}{(\Delta S)^2} (P_{i+1}^{j+1} - 2P_i^{j+1} + P_{i-1}^{j+1}) + \frac{\delta_{EO}}{(\Delta S)^2} (E_{i+1}^{j+1} - 2E_i^{j+1} + E_{i-1}^{j+1}) - i_O I^j \quad (5.76)$$

5.1.3 Numerical Methods for the Stochastic Differential Equation Models

The analytical solutions to the stochastic differential equation models developed in Chapter 4 will be difficult to obtain; therefore, a numerical scheme, namely the Euler-Maruyama method [1], will be used to approximate the analytical solutions to the models. Recall that the stochastic differential equation metal flow models were of the following form

$$\frac{d\mathbf{S}(t)}{dt} = \boldsymbol{\mu}(t, X_1(t), X_2(t), \dots, X_N(t)) + \mathbf{B}(t, X_1(t), X_2(t), \dots, X_N(t)) \frac{d\mathbf{W}}{dt}$$

where $\mathbf{S}(t) = [X_1(t), X_2(t), \dots, X_N(t)]^T$, $\mathbf{B} = \sqrt{\mathbf{C}(t, X_1(t), X_2(t), \dots, X_N(t))}$,

$\mathbf{C}(t, X_1(t), X_2(t), \dots, X_N(t))$ is the covariance matrix, and $\mathbf{W} = [W_1, W_2, \dots, W_N]^T$ are

independent Weiner processes. The Euler-Maruyama scheme for the models will have the form

$$\mathbf{S}_0(\omega) = \mathbf{S}(0, \omega)$$

$$\mathbf{S}_{i+1}(\omega) = \mathbf{S}_i(\omega) + \Delta t \boldsymbol{\mu}(t_i, \mathbf{S}_i(\omega)) + \mathbf{B}(t_i, \mathbf{S}_i(\omega)) \Delta \mathbf{W}_i(\omega)$$

for $i = 0, 1, \dots, N - 1$ with $\mathbf{S}_i(\omega) \approx \mathbf{S}_i(t_i, \omega)$ $\Delta t = \frac{T}{N}$, $t_i = i\Delta t$, $\Delta \mathbf{W}_i(\omega) = \mathbf{W}(t_{i+1}, \omega) -$

$\mathbf{W}_i(t_i, \omega) \sim N(0, \Delta t)$, and ω represents a sample path. In this section, a list of the Euler-Maruyama methods for the stochastic differential equation models is given.

Euler-Maruyama Method for the SDE PUE Model

$$P_0(\omega) = P(0, \omega)$$

$$U_0(\omega) = U(0, \omega)$$

$$E_0(\omega) = E(0, \omega)$$

$$P_{i+1} = P_i - \Delta t \mu P_i U_i + \sqrt{\Delta t} \left(B_{11}^{(i)} W_1^i + B_{12}^{(i)} W_2^i + B_{13}^{(i)} W_3^i \right) \quad (5.77)$$

$$U_{i+1} = U_i + \Delta t (\mu P_i U_i - e U_i) + \sqrt{\Delta t} \left(B_{21}^{(i)} W_1^i + B_{22}^{(i)} W_2^i + B_{23}^{(i)} W_3^i \right)$$

$$E_{i+1} = E_i + \Delta t e U_i + \sqrt{\Delta t} \left(B_{31}^{(i)} W_1^i + B_{32}^{(i)} W_2^i + B_{33}^{(i)} W_3^i \right)$$

$$\text{where } \mathbf{B}^{(i)} = \begin{pmatrix} B_{11}^{(i)} & B_{12}^{(i)} & B_{13}^{(i)} \\ B_{21}^{(i)} & B_{22}^{(i)} & B_{23}^{(i)} \\ B_{31}^{(i)} & B_{32}^{(i)} & B_{33}^{(i)} \end{pmatrix} = \begin{pmatrix} \mu P_i U_i + \varepsilon & -\mu P_i U_i & 0 \\ -\mu P_i U_i & \mu P_i U_i + e U_i + \varepsilon & -e U_i \\ 0 & -e U_i & e U_i + \varepsilon \end{pmatrix}^{\frac{1}{2}},$$

$W_j^i \sim N(0, \Delta t)$ for $j = 1, 2, 3$, and $\varepsilon > 0$. The inclusion of ε will make the square root of the covariance matrix positive definite, allowing for a unique square root covariance matrix.

Euler-Maruyama Method for the SDE PUEP Model

$$\begin{aligned}
P_0(\omega) &= P(0, \omega) \\
U_0(\omega) &= U(0, \omega) \\
E_0(\omega) &= E(0, \omega) \\
P_{i+1} &= P_i + \Delta t (aV_i - \mu P_i U_i + rE_i - l_P P_i) \\
&\quad + \sqrt{\Delta t} \left(B_{11}^{(i)} W_1^i + B_{12}^{(i)} W_2^i + B_{13}^{(i)} W_3^i \right) \\
U_{i+1} &= U_i + \Delta t (\mu P_i U_i - eU_i - l_U U_i) + \sqrt{\Delta t} \left(B_{21}^{(i)} W_1^i + B_{22}^{(i)} W_2^i + B_{23}^{(i)} W_3^i \right) \\
E_{i+1} &= E_i + \Delta t (eU_i - rE_i - l_E E_i) + \sqrt{\Delta t} \left(B_{31}^{(i)} W_1^i + B_{32}^{(i)} W_2^i + B_{33}^{(i)} W_3^i \right)
\end{aligned} \tag{5.78}$$

where

$$\mathbf{B}^{(i)} = \begin{pmatrix} B_{11}^{(i)} & B_{12}^{(i)} & B_{13}^{(i)} \\ B_{21}^{(i)} & B_{22}^{(i)} & B_{23}^{(i)} \\ B_{31}^{(i)} & B_{32}^{(i)} & B_{33}^{(i)} \end{pmatrix} = \begin{pmatrix} aV_i + \mu P_i U_i + rE_i + l_P P_i & -\mu P_i U_i & -rE_i \\ -\mu P_i U_i & \mu P_i U_i + eU_i + l_U U_i & -eU_i \\ -rE_i & -eU_i & rE_i + eU_i + l_E E_i \end{pmatrix}^{\frac{1}{2}}$$

and $W_j^i \sim N(0, \Delta t)$ for $j = 1, 2, 3$.

Euler-Maruyama Method for the SDE PUEP Model with Inflow Domains

$$\begin{aligned}
V_0(\omega) &= V(0, \omega) & U_0(\omega) &= U(0, \omega) \\
P_0(\omega) &= P(0, \omega) & E_0(\omega) &= E(0, \omega) \\
N_0(\omega) &= N(0, \omega) & O_0(\omega) &= O(0, \omega) \\
V_{i+1} &= V_i + \Delta t (-aV_i - l_V V_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{1j} \frac{dW_j}{dt} \\
P_{i+1} &= P_i + \Delta t (aV_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i - l_P P_i) \\
&\quad + \sqrt{\Delta t} \sum_{j=1}^6 B_{2j} \frac{dW_j}{dt}
\end{aligned} \tag{5.79}$$

$$N_{i+1} = N_i + \Delta t (r_p P_i - r_N N_i - l_N N_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{3j} \frac{dW_j}{dt}$$

$$U_{i+1} = U_i + \Delta t (\mu P_i U_i - e U_i - l_U U_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{4j} \frac{dW_j}{dt}$$

$$E_{i+1} = E_i + \Delta t (e U_i - r_E E_i - l_E E_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{5j} \frac{dW_j}{dt}$$

$$O_{i+1} = O_i + \Delta t (r_E E_i - r_O O_i - l_O O_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{6j} \frac{dW_j}{dt}$$

where $\mathbf{B}^{(i)} = \sqrt{\mathbf{C}}$, $\mathbf{C} = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6]$,

$$C_1 = \begin{pmatrix} aV_i + l_V V_i \\ -aV_i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_2 = \begin{pmatrix} -aV_i \\ aV_i + \mu P_i U_i + r_O O_i + r_N N_i + r_p P_i + l_p P_i \\ -r_N N_i - r_p P_i \\ -\mu P_i U_i \\ 0 \\ -r_O O_i \end{pmatrix},$$

$$C_3 = \begin{pmatrix} 0 \\ -r_N N_i - r_p P_i \\ r_N N_i + r_p P_i + l_N N_i \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_4 = \begin{pmatrix} 0 \\ -\mu P_i U_i \\ 0 \\ \mu P_i U_i + e U_i + l_U U_i \\ -e U_i \\ 0 \end{pmatrix},$$

$$C_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -e U_i \\ e U_i + r_E E_i + l_E E_i \\ -r_E E_i \end{pmatrix}, \quad C_6 = \begin{pmatrix} 0 \\ -r_O O_i \\ 0 \\ 0 \\ -r_E E_i \\ r_O O_i + r_E E_i + l_O O_i \end{pmatrix}$$

and $W_j^i \sim N(0, \Delta t)$ for $j = 1, 2, \dots, 6$.

Trade Considerations

$$V_0(\omega) = V(0, \omega) \quad U_0(\omega) = U(0, \omega)$$

$$P_0(\omega) = P(0, \omega) \quad E_0(\omega) = E(0, \omega)$$

$$N_0(\omega) = N(0, \omega) \quad O_0(\omega) = O(0, \omega)$$

$$V_{i+1} = V_i + \Delta t(-aV_i - l_V V_i - x_V V_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{1j} \frac{dW_j}{dt}$$

$$P_{i+1} = P_i + \Delta t \left(aV_i - \mu P_i U_i + r_O O_i + r_N N_i - r_P P_i - l_P P_i - x_P P_i + i_P I_i \right. \\ \left. + S_i \right) + \sqrt{\Delta t} \sum_{j=1}^6 B_{2j} \frac{dW_j}{dt}$$

$$N_{i+1} = N_i + \Delta t \left(r_P P_i - r_N N_i - l_N N_i \right) + \sqrt{\Delta t} \sum_{j=1}^6 B_{3j} \frac{dW_j}{dt}$$

(5.80)

$$U_{i+1} = U_i + \Delta t (\mu P_i U_i - eU_i - l_U U_i - x_U U_i + i_U I_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{4j} \frac{dW_j}{dt}$$

$$E_{i+1} = E_i + \Delta t (eU_i - r_E E_i - l_E E_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{5j} \frac{dW_j}{dt}$$

$$O_{i+1} = O_i + \Delta t (r_E E_i - r_O O_i - l_O O_i - x_O O_i + i_O I_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{6j} \frac{dW_j}{dt}$$

$$\text{where } \mathbf{B} = \sqrt{\mathbf{C}}, \quad \mathbf{C} = (C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6)$$

$$C_1 = \begin{pmatrix} aV_i + l_V V_i + x_V V_i \\ -aV_i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{aligned}
C_2 &= \begin{pmatrix} -aV_i \\ aV_i + \mu P_i U_i + r_O O_i + r_N N_i + r_p P_i + l_p P_i + x_p P_i + i_p P_i + S_i \\ -r_N N_i - r_p P_i \\ -\mu P_i U_i \\ 0 \\ -r_O O_i \end{pmatrix}, \\
C_3 &= \begin{pmatrix} 0 \\ -r_N N_i - r_p P_i \\ r_N N_i + r_p P_i + l_N N_i \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_4 = \begin{pmatrix} 0 \\ -\mu P_i U_i \\ 0 \\ \mu P_i U_i + e U_i + l_U U_i + x_U U_i + i_U I_i \\ -e U_i \\ 0 \end{pmatrix}, \\
C_5 &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ -e U_i \\ e U_i + r_E E_i + l_E E_i \\ -r_E E_i \end{pmatrix}, \quad C_6 = \begin{pmatrix} 0 \\ -r_O O_i \\ 0 \\ 0 \\ -r_E E_i \\ r_O O_i + r_E E_i + l_O O_i + x_O O_i + i_O I_i \end{pmatrix}
\end{aligned}$$

and $W_j^i \sim N(0, \Delta t)$ for $j = 1, 2, \dots, 6$.

Euler-Maruyama Method for the Modified SDE PUEP Model with Inflow Domains and

Stock and Trade Considerations

$$V_0(\omega) = V(0, \omega) \quad U_0(\omega) = U(0, \omega)$$

$$P_0(\omega) = P(0, \omega) \quad E_0(\omega) = E(0, \omega)$$

$$N_0(\omega) = N(0, \omega) \quad O_0(\omega) = O(0, \omega)$$

$$V_{i+1} = V_i + \Delta t (aR_i - fV_i - l_V V_i - x_V V_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{1j} \frac{dW_j}{dt} \quad (5.81)$$

$$\begin{aligned}
P_{i+1} = P_i + \Delta t (fV_i - \mu P_i U_i + r_O O_i + r_N N_i - r_p P_i - l_p P_i - x_p P_i + i_p I_i \\
+ S_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{2j} \frac{dW_j}{dt}
\end{aligned}$$

$$N_{i+1} = N_i + \Delta t (r_p P_i - r_N N_i - l_N N_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{3j} \frac{dW_j}{dt}$$

$$U_{i+1} = U_i + \Delta t (\mu P_i U_i - e U_i - l_U U_i - x_U U_i + i_U I_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{4j} \frac{dW_j}{dt}$$

$$E_{i+1} = E_i + \Delta t (e U_i - r_E E_i - l_E E_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{5j} \frac{dW_j}{dt}$$

$$O_{i+1} = O_i + \Delta t (r_E E_i - r_O O_i - l_O O_i - x_O O_i + i_O I_i) + \sqrt{\Delta t} \sum_{j=1}^6 B_{6j} \frac{dW_j}{dt}$$

$$\text{where } \mathbf{B} = \sqrt{\mathbf{C}}, \quad \mathbf{C} = (C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6)$$

$$C_1 = \begin{pmatrix} aR_i + fV_i + l_V V_i + x_V V_i \\ -fV_i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

$$C_2 = \begin{pmatrix} -fV_i \\ fV_i + \mu P_i U_i + r_O O_i + r_N N_i + r_p P_i + l_p P_i + x_p P_i + i_p P_i + S_i \\ -r_N N_i - r_p P_i \\ -\mu P_i U_i \\ 0 \\ -r_O O_i \end{pmatrix},$$

$$C_3 = \begin{pmatrix} 0 \\ -r_N N_i - r_p P_i \\ r_N N_i + r_p P_i + l_N N_i \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad C_4 = \begin{pmatrix} 0 \\ -\mu P_i U_i \\ 0 \\ \mu P_i U_i + e U_i + l_U U_i + x_U U_i + i_U I_i \\ -e U_i \\ 0 \end{pmatrix},$$

$$C_5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -e U_i \\ e U_i + r_E E_i + l_E E_i \\ -r_E E_i \end{pmatrix}, \quad C_6 = \begin{pmatrix} 0 \\ -r_O O_i \\ 0 \\ 0 \\ -r_E E_i \\ r_O O_i + r_E E_i + l_O O_i + x_O O_i + i_O I_i \end{pmatrix}$$

and $W_j^i \sim N(0, \Delta t)$ for $j = 1, 2, \dots, 6$.

5.2 Graphical User Interface for Metal Flow Simulations

In the previous section, numerical methods for the computation of approximate solutions to the metal flow models were presented. To automate the computation process and to give users the ability to run simulations interactively, graphical user interfaces (GUIs) written in Matlab code are constructed. In this section, an overview of the metal flow simulations is presented.

The layout for the simulations was developed using the Matlab graphical user interface development environment (GUIDE). GUIDE simplifies the construction of interfaces by providing a set of tools for the creation of GUIs. A GUI can be constructed by click and dragging GUI components to a layout view and these components can call Matlab functions or scripts that execute actions [26].

The option screen (*Figure 5.1*) allows the user to choose a simulation for one of the models discussed in this thesis by clicking a push button. The simulations for the three-domain models are self contained: the user parameters and the output are contained on one screen (see *Figure 5.2* and *5.4*). The inflow domain models have a two-screen setup: the first screen contains the user parameters and the second screen contains the simulation (see *Figure 5.3* and *5.5*).

The simulations allow the user to define the model's parameters over a specified time period. The "Compute" button calls a Matlab function that implements a numerical method, as discussed in Section 5.1, to solve the model with user-entered parameters. The

function will not be called if the time parameters are not specified. The entries in the “Time Period” panel are self-generating—entering any 2 entries will generate the third. For example, the “Number of Years” is calculated by entering the “Initial Year” and the “Final Year”.

The simulations have 11 “Metal” push buttons that specifies the type of metal being studied. For the three-domain ODE Models, actual data for the given metal over the time period specified is taken from Excel spreadsheets and is displayed in plots under the simulations. As opposed to selecting actual data from Excel spreadsheets, a modification to the code can be added to obtain data from a database. Selecting a “Metal” button will also populate the parameters with pre-specified data.

In order to solve the partial differential equation models, initial conditions and boundary conditions need to be specified. In the simulations, Dirichlet boundary conditions are implemented. The initial conditions are assumed to be normally distributed. The simulation can be altered to compute initial conditions based on other distributions or functions.

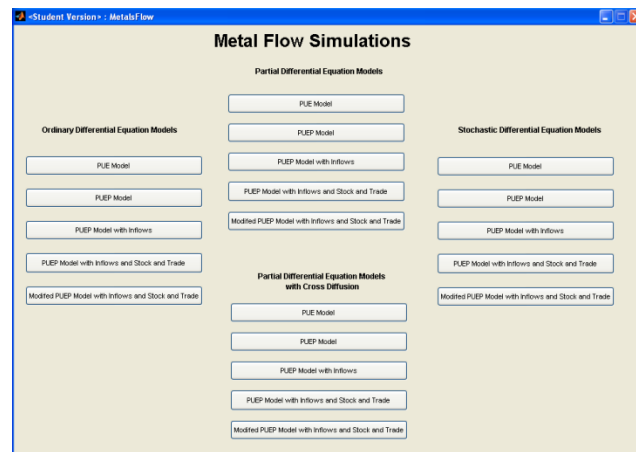


Figure 5.1: Model Selection Screen

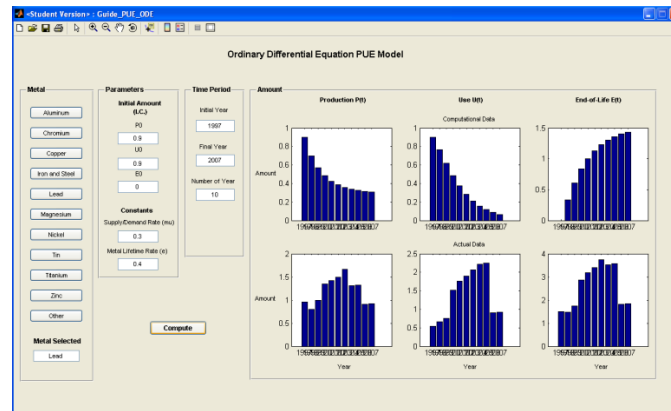
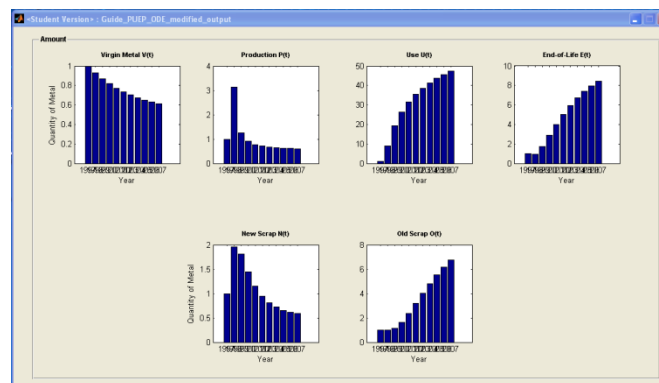


Figure 5.2: Simulation for the *ODE PUE Model*

(a) Parameter Screen



(b) Output Screen

Figure 5.3: Simulation for the *Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations*

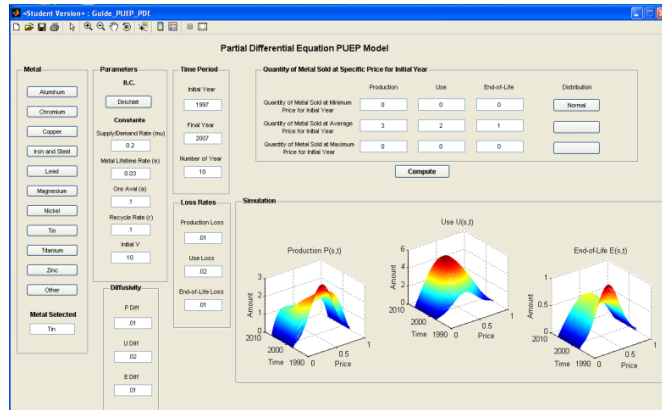
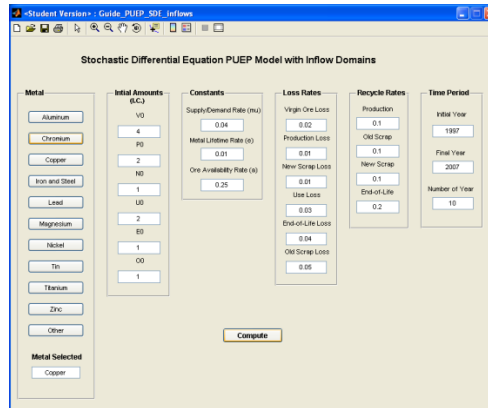
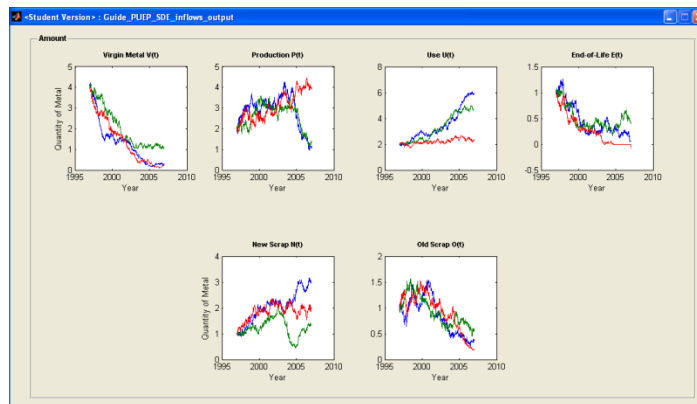


Figure 5.4: Simulation for the *PDE PUEP Model*



(a) Parameter Screen



(b) Output Screen

Figure 5.5: Simulation for the *SDE PUEP Model with Inflow Domains*

Chapter 6: Conclusion and Future Work

In this thesis, deterministic models for metal flows consisting of systems of ordinary differential equations were constructed using compartmental modeling and analysis was performed on the resulting models. Variations to the models which incorporated pricing structures and volatility by utilizing systems of partial and stochastic differential equations were also constructed. Sustainability thresholds, the conditions under which a metal flow system can be sustained, were established for the ordinary differential equation models. Parameter optimization schemes—methods for the acquisition of optimal coefficients that would replicate empirical data—were constructed for the ordinary differential equation models by implementing the Levenberg-Marquardt method. Since the models were built upon assumption taken from epidemiology, modifications using economic principles were introduced. Numerical schemes for the computation of approximate solutions were produced using Euler, BTCS, and Euler-Maruyama methods. Stability criteria for the Euler schemes were established. Standalone GUIs programmed in Matlab to simulate the metal flow models were created. Matlab functions and scripts were also presented for stability analysis, parameter optimization, and the numerical solutions to the models.

As future work, analysis must be performed on the partial and stochastic differential equation models to establish well-posedness for the models. Since the partial

differential equation models consist of systems of reaction-diffusion equations, Turing instability analysis may be utilized to study the stability of the models. Since sustainability thresholds play an integral role in metal flow analysis, they must be established for the partial and stochastic differential equation models. Parameter optimization schemes should be constructed for the partial and stochastic differential equation models in order to verify the validity of the models with empirical data. A maximum likelihood estimation method can be developed as a parameter optimization scheme for the stochastic differential equation models. A vital component of mathematical modeling is to demonstrate the goodness of fit between the resulting models against empirical data. To this front, many case studies need to be performed for various metals over varying temporal and spatial intervals and compared to actual data.

Appendix A: Material Flow Analysis and Metal Flows in Industrial Ecology

The main objective of this thesis is to provide a foundation for the mathematical modeling of metal flows, a discipline in industrial ecology that is concerned with analyzing the manner in which metal propagates through a defined system. In order to provide a working overview of metal flows for readers with various backgrounds in the subject, a short history of material flow analysis is presented in the first section and a listing of two organizations that are currently working in metal flows is provided in the second section of this appendix.

A.1 Short History of Material Flow Analysis

Material flow analysis (MFA) is a systematic assessment of the flows and stocks of material in a well-defined system. MFA is based on a system approach and conservation of mass and has similarities to life cycle assessment and input-output models. Although flow and input-output analysis had been used in engineering and economics throughout the 20th century, flow analysis was only applied to environmental and resource management in the 1970s with the work on city metabolism and pollutant pathways. In 1965, Abel Wolman coined the phrase *city metabolism* to describe the inputs and outputs of products for a U.S. city [37]. Authors, such as Duvigneaud and

Denayeyer-De Smet [16] and Newcombe et al., subsequently provided case studies of city metabolism, demonstrating large resource depletions [30].

By the late 1960s, material balancing was emerging in environmental research, beginning with a study on the accumulation of pollutants in specific regions. Huntzicker et al. studied lead pollution in the Los Angeles basin by using a mass-balancing approach and demonstrated the potential environmental impact [22].

In the early 1980s, Ayres et al. used the current MFA methodology to describe the sinks, sources, and pathways of pollutants in the Hudson-Raritan basin over a hundred year period [5]. Many region-specific papers followed, describing substance flow within waterways and throughout nations [24, 25, 31].

The works by Baccini, Benner, and Bader created a systematic and comprehensive methodology for MFA, which coupled the flow of material with human-driven systems [6, 7]. They coined the phrases “activity” and “metabolism of anthroposphere” to describe the role humans have on material flow systems. The main goals set by these works were to develop a methodology that could be used to analysis, evaluate, and optimize material flow and to improve resource management and reduce environmental degradation [13].

A.2 Metal Flows and Metal Flow Models

The flow of metal in a human-driven system, which is commonly referred to as “metal flows” in the literature, is an application of MFA. Jelinski et al. defined industrial ecology as the study of the interconnection between an industrial system and the

surrounding environment that it influences [23]. Metal flows in this context analyzes the lifecycle of a metal, beginning with the production and consumption of metal ore (mining and processing metal-bearing minerals), the production of metal products, the use of the products in the consumer market, and the disposal and reentry back into the consumer market. The main tenant emerging from this course of study lies in establishing sustainability.

Currently, metal flow studies and data collections are conducted by the U.S. Geological Survey (USGS) and the United Nations (UN), to name a few. The National Mineral Information Center (NMIC) at the USGS collects, analysis, and disseminates information on mineral commodities, illustrating supply and demand for metals and minerals on national and global levels [32]. The Mineral Commodity Section in the NMIC collects and disseminates data on metals, consisting of consumption, production, trade, and reserves of minerals [35]. The Global Minerals Analysis Section in the NMIC produces comprehensive material flow studies on metals, such as aluminum and nickel [36]. Figure A.1 illustrates a metal flow model for aluminum in 2000 constructed by the NMIC.

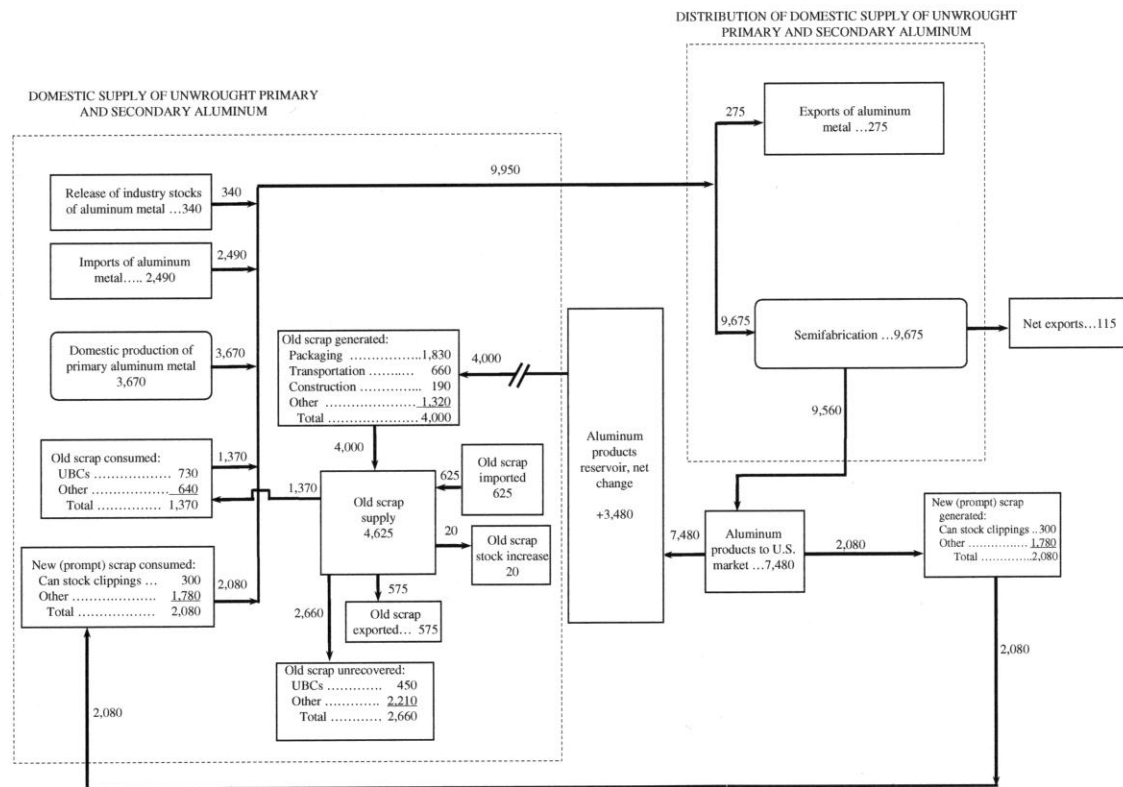


Figure A.1: Aluminum Metal Flow Model in 2000 (Source Note: U.S. Geological Survey)

The Global Metal Flows working group of the International Panel on Sustainable Resource Management, United Nations Environmental Programme, studies the role recyclable material plays in creating a sustainable system [34]. The six reports to be published attempt to provide a critical scientific assessment of metal flows at the national and international levels. Figure A.2 is an illustration constructed by the Global Metal Flows working group depicting the flow of a metal in a global context.

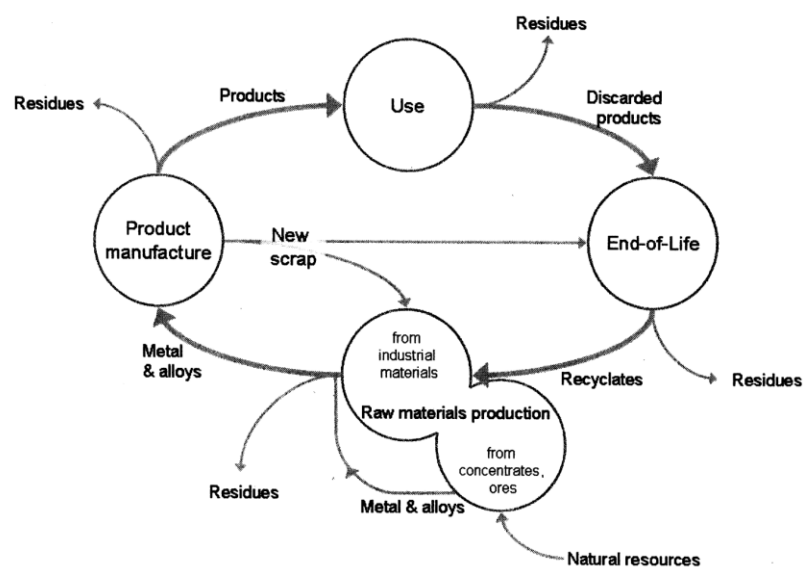


Figure A.2: Metal Flows System (Source Note: UNEP's Global Metal Flows Working Group)

Appendix B: Mathematical Modeling in Epidemiology

The spread of infectious disease has played a vital role in the history of civilizations. The fall of the Roman, Han, and Aztec empires have been directly or indirectly attributed to epidemics. The Black Plague (bubonic plague) that swept throughout Europe and Asia in the 14th century, killing one-third of the European population between 1346 and 1350, created tremendous political and economic upheavals [27]. The rapid global population growth witnessed in the 18th century can be explained by the suppression of epidemics through the improvement of medicine and the development of immunities.

A comprehensive understanding of the dynamics underpinning these outbreaks have provided valuable insight for the mitigation or even the elimination of many outbreaks. The mechanisms of transmission have been well established for many diseases (viral, bacterial, or vector agents). Infection of a viral disease, such as influenza, measles, rubella, and chicken pox, can confer future immunity to that disease, whereas a bacterial disease will not. Vector agents, usually insects, will acquire the disease from an infected human and transmit it to a population. Examples of infectious diseases transmitted by vectors are malaria and lyme disease. The characteristics of the infected and susceptible populations have also been shown to play an important role in the spread of communicable disease, such as demographics, spatial and temporal variables (affecting contact rates and disease incubation times), and proximity to and density of vectors. By

articulating feasible mathematical models that simulate the governing dynamics of epidemics, ways to suppress the spread of infectious diseases have been established, such as the construction of threshold conditions for herd immunity and vector elimination.

The underline assumption of this thesis is that the flow of metals studied in industrial ecology is analogous to the spread of infectious diseases within a population. The mathematical models and formations developed over the past century describing epidemics and endemics can provide a useful basis for the development of dynamical metal flow models. It is the intent of this appendix to describe the mathematical models in epidemiology that provide relevance to the study at hand.

B.1 A Short History of Mathematical Modeling in Epidemiology

The first purported mathematical model for an infectious disease was formulated by the mathematician Daniel Bernoulli in 1760, which studied the effect of smallpox inoculation within a population [10]. The central objective of Bernoulli's probabilistic static model was to calculate the life expectancy of individuals immune to small pox at birth, having important ramification to actuarial work in time-dependent annuities of that period [15]. The solution to the system of differential equations predicted the probability of a newborn being infected at some age t .

Deterministic epidemical modeling did not take root until the 20th century, with fast growth in the field observed after the middle of the century [21]. In 1906, William Hamer developed a discrete time model for the recurrence of measles epidemics, which may have been the first model to define the incidence (number of new infections per unit

time) as the product of the susceptible and infective densities. The British medical doctor, Ronald Ross, demonstrated that *Anopheles* mosquitoes can propagate malaria to humans, garnering him the 1902 Nobel Prize in Medicine [29]. In 1911, Ross constructed a differential equation model which studied malaria as a host-vector disease. Other deterministic models were developed by Ross, Hudson, Martini and Alfred J. Lotka. William O. Kermack and Anderson McKendrick demonstrated in 1926 that the spread of an infectious disease will not develop into an epidemic if the density of the susceptible population did not exceed a set value, known as the epidemic threshold. This result established mathematically the importance of vaccination and quarantine in the fight against epidemics [20].

Between 1900 and 1935, the foundation for the compartmental approach to epidemiology modeling was established. The compartmental approach asserts that a population under study can be broken into disjoint homogeneous domains, dependent on characteristics relevant to the particular disease being studied, such as individuals susceptible, infected, or immune to a particular disease. A variety of models surfaced since the middle of the 20th century, applied to a host of infectious diseases, such as measles, malaria, rabies, and HIV, containing a multitude of parameters, including spatial spread, incubation, and passive immunity.

B.2 Selected Models

Numerous models were developed and analyzed over the past century describing the dynamics in epidemiology, each providing differing layers of complexity based on

the interactions and characteristics of the particular system under studied. This section will only provide an overview of the epidemical models and techniques that will be used to develop models for the flow of metals. For a comprehensive overview of the models of epidemiology, refer to [21].

B.2.1 Standard SIR Models

The standard models of the spread of infectious diseases, commonly referred to as SIR models, computes the number of infected individuals over a time period by assuming that a population in which a disease will propagate can be compartmentalized. The standard SIR model of epidemics, designed by Kermack and McKendrick in 1927, has been applied to epidemics of measles, mumps, and rubella, where birth and death rates are assumed not to have a major influence on the dynamic of the disease. Endemic SIR models assume that disease incubation times are long; thus, birth and death rates are incorporated.

The SIR models compartmentalize the population under study into three homogeneous disjoint domains, which are assumed to be functions of time t : susceptible S , infective I , and removal R . The susceptible domain consists of individuals that are not infected with the disease but can incur the disease at a later time. The infective domain consists of individuals that are infected with the disease and who can transmit it to the individuals in the susceptible domain. The removal domain consists of individuals who were removed from the susceptible-infective dynamic due to recovery, immunity, or death. Individuals from the susceptible domain that acquire infection will enter the

infective domain, and the individual that recovery, are immune or die after infection will transfer into the removal domain (see Figure B.1). Each domain will change with respect to time dependent on a contact rate (number of adequate contacts of a person per unit time) and infection period (the time required for a person to recover from the disease), each parameter being unique to the disease and the population.

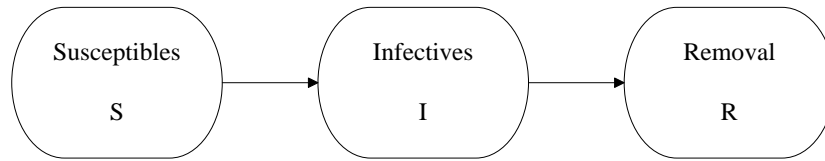


Figure B.1: Standard SIR Models for Epidemics

The following assumptions are made in the construction of the Kermack-McKendrick model for epidemics:

- i. The population is considered to be constant and latency is assumed negligible, meaning, birth and death rates are not considered.
- ii. A gain in the infective domain, referring to the transfer of individuals from the susceptible domain into the infective domain due to infection, is proportional to the product of the susceptible and infective densities: $\beta S(t)I(t)$, where β is referred to as the *contact rate*. Likewise, the amount of individuals removed from the susceptible domain due to the acquisition of the disease is equal to $-\beta S(t)I(t)$.

- iii. The rate of removal from the infective domain into the removal domain is proportional to the density of the infective domain: $-\gamma I(t)$, where γ is referred to as the infectious rate. Likewise, the rate of entry of infected individuals into the removal domain is equal to $\gamma I(t)$.

Collecting these assumptions, the standard SIR epidemic model is obtained:

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta S(t)I(t) & S(0) > 0 \\ \frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t) & I(0) > 0 \\ \frac{dR(t)}{dt} &= \gamma I(t) & R(0) \geq 0\end{aligned}\tag{B.1}$$

Note that

$$\begin{aligned}\frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt} &= 0 \\ \Rightarrow S(t) + I(t) + R(t) &= N \text{ where } N \text{ is constant}\end{aligned}$$

This implies that the population of the system remains constant, satisfying assumption (i).

Assumption (iii) has no epidemiological meaning, and needs further mathematical explanation. Let $u(t)$ represents the amount of individuals that are infected with a disease and $u(s)$ the amount of individuals that stay infected until some time s . Assume that a fraction γ of $u(s)$ will recover from the disease over each time step. Thus, we will obtain the differential equation

$$u' = -\gamma u$$

with the solution

$$u(s) = u(0)e^{-\gamma s}$$

This implies that the number of individuals that stays infected s time units after acquiring the disease is $e^{-\gamma s}$. The infection period is exponentially distributed with mean $\int_0^\infty e^{-\gamma s} ds = 1/\gamma$, called the *average infections period*. Therefore, (iii) assumes that an individual will have an exponential wait time to recover from the disease.

Through the use of this model, the conditions under which an epidemic will occur can be established, known as epidemic thresholds. An epidemic will take root in a population if the rate of infective individuals increases. Mathematically, this implies $\frac{dI(t)}{dt} > 0$, or from (B.1) $\beta S(t)I(t) - \gamma I(t) > 0$. Thus, we will obtain the condition

$$S(t) > \frac{\gamma}{\beta} = \sigma \quad (\text{B.2})$$

where σ is called the *infectious contact number*. This mathematical construct provides a valuable theoretical tool to health organizations and policy makers attempting to the moderate epidemics. Qualitatively, in order to prevent an epidemic, the number of susceptible individuals must be less than σ , which correlates to a decrease in the contact rate β (selective quarantine of the most susceptible to the disease or selective inoculation) or increase the recovery rate γ (providing adequate medical care to the infected). In developing countries where monetary concerns can be paramount in decision making, a non-governmental organization with limited resources can attack epidemics through a calculated approach of contact rate decreases and recovery rate increases.

The basic reproductive number R_0 —the average number of secondary infections produced when introducing single infected individual into the host population—is an

epidemic threshold for many epidemiological models. The basic reproductive number for the SIR model is given by

$$R_0 = \sigma S(0) \quad (\text{B.3})$$

If $R_0 > 1$, then an epidemic will occur. The disease will die out if $R_0 < 1$.

An endemic is the spread of disease within a population that is maintained over generational periods of time without the need for external inputs. In modeling endemics, vital dynamics (births and deaths) of the population must be incorporated. As in the epidemic case, the assumption is made that the population is constant: $S(t) + I(t) + R(t) = N$. Let us also assume that a certain percentage μ of the population will produce an offspring. Thus, the birth rate of the population is given by μN . The offspring are assumed to be susceptible to the disease upon birth; thus, the inflow of newborns will enter the susceptible domain. In each of the domains, a certain percentage μ of population will die; therefore, we will obtain the death rates $-\mu S$, $-\mu I$, and $-\mu R$. To maintain a constant population by balancing the birth and deaths, assume that the birth and death coefficients (μ) are the same. Introducing these inflows and outflows into the epidemic model (B.1), the classic endemic model will be obtained

$$\begin{aligned} \frac{dS(t)}{dt} &= \mu N - \beta S(t)I(t) - \mu S(t) & S(0) > 0 \\ \frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t) - \mu I(t) & I(0) > 0 \\ \frac{dR(t)}{dt} &= \gamma I(t) - \mu R(t) & R(0) \geq 0 \end{aligned} \quad (\text{B.4})$$

B.2.2 SIRS Models: Immunity Temporarily Conferred

If assuming that permanent immunity is not conferred to an individual previously infected with a disease, then the infective will reenter the susceptible domain after recovering from the infection. The literature refers to this model as an SIS model since the infective will reenter the susceptible domain after infection and will not reside in a state of recovery. Constructing a separate recovery domain will be a key component in the metal flow models, since the assumption will be made that a state of recovery will be needed to model the dynamics of discarded and recycled metal. Therefore, a modified SIS model will be presented here.

Let us assume that a certain percentage f of the recovered population will become re-infected after recovering from the disease. Then, the rate of removal from the recovered domain into the susceptible domain will be $-fR(t)$. Likewise, the rate of entry into the susceptible domain will be $fR(t)$. If we apply this inflow and outflow into the epidemic system (B.1), the epidemic SIRS model is obtained:

$$\begin{aligned}\frac{dS(t)}{dt} &= -\beta S(t)I(t) + fR(t) & S(0) > 0 \\ \frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t) & I(0) > 0 \\ \frac{dR(t)}{dt} &= \gamma I(t) - fR(t) & R(0) \geq 0\end{aligned}\tag{B.5}$$

Applying the inflow and outflow to the endemic model (B.4), the endemic SIRS model is obtained:

$$\begin{aligned}
\frac{dS(t)}{dt} &= \mu N - \beta S(t)I(t) + fR(t) - \mu S(t) & S(0) > 0 \\
\frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t) - \mu I(t) & I(0) > 0 \\
\frac{dR(t)}{dt} &= \gamma I(t) - fR(t) - \mu R(t) & R(0) \geq 0
\end{aligned} \tag{B.6}$$

B.2.3 Spatial Diffusion of Infectious Disease

The epidemiological models previously discussed assumed that an infection spreads between individual solely through contact. Within an increasing globalize society, the spread of infectious disease takes on a spatial component, as seen with the 1980s HIV and 2003 Severe Acute Respiratory Syndrome (SARS) pandemics. These two events demonstrated that over time, infectious disease can spread from geographic locations with high infected concentrations to areas of low concentrations. Epidemical models which incorporated a spatial consideration have relied on a reaction-diffusion system of equations to model the spreading of a disease.

Let the spread of disease be governed by time, as seen in the previous models, and by space. Thus, the population domains will become dependent variables with respect to temporal and spatial independent variables. If we assume that a disease residing in a population will spread spatially in a diffusive manner—that is, spreading down the concentration gradient as observe with heat diffusion—we can introduce a diffusivity operator $\frac{\partial^2}{\partial x^2}$ to the previous models.

If the individuals in a particular domain are thought to spreads throughout a geographic location in a diffusive manner, then we can model the rate of change of that

domain D with respect to space x by the term $\delta_{DD} \frac{\partial^2 D(t,x)}{\partial x^2}$, where δ_{DD} is called the spatial diffusivity for the domain D . If, on the other hand, the individuals in one domain D are thought to spread toward or away from individuals in another domain C , then this interaction can be modeled with the cross-diffusion terms $\pm \delta_{CD} \frac{\partial^2 C(t,x)}{\partial x^2}$. For example, a susceptible will move away from an infective in fear of contraction of the disease. (It is important to note that there might be no epidemiological meaning for individuals in one domain to diffuse in the direction of another, but this concept will have pertinent meaning in the metal flow models).

If the individuals in each domain are thought to positively diffuse throughout their respective domain and the susceptible and removal domains are thought to cross-diffuse away from the infective domain, then we will obtain the following endemic SIRS model

$$\begin{aligned}
\frac{\partial S(t,x)}{\partial t} &= \mu N - \beta S(t,x)I(t,x) + fR(t,x) - \mu S(t,x) + \delta_{SS} \frac{\partial^2 S(t,x)}{\partial x^2} - \delta_{IS} \frac{\partial^2 I(t,x)}{\partial x^2} \\
\frac{\partial I(t,x)}{\partial t} &= \beta S(t,x)I(t,x) - \gamma I(t,x) - \mu I(t,x) + \delta_{II} \frac{\partial^2 I(t,x)}{\partial x^2} \\
\frac{\partial R(t,x)}{\partial t} &= \gamma I(t,x) - fR(t,x) - \mu R(t,x) + \delta_{RR} \frac{\partial^2 R(t,x)}{\partial x^2} - \delta_{IR} \frac{\partial^2 I(t,x)}{\partial x^2}
\end{aligned} \tag{B.7}$$

Appendix C: Source Codes and Selected Outputs

C.1 Stability for the ODE PUEP Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: Stability_ODE_PUEP.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Computes the eigenvalues for the Jacobian evaluated at fixed pts
%          to describe the stability of the system at the fixed point
%Inputs: Model coefficients
%Output: The stability at the fixed point
%Date Created: 3 May 2011
%Date Last Modified: 4 May 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function fun = Stability_ODE_PUEP(V,a,e,mu,r,l_P,l_U,l_E)
%find fixed points of the system
FP=[ e/mu+l_U/mu...
      (a*v-l_P/mu*(e+l_U ))*(e+l_U-r*e/(r+l_E ))^(-1),...
      e/(r+l_E )*(a*v-l_P/mu*(e+l_U ))*(e+l_U-r*e/(r+l_E ))^(-1)];

%error message if fixed point is outside domain
if (FP(1)<=0 || FP(2)<= 0 || FP(3)<0)
    disp('Fixed point not defined in domain')
    return;
elseif (isnan(FP(1))==1 || isnan(FP(2))==1 || isnan(FP(3))==1)
    disp('Fixed point not defined in domain')
    return;
elseif (isinf(FP(1))==1 || isinf(FP(2))==1 || isinf(FP(3))==1)
    disp('Fixed point not defined in domain')
```

```

        return;
    end

    %jacobian for the system
    J = [-mu*FP(2), -mu*FP(1), -r;...
        mu*FP(2), mu*FP(1)-e-l_U,0;...
        0,e,-r-l_E];

    str =fprintf('At fixed point (%4.2f,%4.2f,%4.2f)\n',FP(1), FP(2), FP(3));

    lambda = eig(J);          %compute the eigenvalues
    num = size(lambda);

    test = 0;                  %used to check if stability can be assessed
    for i=1:num
        %check if real part is non-zero
        if real(lambda(i)) == 0
            test = test + 1;
        end

        %check stability
        if real(lambda(i)) > 0
            disp('The system is unstable since the eigenvalues are')
            disp(lambda)
            return;
        end
    end

    if test == 0
        disp('The system is stable since the eigenvalues are')
        disp(lambda)
    else
        disp('stability cannot be determined since the real part equals 0')
    end
end

```

C.2 Stability for the ODE PUEP Model with Inflow Domains

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: Stability_ODE_PUEP_Inflow.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Computes the eigenvalues for the Jacobian evaluated at fixed pts
%          to describe the stability of the system at the fixed point
%Inputs: Model coefficients
%Output: The stability at the fixed point
%Date Created: 4 May 2011
%Date Last Modified: 5 May 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function fun = Stability_ODE_PUEP_Inflow(a,e,mu,l_v,l_P,l_N,l_U,l_E,...
    l_O,r_N,r_P,r_O,r_E)

%find fixed points of the system
FP=[0,...
    (e+l_U)/mu,...
    (r_P)/mu*(e+l_U)/(r_N+l_N),...
    (-(e+l_U)/mu*(r_P+l_P )+r_N*(r_P)/mu*(e+l_U)/(r_N+l_N))* ...
    ((e+l_U-(r_O*r_E*e)/(r_E+l_E)*(r_O+l_O) ))^(-1),...
    e/(r_E+l_E)*(-(e+l_U)/mu*(r_P+l_P )+r_N*(r_P)/mu*(e+l_U)/(r_N+l_N))*...
    (e+l_U-(r_O*r_E*e)/(r_E+l_E)*(r_O+l_O) )^(-1),...
    (r_E)/(r_O+l_O)*e/(r_E+l_E)*(-(e+l_U)/mu*(r_P+l_P )+r_N*(r_P)/mu*...
    (e+l_U)/(r_N+l_N))*(e+l_U-(r_O*r_E*e)/(r_E+l_E)*(r_O+l_O) )^(-1) ];

%error message if fixed point is outside domain
if (FP(1)<=0 || FP(2)<= 0 || FP(3)<0)
    disp('Fixed point not defined in domain')
    return;
elseif (isnan(FP(1))==1 || isnan(FP(2))==1 || isnan(FP(3))==1)
    disp('Fixed point not defined in domain')
```

```

        return;
elseif (isinf(FP(1))==1 || isinf(FP(2))==1 || isinf(FP(3))==1)
    disp('Fixed point not defined in domain')
    return;
end

%jacobian for the system
J = [-a-l_V,0,0,0,0,0;...
      a,-mu*FP(4)-r_P-l_P,r_N,-mu*FP(2),0,r_O;...
      0,r_P,-r_N-l_N,0,0,0;...
      0,mu*FP(4),0,mu*FP(2)-e-l_U,0,0;...
      0,0,0,e,-r_E-l_E,0;...
      0,0,0,0,r_E,-r_O-l_O];

str =fprintf('At fixed point (%4.2f,%4.2f,%4.2f)\n',FP(1), FP(2), FP(3));

lambda = eig(J);          %compute the eigenvalues
num = size(lambda);

for i=1:num
    %check if real part is non-zero
    if real(lambda(i)) == 0
        test1 = test1 + 1;
    end

    %check stability
    if real(lambda(i)) > 0
        disp('The system is unstable since the eigenvalues are')
        disp(lambda)
        return;
    end
end

if test1 == 0

```

```
        disp('The system is stable since the eigenvalues are')
        disp(lambda)
    else
        disp('stability cannot be determined since the real part equals 0')
    end
```

C.3 Stability for the ODE PUEP Model with Inflow Domains and Stock and Trade Considerations

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: Stability_ODE_PUEP_Inflow_Trade.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Computes the fixed points and stability at these points for the
%         ODE PUEP Model with Inflow Domains and Trade and Stock
%         Considerations
%Inputs: Model coefficients
%Output: The fixed points and the stability at those fixed points
%Date Created: 24 May 2011
%Date Last Modified: 25 May 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function fun = Stability_ODE_PUEP_Inflow_Trade(I,S,l_V,l_P,l_N,l_U,l_E,...
        l_O,r_O,r_N,r_P,r_E,a,mu,e,i_P,i_U,i_O,x_V,x_P,x_U,x_O)

%defining symbolic metal domains
syms V P N U E O;

%define symbolic function X such that dx/dt = X
E1 = -a*v-l_V*v-x_V*v;
E2 = a*v-mu*p*u+r_O*O+r_N*N-r_P*P-l_P*P-x_P*P+i_P*I+S;
E3 = r_P*P-r_N*N-l_N*N;
E4 = mu*p*u-e*u-l_U*U-x_U*U+i_U*I;
E5 = e*u-r_E*E-l_E*E;
E6 = r_E*E-r_O*O-l_O*O-x_O*O+i_O*I;

%find fixed points of the system
[FP1,FP2,FP3,FP4,FP5,FP6]=solve(E1,E2,E3,E4,E5,E6,'V','P','N','U','E','O');

[row,col] = size(FP1);

```

```

for i = 1:row
    FP(i,:)=[double(FP1(i)),double(FP2(i)),double(FP3(i)),...
        double(FP4(i)),double(FP5(i)),double(FP6(i))];
end

for i = 1:row
    proceed = 1;    %variable used to check if fp is in domain and stable

    %error message if fixed point is outside domain
    if (FP(i,1)<=0 || FP(i,2)<= 0 || FP(i,3)<0 || FP(i,4)<0 ||...
        FP(i,5)<0 || FP(6)<0 || ...
        isnan(FP(i,1))==1 || isnan(FP(i,2))==1 || ...
        isnan(FP(i,3))==1 || isnan(FP(i,4))==1 || ...
        isnan(FP(i,5))==1 || isnan(FP(i,6))==1 || ...
        isinf(FP(i,1))==1 || isinf(FP(i,2))==1 || ...
        isinf(FP(i,3))==1 || isinf(FP(i,4))==1 ||...
        isinf(FP(i,5))==1 || isinf(FP(i,6))==1)
        disp('The fixed point (V,P,N,U,E,O)=')
        str =fprintf('%4.2f,%4.2f,%4.2f,%4.2f,%4.2f,%4.2f) ',...
            FP(i,1), FP(i,2), FP(i,3), FP(i,4), FP(i,5), FP(i,6));
        disp('is not defined in domain.')
        proceed = 0;
    end

    if proceed ~= 0

        %Find the Jacobian for the system evaluated at the fixed points
        %jacobian for the system
        J=[diff(E1,V),diff(E1,P),diff(E1,N),...
            diff(E1,U),diff(E1,E),diff(E1,O);
            diff(E2,V),diff(E2,P),diff(E2,N),...
            diff(E2,U),diff(E2,E),diff(E2,O);
            diff(E3,V),diff(E3,P),diff(E3,N),...
            diff(E3,U),diff(E3,E),diff(E3,O);

```



```

diff(E4,V),diff(E4,P),diff(E4,N),...
diff(E4,U),diff(E4,E),diff(E4,O);
diff(E5,V),diff(E5,P),diff(E5,N),...
diff(E5,U),diff(E5,E),diff(E5,O);
diff(E6,V),diff(E6,P),diff(E6,N),...
diff(E6,U),diff(E6,E),diff(E6,O)];

%evaluate J at the fixed point
J = subs(J,{V,P,N,U,E,O},{FP(i,1),FP(i,2),FP(i,3),FP(i,4),...
    FP(i,5),FP(i,6)});

%casting Jacobian from symbolic notations to double precision
for r = 1:6
    for s = 1:6
        J(r,s) = double(J(r,s));
    end
end

%display results
disp('At fixed point (V,P,N,U,E,O)=')
str =fprintf('%4.2f,%4.2f,%4.2f,%4.2f,%4.2f,%4.2f) \n',...
    FP(i,1), FP(i,2), FP(i,3), FP(i,4), FP(i,5), FP(i,6));

lambda = eig(J);          %compute the eigenvalues
num = size(lambda);

for r=1:num
    %check if real part is non-zero
    if real(double(lambda(r))) == 0
        proceed = -1;
    %check if unstability
    elseif real(double(lambda(r))) > 0
        proceed = 0;
    end
end

```

```
end

if proceed == -1
    disp('stability cannot be determined');
    disp('since the real part equals 0.');
```

elseif proceed == 0

```
    disp('The system is unstable.')
else
    disp('the system is stable.')
end

end

end
```

C.4 Stability for the Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations

The Matlab function `Stability_ODE_PUEP_Inflow_Trade_Modified.m`, which was written to find the fixed points to the model and analyze the stability of the system at those points, is the same as the `Stability_ODE_PUEP_Inflow_Trade.m` function discussed in C.3, with modifications to lines 14-22, which is presented here.

```
function fun = Stability_ODE_PUEP_Inflow_Trade_Modified(I,S,R,l_V,l_P,l_N,...
    l_U,l_E,l_O,r_O,r_N,r_P,r_E,a,f,mu,e,i_P,i_U,i_O,x_V,x_P,x_U,x_O)

%defining symbolic metal domains
syms V P N U E O;

%define symbolic function E such that dx/dt = E
E1 = a*R-f*V-l_V*V-x_V*V;
E2 = f*V-mu*P*U+r_O*O+r_N*N-r_P*P-l_P*P-x_P*P+i_P*I+S;
```

C.5 Parameter Optimization for the ODE PUE Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: LM_PUE_ODE.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Matlab script that computes the best (mu,e) for an empirical data
%          for the ODE PUE Model using the Levenburg-Marquardt Algorithm
%Output: The optimal parameters (mu,e)
%Date Created: 15 March 2011
%Date Last Modified: 16 March 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Generate empirical data
%This section should be modified to allow the input of actual data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mu=.1;                                %first parameter
e=.2;                                %second parameter
t0=0;                                %initial time
tF=1;                                %final time
Num=10;                               %number of steps

P0=1; U0=1; E0=1;                    %initial conditions

[P,U,E]=PUE_ODE(P0,U0,E0,mu,e,t0,tF,Num); %calling PUE_ODE function

P_emp = P +.01*rand(length(P),1);    %adding noise
U_emp = U +.01*rand(length(U),1);
E_emp = E +.01*rand(length(E),1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LM Algorithm
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

mu=.3; e=.3;           %initial parameter guess
t0=0;                  %initial time
tF=1;                  %final time
Num=10;                %number of steps
del_t = (tF-t0)/Num;   %time step
t=[t0:del_t:tF]';      %time domain
P0=1; U0=1; E0=1;      %initial condition

%solve for initial guess
%PUE_ODE() is a function that solves the IVP using finite different
[P_initial,U_initial,E_initial]=PUE_ODE(P0,U0,E0,mu,e,t0,tF,Num);

J_Rows=3*(length(P_initial)-1); %number of rows for the Jacobian
J_Cols=2;               %number of parameters to be estimated
num_iters=100;          %number of iterations of LM
lambda=0.01;            %initial dampering term
updateJ=1;              %if evalJ==1 then Jacobian is evaluated for
                        %current parameters; if evalJ==0 then dampering
                        %factor lambda needs to be increased

for iters=1:num_iters
    if updateJ==1
        %evaluate the Jacobian matrix at the current (mu,e)
        J=zeros(J_Rows,J_Cols);
        [P_est,U_est,E_est]=PUE_ODE(P0,U0,E0,mu,e,t0,tF,Num);

        for i = 1:(J_Rows/3)
            J(i,:)= del_t*[-P_est(i)*U_est(i),0];
            J(i+J_Rows/3,:)= del_t*[P_est(i)*U_est(i),-U_est(i)];
            J(i+2*J_Rows/3,:)= del_t*[0,U_est(i)];
        end
    end
end

```

```

%evaluate epsilon at the current parameters
epsilon = zeros(J_Rows,1);
for i =1:(J_Rows/3)
    epsilon(i) = P_emp(i+1) - P_est(i+1);
    epsilon(i+J_Rows/3) = U_emp(i+1) - U_est(i+1);
    epsilon(i+2*J_Rows/3) = E_emp(i+1) - E_est(i+1);
end

%compute the Hessian matrix
Hess = J'*J;

%compute the error at the first iterate
if iters == 1
    cost = norm(epsilon);
end
end

%apply the dampering term
N=Hess+(lambda*diag(diag(Hess)));

%compute the updated parameters
delta = N\ (J'*epsilon);
mu_lm = mu+delta(1);
e_lm = e+delta(2);

%compute epsilon and cost function for the new parameters
[P_lm,U_lm,E_lm]=PUE_ODE(P0,U0,E0,mu_lm,e_lm,t0,tF,Num);

epsilon_update = zeros(J_Rows,1);
for i =1:(J_Rows/3)
    epsilon_update(i) = P_emp(i+1) - P_lm(i+1);
    epsilon_update(i+J_Rows/3) = U_emp(i+1) - U_lm(i+1);
    epsilon_update(i+2*J_Rows/3) = E_emp(i+1) - E_lm(i+1);
end

```

```

cost_update = norm(epsilon_update);

%adjusting the dampering term
if cost_update < cost
    lambda = lambda/10;
    mu = mu_lm;
    e = e_lm;
    cost = cost_update;
    updateJ = 1;
else
    updateJ = 0;
    lambda=lambda*10;
end
end

disp([mu,e])

```

C.6 Euler's Method for the Modified ODE PUEP Model with Inflow Domains and Stock and Trade Considerations

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: PUEP_inflow_ODE_trade_reserves.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Matlab function that computes the finite difference method for
%         the PUEP ODE model with inflow domains, trade, reserves
%Date Created: 14 Jan 2011
%Date Last Modified: 20 Jan 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%defining time domain and intervals
t0 = 0;
t1 = 1;
n = 100;
del_t = (t1-t0)/n;
Time = 0:del_t:t1;

%defining populations
V = zeros(1,n+1);
P = zeros(1,n+1);
N = zeros(1,n+1);
U = zeros(1,n+1);
E = zeros(1,n+1);
O = zeros(1,n+1);

%initial conditions
V(1,1)=2;
P(1,1)=2;
N(1,1)=2;
U(1,1)=2;
E(1,1)=1;
```



```

o(1,1)=2;

%import and stock release functions
I = zeros(1,n+1)+1;
S = zeros(1,n+1)+.5;
R = zeros(1,n+1)+100;

%initializing constants
%metal loss rates
l_V = .2;
l_P = .1;
l_N = .3;
l_U = .2;
l_E = .2;
l_O = .2;

%metal recyclable rates
r_O = .25;
r_N = .25;
r_P = .1;
r_E = .2;

%availability, supply/demand, end-of-life, fabrication, import, export
a=.1;
mu=.1;
e=.1;
f=.1;
i_P=.2;
i_U=.5;
i_O=.3;
x_P=.1;
x_U=.1;
x_O=.1;

```

```

%Calculating populations
for i=1:n
    V(1,i+1)=V(1,i)+del_t*(-l_v*V(1,i)+a*R(1,i)-f*V(1,i));
    P(1,i+1)=P(1,i)+del_t*(-l_p*P(1,i)-mu*P(1,i)*U(1,i)+r_o*O(1,i)...
        +r_n*N(1,i)-r_p*P(1,i)+f*V(1,i)-x_p*P(1,i)+i_p*I(1,i)+S(i));
    N(1,i+1)=N(1,i)+del_t*(-l_n*N(1,i)+r_p*P(1,i)-r_n*N(1,i));
    U(1,i+1)=U(1,i)+del_t*(-l_u*U(1,i)+mu*P(1,i)*U(1,i)-e*U(1,i)...
        -x_u*U(1,i)+i_u*I(1,i));
    E(1,i+1)=E(1,i)+del_t*(-l_e*E(1,i)+e*U(1,i)-r_e*E(1,i));
    O(1,i+1)=O(1,i)+del_t*(-l_o*O(1,i)+r_e*E(1,i)-r_o*O(1,i)-x_o*O(1,i)...
        +i_o*I(1,i));
end

%Plot populations
scrsz = get(0,'ScreenSize');
figure('Position',[0 0 scrsz(3)/2 scrsz(3)/2])

%Plot V(t)
subplot(3,2,1)
plot(Time,V);
title('Virgin Metal V(t)')
xlabel('Time')
ylabel('Amount')

%Plot P(t)
subplot(3,2,2)
plot(Time,P);
title('Production P(t)')
xlabel('Time')
ylabel('Amount')

%Plot N(t)
subplot(3,2,3)
plot(Time,N);

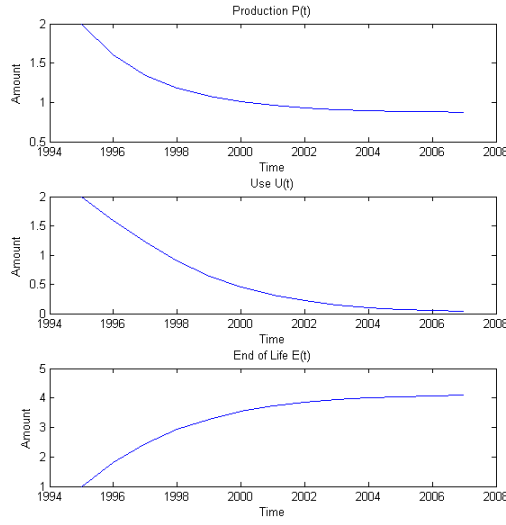
```

```
title('New Scrap N(t)')  
xlabel('Time')  
ylabel('Amount')
```

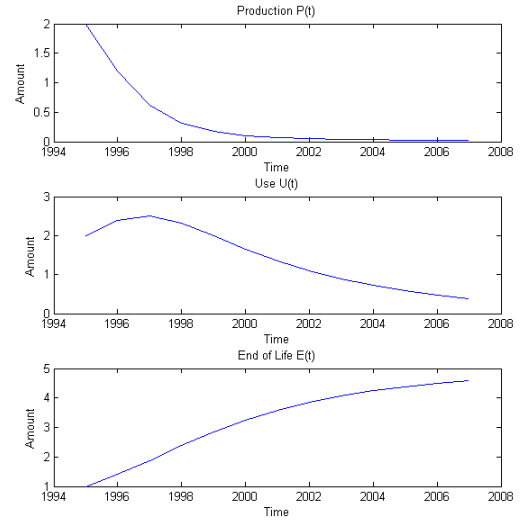
```
%Plot U(t)  
subplot(3,2,4)  
plot(Time,U);  
title('Use U(t)')  
xlabel('Time')  
ylabel('Amount')
```

```
%Plot E(t)  
subplot(3,2,5)  
plot(Time,E);  
title('End of Life E(t)')  
xlabel('Time')  
ylabel('Amount')
```

```
%Plot O(t)  
subplot(3,2,6)  
plot(Time,O);  
title('Old Scrap O(t)')  
xlabel('Time')  
ylabel('Amount')
```

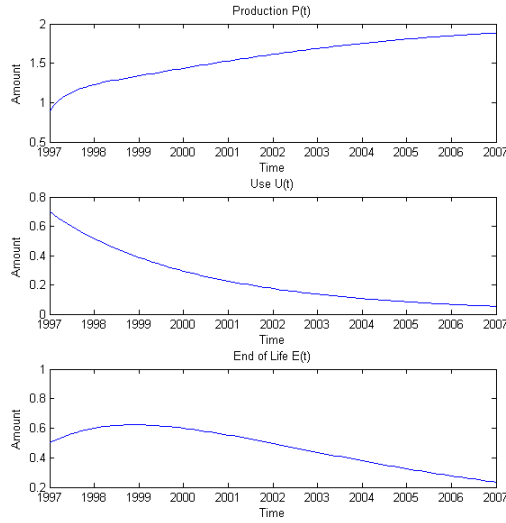


(a) $\mu = 0.1, e = 0.4$

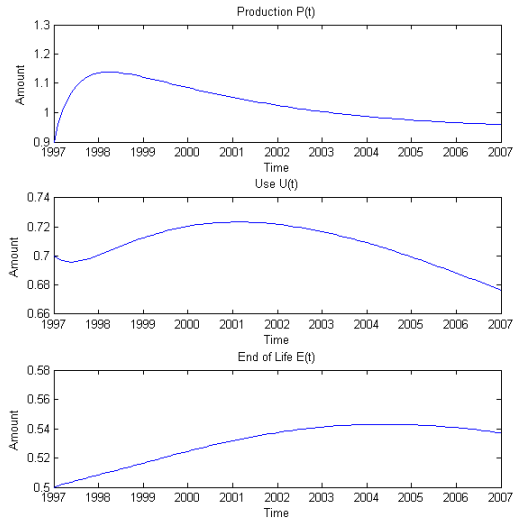


(b) $\mu = 0.2, e = 0.2$

Figure C.1: Comparison of two solutions to the *ODE PUE Model*



(a) $a = 0.1, \mu = 0.1, r = 0.25,$
 $e = 0.4, l_P = l_U = l_E = 0.01$



(b) $a = 0.2, \mu = 0.2, r = 0.25,$
 $e = 0.2, l_P = l_U = l_E = 0.01$

Figure C.2: Comparison of two solutions to the *ODE PUEP Model*

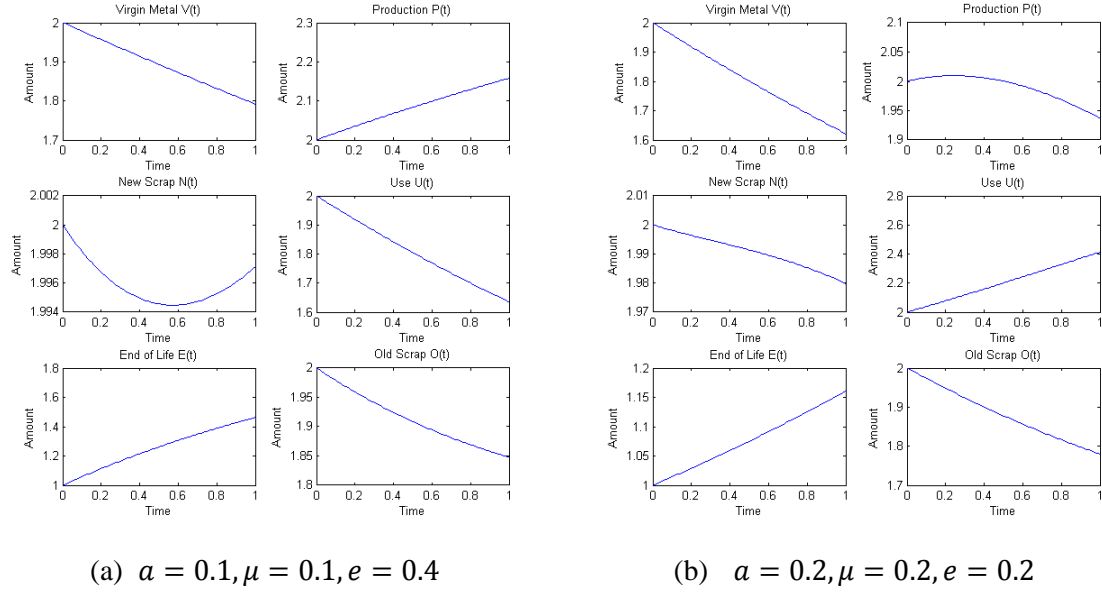


Figure C.3: Comparison of two solutions to the *ODE PUEP Model with Inflow Domains*

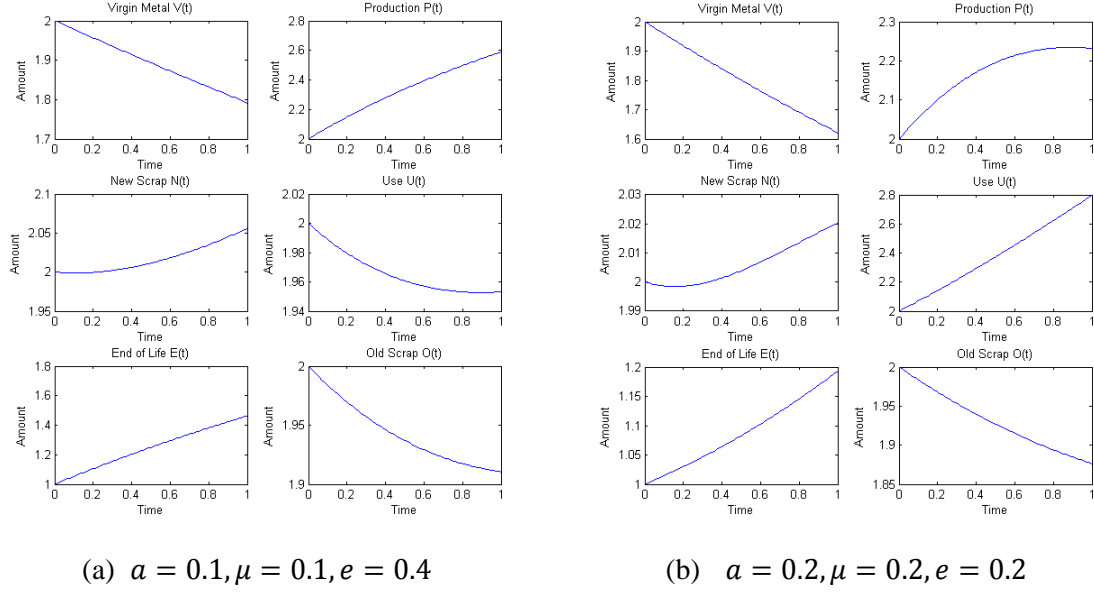


Figure C.4: Comparison of two solutions to the *ODE PUEP Model with Inflow Domains and Stock and Trade Considerations*

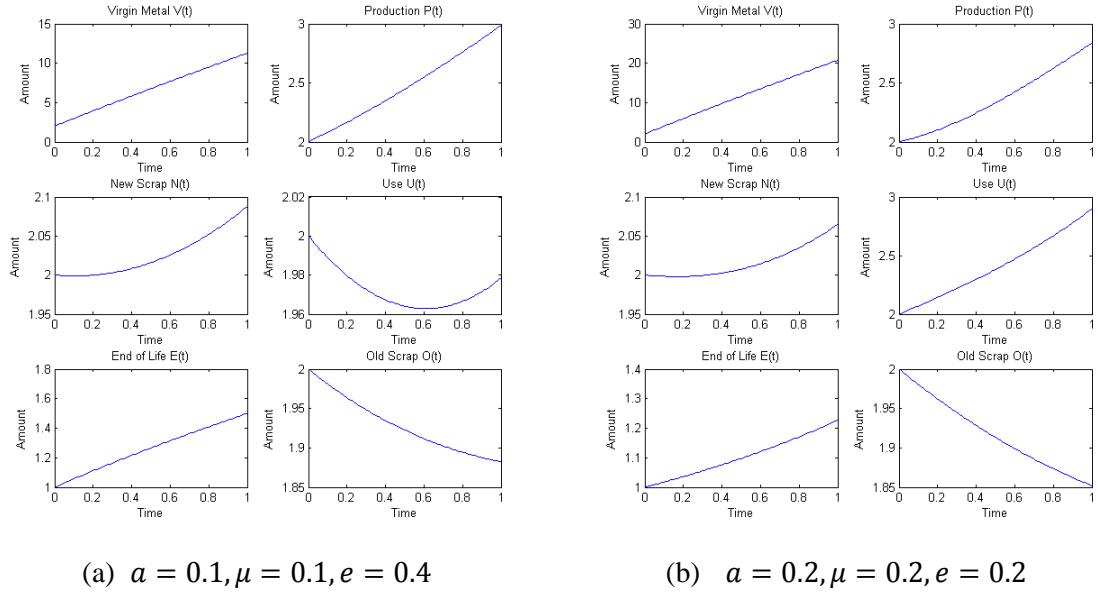


Figure C.5: Comparison of two solutions to the *ODE PUEP Model with Inflow Domains and Stock and Trade Considerations*

C.7 BTCS Method for the Modified PDE PUEP Model with Inflow Domains, Trade and Stock Considerations and Price Cross Diffusion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: PUEP_inflow_PDE_crossdiffusion_trade_reserves.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Matlab function that computes the finite difference method for
%         the PDE PUEP model with inflow domains, cross diffusion terms,
%         trade, and reserves
%Date Created: 14 Jan 2011
%Date Last Modified: 20 Jan 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%defining time and price domains and intervals
t0 = 0;
t1 = 1;
m = 100;
del_t = (t1-t0)/m;
Time = 0:del_t:t1;

s0 = 0;
s1 = 1;
n = 100;
del_s = (s1-s0)/n;
Price = 0:del_s:s1;

%defining populations
V = zeros(n+1,m+1);
P = zeros(n+1,m+1);
N = zeros(n+1,m+1);
U = zeros(n+1,m+1);
E = zeros(n+1,m+1);
O = zeros(n+1,m+1);
```

%import, stock release, and reserves functions

```
I = zeros(n+1,1)+1;  
S = zeros(n+1,1)+.5;  
R = zeros(n+1,1)+100;
```

%initial conditions

```
V(:,1)=(Price-Price.^2);  
P(:,1)=2*(Price-Price.^2);  
N(:,1)=(1/8)*(Price-Price.^2);  
U(:,1)=(2/3)*(Price-Price.^2);  
E(:,1)=(1/3)*(Price-Price.^2);  
O(:,1)=(1/8)*(Price-Price.^2);
```

%boundary values

```
V(1,:)=0;  
P(1,:)=0;  
N(1,:)=0;  
U(1,:)=0;  
E(1,:)=0;  
O(1,:)=0;
```

```
V(n+1,:)=0;  
P(n+1,:)=0;  
N(n+1,:)=0;  
U(n+1,:)=0;  
E(n+1,:)=0;  
O(n+1,:)=0;
```

%availability, supply/demand, end-of-life, fabrication, import, export

```
alpha=.01;  
mu=.2;  
e=.1;  
f=.9;
```



```

i_P=.2;
i_U=.5;
i_O=.3;
x_P=.1;
x_U=.1;
x_O=.1;

%recyclible rates
r_O=.25;
r_E=.25;
r_N=.75;
r_P=.05;

%loss rates
l_V=.4;
l_P=.1;
l_N=.1;
l_U=0;
l_E=.1;
l_O=.1;

%diffusion terms
delta_VV=.2;
delta_PP=.2;
delta_NN=.2;
delta_UU=.2;
delta_EE=.2;
delta_OO=.2;

%cross diffusion terms
delta_PV=.02;
delta_VP=.02;
delta_OP=.02;
delta_PN=.02;

```

```

delta_PU=.02;
delta_PE=.02;
delta_UE=.02;
delta_PO=.02;
delta_EO=.02;

%Matrix system terms for V
aV = zeros(1,n-2)+delta_VV/(del_s^2);
bV = zeros(1,m-1)-1/del_t-2*delta_VV/(del_s^2);
A_V = (zeros(n-1,m-1)+diag(bV,0)+diag(aV,1)+diag(aV,-1))^(-1);

%Matrix system terms for P
aP = zeros(1,n-2)+delta_PP/(del_s^2);
bP = zeros(1,m-1)-1/del_t-2*delta_PP/(del_s^2);
A_P = (zeros(n-1,m-1)+diag(bP,0)+diag(aP,1)+diag(aP,-1))^(-1);

%Matrix system terms for N
aN = zeros(1,n-2)+delta_NN/(del_s^2);
bN = zeros(1,m-1)-1/del_t-2*delta_NN/(del_s^2);
A_N = (zeros(n-1,m-1)+diag(bN,0)+diag(aN,1)+diag(aN,-1))^(-1);

%Matrix system terms for U
aU = zeros(1,n-2)+delta_UU/(del_s^2);
bU = zeros(1,m-1)-1/del_t-2*delta_UU/(del_s^2);
A_U = (zeros(n-1,m-1)+diag(bU,0)+diag(aU,1)+diag(aU,-1))^(-1);

%Matrix system terms for E
aE = zeros(1,n-2)+delta_EE/(del_s^2);
bE = zeros(1,m-1)-1/del_t-2*delta_EE/(del_s^2);
A_E = (zeros(n-1,m-1)+diag(bE,0)+diag(aE,1)+diag(aE,-1))^(-1);

%Matrix system terms for O
aO = zeros(1,n-2)+delta_OO/(del_s^2);
bO = zeros(1,m-1)-1/del_t-2*delta_OO/(del_s^2);

```

```

A_0 = (zeros(n-1,m-1)+diag(b0,0)+diag(a0,1)+diag(a0,-1))^( -1);

%Calculating populations
for j=2:m+1
    %calculating lambdas
    lambda_V=-1/del_t+l_V+f;
    lambda_P=-1/del_t+mu*U(2:n,j-1)+l_P+r_P+x_P;
    lambda_N=-1/del_t+l_N+r_N;
    lambda_U=-1/del_t-mu*P(2:n,j-1)+e+l_U+x_U;
    lambda_E=-1/del_t+l_E+r_E;
    lambda_O=-1/del_t+l_O+r_O+x_O;

    %calculating gammas
    gamma_V=delta_PV/(del_s^2)*(P(3:n+1,j-1)-2*P(2:n,j-1)+P(1:n-1,j-1))...
        -alpha*R(2:n,1);
    gamma_P=-r_O*O(2:n,j-1)-r_N*N(2:n,j-1)-f*V(2:n,j-1)+...
        delta_VP/(del_s^2)*(V(3:n+1,j-1)-2*V(2:n,j-1)+V(1:n-1,j-1))+...
        delta_OP/(del_s^2)*(O(3:n+1,j-1)-2*O(2:n,j-1)+O(1:n-1,j-1))...
        -i_P*I(2:n,1)-S(2:n,1);
    gamma_N=-r_P*P(2:n,j-1)+...
        delta_PN/(del_s^2)*(P(3:n+1,j-1)-2*P(2:n,j-1)+P(1:n-1,j-1));
    gamma_U=delta_PU/(del_s^2)*(P(3:n+1,j-1)-2*P(2:n,j-1)+P(1:n-1,j-1))...
        -i_U*I(2:n,1);
    gamma_E=-e*U(2:n,j-1)+...
        delta_PE/(del_s^2)*(P(3:n+1,j-1)-2*P(2:n,j-1)+P(1:n-1,j-1))+...
        delta_UE/(del_s^2)*(U(3:n+1,j-1)-2*U(2:n,j-1)+U(1:n-1,j-1));
    gamma_O=-r_E*E(2:n,j-1)+...
        delta_PO/(del_s^2)*(P(3:n+1,j-1)-2*P(2:n,j-1)+P(1:n-1,j-1))+...
        delta_EO/(del_s^2)*(E(3:n+1,j-1)-2*E(2:n,j-1)+E(1:n-1,j-1))...
        -i_O*I(2:n,1);

    %declaring b vectors
    b_V = lambda_V.*V(2:n,j-1)+gamma_V;
    b_P = lambda_P.*P(2:n,j-1)+gamma_P;

```

```

b_N = lambda_N.*N(2:n,j-1)+gamma_N;
b_U = lambda_U.*U(2:n,j-1)+gamma_U;
b_E = lambda_E.*E(2:n,j-1)+gamma_E;
b_O = lambda_O.*O(2:n,j-1)+gamma_O;

%adding boundary values
b_V(1,1)=b_V(1,1)-aV(1,1)*V(1,j);
b_V(n-1,1)=b_V(n-1,1)-aV(1,1)*V(n+1,j);

b_P(1,1)=b_P(1,1)-aP(1,1)*P(1,j);
b_P(n-1,1)=b_P(n-1,1)-aP(1,1)*P(n+1,j);

b_N(1,1)=b_N(1,1)-aN(1,1)*N(1,j);
b_N(n-1,1)=b_N(n-1,1)-aN(1,1)*N(n+1,j);

b_U(1,1)=b_U(1,1)-aU(1,1)*U(1,j);
b_U(n-1,1)=b_U(n-1,1)-aU(1,1)*U(n+1,j);

b_E(1,1)=b_E(1,1)-aE(1,1)*E(1,j);
b_E(n-1,1)=b_E(n-1,1)-aE(1,1)*E(n+1,j);

b_O(1,1)=b_O(1,1)-aO(1,1)*O(1,j);
b_O(n-1,1)=b_O(n-1,1)-aO(1,1)*O(n+1,j);

%calculate V,P,N,U,E,O
V(2:n,j)=A_V*b_V;
P(2:n,j)=A_P*b_P;
N(2:n,j)=A_N*b_N;
U(2:n,j)=A_U*b_U;
E(2:n,j)=A_E*b_E;
O(2:n,j)=A_O*b_O;

end

%Plot populations

```

```
scrsz = get(0,'ScreenSize');
figure('Position',[0 0 scrsz(3) scrsz(3)/2])
```

```
%Plot V(t)
subplot(3,2,1)
mesh(Price,Time,V')
title('Virgin Metal V(s,t)')
xlabel('Price')
ylabel('Time')
zlabel('Amount')
```

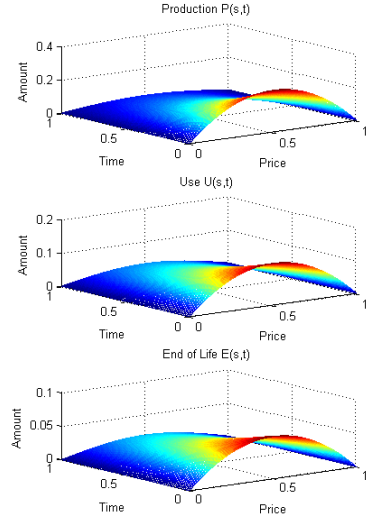
```
%Plot P(s,t)
subplot(3,2,2)
mesh(Price,Time,P')
title('Production P(s,t)')
xlabel('Price')
ylabel('Time')
zlabel('Amount')
```

```
%Plot N(s,t)
subplot(3,2,3)
mesh(Price,Time,N')
title('New Scrap N(s,t)')
xlabel('Price')
ylabel('Time')
zlabel('Amount')
```

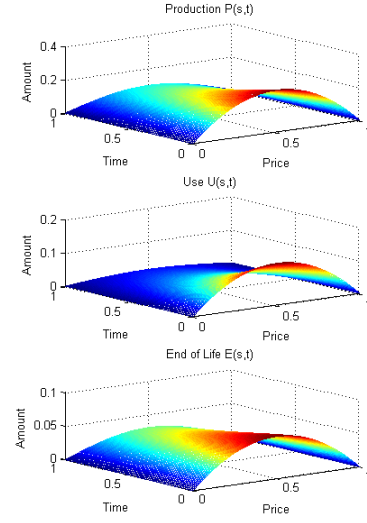
```
%Plot U(s,t)
subplot(3,2,4)
mesh(Price,Time,U')
title('Use U(s,t)')
xlabel('Price')
ylabel('Time')
zlabel('Amount')
```

```
%Plot O(s,t)
subplot(3,2,5)
mesh(Price,Time,O')
title('Old Scrap O(s,t)')
xlabel('Price')
ylabel('Time')
zlabel('Amount')

%Plot E(s,t)
subplot(3,2,6)
mesh(Price,Time,E')
title('End of Life E(s,t)')
xlabel('Price')
ylabel('Time')
zlabel('Amount')
```

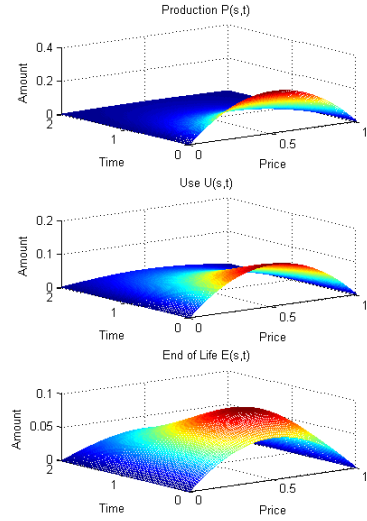


(a) $\mu = 0.1, e = 0.4, \delta_{PP} = 0.2,$
 $\delta_{UU} = 0.1, \delta_{EE} = 0.2$

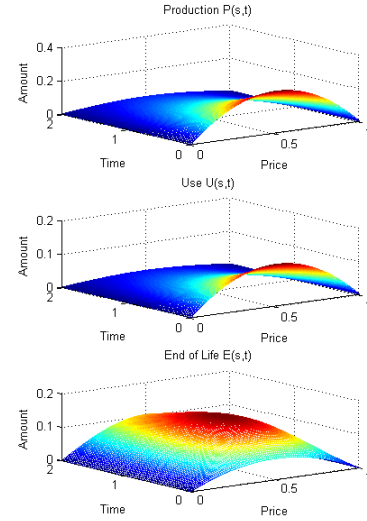


(b) $\mu = 0.2, e = 0.2, \delta_{PP} = 0.1,$
 $\delta_{UU} = 0.2, \delta_{EE} = 0.1$

Figure C.6: Comparison of two solutions to the *PDE PUE Model*

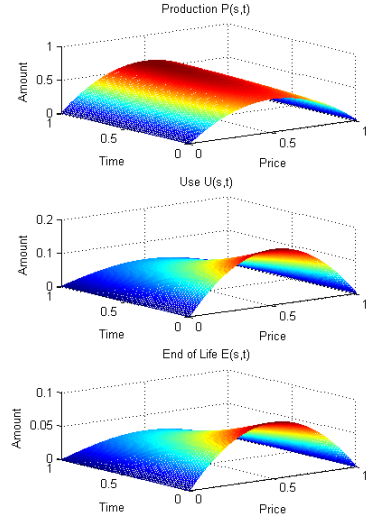


(a) $\mu = 0.1, e = 0.4, \delta_{PP} = 0.2, \delta_{UU} =$
 $0.1, \delta_{EE} = 0.2, \delta_{PU} = 0.05, \delta_{PE} =$
 $0.05, \delta_{UE} = 0.1$

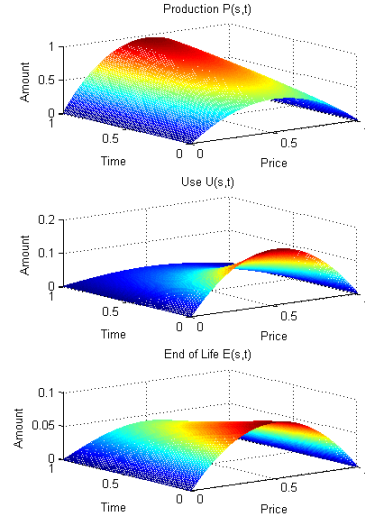


(b) $\mu = 0.2, e = 0.2, \delta_{PP} = 0.1, \delta_{UU} =$
 $0.2, \delta_{EE} = 0.1, \delta_{PU} = 0.05, \delta_{PE} =$
 $0.1, \delta_{UE} = 0.05$

Figure C.7: Comparison of two solutions to the *PDE PUE Model with Price Cross Diffusion*

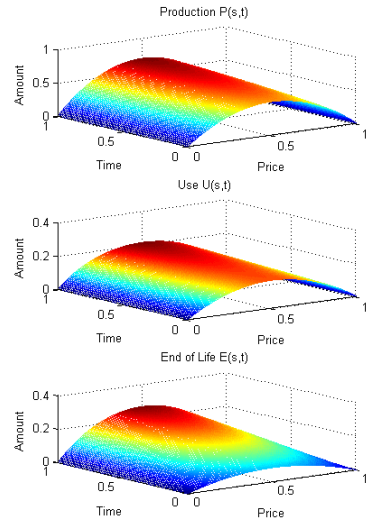


(a) $\mu = 0.1, e = 0.4, \delta_{PP} = 0.2, \delta_{UU} = 0.1, \delta_{EE} = 0.2$

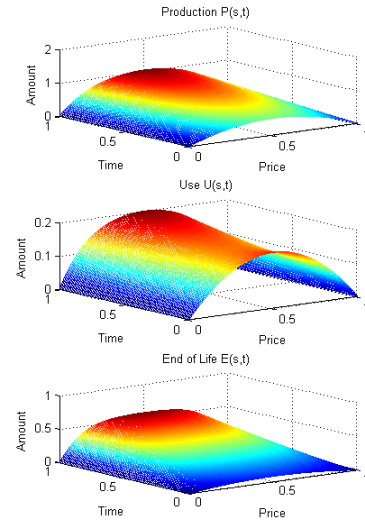


(b) $\mu = 0.2, e = 0.2, \delta_{PP} = 0.1, \delta_{UU} = 0.2, \delta_{EE} = 0.1$

Figure C.8: Comparison of two solutions to the *PDE PUEP Model*

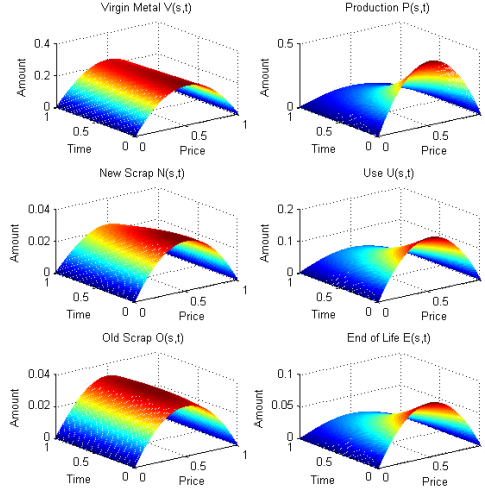


(a) $\mu = 0.1, e = 0.4, \delta_{PP} = 0.2, \delta_{UU} = 0.1, \delta_{EE} = 0.2, \delta_{EP} = 0.1, \delta_{PU} = 0.05, \delta_{PE} = 0.05, \delta_{UE} = 0.1$

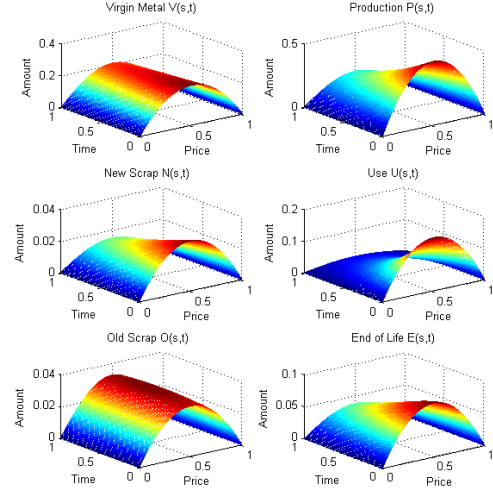


(b) $\mu = 0.2, e = 0.2, \delta_{PP} = 0.1, \delta_{UU} = 0.2, \delta_{EE} = 0.1, \delta_{EP} = 0.05, \delta_{PU} = 0.05, \delta_{PE} = 0.1, \delta_{UE} = 0.05$

Figure C.9: Comparison of two solutions to the *PDE PUEP Model with Price Cross Diffusion*

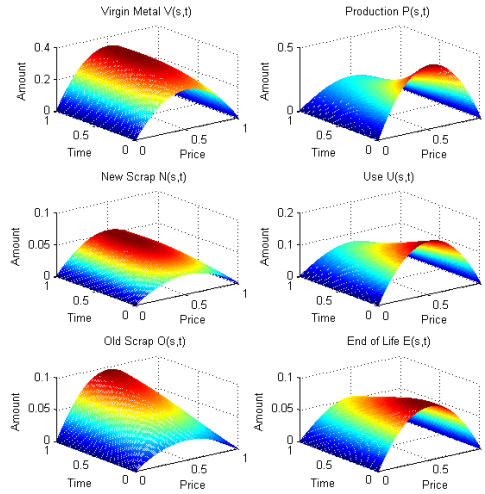


(a) $\mu = 0.1, e = 0.4, \delta_{VV} = 0.01, \delta_{PP} = 0.2, \delta_{NN} = 0.02, \delta_{UU} = 0.1, \delta_{EE} = 0.2, \delta_{OO} = 0.01$

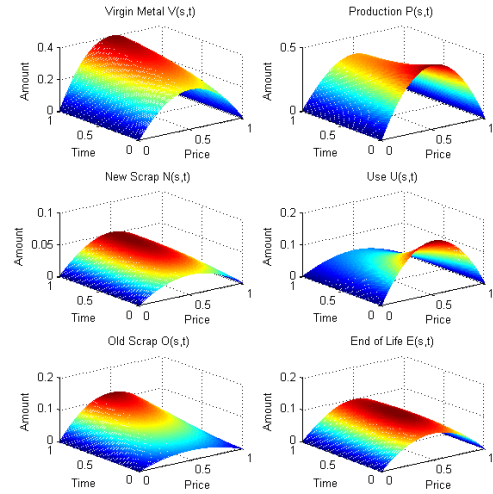


(b) $\mu = 0.1, e = 0.4, \delta_{VV} = 0.02, \delta_{PP} = 0.1, \delta_{NN} = 0.1, \delta_{UU} = 0.2, \delta_{EE} = 0.1, \delta_{OO} = 0.02$

Figure C.10: Comparison of two solutions to the *PDE PUEP Model with Inflow Domains*



(a) $\mu = 0.1, e = 0.4, \delta_{VV} = 0.01, \delta_{PP} = 0.2, \delta_{NN} = 0.02, \delta_{UU} = 0.1, \delta_{EE} = 0.2, \delta_{OO} = 0.01, \delta_{PV} = 0.1, \delta_{VP} = \delta_{OP} = 0.01, \delta_{PN} = \delta_{PU} = \delta_{PE} = \delta_{UE} = \delta_{PO} = \delta_{EO} = 0.2$



(b) $\mu = 0.1, e = 0.4, \delta_{VV} = 0.02, \delta_{PP} = 0.1, \delta_{NN} = 0.1, \delta_{UU} = 0.2, \delta_{EE} = 0.1, \delta_{OO} = 0.02, \delta_{PV} = 0.1, \delta_{VP} = \delta_{OP} = 0.01, \delta_{PN} = \delta_{PU} = \delta_{PE} = \delta_{UE} = \delta_{PO} = \delta_{EO} = 0.2$

Figure C.11: Comparison of two solutions to the *PDE PUEP Model with Inflow Domains and Price Cross Diffusion*

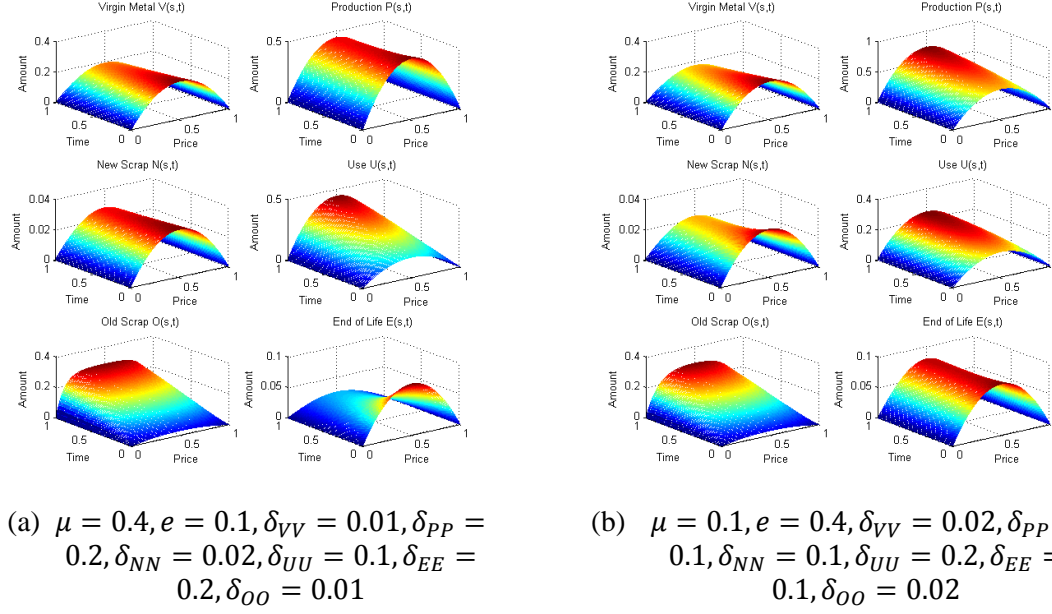


Figure C.12: Comparison of two solutions to the *PDE PUEP Model with Inflow Domains and Trade and Stock Considerations*

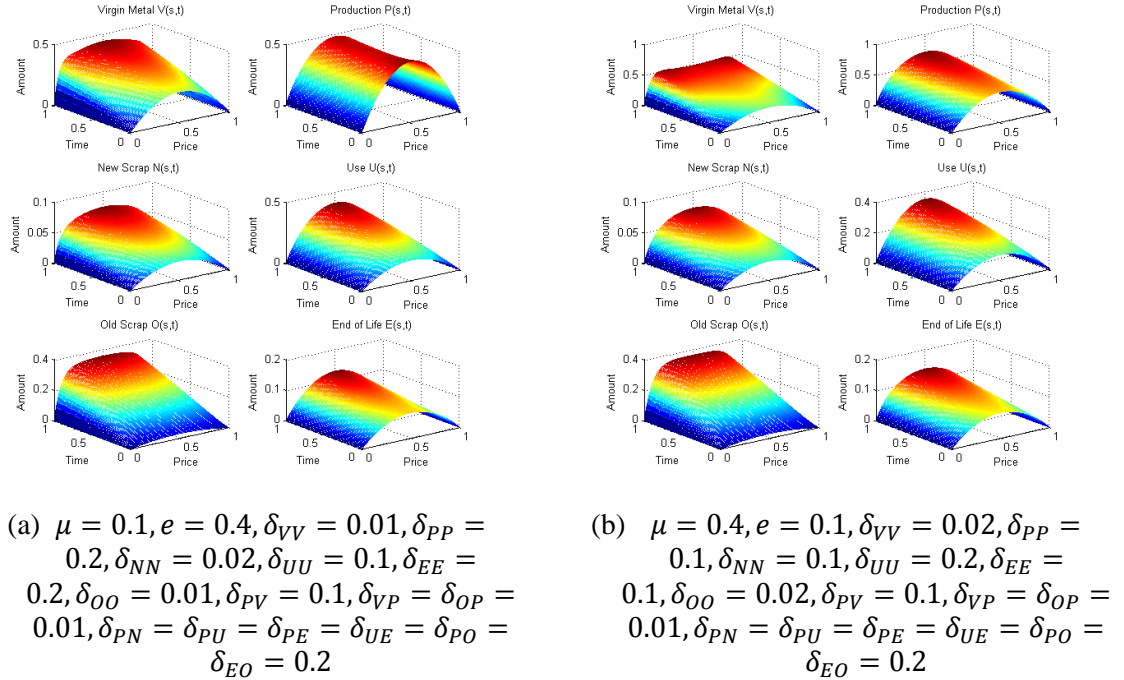


Figure C.13: Comparison of two solutions to the *PDE PUEP Model with Inflow Domains, Trade and Stock Considerations and Price Cross Diffusion*

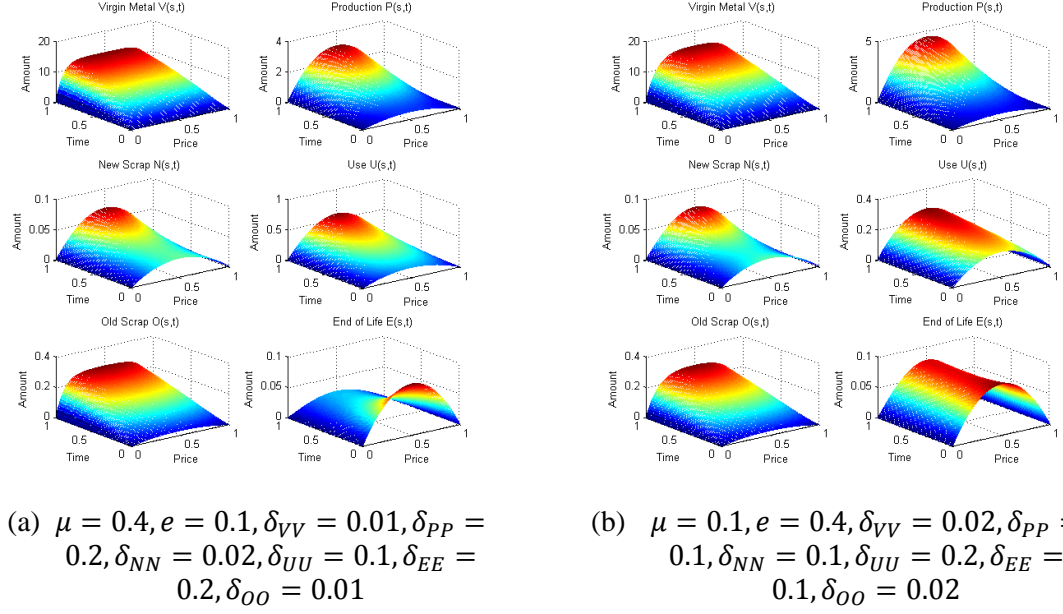


Figure C.14: Comparison of two solutions to the *Modified PDE PUEP Model with Inflow Domains and Trade and Stock Considerations*

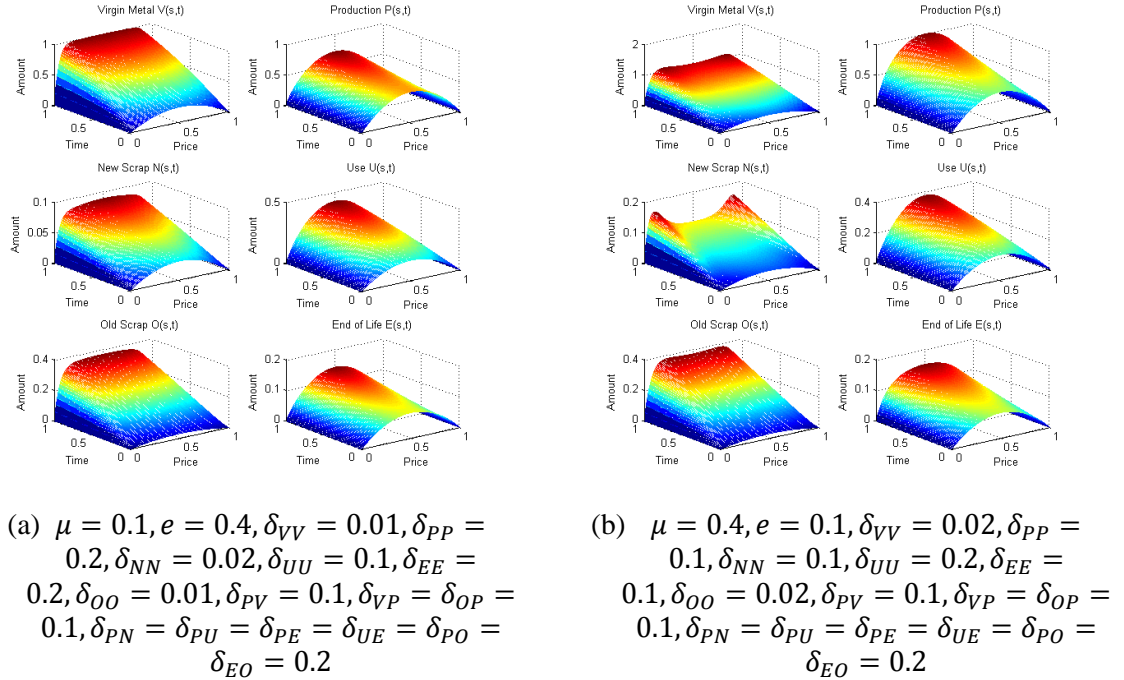


Figure C.15: Comparison of two solutions to the *Modified PDE PUEP Model with Inflow Domains, Trade and Stock Considerations and Price Cross Diffusion*

C.8 Euler-Maruyama Method for the SDE PUE Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: PUE_SDE.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Matlab function that computes the Euler's Method for
%         the SDE PUE model
%Date Created: 8 April 2011
%Date Last Modified: 8 April 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;

%defining time domain and intervals
t0 = 2000;
t1 = 2007;
n = 100;
del_t = (t1-t0)/n;
Time = t0:del_t:t1;
orderT=0.000000001;    %used to create a positive definite covariance matrix

%defining populations
P = zeros(n+1,1);
U = zeros(n+1,1);
E = zeros(n+1,1);

%initial conditions
P(1)=2;
U(1)=2;
E(1)=1;

%initializing constants
mu=0.1;
e=0.1;
```

```

%3 individual plots
scrsz = get(0,'ScreenSize');
figure('Position',[0 0 scrsz(3)/2 scrsz(3)/2])

for j = 1:3
%Calculating populations
    for i=1:n
        dw1=.25*randn;
        dw2=.25*randn;
        dw3=.25*randn;

        C=[mu*P(i)*U(i)+orderT, -mu*P(i)*U(i), 0;...
            -mu*P(i)*U(i), mu*P(i)*U(i)+e*U(i)+orderT,-e*U(i);...
            0, -e*U(i), e*U(i)+orderT];
        B=sqrtm(C);

        P(i+1)=P(i)-del_t*mu*P(i)*U(i)+...
            sqrt(del_t)*B(1,1)*dw1+...
            sqrt(del_t)*B(1,2)*dw2+...
            sqrt(del_t)*B(1,3)*dw3;
        U(i+1)=U(i)+del_t*(mu*P(i)*U(i)-e*U(i))+...
            sqrt(del_t)*B(2,1)*dw1+...
            sqrt(del_t)*B(2,2)*dw2+...
            sqrt(del_t)*B(2,3)*dw3;
        E(i+1)=E(i)+del_t*e*U(i)+...
            sqrt(del_t)*B(3,1)*dw1+...
            sqrt(del_t)*B(3,2)*dw2+...
            sqrt(del_t)*B(3,3)*dw3;

%eliminate negative amounts
        if P(i+1)<=0
            P(i+1)=0;
        end
    end
end

```

```

        if U(i+1)<=0
            U(i+1)=0;
        end

        if E(i+1)<=0
            E(i)=0;
        end
    end

    %Plot P(t)
    hold on;
    subplot(3,1,1)
    if j == 1
        plot(Time,P,'b');
    elseif j == 2
        plot(Time,P,'r');
    elseif j ==3
        plot(Time,P,'g');
    end
    title('Production P(t)')
    xlabel('Time')
    ylabel('Amount')

    %Plot U(t)
    hold on;
    subplot(3,1,2)
    if j == 1
        plot(Time,U,'b');
    elseif j == 2
        plot(Time,U,'r');
    elseif j ==3
        plot(Time,U,'g');
    end
end

```

```

title('Use U(t)')
xlabel('Time')
ylabel('Amount')

%Plot E(t)
hold on;
subplot(3,1,3)
if j == 1
    plot(Time,E,'b');
elseif j == 2
    plot(Time,E,'r');
elseif j ==3
    plot(Time,E,'g');
end
title('End of Life E(t)')
xlabel('Time')
ylabel('Amount')
end

```

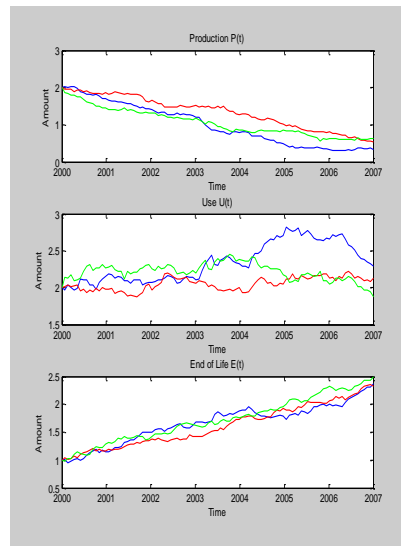


Figure C.16: Comparison of three trajectories for the *SDE PUE Model*

C.9 Euler-Maruyama Method for the SDE PUEP Model

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: PUEP_SDE.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Matlab function that computes the Euler's method for
%         the PUEP SDE model with metal losses
%Date Created: 8 April 2011
%Date Last Modified: 8 April 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;

%defining time domain and intervals
t0 = 1997;
t1 = 2007;
n = 100;
del_t = (t1-t0)/n;
Time = t0:del_t:t1;

%defining populations
P = zeros(1,n+1);      %production
U = zeros(1,n+1);      %use
E = zeros(1,n+1);      %end-of-life

%initial conditions
P(1,1)=2;
U(1,1)=2;
E(1,1)=1;
v = 10;                %ore reserve constant

%initializing constants
l_P = .01; %production lose rate
l_U = .02; %use lose rate
```



```

l_E = .03; %end-of-life lose rate

a=.02;      %ore availibility
mu=.2;      %supply/demand coefficient
r=.25;      %recyclible rate
e=.2;      %end-of-life coefficient

for j = 1:3
    %Calculating populations
    for i=1:n
        dw1=.25*randn;
        dw2=.25*randn;
        dw3=.25*randn;

        C=[a*v+mu*P(i)*U(i)+r*E(i)+l_P*P(i), -mu*P(i)*U(i), -r*E(i);...
            -mu*P(i)*U(i), mu*P(i)*U(i)+e*U(i)+l_U*U(i),-e*U(i);...
            -r*E(i), -e*U(i), r*E(i)+e*U(i)+l_E*E(i)];
        B=sqrtm(C);

        P(i+1)=P(i)+del_t*(a*v-mu*P(i)*U(i)+r*E(i)-l_P*P(i))+...
            sqrt(del_t)*B(1,1)*dw1+...
            sqrt(del_t)*B(1,2)*dw2+...
            sqrt(del_t)*B(1,3)*dw3;
        U(i+1)=U(i)+del_t*(mu*P(i)*U(i)-e*U(i)-l_U*U(i))+...
            sqrt(del_t)*B(2,1)*dw1+...
            sqrt(del_t)*B(2,2)*dw2+...
            sqrt(del_t)*B(2,3)*dw3;
        E(i+1)=E(i)+del_t*(e*U(i)-l_E*E(i)-r*E(i))+...
            sqrt(del_t)*B(3,1)*dw1+...
            sqrt(del_t)*B(3,2)*dw2+...
            sqrt(del_t)*B(3,3)*dw3;
        v = v-a*v;

        %eliminate negative amounts

```

```

        if P(i+1)<=0
            P(i+1)=0;
        end

        if U(i+1)<=0
            U(i+1)=0;
        end

        if E(i+1)<=0
            E(i+1)=0;
        end

        if v<=0
            v=0;
        end

    end

%Plot P(t)
hold on;
subplot(3,1,1)
if j == 1
    plot(Time,P,'b');
elseif j == 2
    plot(Time,P,'r');
elseif j ==3
    plot(Time,P,'g');
end
title('Production P(t)')
xlabel('Time')
ylabel('Amount')

%Plot U(t)
hold on;

```

```

subplot(3,1,2)
if j == 1
    plot(Time,U,'b');
elseif j == 2
    plot(Time,U,'r');
elseif j ==3
    plot(Time,U,'g');
end
title('Use U(t)')
xlabel('Time')
ylabel('Amount')

%Plot E(t)
hold on;
subplot(3,1,3)
if j == 1
    plot(Time,E,'b');
elseif j == 2
    plot(Time,E,'r');
elseif j ==3
    plot(Time,E,'g');
end
title('End of Life E(t)')
xlabel('Time')
ylabel('Amount')
end

```

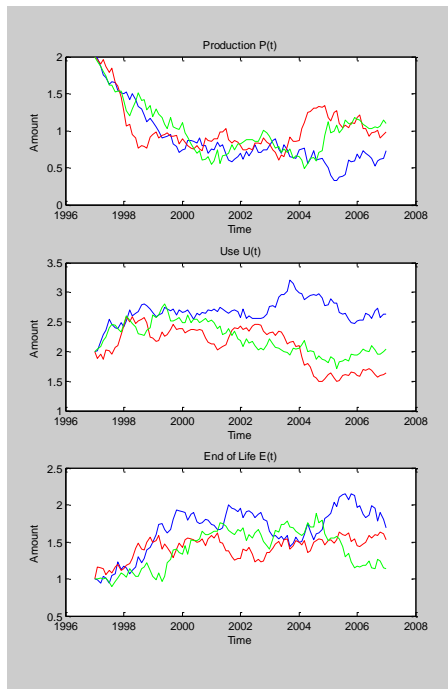


Figure C.17: Comparison of three trajectories for the *SDE PUEP Model*

C.10 Euler-Maruyama Method for the SDE PUEP Model with Inflow Domains

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%File Name: PUEP_inflow_SDE.m
%Author: Marc A Angulo
%Course: Thesis
%Purpose: Matlab function that computes the Euler's method for
%         the PUEP SDE model with inflow domains
%Date Created: 8 April 2011
%Date Last Modified: 8 April 2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%defining time domain and intervals
t0 = 0;
t1 = 1;
n = 100;
del_t = (t1-t0)/n;
Time = 0:del_t:t1;

%defining populations
V = zeros(n+1,1);
P = zeros(n+1,1);
N = zeros(n+1,1);
U = zeros(n+1,1);
E = zeros(n+1,1);
O = zeros(n+1,1);

%initial conditions
V(1)=2;
P(1)=2;
N(1)=2;
U(1)=2;
E(1)=1;
O(1)=2;
```

```

%initializing constants

%metal loss rates
l_V = .1;
l_P = .1;
l_N = .1;
l_U = .1;
l_E = .1;
l_O = .1;

%metal recyclable rates
r_O = .5;
r_N = .5;
r_P = .5;
r_E = .5;

a=.1;
mu=.1;
e=.1;

%Plot populations
scrsz = get(0,'ScreenSize');
figure('Position',[0 0 scrsz(3)/2 scrsz(3)/2])

for j = 1:3
    %Calculating populations
    for i=1:n
        dw1=.5*randn;
        dw2=.5*randn;
        dw3=.5*randn;
        dw4=.5*randn;
        dw5=.5*randn;
        dw6=.5*randn;
    end
end

```

```

C=[a*v(i)+l_v*v(i),-a*v(i),0,0,0,0;...
   -a*v(i),a*v(i)+mu*p(i)*u(i)+r_o*o(i)+r_n*n(i)+r_p*p(i)+l_p*p(i),...
   -r_n*n(i)-r_p*p(i),-mu*p(i)*u(i),0,-r_o*o(i);...
   0,-r_n*n(i)-r_p*p(i),r_n*n(i)+r_p*p(i)+l_n*n(i),0,0,0;...
   0,-mu*p(i)*u(i),0,mu*p(i)*u(i)+e*u(i)+l_u*u(i),-e*u(i),0;...
   0,0,0,-e*u(i),e*u(i)+r_e*e(i)+l_e*e(i),-r_e*e(i);...
   0,-r_o*o(i),0,0,-r_e*e(i),r_o*o(i)+r_e*e(i)+l_o*o(i)];
B=sqrtm(C);

A=sqrt(del_t)*...
   [dw1*B(:,1) dw2*B(:,2) dw3*B(:,3) dw4*B(:,4) dw5*B(:,5) dw6*B(:,6)];

V(i+1)=V(i)+del_t*(-l_v*v(i)-a*v(i))+sum(A(1,:));
P(i+1)=P(i)+del_t*(-l_p*p(i)-mu*p(i)*u(i)+r_o*o(i)...
   +r_n*n(i)-r_p*p(i)+a*v(i))+sum(A(2,:));
N(i+1)=N(i)+del_t*(-l_n*n(i)+r_p*p(i)-r_n*n(i))+sum(A(3,:));
U(i+1)=U(i)+del_t*(-l_u*u(i)+mu*p(i)*u(i)-e*u(i))+sum(A(4,:));
E(i+1)=E(i)+del_t*(-l_e*e(i)+e*u(i)-r_e*e(i))+sum(A(5,:));
O(i+1)=O(i)+del_t*(-l_o*o(i)+r_e*e(i)-r_o*o(i))+sum(A(6,:));
end

%Plot V(t)
hold on;
subplot(3,2,1)
if j == 1
    plot(Time,V,'b');
elseif j == 2
    plot(Time,V,'r');
elseif j ==3
    plot(Time,V,'g');
end
title('Virgin Metal V(t)')
xlabel('Time')
ylabel('Amount')

```

```

%Plot P(t)
hold on;
subplot(3,2,2)
if j == 1
    plot(Time,P,'b');
elseif j == 2
    plot(Time,P,'r');
elseif j ==3
    plot(Time,P,'g');
end
title('Production P(t)')
xlabel('Time')
ylabel('Amount')

```

```

%Plot N(t)
hold on;
subplot(3,2,3)
if j == 1
    plot(Time,N,'b');
elseif j == 2
    plot(Time,N,'r');
elseif j ==3
    plot(Time,N,'g');
end
title('New Scrap N(t)')
xlabel('Time')
ylabel('Amount')

```

```

%Plot U(t)
hold on;
subplot(3,2,4)
if j == 1
    plot(Time,U,'b');

```



```

elseif j == 2
    plot(Time,U,'r');
elseif j ==3
    plot(Time,U,'g');
end
title('Use U(t)')
xlabel('Time')
ylabel('Amount')

%Plot E(t)
hold on;
subplot(3,2,5)
if j == 1
    plot(Time,E,'b');
elseif j == 2
    plot(Time,E,'r');
elseif j ==3
    plot(Time,E,'g');
end
title('End of Life E(t)')
xlabel('Time')
ylabel('Amount')

%Plot O(t)
hold on;
subplot(3,2,6)
if j == 1
    plot(Time,O,'b');
elseif j == 2
    plot(Time,O,'r');
elseif j ==3
    plot(Time,O,'g');
end
title('Old Scrap O(t)')

```

```

xlabel('Time')
ylabel('Amount')
end

```

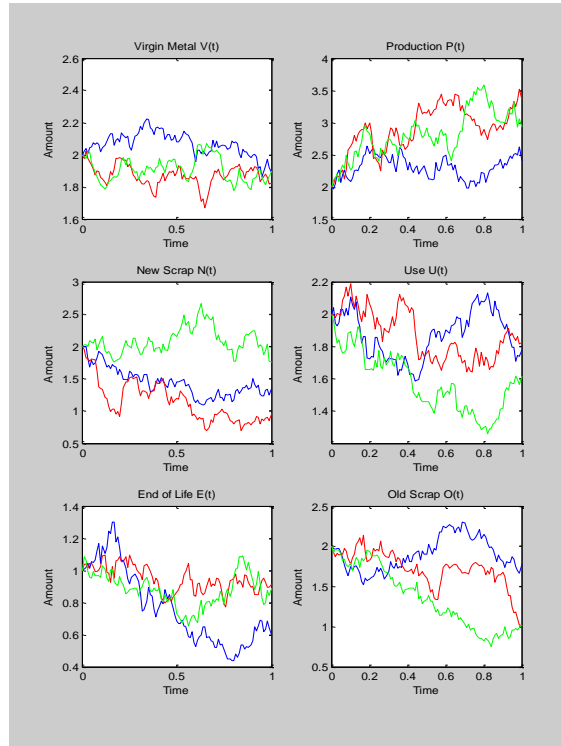


Figure C.18: Comparison of three trajectories for the *SDE PUEP Model with Inflow Domains*

Bibliography

Bibliography

- [1] E. Allen. Modeling with Ito stochastic differential equations, Springer, Netherlands, 2007.
- [2] K. T. Alligood, T. D. Sauer, J.A. Yorke. Chaos: and introduction to dynamical systems, Springer, New York, 1996.
- [3] M. A. Angulo. Mineral commodity summary: cesium, U.S. Geological Survey, January 2011. (Accessed June 6, 2011, at <http://minerals.usgs.gov/minerals/pubs/commodity/cesium/mcs-2011-cesiu.pdf>)
- [4] M. A. Angulo. 2009 Minerals yearbook: strontium, U.S. Geological Survey, December 2010. (Accessed June 6, 2011, at <http://minerals.usgs.gov/minerals/pubs/commodity/strontium/myb1-2009-stron.pdf>)
- [5] R. U. Ayes. Industrial metabolism technology and environment, National Academy Press, Washington, DC, 1989.
- [6] P. Baccini, H. P. Bader. Regionaler stoffhaushalt: Erfassung, Bewertung, Steuerung, Spektrum Askademischer Verlag GmbH, Heidelberg, 1996.
- [7] P. Baccini, P. H. Brunner. Metabolism of the anthroposphere, Springer, New York, 1991.
- [8] G. I. Barenblatt. Scaling, self-similarity, and intermediate asymptotics, Cambridge University Press, New York, 1996.
- [9] M. S. Bazaraa, H. D. Sherali, C. M. Shetty. Nonlinear programming: theory and application, third ed., Wiley, New Jersey, 2006.
- [10] D. Bernoulli. Essai d'une nouvelle analyse de la mortalité causée par la petite vérole, Mem Math Phy Acad Roy Sci Paris (1760) 1-45.
- [11] D. Betounes. Differential equations: theory and applications, second ed., Springer, New York, 2010.

- [12] R. L. Burden, J. D. Faires. Numerical analysis, eighth ed., Thomson, California, 2005.
- [13] P. H. Brunner, H. Rechberger. Practical handbook of material flow analysis, Lewis Publishers, New York, 2004.
- [14] C. Chicone. Ordinary differential equations with applications, Springer, New York, 1991.
- [15] R. H. Daw. Smallpox and the double decrement table: a piece of actuarial pre-history, *Jia*, 106 (1979), 299-318.
- [16] P. Duvigneaud, S. Denayeyer-De Smet. L'écosystème urbs. l'écosystème urbain bruxellois, *Productivité Biologique en Belgique*, (1975).
- [17] R. C. Free, ed. 21st century economics: a reference book, vol. 1, Sage, California, 2010.
- [18] A. Iserles. A first course in the numerical analysis of differential equations, second ed., Cambridge University Press, New York, 2009.
- [19] J. S. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique, *Princeton University Bulletin*. 13 (1902), 49-52.
- [20] H. W. Hethcote. Three Basic epidemiological models. *Biomath* 18 (1989), 119-144.
- [21] H. W. Hethcote. The mathematics of infectious diseases: *SIAM Rev*, 42, 4 (2000) 599-653.
- [22] J. J. Huntzicker, S. K. Friedlander, C. I. Davidson. Material balance for automobile-emitted lead in Los Angeles Basin, *Environ Sci Tech*, 9, 5 (1975), 448-457.
- [23] L. W. Jelinski, T. E. Graedel, R. A. Laudise, D. W. McCall, C. K. Patel. Industrial ecology: concepts and approaches, *Proc Nat Acad Sci USA*, 89 (1992), 793-797.
- [24] R. Keijin, E. Van der Voet, H. A. Udo de Haes. Controlling substance flows: the case of chlorine, *Environ Manage*, 8, 4 (1994), 523-542.
- [25] R. J. Lantzy, F. T. McKenzie. Atmospheric trace metals: global cycles and assessment of man's impact, *Geochim Cosmochim Acta*, 43, 4 (1979), 511-525.

- [26] MathWorks. About the GUIs in Matlab software, *undated*. (Accessed June 1, 2011, at http://www.mathworks.com/help/techdoc/creating_guis/f2-998197.html)
- [27] W. H. McNeill. Plagues and people, Anchor Press, New York, 1976.
- [28] M. M. Meerschaert. Mathematical modeling, third ed., Elsevier Academic Press, 2007.
- [29] Nobelprize.org. Ronald Ross - biography, *undated* (Accessed Jun 9, 2011, at http://nobelprize.org/nobel_prizes/medicine/laureates/1902/ross-bio.html)
- [30] K. Newcombe, I. D. Kalma, A. R. Aston. The metabolism of a city: the case of Hong Kong, *Ambio*, 7, 1 (1978), 3-15.
- [31] W. M. Stigliani, P. R. Jaffe, S. Anderberg. Heavy metal pollution in the Rhine Basin, *Environ Sci Technol*, 27, 5 (1993), 786-793.
- [32] S. F. Sibley. Using U.S. Geological Survey data in material flow analysis: an introduction. *J Ind Ecol*, 13, 5 (2009), 670-673.
- [33] I. Sushko, T. Puu, eds. Business cycle dynamics: models and tools, Springer, Berlin, 2006.
- [34] United Nations Environment Programme. Metal stocks in society: scientific synthesis, *undated* (Accessed May 23, 2011, at <http://www.unep.org/metalstocks/>)
- [35] U.S. Geological Survey. Commodity statistics and information, April 2011. (Accessed May 23, 2011, at <http://minerals.usgs.gov/minerals/pubs/commodity/>)
- [36] U.S. Geological Survey. Materials flow, May 2011. (Accessed May 23, 2011, at <http://minerals.usgs.gov/minerals/mflow/>)
- [37] A. Wolman. The metabolism of cities, *Sci Amer*, 213 (1965), 179-190.
- [38] D. Zwillinger. Handbook of differential equations, third ed., Academic Press, New York, 1998.

Curriculum Vitae

Marc Anthony Angulo earned an Associate in Applied Sciences in Aviation Technology from the Community College of the Air Force in 2005, a Bachelor of Science in Applied Mathematics and a Certification in Elements of Computing from the University of Texas at Austin in 2006, and a Graduate Certification in Computational Techniques and Applications from George Mason University in 2011. He is currently a Physical Scientist (Mineral Commodity Specialist) with the U.S. Geological Survey.