# APPLICATIONS AND NUMERICAL INVESTIGATION OF DIFFERENTIAL ALGEBRAIC EQUATIONS

by

David Ian Murray Milton

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science

in

The Faculty of Science
Modelling and Computational Science
University of Ontario Institue of Technology

May 2010

# Abstract

Differential-algebraic equations (DAEs) result in many areas of science and engineering. In this thesis, numerical methods for solving DAEs are compared for two problems, energy-economic models and traffic flow models. An energy-economic model is presented based on the Hubbert model of oil production and is extended to include economic factors for the first time. Using numerical methods to simulate the DAE model, the resulting graphs break the symmetry of the traditional Hubbert curve. For the traffic flow models, a numerical method is developed to solve the steady-state flow pattern including the linearly unstable regime, i.e. solutions which cannot be found with an initial value solver.

Key words: numerical method, differential-algebraic equation, oil production, traffic modelling

# Acknowledgements

Thanks to the professors and staff of the UOIT Science department for making the MSc program such an enjoyable experience. A special thank-you to Dr. Mark Staley for his help in developing the energy-economic models. Dr. Peter Berg, thank you for the mentorship and long meetings in your office over the past few years. Your guidance certainly has made this endeavour a success. Tanya, I appreciate your love and support. I love you too.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Differential Algebraic Equations

Differential algebraic equations (DAEs) are common in science and engineering problems. For a detailed introduction to DAEs see Ascher *et al.* [3] and Brenan *et al.* [8]. In general, the implicit form is

$$F(t, y, y') = 0, \tag{1.1.1}$$

where the prime denotes the derivative of $y$ with respect to $t$. This form looks similar to implicit ordinary differential equations (ODEs). The difference comes when we consider partitioning $y$, typically a vector-valued function, into two subvectors: $y_A$, the algebraic components and $y_D$, the differential components. Since $y_A$ is purely

algebraic, the system will not contain any terms $y'_A$. The system $F$ can be separated into two equations, one implicit ODE and an algebraic constraint

$$0 = f(t, y_A, y_D, y'_D), \tag{1.1.2}$$

$$0 = g(t, y_A, y_D). \tag{1.1.3}$$

In some cases, the algebraic constraint $g$ can be substituted into $f$. In other cases, the algebraic constraint can be solved for $y_A$ and then the derivative can be taken. For both of these cases, the DAE system is transformed into an ODE system so that it can be solved using ODE methods. Since this is not always possible, having methods to solve DAEs directly and without manipulation is desirable.

### 1.1.1 Classification of DAEs

DAEs are classified similarly to ODEs. The systems of equations can be implicit, explicit, or even a mix called semi-explicit. Other classifications, such as linear constant coefficient, linear time varying and nonlinear, also apply.

The index of a DAE is the minimum number of times all or part of the DAE system must be differentiated with respect to time in order to convert the DAE into an explicit ODE [9]. ODEs are therefore classified as index zero DAEs. Take for example the following semi-explicit DAE system

$$y'_D = f(t, y_A, y_D), \tag{1.1.4}$$

$$0 = g(t, y_A, y_D). \tag{1.1.5}$$

After differentiating the constraint, we get

$$y'_D = f(t, y_A, y_D), \tag{1.1.6}$$

$$g_{y_D}(t, y_A, y_D)y'_D + g_{y_A}(t, y_A, y_D)y'_A = -g_t(t, y_A, y_D). \tag{1.1.7}$$

If $g_{y_A}(t, y_A, y_D)$ is non-singular we can in principle solve (1.1.7) for $y'_A$ and the DAE is classified as index 1. If not, then the procedure of differentiation is continued until an explicit ODE is obtained. Often DAEs with an index greater than one are called higher-index systems. In general, numerical methods perform better for systems with lower indices.

### 1.1.2 Numerical Methods for DAEs

There are several numerical methods to solve DAEs. **Backward differentiation formulas** (BDFs) take the following form

$$\sum_{i=0}^{k} \alpha_i y_{n-i} = h\beta_0 y'_n. \tag{1.1.8}$$

Here, $k$ denotes the number of "steps" of this method. The first one-step method ($k = 1$) is the backward Euler method

$$y'_n = \frac{y_n - y_{n-1}}{h}. \tag{1.1.9}$$

The $k$-step BDF method is stable for $k < 7$ [10]. Whereas for $k \geq 7$, the method may be stable or unstable, depending on the problem. However, the accuracy of the method increases with $k$ ($k < 7$). BDF methods are used in conjunction with Newton's method to solve the system at each time step using past steps. A challenge for higher-order BDF methods is that more than one set of initial conditions are required. For a $k$-step method, $k$ initial conditions are required. These initial conditions need to be $O(h^p)$ accurate for a method of order $p$ and if an error tolerance is used, it must also be accurate to the given error. One approach to generate the initial conditions is to recursively use a $(k-1)$-step method to obtain the starting points for the $k$-step method.

Now consider the implicit DAE in standard form from equation (1.1.1), by substituting the BDF method

$$F(t_n, y_n, \frac{1}{h\beta_0}\sum_{i=0}^{k}\alpha_i y_{n-i}) = 0. \tag{1.1.10}$$

This DAE system can be solved using Newton's method at each time step using a $k$-step backward differentiation formula.

**DASPK** and **DASSL** are software packages written by Linda Petzold [24] in the

1980s to solve implicit ODEs and DAEs. The code is written in Fortran and is freely available for download.

The codes replace $y'_n$ with a BDF of order 1 to 5. Since the step size $h$ is not necessarily constant, a fixed leading coefficient form is used. A fixed-coefficient strategy has the advantage of simplicity but it introduces an error due to interpolation. A variable-coefficient strategy is much better in terms of error but at the expense of many evaluations of the Jacobian of the system. Fixed leading coefficients are a trade off between the two approaches. Details of the three different strategies are discussed by Ascher *et al.* [2].

As an initial guess for $y_n^0$, polynomial interpolation of the last $k+1$ values is used to obtain an accurate starting point $(y_{n-1}...y_{n-(k+1)})$.

DASPK has many features and options. For a small system, it is not too hard to write a function $F$ and the corresponding Jacobian function $JAC$. However, for a larger or more complicated system it may be difficult. By setting $INFO(5) = 0$, the DASPK code will calculate all the partial derivatives by numerical differences. The trade offs are the reduced accuracy in the generated Jacobian and the increased run time due to the additional function evaluations required to compute the Jacobian.

Another useful option helps with getting a consistent set of initial conditions, a crucial aspect of all DAEs. Since DAEs have an algebraic system we should not have to specify all the initial conditions. As can be seen from equations (1.1.4) and (1.1.5), by providing $y_D$, the initial conditions for $y_A$ and $y'_D$ can be solved for. Similarly by providing $y'_D$ the initial conditions for $y_A$ and $y_D$ can be solved for. DASPK provides the option to solve for either case by setting $INFO(11)$ appropriately. Initial guesses

are still required for all variables. Without a set of consistent initial conditions, DASPK is unable to start to solve the system and exits with an error.

## 1.2 Problems

In this thesis, we will study two mathematical models, which employ DAEs, by use of both analytical and numerical techniques. The main goal is to assess the accuracy and efficiency of various schemes.

### 1.2.1 Energy-Economic Model

Chapter 2 entails an approach to energy-economic modeling. The system presented is a non-linear, index 1, semi-explicit DAE. There are two equations, one for the rate of production of oil, $\dot{Q}$, and another for the price of oil, $p_E$. To the best of our knowledge, no other established (Hubbert) models of oil production in the literature include economic aspects.

### 1.2.2 Traffic Flow Model

Chapter 3 presents methods for solving macroscopic traffic flow models. The system of partial differential equations is solved on a spatial mesh creating a large system of ODEs.

When solving for the steady-state flow of the system, there are an infinite number of solutions. With addition of one algebraic equation and a simple parameter, a unique

solution can be selected. The novelty in our approach lies in the computation of linearly unstable, stationary traffic patterns that cannot be detected with corresponding initial value solvers.

# Chapter 2

# Energy-Economic Model

In this chapter, an energy-economic model is presented with a Hubbert-like oil production function and a classical economic production function. This work has been submitted for publication [6]. The focus of this chapter is on the numerical techniques for solving the DAE model and the results. An analytical solution does exist for a particular case and is used to assess the accuracy of the numerical results.

## 2.1 The Hubbert Curve and Peak Oil Theory

In the 1950s, Hubbert proposed a model to predict the rate of oil production $\frac{dQ}{dt}$, where $Q(t)$ is the amount in barrels extracted up until year $t$ [16]. The model was a bell-shaped curve given by the following differential equation

$$\frac{dQ}{dt} = b\,Q\,(R - Q)\,, \tag{2.1.1}$$

i.e., a logistic curve

$$Q(t) = \frac{R}{1 + ae^{-bRt}}, \tag{2.1.2}$$

where $R$ represents the initial reserves in place, and $a$ and $b$ are positive parameters used to fit the model to historical data, thereby allowing extrapolation of oil production. Equation (2.1.1) has two parts, an exponential growth term, $bQ$, and a geological constraint reflecting that it becomes increasingly difficult to extract oil as $Q \to R$. Obviously, we impose $0 \leq Q \leq R$. Differentiating equation (2.1.2) with respect to $t$ gives the oil production curve $E(t)$

$$E(t) = \frac{dQ}{dt} = \frac{bR^2}{4\cosh^2[bRt/2 - \ln(\sqrt{a})]} \tag{2.1.3}$$

which is symmetric about the peak at $Q = R/2$.

Using this model in 1956, Hubbert predicted the US oil peak of 1970 very accurately. Since then, the model has been applied to other regions with surprising accuracy. If the model can be applied to global oil production, the natural question is, when will the global peak in oil production occur?

## 2.2 The Model

Historical data shows that world oil demand increases exponentially, except during times of global recession. In order to meet this demand, production must also increase at an exponential rate. Since oil is only a finite resource, at some point the production rate must reach a plateau and then eventually the rate will also decrease. Oil production in Norway, the UK, and the USA has already followed this trend at a

| Variable | Description |
|---|---|
| $E$ | Rate of oil production (barrels / year) |
| $K$ | Capital, money invested ($ / year) |
| $L$ | Labour, population |
| $p_E$ | Price of oil ($ / barrel) |
| $Q$ | Cummulative oil production (barrels) |
| $Y$ | Economic output, GDP ($ / year) |

Table 2.1: Variables used in the model.

national level.

Our goal is to formulate equations for the world oil production that model this trend and also incorporate economic factors such as capital investment in oil infrastructure and labour. The following three sections cover the economic production function, the price for oil and oil production.

Table 2.1 lists the variables which will be used in the model in the sections that follow.

### 2.2.1 Production Function

A production function specifies the output of a company, an industry or country as an expression of all its inputs. For example, the revenues of a corporation (the output) are modeled as a function of the labour, energy, materials and other costs (the inputs).

The relationships between the inputs of a production function are an important aspect. One relationship is the elasticity of substitution, $\sigma$, a non-dimensional quantity that models the ability to replace one input by another while maintaining the same output.

An example of substitution is the balance of labour and machinery in a factory. If machinery could do all the work and no labour would be needed, then the elasticity of substitution would be $\sigma = \infty$. This also implies that labour could completely replace machinery. If on the other hand both labour and machinery are required and one could not replace the other, then the elasticity of substitution would be $\sigma = 0$.

Regarding the relationship between capital, $K$, and energy, $E\left(\frac{dQ}{dt}\right)$, it is defined as

$$\sigma^{-1} = -\frac{d\log\left[\left(\frac{\partial K}{\partial E}\right)_Y\right]}{d\log\left(\frac{K}{E}\right)} = \frac{d\log\left[\frac{\partial Y/\partial E}{\partial Y/\partial K}\right]}{d\log\left(\frac{K}{E}\right)}, \tag{2.2.1}$$

where $Y$ is output [22]. Later we make the assumption that all energy comes from oil so $E$ will be equivalent to the oil production rate.

Constant elasticity of substitution (CES) is a property of a production function. It means that the elasticity of substitution remains constant within the model, independent of the values of the input variables, $K$ and $E$ in our case.

Kemfert [18] used a two-level nested CES production function. Here, we will assume a slightly simpler CES production function [17]

$$Y = Y(E, K, L) = [\beta(BE)^\rho + (1-\beta)K^\rho]^{\frac{\alpha}{\rho}}(AL)^{1-\alpha}, \tag{2.2.2}$$

describing $Y$, the economic output or global GDP, where

$$\rho < 1, \quad 0 < \alpha, \quad \beta < 1. \tag{2.2.3}$$

In the above production function, $A$ is labour productivity, $B$ is energy efficiency,

and $L$ is labour. All three are assumed to grow exponentially with time as follows:

$$A(t) = A_0 e^{g_a t}, \tag{2.2.4}$$

$$B(t) = B_0 e^{g_b t} \tag{2.2.5}$$

and

$$L(t) = L_0 e^{g_l t}. \tag{2.2.6}$$

Exponential growth is a reasonable assumption since that matches with the trend in historical data and will likely occur for the foreseeable future. It can be shown that the constant parameter $\rho$ in equation (2.2.2) determines the elasticity of substitution between capital, $K$, and energy, $E$ and relates to $\sigma$ as follows

$$\rho = 1 - 1/\sigma. \tag{2.2.7}$$

Since $\sigma$ is defined on the domain $0 \leq \sigma < \infty$, $\rho$ is defined on the domain $-\infty < \rho \leq 1$. There are three interesting limiting cases for $\rho$. For $\rho \to -\infty$, the substitution of capital for energy is almost impossible and the resulting production function is

$$Y = [\min\{BE, K\}]^{\alpha} (AL)^{1-\alpha}. \tag{2.2.8}$$

When $\rho \approx 0$, the substitution of capital for energy is fairly easy. The resulting production function leads to a Cobb-Douglas model [12] which is solved analytically in section 2.3

$$Y = [(BE)^{\beta} K^{1-\beta}]^{\alpha} (AL)^{1-\alpha}. \tag{2.2.9}$$

The most unlikely case within the current world economy is $\rho \to 1$, which implies that money (capital) is a perfect substitute for oil

$$Y = [\beta BE + (1-\beta)K]^{\alpha}(AL)^{1-\alpha}. \tag{2.2.10}$$

### 2.2.2 Price of Oil

In order to determine the price of energy, $p_E$, we maximize the profit, $M_Y$, related to the production function (2.2.2). The profit is given by the output minus all inputs (or costs)

$$M_Y = Y - p_E E - r_K K - wL, \tag{2.2.11}$$

where $p_E$ denotes the price of oil, $r_K$ the rent on capital, and $w$ the wage for labour. Next we maximize the profit with respect to oil and capital to get two equations,

$$\frac{\partial M_Y}{\partial E} = 0 \Rightarrow p_E = \frac{\partial Y}{\partial E} \tag{2.2.12}$$

and

$$\frac{\partial M_Y}{\partial K} = 0 \Rightarrow r_K = \frac{\partial Y}{\partial K}. \tag{2.2.13}$$

Taking the partial derivative of the CES production function (2.2.2) with respect to $E$ and $K$ yields the following two relations

$$p_E = \alpha[\beta(BE)^{\rho} + (1-\beta)K^{\rho}]^{\frac{\alpha}{\rho}-1}(AL)^{1-\alpha}\beta B^{\rho}E^{\rho-1}\rho \tag{2.2.14}$$

and

$$r_K = \alpha[\beta(BE)^\rho + (1-\beta)K^\rho]^{\frac{\alpha}{\rho}-1}(AL)^{1-\alpha}(1-\beta)K^{\rho-1}\rho. \tag{2.2.15}$$

After dividing equation (2.2.14) by equation (2.2.15), we have an equation describing $K$ in terms of $E$,

$$K = E\left(\frac{(1-\beta)p_E}{\beta B^\rho r_K}\right)^{\frac{1}{1-\rho}}. \tag{2.2.16}$$

Substituting this expression back into equation (2.2.14) yields the price of oil as a function of oil consumption, $E$,

$$p_E = \frac{1}{E^{1-\alpha}}\alpha\beta B^\rho (AL)^{1-\alpha}\left[\beta B^\rho + (1-\beta)\left(\frac{(1-\beta)p_E}{\beta B^\rho r_k}\right)^{\frac{\rho}{1-\rho}}\right]^{\frac{\alpha}{\rho}-1}. \tag{2.2.17}$$

This equation will be the algebraic constraint of the DAE system to follow. Note that this implicit function for $p_E$ has a highly nonlinear $p_E$ term on the right hand side. In section 2.4, the applied numerical methods incorporate a root solver and use the rearranged explicit version of equation (2.2.17).

### 2.2.3 Supply of Oil

Based on Hubbert's model and supported by Pickering's recent research [25], oil production should scale with remaining reserves, $R - Q$, which represents the geological constraint. Oil supply should also scale with the amount of oil infrastructure, $K_E$, representing the capital investment in infrastructure such as pipelines and refineries. These dependencies suggest a model for oil production as follows

$$E = \frac{dQ}{dt} = C\,K_E^{\epsilon}(R-Q)^{1-\epsilon},\ 0<\epsilon<1. \tag{2.2.18}$$

The oil extraction efficiency (or productivity) parameter, $C$, increases exponentially with time,

$$C(t) = C_0 e^{g_c t}, \tag{2.2.19}$$

which represents technological advancement in the industry which is certainly continuing into the near future. This is not efficiency in the strict sense which varies between 0 to 100 percent but rather the number of barrels of oil that can be extracted per dollar of capital investment. Again, $R$ are the initial oil reserves, $Q$ is the total oil extracted to date, and $\dfrac{dQ}{dt} = E$ is the production rate, which are the same as the terms in the Hubbert model from equation (2.1.1).

The exponents of $\epsilon$ and $1-\epsilon$ are introduced to model diminishing returns on capital and reserves. Oil production does not scale linearly with capital. As $Q \to R$, it becomes increasingly more difficult to produce oil. Hence, more capital is required to extract every last drop. Pickering [25] proposed that $\epsilon$ be close to zero. A value of $\epsilon = 0.1$ will be used in this study unless stated otherwise. Figure 2.1 shows the diminishing returns for two cases, fixed reserves and fixed capital.

Our model has the correct scaling characteristics, sometimes referred to as the replication argument [15]. The hypothesis is that if there were two identical copies of planet Earth, the expectation is that there would be twice the oil production of one single planet. In other words, doubling $K_E$ and $R-Q$ results in doubling $E$.

Using the oil supply function (2.2.18), we can now derive a formula for the rate of oil

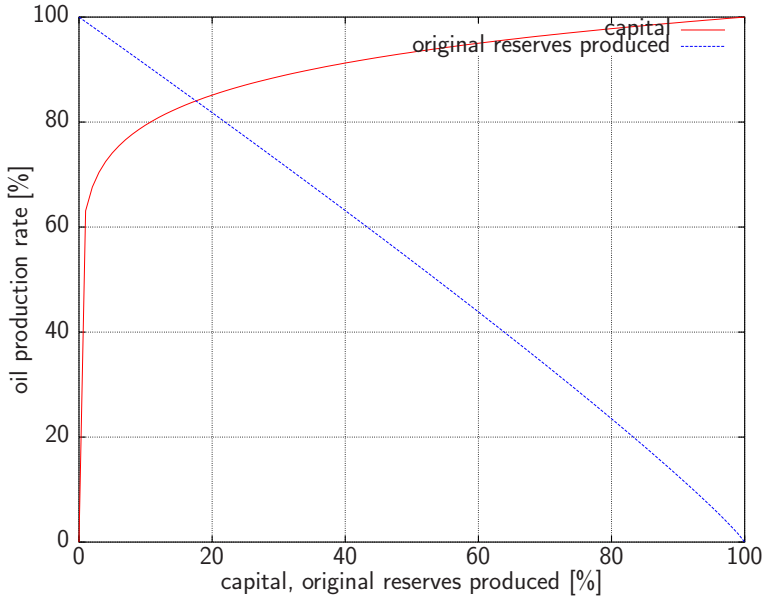



Figure 2.1: Diminishing returns of the oil supply function (2.2.18): Solid line - oil production rate, $E$, *vs.* oil capital, $K_E$, for fixed remaining reserves, $R - Q$; dashed line - production rate *vs.* remaining reserves, for fixed capital.

production, $\dfrac{dQ}{dt}$, as a function of the price of oil, $p_E$. Oil industry profits, $M_E$, are determined by the revenues minus the costs

$$M_E = p_E E - r_E K_E, \tag{2.2.20}$$

where $r_E$ is the rent on oil capital. Compared to the profit $M_Y$ in equation (2.2.11), $M_E$ is not a function of labour, $L$, and wage, $w$. It is assumed that labour costs are insignificant compared to the capital requirements for the oil industry. The oil industry is capital intensive, not labour intensive.

The result of maximizing profits with respect to oil capital, $K_E$, is

$$\frac{\partial M_E}{\partial K_E} = 0 \Rightarrow \frac{\partial E}{\partial K_E} = \frac{r_E}{p_E}. \tag{2.2.21}$$

Differentiating equation (2.2.18) with respect to $K_E$ and solving for $K_E$ yields,

$$K_E = \left(\frac{p_E\,\epsilon C(R-Q)^{1-\epsilon}}{r_E}\right)^{\frac{1}{1-\epsilon}} \tag{2.2.22}$$

$$= (R-Q)\left(\frac{p_E\,\epsilon C}{r_E}\right)^{\frac{1}{1-\epsilon}}. \tag{2.2.23}$$

Now we substitute $K_E$ into equation (2.2.18),

$$E = C\left(\frac{p_E\,\epsilon C}{r_E}\right)^{\frac{\epsilon}{1-\epsilon}} (R-Q)^{\epsilon}(R-Q)^{1-\epsilon}, \tag{2.2.24}$$

which can be simplified to

$$E = \frac{dQ}{dt} = C^{\frac{1}{1-\epsilon}}\left(\frac{p_E\,\epsilon}{r_E}\right)^{\frac{\epsilon}{1-\epsilon}} (R-Q). \tag{2.2.25}$$

The oil production function (2.2.25) is not only dependent on the usual variables, initial reserves, $R$, and cumulative production, $Q$, but now also a function of the price of oil, $p_E$. This in turn makes oil production dependent on the variables from equation (2.2.17), such as labour availability and productivity, which affect energy demand. We also note that there is a linear scaling, $E \sim (R-Q)$, which is in line with Pickering's research [25].

We now have a model for the price of oil and oil production. Equations (2.2.17) and

(2.2.25) can be re-written in semi-explicit form as follows:

$$\frac{dQ}{dt} = C^{\frac{1}{1-\epsilon}} \left( \frac{p_E \, \epsilon}{r_E} \right)^{\frac{\epsilon}{1-\epsilon}} (R - Q), \tag{2.2.26}$$

$$0 = p_E - \frac{\alpha\beta B^\rho [AL]^{1-\alpha}}{E(Q, p_E)^{1-\alpha}} \left[ \beta B^\rho + (1-\beta) \left( \frac{(1-\beta) p_E}{\beta B^\rho r_k} \right)^{\frac{\rho}{1-\rho}} \right]^{\frac{\alpha}{\rho}-1}. \tag{2.2.27}$$

The DAE system is comprised of a first-order differential equation (2.2.26) for $Q$ and an algebraic constraint (2.2.27).

In section 2.3, we will see how the model can be solved analytically for $\rho = 0$. In all other cases, the numerical methods presented in section 2.4 will be needed.

### 2.2.4 Parameter Values

In order to make use of the derived model, numerical values are required for the system parameters such as the labour productivity $A$, energy efficiency $B$, and productivity parameter $C$. Hanz determined the values as follows [14].

Kemfert made use of economic and industrial data to estimate substitution of elasticities for the German chemical industry [18]. Although they are not global values, they should provide an estimate for the energy efficiency parameter $B$ to within an order of magnitude and should allow reasonable results from the model.

The labour productivity parameter, $A$ can be estimated by using the CES production function (2.2.2), the previously determined value for $B$ and some current oil industry estimates for the remainder of the unknowns.

Using the oil production function (2.2.18) and oil industry estimates from British

Petroleum [7], for the current production of $E = 85\,\text{mbpd}$, capital invested, $K_E = \$\,1.4 \times 10^{12}$, and cumulative production, $Q = 10^{12}$ barrels, the productivity parameter $C$ can be determined.

Initial conditions required to obtain a solution are determined as follows. $Q_0(t_0)$ is chosen to be fixed for $t_0 = 1960$ and $Q_0 = Q(t_0) = 0.2\,R$. The initial value for $p_E(t_0)$ can then be determined by solving the algebraic constraint (2.2.27). Evaluating the oil production function (2.2.26) determines the initial production rate. Only one of the three variables $p_E$, $Q$ or $\dfrac{dQ}{dt}$ can be chosen and then the other two can be determined by the system. This could be considered to be a weakness of the model. We chose to select $Q$.

Table 2.2 lists all the parameters and their values for the model.

## 2.3 Results: An Analytical Model ($\rho \to 0$)

There are several limiting cases for the CES production function shown in equations (2.2.8) to (2.2.10). The case of $\rho \to 0$ is particularly interesting because it is possible to be solved analytically.

After taking the limit as $\rho \to 0$ for equation (2.2.17), we obtain (proof by L'Hôpital's rule [6])

$$p_E = \beta \left( \frac{\alpha B^{\alpha\beta} (1-\beta)^{\alpha(1-\beta)}}{r_K^{\alpha(1-\beta)}} \right)^{\frac{1}{1-\alpha(1-\beta)}} \left( \frac{AL}{E} \right)^{\frac{1-\alpha}{1-\alpha(1-\beta)}}. \tag{2.3.1}$$

Substituting this result into the our production function (2.2.25) we get

$$\frac{dQ}{dt} = D\,(R-Q) E^{-\frac{1-\alpha}{1-\alpha(1-\beta)}\frac{\epsilon}{1-\epsilon}} e^{gt}, \tag{2.3.2}$$

| Parameter | Description | Value |
|---|---|---|
| $A$ | Labour productivity | $2.98 \times 10^5$ \$ |
| $B$ | Energy efficiency | $273 \times 10^6$ \$/mega barrels |
| $C$ | Oil extraction efficiency/productivity | $1.5 \times 10^{-3}$ \$$^{1-\epsilon}$ mega barrels$^{\epsilon-1}$ |
| $g_a$ | Growth rate of labour productivity | 0.03 year$^{-1}$ |
| $g_b$ | Growth rate of energy efficiency | 0.015 year$^{-1}$ |
| $g_c$ | Growth rate of oil extraction efficiency/technology | 0.015 year$^{-1}$ |
| $g_l$ | Population growth rate | 0.01 year$^{-1}$ |
| $Q_0$ | Initial cumulative production | $0.2\,R$ |
| $r_E$ | Rent on oil capital | 0.1 |
| $r_K$ | Rent on capital | 0.1 |
| $R$ | Initial reserves | 2.2 trillion barrels |
| $\alpha$ | | 0.3 |
| $\beta$ | | 0.066 |
| $\epsilon$ | | 0.1 |

Table 2.2: Parameter values used in the model.

where $D$ is a collection of parameters and $g$ is a function of parameters as follows:

$$D = D(A_0, B_0, C_0, \alpha, \beta, \epsilon, r_E, r_K), \tag{2.3.3}$$

$$g = g(g_a, g_b, g_c, g_l; \alpha, \beta, \epsilon). \tag{2.3.4}$$

Equation (2.3.2) can further be simplified by defining a new parameter $\gamma$ as

$$\gamma = \frac{1-\alpha}{1-\alpha(1-\beta)} \frac{\epsilon}{1-\epsilon}, \tag{2.3.5}$$

which results in the simplified equation

$$\frac{dQ}{dt} = D\,(R-Q) E^{-\gamma} e^{gt}. \tag{2.3.6}$$

$E$ can be moved to the left-hand size by using the identity, $E = \dfrac{dQ}{dt}$, yielding

$$\left(\frac{dQ}{dt}\right)^{1+\gamma} = D\,(R-Q) e^{gt}. \tag{2.3.7}$$

Finally, a first-order ODE for oil production is given by

$$\frac{dQ}{dt} = D^{\frac{1}{1+\gamma}} (R-Q)^{\frac{1}{1+\gamma}} e^{Gt}, \tag{2.3.8}$$

where $G = g/(1+\gamma)$.

By separating $Q$ to the left hand side and $t$ to the right hand side

$$\frac{dQ}{(R-Q)^{\frac{1}{1+\gamma}}} = D^{\frac{1}{1+\gamma}} e^{Gt} dt, \tag{2.3.9}$$

this can be integrated as follows

$$\int_{Q_0}^{Q(t)} \frac{dQ'}{(R-Q')^{\frac{1}{1+\gamma}}} = \int_{t_0}^{t} D^{\frac{1}{1+\gamma}} e^{Gt'} dt', \tag{2.3.10}$$

where $Q_0 = Q(t_0)$. After integration, solving for $Q(t)$ results in the cumulative oil production function

$$Q(t) = R - \left[ (R-Q_0)^{\frac{\gamma}{1+\gamma}} + \frac{D^{\frac{1}{1+\gamma}}}{G} \left( e^{Gt_0} - e^{Gt} \right) \right]^{\frac{1+\gamma}{\gamma}}, \tag{2.3.11}$$

which cannot exceed $R$.

The oil production function can be determined by differentiating equation (2.3.11) with respect to time

$$E(t) \;=\; \frac{1+\gamma}{\gamma} \left[ (R-Q_0)^{\frac{\gamma}{1+\gamma}} + \frac{D^{\frac{1}{1+\gamma}}}{G} \left( e^{Gt_0} - e^{Gt} \right) \right]^{\frac{1}{\gamma}} D^{\frac{1}{1+\gamma}} e^{Gt}. \tag{2.3.12}$$

Figure 2.2 shows a graph of the analytical solution for the oil production function (2.3.12). Total reserves of $R = 2.2$ trillion barrels from [7] and the initial cumulative production in $t_0 = 1960$ is chosen to be $Q_0 = 0.2\,R$, which is in line with historical data. All other parameter values are from Table 2.2.

Unlike the Hubbert graph, this curve is certainly not symmetric about the production maximum. After peak production, the decline is much steeper than the approach to the peak. Also, Hubbert's model required an infinite amount of time for all the reserves to be extracted. In contrast, $Q(t) = R$ in a finite amount of time, given equation (2.3.11), at which point oil production will be zero.

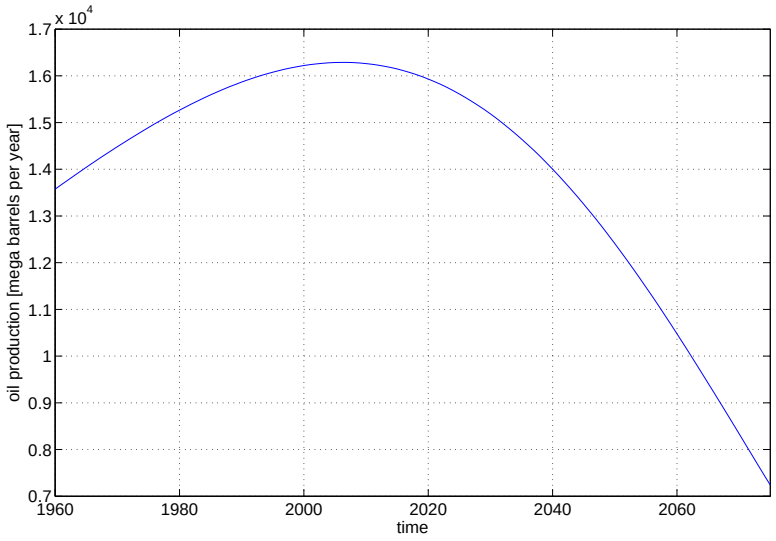



Figure 2.2: Analytical solution: Oil production *vs.* time for $\rho = 0$.

Figure 2.3 shows how sensitive the production curve is to the parameter for diminishing returns on capital, $\epsilon$. A slight change from the reference value $\epsilon = 0.10$ changes the height and year of the peak quite dramatically. Smaller values of $\epsilon$ cause earlier and higher peaks. This can be explained by looking at the relationship between oil supply and $\epsilon$ in equation (2.2.25) for $\epsilon \ll 1$,

$$E \sim p_E^{\frac{\epsilon}{1-\epsilon}} \approx p_E^{\epsilon}. \tag{2.3.13}$$

For small values of $\epsilon$, supply is not sensitive to the price, $p_E$. With this inelastic demand, there is higher consumption so the peak occurs earlier. The area under the curves in Figure 2.3 are the same which means that a higher peak will imply a steeper decline.

Additionally, the analytical solution can be used to observe the sensitivity of the

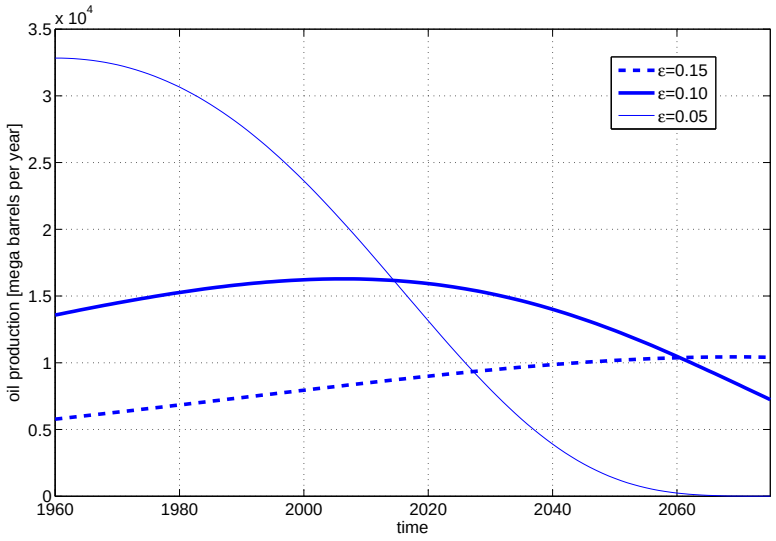



Figure 2.3: Oil production for $\rho = 0$ and three different values of $\epsilon$.

growth parameters, $g_i$. Figure 2.4 shows the resulting oil production curves for the growth parameter values listed in Table 2.2 and for a second case where all growth rates are zero for the no-growth case. Most notable when comparing the two curves is that the rate of decline of oil production is slow for the no-growth case and much steeper for the positive growth rates. Higher energy efficiency, oil extraction efficiency and labour productivity result in the production peak being pushed out further in time. However, because there is only a fixed amount of oil in the ground, higher efficiency and productivity result in a faster decline past the peak.

## 2.4 Numerical Methods

In the previous section, the model was solved analytically for the case $\rho \to 0$. This implies that the substitution of capital for energy is easy, which is not true in most

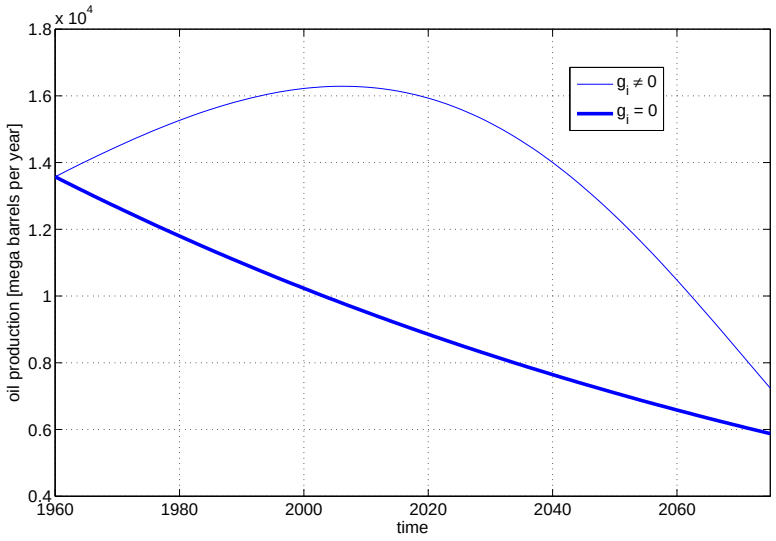



Figure 2.4: Oil production with non-zero exponential growth factors, $g_i \neq 0$, and with zero exponential growth factors, $g_i = 0$.

cases. Thus solving the model for arbitrary values of $\rho$ is desirable to see the impact on oil production and energy prices. Since solving the model analytically is not possible, the DAE system will need to be solved numerically.

We first start by characterizing the DAE system. The system can be quickly determined to be a non-linear, semi-explicit DAE. It is harder to determine the index at first glance. We start by re-writing the system of equations (2.2.17) and (2.2.25) in terms of the time-varying variables,

$$E = f(C, p_E, Q), \tag{2.4.1}$$

$$p_E = g(A, B, E, L, p_E), \tag{2.4.2}$$

with

$$f(C, p_E, Q) = \frac{dQ}{dt} = C^{\frac{1}{1-\epsilon}} \left( \frac{p_E \, \epsilon}{r_E} \right)^{\frac{\epsilon}{1-\epsilon}} (R - Q), \tag{2.4.3}$$

$$g(A, B, E, L, p_E) = \frac{1}{E^{1-\alpha}} \alpha\beta B^{\rho} (AL)^{1-\alpha} \left[ \beta B^{\rho} + (1-\beta) \left( \frac{(1-\beta) p_E}{\beta B^{\rho} r_k} \right)^{\frac{\rho}{1-\rho}} \right]^{\frac{\alpha}{\rho}-1}. \tag{2.4.4}$$

To find the index we take the derivative of equations (2.4.1) and (2.4.2) with respect to time

$$E' = \frac{\partial f}{\partial C} C' + \frac{\partial f}{\partial p_E} p'_E + \frac{\partial f}{\partial Q} Q', \tag{2.4.5}$$

$$p'_E = \frac{\partial g}{\partial A} A' + \frac{\partial g}{\partial B} B' + \frac{\partial g}{\partial E} E' + \frac{\partial g}{\partial L} L' + \frac{\partial g}{\partial p_E} p'_E. \tag{2.4.6}$$

Substituting equation (2.4.5) into equation (2.4.6) and rearranging all the $p'_E$ terms to the left hand side, we get

$$\left( 1 - \frac{\partial g}{\partial E} \frac{\partial f}{\partial p_E} - \frac{\partial g}{\partial p_E} \right) p'_E = \frac{\partial g}{\partial A} A' + \frac{\partial g}{\partial B} B' + \frac{\partial g}{\partial E} \left( \frac{\partial f}{\partial C} C' + \frac{\partial f}{\partial Q} Q' \right) + \frac{\partial g}{\partial L} L'. \tag{2.4.7}$$

In order for the DAE to be of index 1, we need

$$1 - \frac{\partial g}{\partial E}\frac{\partial f}{\partial p_E} - \frac{\partial g}{\partial p_E} \neq 0. \tag{2.4.8}$$

Next we determine the three partial derivatives in equation (2.4.8) by taking partial derivatives of equations (2.4.3) and (2.4.4).

$$\frac{\partial f}{\partial p_E} = \frac{\epsilon}{(1-\epsilon)p_E}E, \tag{2.4.9}$$

$$\frac{\partial g}{\partial E} = \frac{\alpha - 1}{E}p_E, \tag{2.4.10}$$

$$\frac{\partial g}{\partial p_E} = \left(\frac{\alpha}{\rho} - 1\right)\left(\frac{\rho}{1-\rho}\right)\frac{(1-\beta)\left(\frac{(1-\beta)p_E}{\beta B^\rho r_k}\right)^{\frac{\rho}{1-\rho}}}{\beta B^\rho + (1-\beta)\left(\frac{(1-\beta)p_E}{\beta B^\rho r_k}\right)^{\frac{\rho}{1-\rho}}}. \tag{2.4.11}$$

To determine the constraint, substitute equations (2.4.9)-(2.4.11) and simplify to yield

$$1 - (\alpha - 1)\frac{\epsilon}{(1-\epsilon)} - \left(\frac{\alpha}{\rho} - 1\right)\left(\frac{\rho}{1-\rho}\right)\frac{(1-\beta)\left(\frac{(1-\beta)p_E}{\beta B^\rho r_k}\right)^{\frac{\rho}{1-\rho}}}{\beta B^\rho + (1-\beta)\left(\frac{(1-\beta)p_E}{\beta B^\rho r_k}\right)^{\frac{\rho}{1-\rho}}} \neq 0. \tag{2.4.12}$$

The left side of equation 2.4.12 is always positive for the parameters in Table 2.2 and so the DAE system is of index 1. One case where the equation (2.4.12) could be zero is when $\alpha = \rho = \frac{1}{\epsilon}$. Since $\rho \leq 1$ and $0 \geq \epsilon \geq 1$, the only case that is valid is when all three parameters are 1. This is highly unlikely given that $\rho = 1$ implies that capital is a perfect substitute for oil.

Since we have already taken the derivative of the algebraic constraint and found that

the system can be made into an explicit ODE, we can find the solution using an ODE code like **ODE45**. The initial conditions must satisfy the original DAE system. The process of transforming the DAE into an explicit ODE is certainly tedious and certainly not desirable. In most cases and for large high-order DAE systems, it is not even feasible.

In Hanz's research [14], a **split-step** method is used. First the ODE production function is solved with a fourth-order Runge-Kutta method, assuming $p_E$ is constant. Then after solving for $Q$, $p_E$ is determined by solving the equation (2.2.27) with a non-linear root solver. For a small enough time step, the split-step method does yield an equivalent solution. However, this is not a solid numerical approach, since both ODE and the algebraic constraint are not solved simultaneously.

An enhancement to the split-step method is to satisfy the algebraic constraint (2.2.27) for each stage of the Runge-Kutta method, breaking the assumption that $p_E$ is constant while calculating $Q$. In general, explicit Runge-Kutta methods take on the following form [1]

$$y_{n+1} = y_n + \Delta t \sum_{i=1}^{s} b_i k_i, \tag{2.4.13}$$

$$k_i = f(t_n + c_i \Delta t, y_n + \Delta t \sum_{j=1}^{s} a_{i,j} k_j). \tag{2.4.14}$$

The split-step method keeps $p_E$ constant for all $k_i$ which are the outputs of each stage of the Runge-Kutta method which are values of $\dot{Q}$. However, this enhanced method solves the complete DAE (2.2.26) and (2.2.27) using the function $f$ in equation (2.4.14)

for each $k_i$. That is, each $k_i$ is a two element vector of $p_E$ and $\dot{Q}$. There is no general code for this method so it was implemented for a classical RK4 method with a fixed step. We call this method a **one-step** method because it has a form similar to the class of ODE methods of the same name.

As described in section 1.1.2, **DASSL** is a code that solves implicit DAEs using a BDF method. To maximize efficiency, DASSL uses polynomial extrapolation to get the initial guess for the next step. This reduces the number of iterations that need to be done in the non-linear solver. It also handles step-size reduction as well as setting the order of the BDF method appropriately.

To get an understanding of the intricacies of a DAE solver, we wrote a fourth-order fixed-step **BDF method**. The method followed the generic DAE BDF method from equation (1.1.10).

The challenge with an order $k$ BDF method is that we require $k$ initial values which are accurate to order $k$. We solve this problem by initializing the solution with an order 1 BDF with a reduced step-size and then grow the step-size and increase the order. The following relation must hold: $h^k \geq h_r^{k_r}$, where $h$ and $k$ are the desired step-size and order and $h_r$ and $k_r$ are the reduced step-size and order, respectively. For a step-size of $h = 0.1$ years and the desired order $k = 4$, the initial step-size was $h_r = 0.0001$ and $k_r = 1$ until there were enough points to move up to order 2, 3 and eventually 4. Overall this code did not implement any of the complicated options provided in DASSL, such as variable step-size adjustment or local error control, just to name a few.

Having an analytical solution for $\rho = 0$ provides a benchmark to compare all the

solutions from the different numerical methods. The analytical solution provides a metric to check how large the global error is for each method. Figure 2.5 shows the relative error from the analytical solution to the DASSL solution, the ODE45 solution, the fixed-step BDF solution and the fixed-step one-step method. Both fixed-step methods used a time step of $\Delta t = 0.1$ years, or equivalently 900 evaluations. Both adaptive-step methods used a local relative error tolerance of $1 \times 10^{-6}$. This resulted in ODE45 using 105 evaluations while DASSL used only 91 evaluations. When comparing relative error, all methods did equally well as shown in Figure 2.5. Overall DASSL is superior to the fixed-step methods when considering accuracy per computation. DASSL and ODE45 performed similarly, however DASSL has the advantage that it can solve the DAE directly without transforming the system of equations into an ODE.

## 2.5 Results: Arbitrary $\rho$

Using DASSL, the solution curves for the rate of oil production, $\frac{dQ}{dt}$, and the price of energy, $p_E$, can be determined for various values of $\rho$. Figure 2.6 shows the rate of oil production which has the right order of magnitude (oil production in 2007 was approximately 85 mbpd, or 31 giga barrels per year or $3.1 \times 10^4$ mega barrels). As $\rho$ gets smaller, that is smaller elasticity of substitution, the production peak is higher and the decline afterwards is steeper. For larger $\rho$, the peak is pushed out later in time. None of the curves exhibit the symmetry seen in the Hubbert curve.

Figure 2.7 shows the corresponding price of energy which is too high by a factor of 10-50. The model assumes that energy is supplied entirely by oil, compared to 38

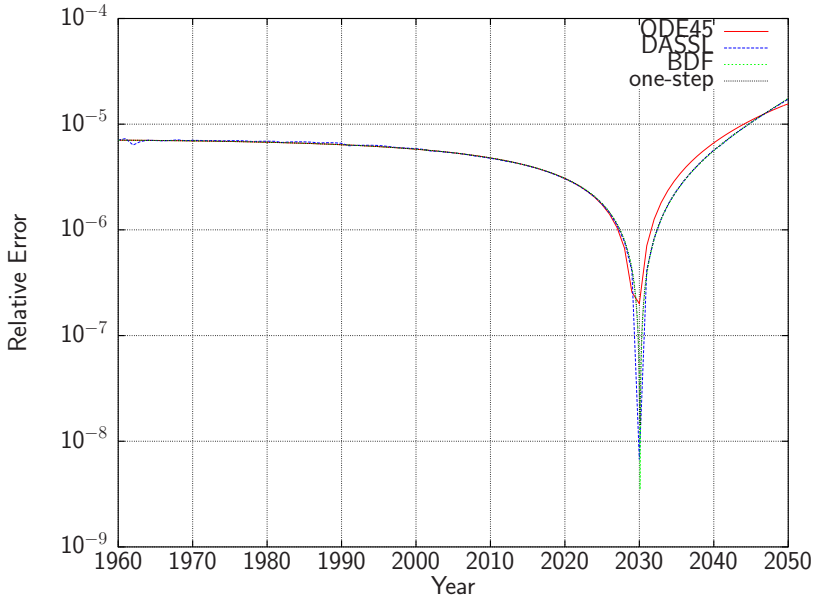



Figure 2.5: Relative error from the analytical model of oil production for $\rho = 0$ for four numerical solutions.

percent in reality according to British Petroleum [7]. When selecting parameters, the German chemical industry was used as a proxy for the average world economy. The result of this assumption is higher energy prices since the demand in our model is much higher than in reality. Note that all curves in Figure 2.7 are monotonically increasing. A more advanced model is required to model price decrease when another form of energy replaces oil.

By increasing the value of the parameter $C$ compared to the previous set of simulations, the production peaks occur earlier and their magnitude approaches current production levels. Results are shown in Figure 2.8. An interesting property of this set of parameters is that all the production curves intersect at year 2020. There does not seem to be an apparent or obvious explanation for this effect.

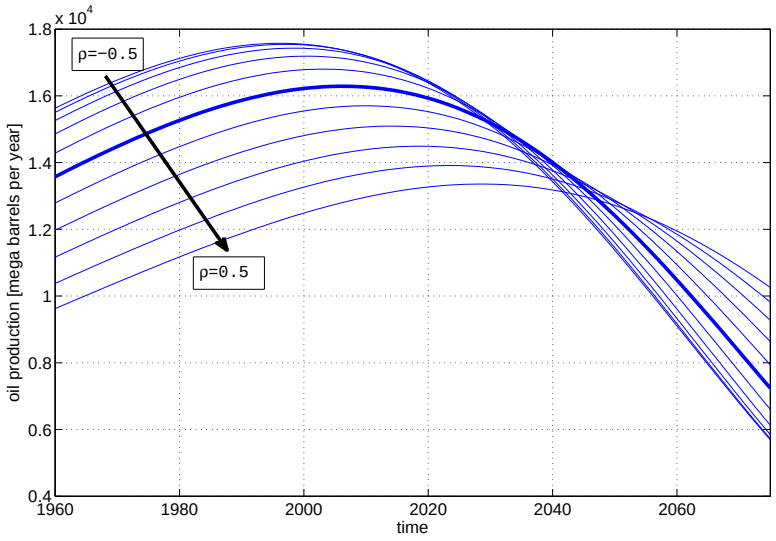



Figure 2.6: Oil production curves for various values of $\rho$, ranging from $-0.5$ to 0.5 in increments of 0.1, and $C = 1.5 \times 10^{-3}$. The thick line represents $\rho = 0$.

Figure 2.9 shows the corresponding price of energy curves. Compared to Figure 2.7, the price of energy decreases with an increase in $C$.

For some conclusions regarding this model, we refer to chapter 4.

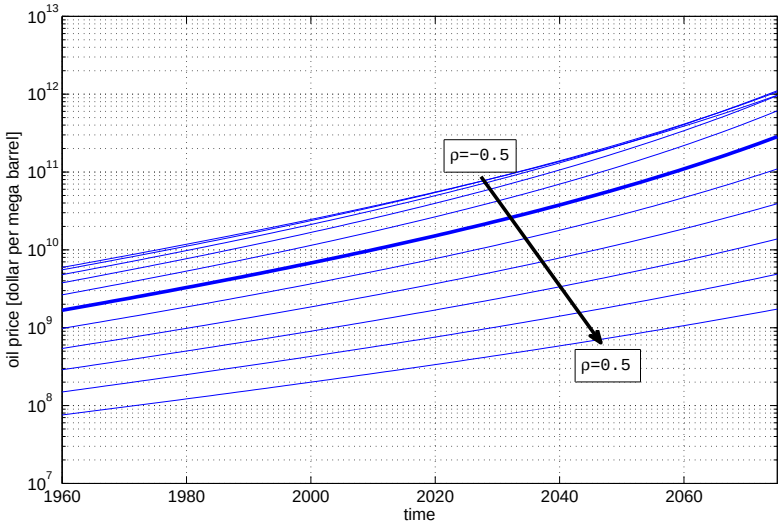



Figure 2.7: The price of oil corresponding to the production curves in Figure 2.6. The thick line represents $\rho = 0$.

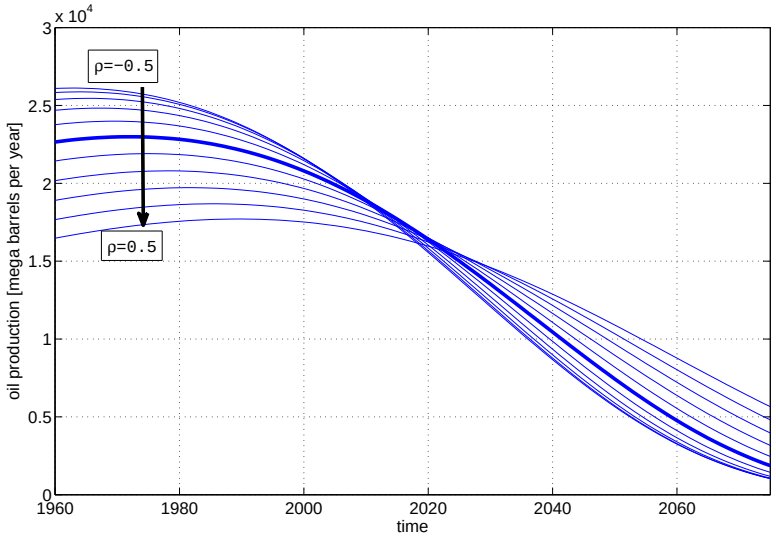



Figure 2.8: Oil production curves for various values of $\rho$, ranging from $-0.5$ to 0.5 in increments of 0.1, and $C = 2.5 \times 10^{-3}$. The thick line represents $\rho = 0$.

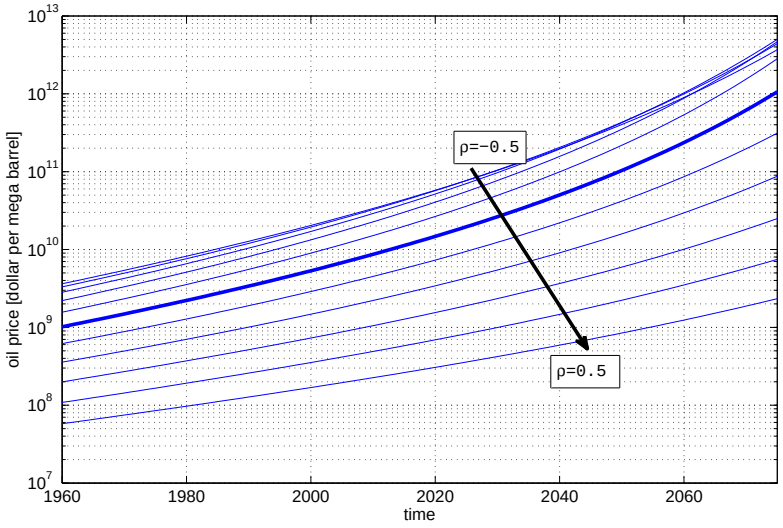



Figure 2.9: The price of oil corresponding to the production curves in Figure 2.8. The thick line represents $\rho = 0$.

# Chapter 3

# Traffic Flow Modeling

In this chapter, microscopic and macroscopic models for traffic flow are presented. Numerical methods for transient and steady-state analysis of the macroscopic model are provided with solutions for traffic flow on a loop with a bottleneck. In particular, we aim to produce results of the microscopic model by use of an analogous macroscopic model. The chapter concludes with a discussion of solutions for cases when the system is linearly unstable.

## 3.1 Microscopic Model

A certain type of microscopic model for vehicle traffic on a road is introduced by Bando *et al.* [4]. Each vehicle is modeled as a point, that is, having no length, and traveling along a one-dimensional road. Cars follow the car in front and passing is not allowed. The equations that govern the movements of vehicle $n$, are

$$\dot{x}_n = v_n, \tag{3.1.1}$$

$$\dot{v}_n = \lambda[V_{op}(h_n) - v_n]. \tag{3.1.2}$$

Here, $h_n$ is called the headway, which is the length of open road ahead of car $n$. For all cars but the lead car, we have

$$h_n = x_{n+1} - x_n, \tag{3.1.3}$$

which provides a uni-directional coupling between vehicles in equation (3.1.2). Moreover, $V_{op}$ is called the non-dimensional optimal velocity (OV) function and is introduced by Bando *et al.* as

$$V_{op}(h) = \tanh(h-2) + \tanh(2). \tag{3.1.4}$$

It represents the velocity that a driver would like to follow a vehicle with, given a headway $h$. Figure 3.1 shows the optimal velocity function. It is always positive, so cars always try to move forward. The fastest velocity a car can assume is $1 + \tanh(2)$.

The sensitivity parameter $\lambda$ determines how fast a car responds or how much it is coupled to the vehicle it is following. Bando *et al.* provide the linear stability criteria for $\lambda$ for uniform car spacing $h$ as

$$V'_{op}(h) < \frac{\lambda}{2}, \tag{3.1.5}$$

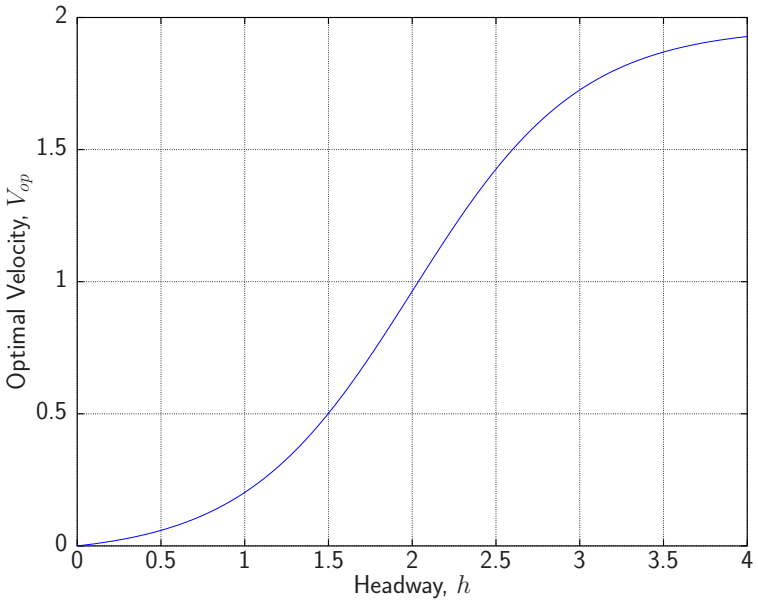



Figure 3.1: Optimal velocity function.

by performing Fourier mode analysis for the linearized system about the steady-state flow on a loop without a bottleneck. By taking the derivative of the optimal velocity function and finding the maximum, a requirement for unconditional linear stability independent of $h$ can be determined to be

$$\lambda > 2. \tag{3.1.6}$$

In general, however, the critical value, $\lambda_{crit}$, for the given optimal velocity function (3.1.4) and uniform spacing $h$ is

$$\lambda_{crit} = 2\,\mathrm{sech}^2(h-2). \tag{3.1.7}$$

All values of $\lambda$ above the curve shown in Figure 3.2 equate to a linearly stable model while values of $\lambda$ below the line correspond to instability. In the latter case, small perturbations to the initial uniform flow result in the formation of stop-and-go patterns. Indeed if equation (3.1.6) holds, then the flow is linearly stable for all $h$.

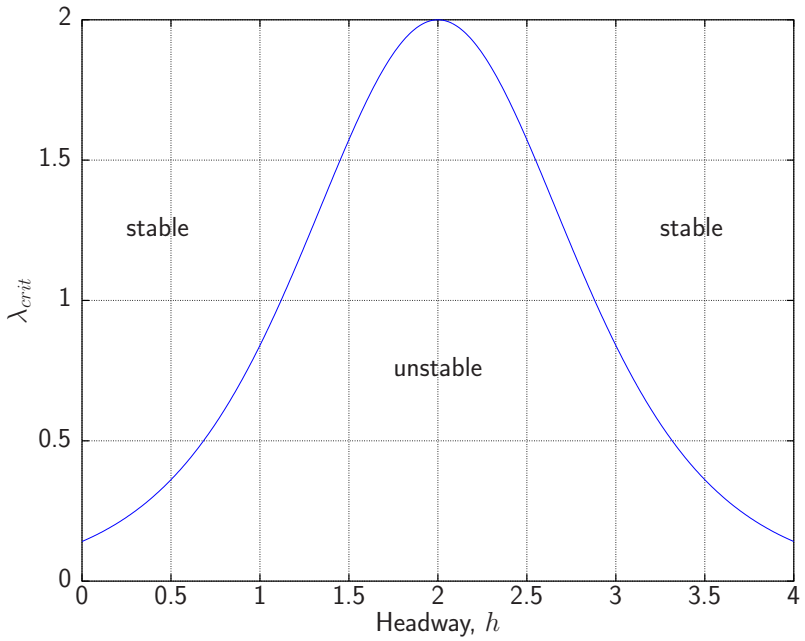



Figure 3.2: Critical values of $\lambda$ for linear stability.

### 3.1.1 On a Loop

Ward *et al.* [26] looked at the one-dimensional car-following model on a loop or ring road including a bottleneck. $N$ cars travel around the loop of length $L$. When cars reach the position $x_n = L$, they start back again at the beginning of the road at position $x_n = 0$. The headway $h_n$ is the same as equation (3.1.3) except that the lead car, $N$, has a headway with respect to car 1 given by

$$h_N = L + x_1 - x_N. \tag{3.1.8}$$

A bottle neck is added to a segment of the road, where all vehicles must reduce their speed. The bottle neck is of length $\hat{L}L$ where $0 < \hat{L} < 1$. The reduced speed is achieved with a reduction factor $r_B$, where $0 < r_B < 1$. The optimal velocity function from equation (3.1.4) is modified to also be a function of position as follows

$$V_{op}(h, x) = \begin{cases} r_B(\tanh(h-2) + \tanh(2)), & 0 \leq x < \hat{L}L, \\ (\tanh(h-2) + \tanh(2)), & \hat{L}L \leq x < L. \end{cases} \tag{3.1.9}$$

Ward *et al.* showed the dependence of the emerging steady-state traffic patterns on three key parameters, the mean headway, $h_* = L/N$, the length of the bottle neck, $\hat{L}$, and the reduction factor $r_B$. Results are presented for $r_B = 0.6$, $N = 100$, $\hat{L} = 0.25$, $\lambda = 3$ and $h_* = 7$, 2.5 and 1, corresponding to light, medium and heavy traffic, respectively.

In steady-state, found as $t \to \infty$ the traffic pattern does no longer change its shape. The flux of cars, $Q$, is constant and given by the following equation

$$Q = \rho v, \tag{3.1.10}$$

where

$$\rho = \frac{1}{h}. \tag{3.1.11}$$

At steady-state, the vehicles' velocities approach the optimal velocity, $V_{op}$, except near shock fronts. Substituting $V_{op}\left(\frac{1}{\rho}\right)$ for $v$, we generate the flux of vehicles in the bottle neck and for the open road.

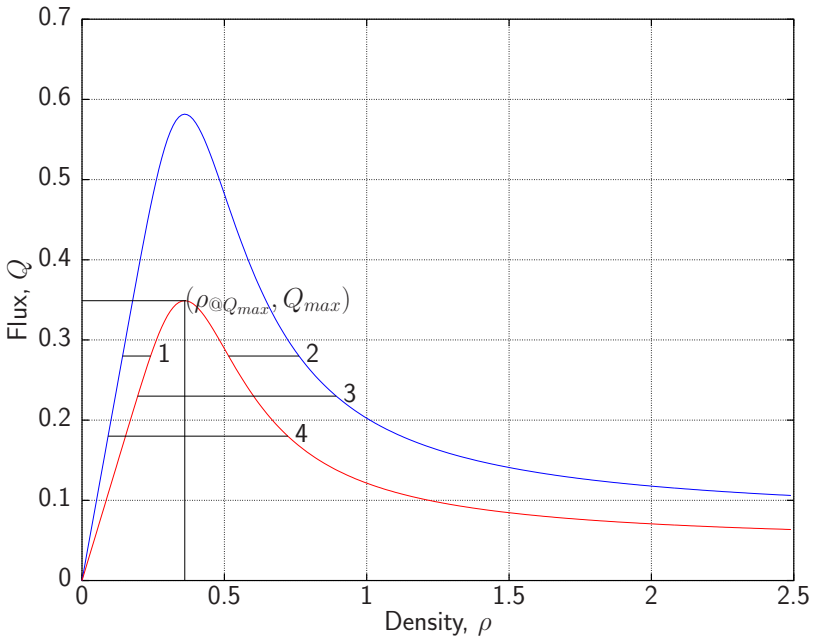



Figure 3.3: Four constant-flux scenarios for traffic flow on a loop with a bottle neck: Case 1: light traffic, Case 2: heavy traffic, Case 3: light traffic in bottleneck, heavy traffic on the open road, Case 4: heavy traffic in bottleneck, light traffic on the open road.

Figure 3.3 presents four possible scenarios for steady-state flux. Ward *et al.* showed that cases 1 and 2 represent light and heavy traffic, while cases 3 and 4 can be disregarded since each has a non-stationary expansion fan, i.e. a dispersive shock

front that does not allow for a stationary flow pattern. Cases 1 and 2 are two-plateau solutions. For example, light density traffic can be described by line 1. The open road traffic density plateau has the value on the left side of line 1, where it intersects with the upper flux curve for the open road. Similarly the bottle neck traffic density is given by the point on the right side of line 1 which intersects with the lower flux curve representing the flux in the bottle neck. $Q_{max}$ represents the highest allowable flux in the system at steady-state. This is the maximum flux allowable based on the reduction factor $r_B$ in the bottle neck. $\rho_{@Q_{max}}$ is the corresponding density and is approximately 0.36. It is independent of $r_B$. The upper limit of light traffic and the lower limit of heavy traffic are determined by a bottle neck density $\rho_{@Q_{max}}$ of maximum flux. Medium traffic is in between light and heavy. It has a constant flux of $Q_{max}$ inside the bottleneck and a density of $\rho_{@Q_{max}}$. Medium traffic has three plateau values which remain constant. However the widths of the plateau values depends on the average density, $\rho_*$.

## 3.2 Macroscopic Model

We will now consider a continuum description of cars, based on PDEs for velocity and density, the latter of which replaces the headway $h$ in discrete models. Berg *et al.* [5] and Lee *et al.* [20] both developed macroscopic models based on Bando *et al.*'s microscopic model by coarse graining the microscopic variables to obtain a macroscopic description of traffic flow. The macroscopic models preserve the stability criteria as presented in equation (3.1.5).

The models were developed by means of an approximation of the headway in terms

of the density. When doing so, it is unreasonable to assume $h_n(x,t) = \rho^{-1}(x,t)$. Lee *et al.* showed with a counterexample that this relation does not take directionality into account. Instead the density should be evaluated at the midpoint between two vehicles

$$h_n(x,t) = \rho^{-1}\left(x + \frac{1}{2\rho(x,t)}, t\right). \tag{3.2.1}$$

The corresponding macroscopic traffic flow model is a system of partial differential equations (PDEs), given by a conservation equation

$$\rho_t + (\rho v)_x = 0 \tag{3.2.2}$$

and a momentum equation

$$v_t + v v_x = \lambda[V_{op}(\rho^{-1}) - v] - \frac{\lambda V'_{op}}{2\rho^3}\rho_x + \frac{\lambda}{6\rho^2} v_{xx}. \tag{3.2.3}$$

Numerical comparisons show a good match between the velocity profiles of the microscopic model and the macroscopic model for certain flow regimes. We aim to use this model for the study of the bottleneck.

## 3.3 Numerical Techniques

In order to solve the macroscopic equations, we need to discretize the PDE system on a grid. Assuming non-homogeneous grid spacing, a spatial function $y$ can be

discretized regarding its derivative $y_x$ for grid point $i$ as

$$y_x \approx \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \tag{3.3.1}$$

and the discretization of $y_{xx}$ for grid point $i$ is

$$y_{xx} \approx \frac{\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}}{\frac{x_{i+1} - x_{i-1}}{2}}. \tag{3.3.2}$$

Both expressions are $O((\Delta x)^2)$. Since we are interested in simulating on a loop, for the last grid point $N$, $i + 1$ will coincide with the first grid point. Similarly, for the first grid point, $i - 1$ will coincide with the last grid point $N$. This can be achieved in the code by either direct implementation with condition statements based on the grid point index, or the vector can be increased by two and ghost cells can be added. For a vector index from 1 to $N$, ghost cells would be the two additional cells indexed by 0 and $N + 1$. Whenever index 1 is updated so would the ghost cell with index $N + 1$ and similarly $N$ and 0.

### 3.3.1 Transient Analysis

We will now summarize several numerical techniques that were employed in the transient case. The simplest numerical method for solving initial value problems is the explicit **Euler** method. The system can advance by a simulation step-size of $\Delta t$ with only the previous state of the system and a single evaluation of the derivative as

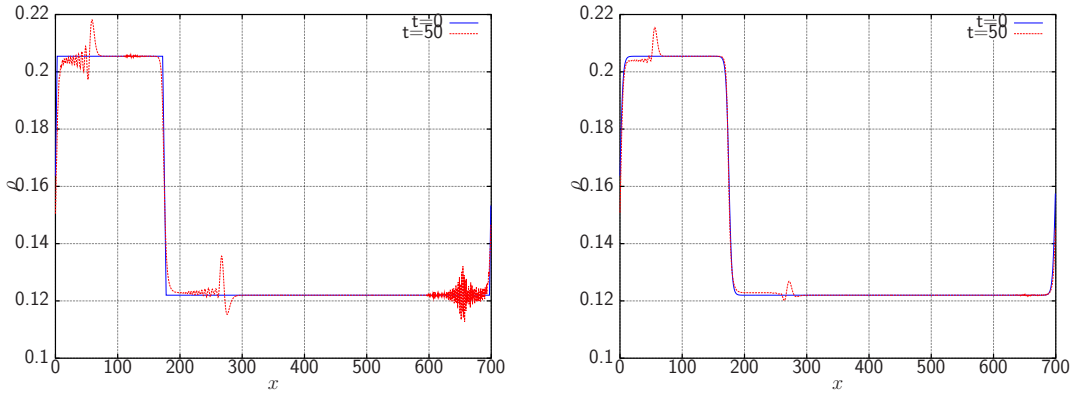
follows

$$y_{n+1} = y_n + \Delta t\, f(t_n, y_n). \tag{3.3.3}$$

For our initial conditions we can use the plateau values of $\rho$ based on Figure 3.3. To transition between the plateau values, we can use a piecewise linear function or a smooth function like $\tanh(x)$. Figure 3.4 shows the difference between piecewise linear initial conditions and smooth initial conditions after simulating up to $t = 50$. Figure 3.4a shows some high frequency noise which is introduced because of the discretization of the grid and the non-smooth initial conditions. It is important that initial conditions are smooth as in Figure 3.4b so that there are no discontinuities in the finite difference schemes regarding $y_x$ and $y_{xx}$. Piecewise linear initial conditions are undesirable since the discretized first derivative, $y_x$, is discontinuous so the second derivative, $y_{xx}$, grows as the mesh becomes finer with the spatial mesh distance becoming smaller.

With a higher-order method, a large step-size can be taken and thus a shorter simulation time to compute the simulation can be achieved. Another simple method to implement is the classical fourth-order **Runge Kutta** method (RK4), described in 2.4.

Figure 3.5 compares the results for the Euler and RK4 fixed step methods. Figure 3.5a shows the Euler method for a fixed step-size of $\Delta = 0.001$. This is the largest time-step for which the method is stable. Figure 3.5b shows the obvious instability of the Euler method for a larger step size of $\Delta = 0.005$. The RK4 method is stable



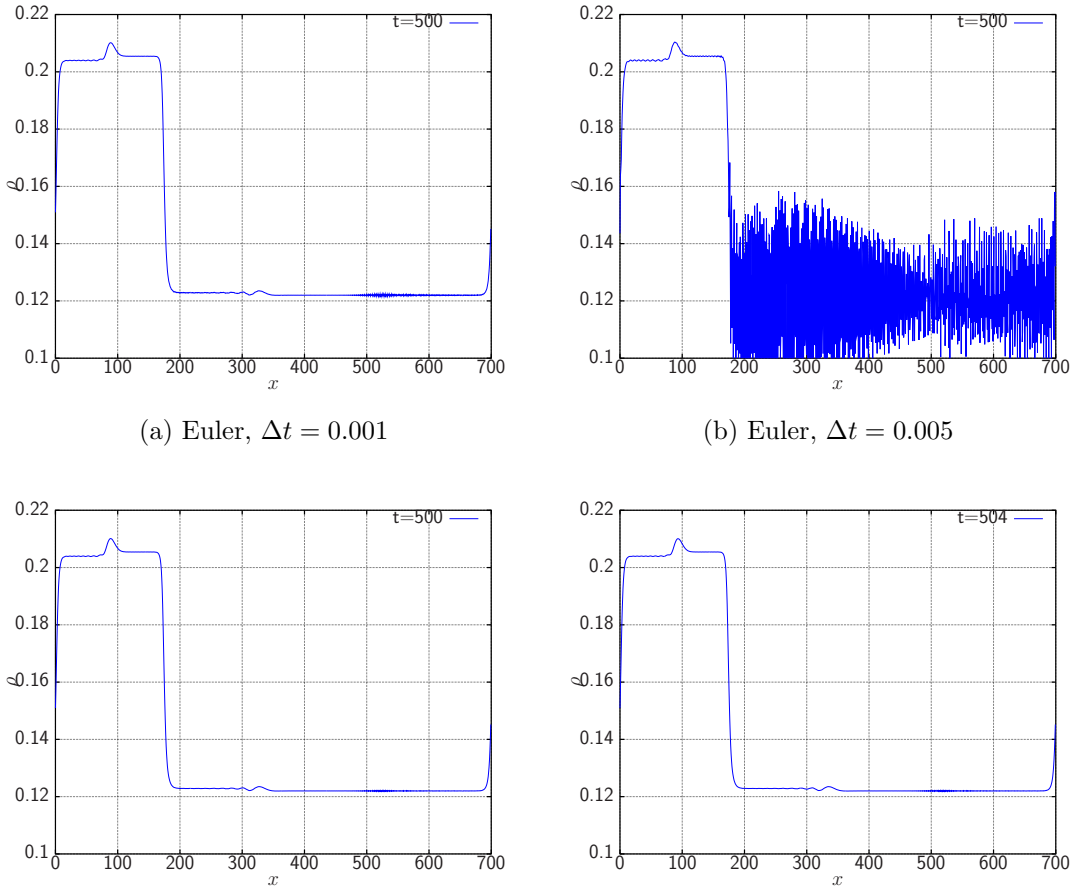

(a) Piecewise linear

(b) Smooth

Figure 3.4: Simulations of light traffic for piecewise linear and smooth initial conditions (N=1000).

up to $\Delta = 0.009$, determined by empirical observation, as shown by Figures 3.5c and 3.5d. The RK4 method requires 3 function evaluations per step, where the Euler method needs only one. However, since the RK4 method can take a step which is 9 times larger, it is still more efficient. This comes as no surprise that the RK4 method outperforms the most simple method, the Euler method.

Both the explicit Euler method and the RK4 method are fixed step methods, meaning that a fixed step size is selected at the beginning of the simulation, based on the system, such that it is stable. In contrast, **embedded methods** provide local error control and are able to adjust the step size based on a provided error tolerance. An estimate for the local error can be determined by taking the difference between two methods, one of order $p$ and the other of order $p-1$. Embedded methods make use of the same function evaluations for both methods, the order $p-1$ method is embedded in the order $p$ method.




(a) Euler, $\Delta t = 0.001$

(b) Euler, $\Delta t = 0.005$

(c) RK4, $\Delta t = 0.005$

(d) RK4, $\Delta t = 0.009$

Figure 3.5: Simulations of light traffic using fixed step methods.

The **Fehlberg** method is an order 4-5 method. It has six stages, however, the first stage is the same as the previous time step, so it only requires five function evaluations. The coefficients have been optimized to reduce the error in the fourth-order method. ODE45, often used in MatLab, uses the Dormand-Prince method. It is an order 4-5 method, however, it has seven stages but requires only six function evaluations and is optimized to reduce the error of the fifth-order method.

The GNU Scientific Library (GSL) [13] is a numerical library with many routines and a large reference manual intended for C and C++ programmers. **RKF45** is a routine which implements Felhberg method.

Figure 3.6 presents similar results to Ward *et al.*'s findings for $\lambda = 3$ for the RKF45 method for a homogeneous spatial mesh of 1000 points. Medium and heavy traffic density settle to steady-state quickly. Light traffic in Figure 3.6a still has not converged to a steady-state. There is still a small traveling wave at time $t = 100,000$ at the position $x = 500$.

One way to reduce the run time of the transient simulations is to coarsen the spatial mesh by increasing the step-size. Given the steep transitions between the plateaus, reducing the number of mesh points is not a reliable option. A non-homogeneous mesh with a higher concentration of grid points in the vicinity of the transitions seems like a good idea. However, there are traveling waves, which can be seen in Figure 3.4b which would be hard to capture in the less dense areas of the mesh. Another option would be to allow the grid to move to capture the traveling wave. Fewer grid points could be used but more equations would be needed.

Petzold [11] generalized the **moving mesh** method for the following general system

$$F(u_t, u, u_x, u_{xx}) = 0. \tag{3.3.4}$$

The system of equations (3.2.2) and (3.2.3) can easily be manipulated to match this general form. It is important to note that since $x$ is now moving, $x'$, the mesh velocity, is no longer zero. Accordingly, we find

$$\frac{du(x,t)}{dt} = \frac{\partial u(x,t)}{\partial x}\frac{dx}{dt} + \frac{\partial u(x,t)}{\partial t}. \tag{3.3.5}$$

Rearranging for $u_t$ and substituting into the general system (3.3.4) leads to

$$F(u' - u_x x', u, u_x, u_{xx}) = 0. \tag{3.3.6}$$

Petzold also suggested an equation to move the mesh points as follows

$$\alpha x' + u' u_x = 0, \tag{3.3.7}$$

for an appropriate value of $\alpha$. Allowing the mesh points to move requires an additional $N$ equations, bringing the total to $3N$: one for the density, velocity and position of each grid point. For a static mesh of size $N = 1000$, the equivalent moving mesh is chosen to be $N = 667$ to offset the extra function evaluations. In total, both systems have 2000 variables. Figure 3.7 shows the result for the moving mesh system ($\alpha = 10$). The majority of mesh points do not move and stay in their original position. The points associated with the transition regions move along with the density function $\rho$. Unfortunately, the final solution has some noise and is not as smooth as desired. Figure 3.8 shows the result for a homogeneous mesh of size $N = 667$ which is much smoother. In this sense, this moving mesh approach must be considered unsuccessful.

### 3.3.2 Steady-State Analysis

Since the transient simulations take a very long time to converge to a steady-state, another method for quick steady-state analysis is desirable. The steady-state is obtained in the limit as $\rho_t \to 0$ and $v_t \to 0$. By rearranging the equations (3.2.2) and (3.2.3), we find

$$\rho_t = -(\rho v)_x, \tag{3.3.8}$$

$$v_t = -vv_x + \lambda[V_{op}(\rho^{-1}) - v] - \frac{\lambda V'_{op}}{2\rho^3}\rho_x + \frac{\lambda}{6\rho^2}v_{xx}. \tag{3.3.9}$$

Newton's method, as shown below, can be used to solve for the steady-state solution. We write formally for the Newton scheme to solve $F(y) = 0$

$$\frac{\partial F}{\partial y}(y)\delta = -F(y), \tag{3.3.10}$$

$$y^{v+1} = y^v + \delta. \tag{3.3.11}$$

Using the equations (3.3.8) and (3.3.9) directly does not achieve the correct result. Figure 3.9 shows the number of cars is no longer being conserved. The key is to treat equation (3.3.8) simply as a constraint that can be converted to

$$Q = \rho v, \tag{3.3.12}$$

where $Q$ is a constant. There are an infinite number of solutions to our steady-state problem. To select a unique solution, we seek a parametrized equation. In fact, there are two reasonable choices for the parameter: $Q$, the flux, or $\rho_*$, the average density. Since we are trying to conserve density and $\rho_*$ can easily be determined from the initial condition, the following equation is added to the system

$$\rho_* = \frac{1}{L} \int \rho \, dx. \tag{3.3.13}$$

Moreover, $\rho_*$ defines the system uniquely where $Q$ leaves at least two possible solutions (see Figure 3.3).

The problem now is that our system seems overdetermined. The system is still consistent as the additional constraint (3.3.13) has only been introduced to limit the number of solutions. For a mesh of size $N$, there are $2N$ variables to be solved for $2N + 1$ equations. In Newton's method, we are trying to solve a system of the form $Ax = b$. DGELS is a routine provided in the Linear Algebra PACKage (LAPACK) [19] which finds the linear least squares solution for problems of that form for overdetermined systems using QR factorization. Remember that without the constraint (3.3.13), we have infinitely many solutions.

QR factorization can be performed as follows: $A = QR$ is an $m \times n$ matrix where $m \geq n$, $Q$ is an orthogonal $m \times m$ matrix and $R$ is an upper triangular $m \times n$ matrix where the last $m - n$ rows are zero

$$\begin{aligned} Ax &= b, \\ QRx &= b, \\ Q^TQRx &= Q^Tb, \\ Rx &= Q^Tb. \end{aligned}$$

Since $R$ is an upper triangular matrix, elements of $x$ can easily be determined in reverse order, using only the previously determined elements.

Another modification that can be made to Newton's method is to include dynamic step scaling. Equation (3.3.11) takes a full step of $\delta$. With a small modification, the step size is modified as follows

$$y^{v+1} = y^v + \alpha\delta. \tag{3.3.14}$$

$\alpha$ is chosen such that no element of the solution vector $y$ changes by more than 5% every iteration

$$0.05 \geq \alpha\frac{\delta_i}{y_i} \quad i = 1 \ldots 2N \tag{3.3.15}$$

Figure 3.10 shows how the scaling factor $\alpha$ changes over the first six iterations of

the heavy traffic simulation. As the solution converges, the steps are smaller and the scaling is increased to 1, so that full steps are taken.

Iterative methods require a convergence check. Rather than simply checking the absolute error, instead the maximum absolute error for the last three iterations are required to be within 0.5%. Figure 3.11 shows the convergence of the algorithm for light traffic.

Despite all the improvements, Figure 3.12 shows that there are some high frequency oscillations. Note that the largest absolute error is $3 \times 10^{-13}$ so the solution has converged but yet this solution with the high frequency oscillations is not expected. We expect to see a smooth solution for the density of vehicles. Upon further investigation, Figure 3.12b shows the oscillations at the grid points. These oscillations are introduced by the discretization of $y_x$ from equation (3.3.1). By blending the centered finite difference and the forward finite difference, the oscillations due to discretization can be removed

$$y_x \approx \beta \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} + (1 - \beta) \frac{y_{i+1} - y_i}{x_{i+1} - x_i}. \tag{3.3.16}$$

This is almost but not exactly an upwinding technique [21]. A true upwinding scheme takes into account the direction of the movement and applies either the forward or backward biased finite difference. Figure 3.13 shows the solutions for $\beta = 0.001$ for light, medium and heavy traffic flows.

The solutions for steady-state match well with the final time step profiles for the transient analysis for medium and heavy traffic densities. For light densities, the cars are weakly coupled so it takes a long transient simulation for a stationary profile to

appear. Figure 3.6a does not match exactly Figure 3.13a since even at $t = 10000$ the transient traveling wave still has not completely dampened out.

A transient simulation does provide more information than just the final steady-state profile. However, if all that is desired is the final stationary profile then the steady-state simulation is far more efficient. Table 3.1 shows the number of function evaluations required to compute the steady-state profile when performing transient analysis and steady-state analysis. On average, the transient simulation requires 100 times more function evaluations. The steady-state simulation method could be further reduced with the implementation of a real Jacobian function. Currently the majority of function evaluations are performed to determine the Jacobian by numerical differences.

Runtime of the simulations takes on the order of minutes to hours. In order to achieve the same result in a shorter amount of time, parallel computations are desirable. The system of $2N+1$ functions can be easily broken up and evaluated in parallel. All evaluations require only data from the previous time-step so there are no dependencies. The function can be considered perfectly parallel. OpenMP [23] provides a simple API to automatically transform a serial loop into a block of parallel code with one simple line.

```
#pragma omp parallel for shared(<variables>) private(<variables>)
```

The `#pragma` statement tells the compiler that the next `for` loop should be made parallel and lists the shared and private variables.

| | Transient | Steady-state |
|---|---|---|
| Light | 6,087,383 | 23,995 |
| Medium | 1,118,626 | 63,985 |
| Heavy | 4,380,370 | 43,990 |

Table 3.1: Number of system function evaluations required.

## 3.4 The Linearly Unstable Regime

In the previous sections, $\lambda = 3$ so the model always remained linearly stable. By using equation (3.1.7) the critical value for $\lambda$ can be determined for each plateau density.

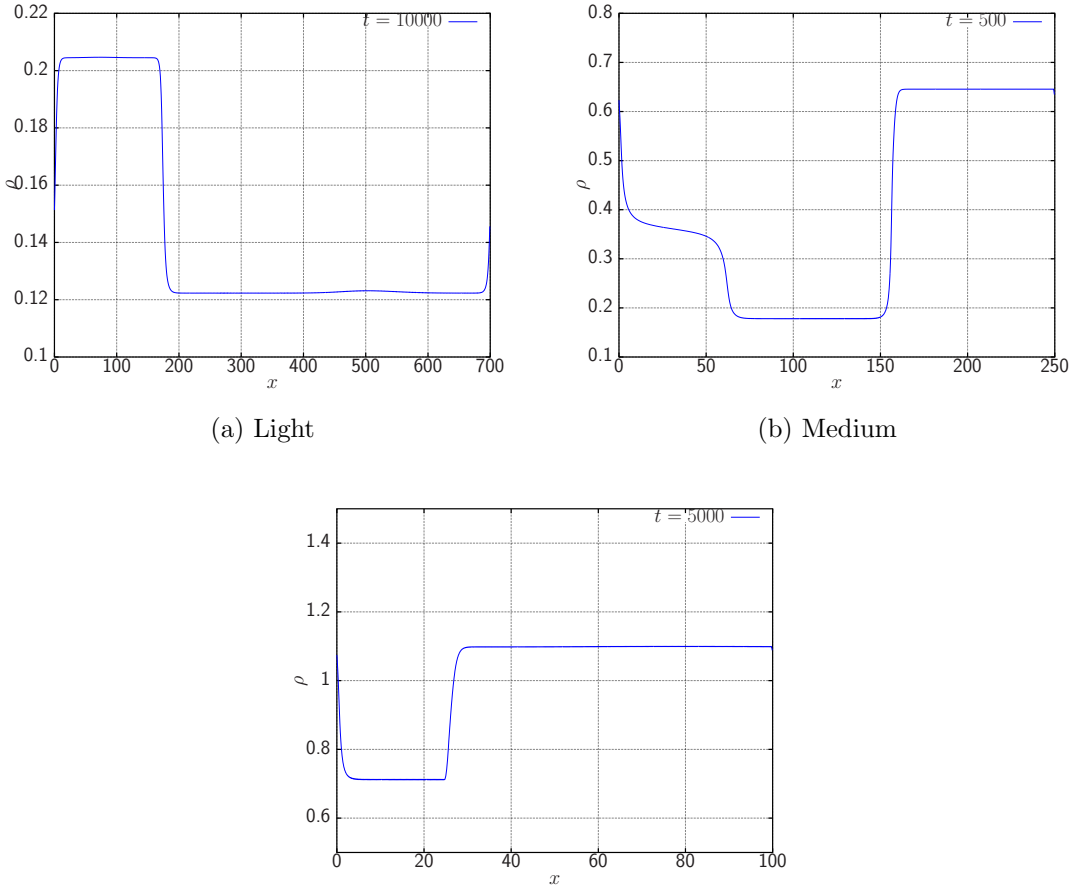
For heavy traffic flow, the $\lambda$ values associated with the plateau densities are $\lambda = 0.86$ for the bottle neck and $\lambda = 0.733$ for the open road. Figure 3.14 shows the steady-state solutions are very close for all three values of $\lambda$. Transient simulation converges for $\lambda = 0.86$ but fails for $\lambda = 0.733$. Figure 3.15 shows that for $\lambda = 0.733$ the transient simulation is close to the steady-state solve at $t = 4000$ but by $t = 8000$ the simulation has started to diverge. It is interesting how similar the steady-state simulations are for the various $\lambda$ values in Figure 3.14 while the transient simulations are unstable and, hence, very sensitive to a change in $\lambda$. Hence, we have developed a numerical tool, i.e. the steady-state solve, that allows us to determine steady-state solutions that are unstable and, hence, never observed in transient simulations.

For medium traffic flow, the values of $\lambda$ for each of the plateaus are $\lambda = 0.698$ for the bottle neck, $\lambda = 0.006$ for the first section of the open road and $\lambda = 1.64$ for the denser final section of the open road which is the queue of traffic waiting to get into the bottle neck. Figure 3.16 shows the flow for $\lambda = 3, 1.64$ and $0.698$ using the steady-state solver. $\lambda = 0.006$ is far too small. It is important to note that as

$\lambda \to 0$, the vehicles are no longer coupled and do not adjust their speed based on the vehicle they are following. There are infinitely many solutions for $\lambda = 0$ determined by $\rho_x = 0$ and $v_x = 0$.

The transient simulation is successful in determining the steady-state solution for $\lambda = 1.64$. Figure 3.17 shows that for $\lambda = 0.698$, the density becomes unstable at the beginning of the bottle neck, which is the transition between the two linearly unstable plateaus.

Light traffic has very low density and small $\lambda$ values associated with the plateaus. In the bottle neck, $\lambda = 0.015$ and for the open road $\lambda = 3.5 \times 10^{-5}$. Figure 3.18 shows that $\lambda = 0.015$ is so small that the plateau value is never reached. $\lambda = 0.05$ is approximately the smallest $\lambda$ to reach the plateau value in the bottle neck. In terms of transient simulation, $\lambda = 0.015$ is too small.




(a) Light

(b) Medium

(c) Heavy

Figure 3.6: Transient traffic simulation for light, medium and heavy traffic densities ($\lambda = 3$).

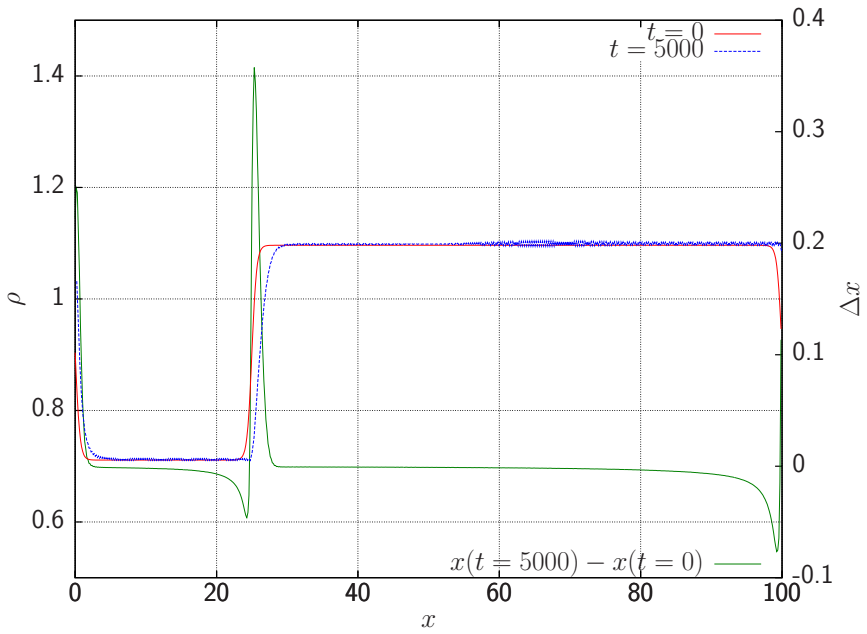



Figure 3.7: Heavy traffic simulation for a moving spatial mesh of $N = 667$ points.

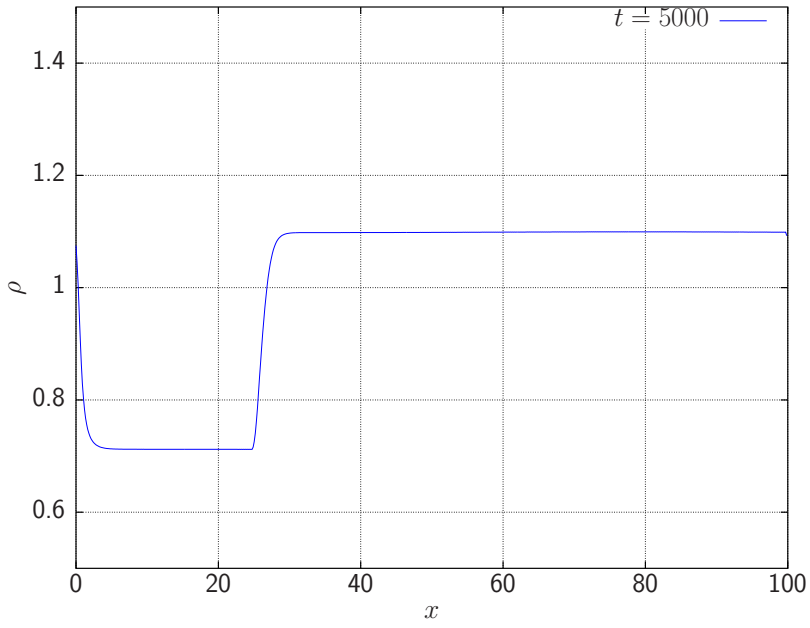



Figure 3.8: Heavy traffic simulation for a homogeneous spatial mesh of $N = 667$ points.

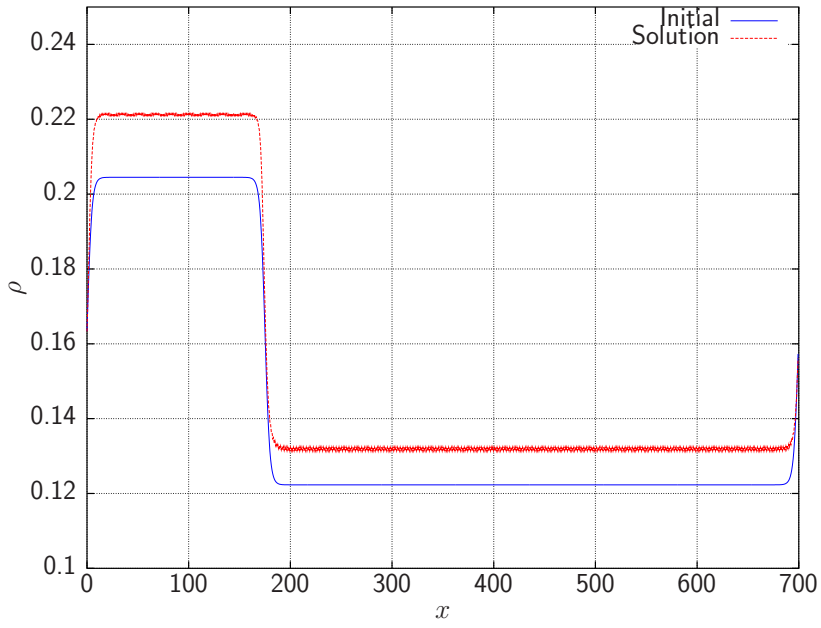



Figure 3.9: Steady-state simulation of naive model shows density is no longer conserved for light traffic.

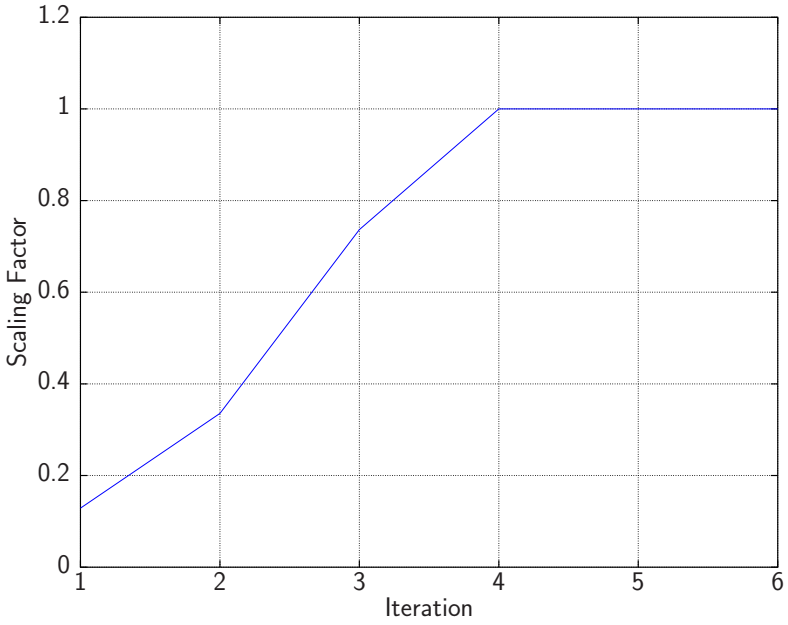



Figure 3.10: Dynamic step size scaling for heavy traffic.

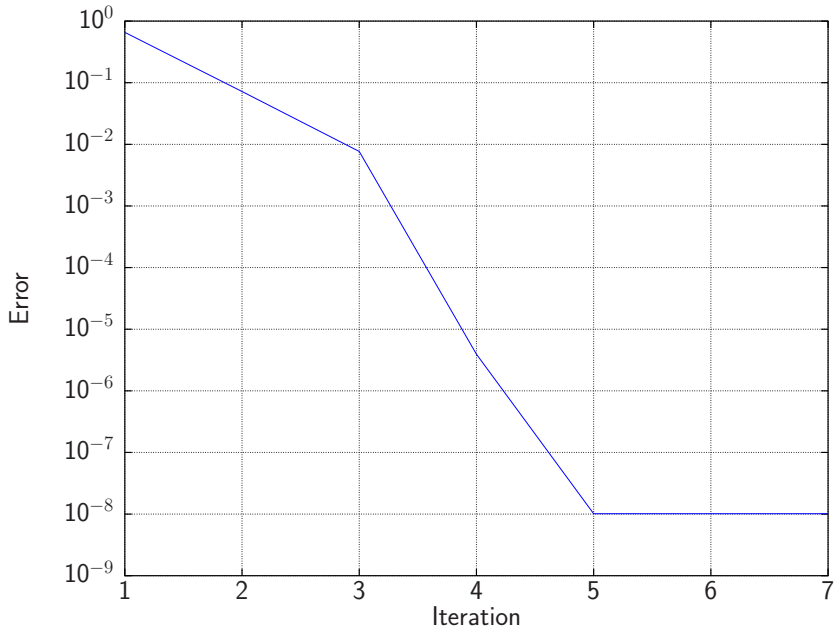
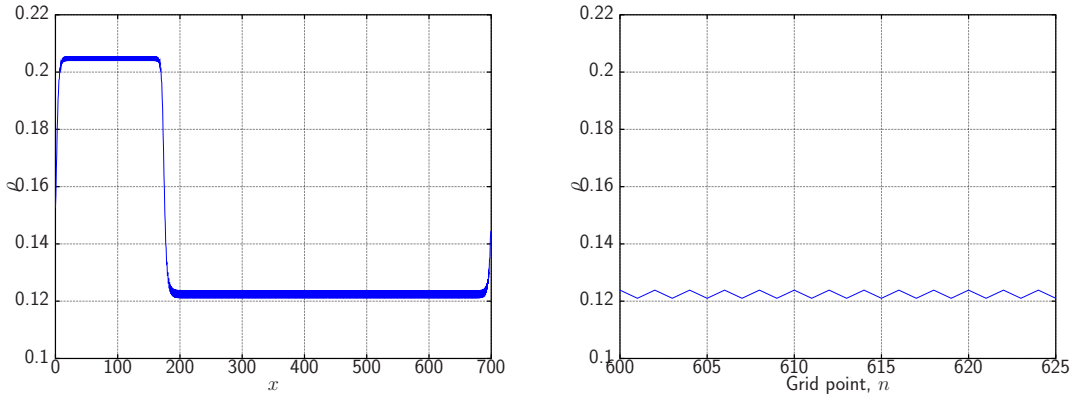


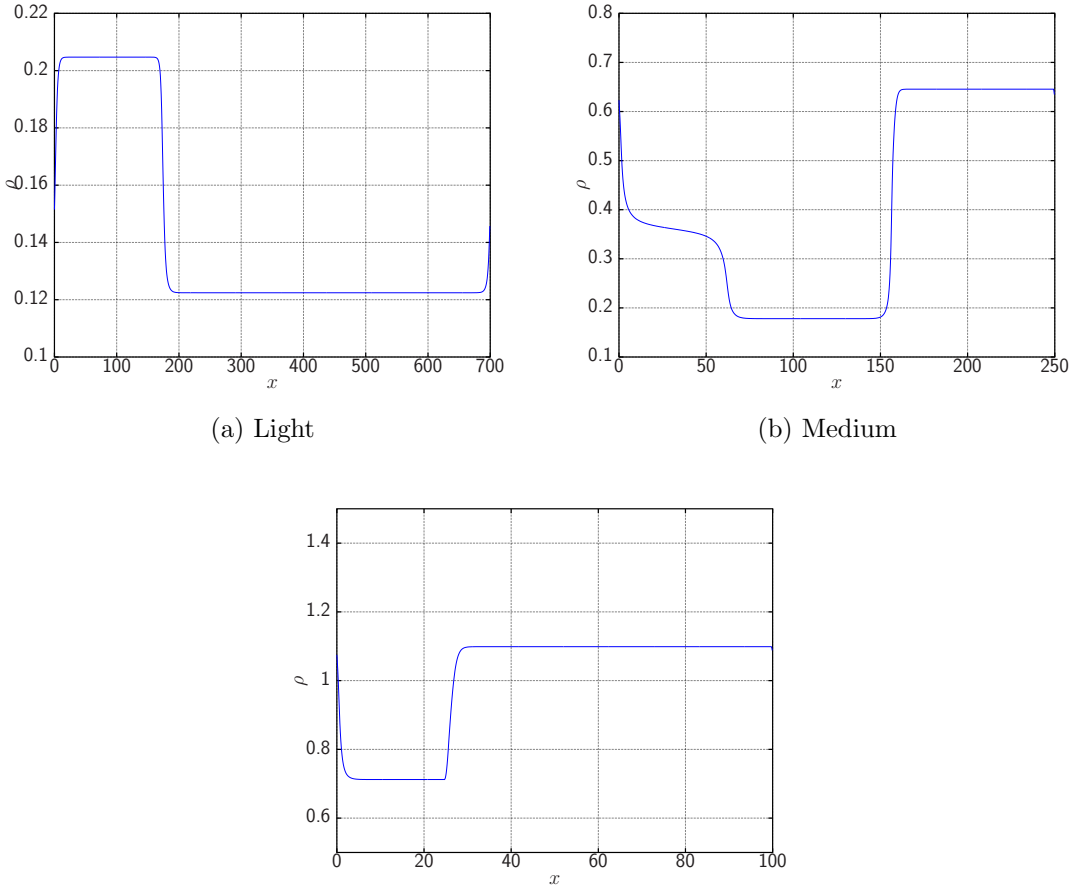

Figure 3.11: Error convergence for light traffic.




(a) Piecewise linear

(b) Smooth

Figure 3.12: Simulations of light traffic using Newton's method.




(a) Light

(b) Medium

(c) Heavy

Figure 3.13: Steady-state traffic simulation for light, medium and heavy traffic densities ($\lambda = 3$).

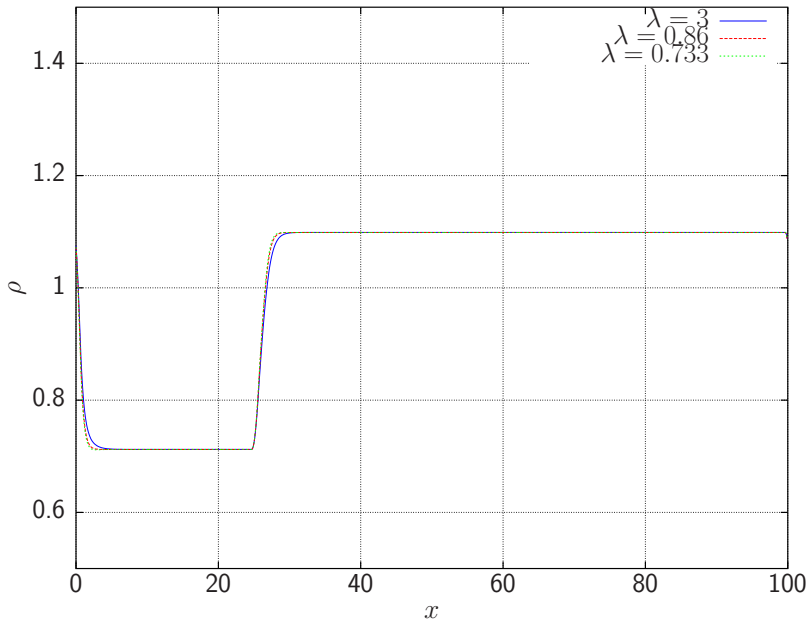



Figure 3.14: Steady-state solutions for heavy traffic flow for $\lambda = 3$, 0.86 and 0.733.

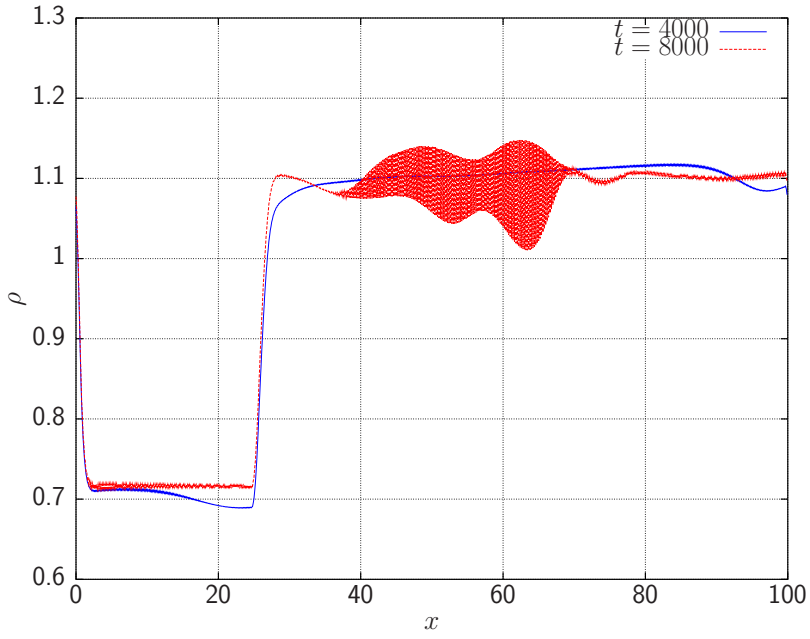



Figure 3.15: Transient simulation for linearly unstable heavy traffic flow ($\lambda = 0.733$).

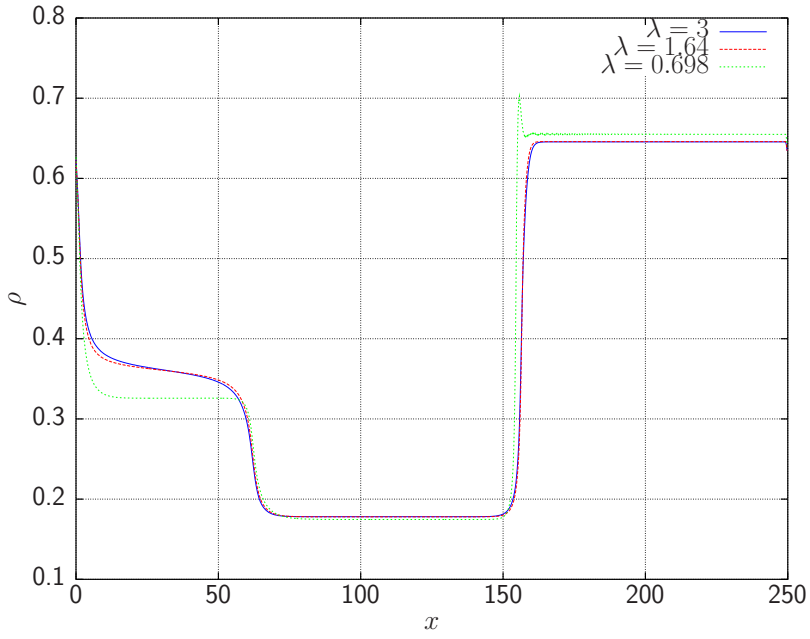



Figure 3.16: Steady-state solutions for medium traffic flow for $\lambda = 3$, 1.64 and 0.698.

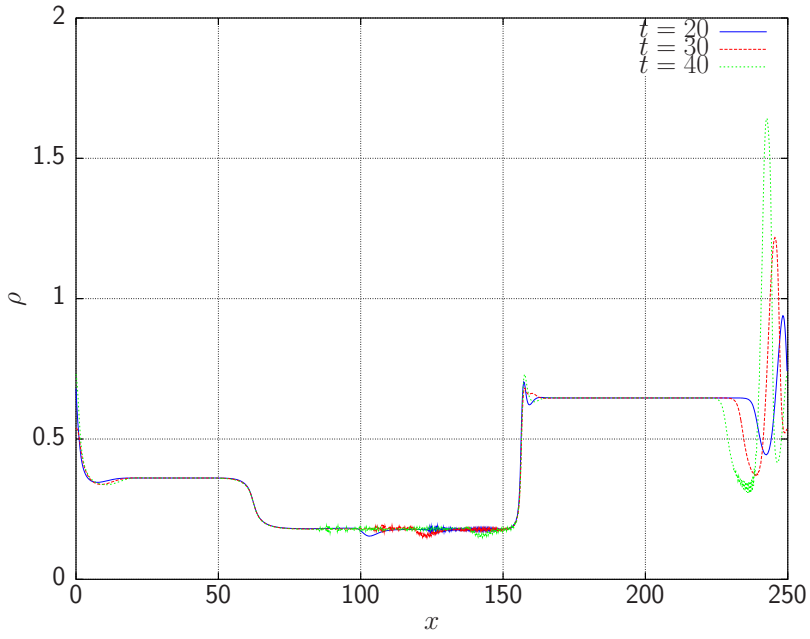



Figure 3.17: Transient simulation for linearly unstable medium traffic flow ($\lambda = 0.698$).

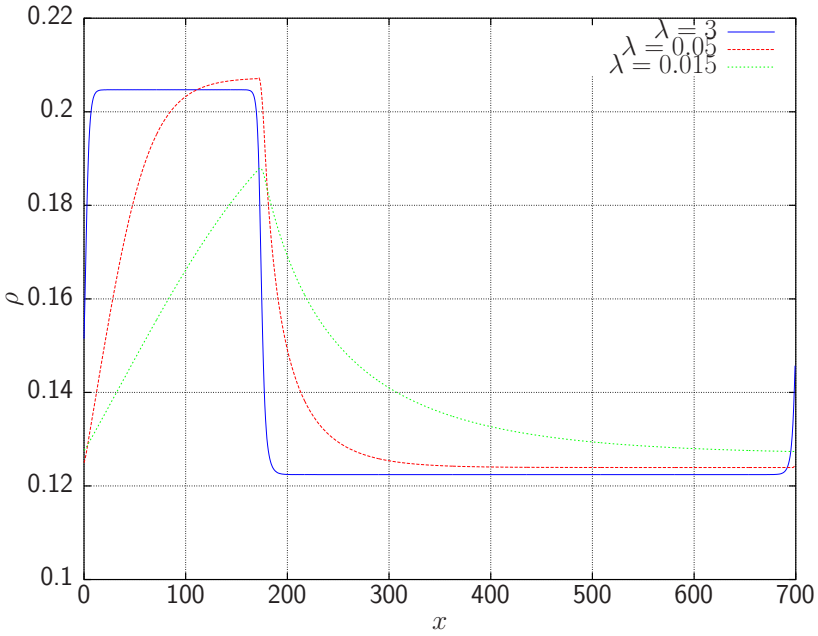



Figure 3.18: Steady-state solutions for light traffic flow for $\lambda = 3$, 0.05 and 0.015.

# Chapter 4

# Conclusion

## 4.1 Energy-Economic Models

From a numerical methods perspective, the energy-economic models offered an opportunity to investigate and implement DAE codes. Being able to work directly with the DAE system without manipulation is certainly helpful, since algebraic manipulation and differentiation was cumbersome, given all the parameters. DASSL is a user-friendly code to use with many handy features, which make it easy to solve systems of equations quickly and efficiently so that researchers can spend more time creating models and analyzing results.

The analytical solution for $\rho = 0$ and the numerical simulation of the model for the same case matched, as expected, and added confidence to numerical simulation results. The production curves resemble those of the Hubbert model except for the symmetry about the peak. The model predicts that a higher peak is followed by a

steeper decline in production.

The model, as presented in this thesis, is very sensitive to the chosen parameters. Hence, more work is needed in this area. Kemfert's [18] data was the most relevant and complete data set available. However, the results were certainly skewed by the fact that German chemical industry is less energy intensive (GDP/Joule) than the overall global average, yet it consumes more energy that the global average economy.

The biggest area in which the model could be improved is the assumption that 100 percent of the energy comes from oil. This leads to high oil prices as seen Figure 2.7 and Figure 2.9. More advanced models are required to add other energy sources such as nuclear and hydro-electric. The price for oil, $p_E$, is also monotonically increasing in the model. With the addition of other sources of energy, care should be taken to allow the price for energy to decrease.

Based on the sensitivity of the model, it seems reasonable to believe that with the proper choice of parameters and some change in the model for the assumptions mentioned above, it could match closely with historical production data.

## 4.2 Traffic Modeling

The steady-state simulation method was a computational success if its sole purpose is to determine the final steady solution. Table 3.1 showed a dramatic decrease in the number of function evaluations for the steady-state method versus the transient simulation methods. Moreover, steady-state solutions in the linearly unstable regime were derived that cannot be found with an initial value solver.

Transient simulations do provide more information than simply the final solution so methods should still be improved to reduce the simulation time. The mesh size is one area that could be improved. The moving mesh method did reduce the size but mesh points only moved slightly as seen in Figure 3.7. Another method to move the mesh would be to consider the arc length of $\rho$ along the road of length $L$ and then to position the grid points so that the arc length of each of the segments is equal. This would still have more mesh nodes than necessary in the plateau regions. Adding an attraction term so the nodes move towards areas were the $\rho_x$ and $\rho_{xx}$ are non-zero would certainly help to pull mesh points into the desired regions.

Heavy, medium and light traffic each exhibited different behaviours for the linearly unstable cases. Steady-state simulations showed that there is very little difference for heavy traffic. Medium traffic showed large spikes in density that grew as $\lambda$ was lowered. Light traffic in fact never reached the plateau values for small $\lambda$. Regardless of the density, the transient simulations were not able to converge to the steady-state solution in reasonable time. That shows the numerical stability of the steady-state method, which in essence was an error minimization code.

Another method for determining the linear unstable steady-state would be to decrease $\lambda$ over time. This would work for both the transient simulation and the steady simulation methods. This idea is inspired by the result in Figure 3.14 which shows steady-state solutions for heavy traffic. Starting with $\lambda = 3$, the code could wait for convergence, and then decrease $\lambda$ to 0.86 and then to 0.733 and so on (or any other values). This change in $\lambda$ over time has no real physical significance but would aide in computational efficiency, since the more unstable the model is, the smaller the steps that are taken and the longer the time until convergence.

In terms of the traffic models themselves, the following ideas could be investigated in future studies. Instead of scaling the optimal velocity function $V_{op}$ in the bottle neck by the reduction factor $r_B$, a clip on the velocity could be imposed. The new optimal velocity function would look like

$$V_{op}(h,x) = \begin{cases} \min(r_B(1+\tanh(2)), \tanh(h-2)+\tanh(2)), & 0 \le x < \hat{L}L, \\ (\tanh(h-2)+\tanh(2)), & \hat{L}L \le x < L. \end{cases} \tag{4.2.1}$$

In this case, vehicles would have the same acceleration profile when traveling at speeds lower than the imposed speed limit. This would maintain the drivers aggressiveness, model the observation that most drivers do not change their behaviour in traffic congestion. Numerically, this would add a slight challenge since the optimal velocity function would no longer be smooth.

Another topic for investigation would be to simulate traffic flow where each vehicle has its own value of $\lambda$, $\lambda_i$. This would be easiest to model using the microscopic model. Different distributions for $\lambda$ would cause different flow behaviour. A bimodal or trimodal distribution would allow for groups of drivers such as aggressive drivers, normal drivers and slow or large vehicles. Of course, this model would be most interesting for multi-lane traffic, since on a single lane without passing, eventually all the vehicles would catch up to the pace car vehicle with the smallest $\lambda$.

# Bibliography


[1] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, (1998) 82.

[2] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, (1998) 147-149.

[3] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, (1998) 231-256.

[4] M. Bando, K. Hasebe, A. Nakayama, A. Shibata and Y. Sugiyama. *Dynamical model of traffic congestion and numerical simulation*. Physical Review E 51 (1995) 1035-1041.

[5] P. Berg, A. Mason and A. Woods. *Continuum approach to car-following models*. Physical Review E 61 (2000) 1056-1066.

[6] P. Berg, P. Hanz and I. Milton. *An energy-economic Hubbert model*. Submitted to Energy Economics, January 2010.

[7] British Petroleum. *Statistical review of world energy. 2008 and preceding years*. http://www.bp.com

[8] K. E. Brenan, S. L. Campbell and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, (1996) 1-39.

[9] K. E. Brenan, S. L. Campbell and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, (1996) 16-18.

[10] K. E. Brenan, S. L. Campbell and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, (1996) 42.

[11] K. E. Brenan, S. L. Campbell and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* SIAM, (1996) 182-185.

[12] P. H. Douglas. *The Cobb-Douglas production function once again: Its history, its testing, and some new empirical values.* The Journal of Political Economy 84 (1976) 903-916.

[13] *GNU Scientific Library (GSL).* http://www.gnu.org/software/gsl/

[14] P. Hanz. *Energy Economic Models.* Bachelor of Science thesis, UOIT, April 2009.

[15] J. Henderson and R. Quandt. *Microeconomic Theory: A Mathematical Approach (Second Edition).* McGraw-Hill Inc. (1971) 81.

[16] M. Hubbert. *Nuclear energy and the fossil fuels.* Drilling and Production Practice 1956; American Petroleum Institute

[17] C. Jones. *Introduction to Economic Growth (Second Edition).* W. W. Norton & Compan (2001) 192.

[18] C. Kemfert. *Estimated substitution elasticities of a nested CES production function approach for Germany.* Energy Economics 20 (1998) 249-264.

[19] *Linear Algebra PACKage (LAPACK).* http://www.netlib.org/lapack/

[20] H. K. Lee, H.-W. Lee and D. Kim. *Macroscopic traffic models from microscopic car-following models.* Physical Review E 64 (2001) 056126.

[21] S. Li and L. Petzold. *Moving mesh methods with upwinding schemes for time-dependent PDEs.* Journal of Computational Physics 131 (1997) 368-377.

[22] D. McFadden. *Constant elasticity of substitution production functions.* The Review of Economic Studies 30 (1963) 73-83.

[23] *OpenMP.* http://openmp.org/wp/

[24] L. Petzold. *A Description of DASSL: A Differential/Algebraic System Solver.* Proceedings IMACS World Congress 1982.

[25] A. Pickering. *The oil reserves production relationship.* Energy Economics 30 (2007) 352-370.

[26] J. Ward, R. E. Wilson and P. Berg. *Multiscale analysis of a spatially heterogeneous microscopic traffic model.* Physica D 236 (2007) 1-12.