NTNU – Trondheim
Norwegian University of
Science and Technology

Numerical Methods for Valuation and Optimal Operation of Natural Gas Storage

Erik Magnus G. Følstad

Master of Science in Physics and Mathematics
Submission date: March 2015
Supervisor: Espen Robstad Jakobsen, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

The thesis describes different approaches for solving numerically a PDE model for the valuation and optimal operation of natural gas storage, characterized as a Hamilton Jacobi Bellman (HJB) equation. The HJB equation is derived by formulating the given natural gas storage problem as a stochastic control problem and then applying the dynamic programming principle. We present three separate numerical methods for solving the HJB equation, namely a standard upwind finite difference method, and two new methods characterized as: (i) a semi-Lagrangian time stepping method combined with a one dimensional finite element method, and (ii) a two dimensional finite element method combined with finite difference discretization in time. The upwind finite difference method is shown to be consistent, stable and monotone. These properties guarantee that the numerical solution converge to the viscosity solution of the HJB equation, [19]. Numerical results suggest that the two new methods converge to the same solution as the finite difference method for a given test case.

Sammendrag

Denne oppgaven beskriver ulike numeriske metoder for å løse en PDE modell for verdisetting og optimal styring av naturgasslagring, gitt som en Hamilton Jacobi Bellman (HJB) likning. Likningen er utledet ved å først formulere problemstillingen som et stokastisk kontrollproblem og deretter benytte dynamisk programmeringsprinsippet. Vi presenterer tre ulike metoder for å løse denne likningen, nærmere bestemt en standard oppvind differanse metode, og to nye metoder beskrevet som: (i) en semi-Lagrangian tids-diskretiseringsmetode kombinert med endelig element metode i en romlig retning, og (ii) en endelig elementmetode i to romlige retninger kombinert med endelig differanse i tid. Det blir vist at differansemetoden er konsistent, stabil og monoton, hvilket impliserer at den numeriske løsningen konvergerer til viskositetsløsningen av HJB likningen, [19]. Numeriske resultater tyder på at de to nye metodene konvergerer mot samme løsning som differansemetoden for et gitt test-problem.

Contents

1 Introduction 9

1.1 Previous Work . . . 9

1.2 The purpose of this thesis . . . 9

1.3 Contribution . . . 10

1.4 Outline . . . 10

2 Background 11

2.1 Stochastic control theory . . . 11

2.2 Dynamic Programming Principle and the Hamilton Jacobi Bellman Equation . . . 12

2.3 Derivation of the Natural Gas Storage Model . . . 13

2.4 Important assumptions about the NGS model . . . 17

2.4.1 Boundary . . . 17

2.4.2 Bang-bang controls . . . 19

3 A Monotone Finite Difference Scheme 21

3.1 Grid . . . 21

3.2 The upwinding technique . . . 22

3.3 Numerical scheme . . . 23

3.3.1 Boundary . . . 23

3.4 Monotonicity . . . 24

3.5 Stability . . . 26

3.6 Consistency . . . 28

3.7 Implementation . . . 29

4 The Finite Element method (1D) 33

4.1 Weak Formulation . . . 33

4.2 Approximation . . . 34

4.3 Stabilization . . . 36

4.4 Matrix Formulation . . . 36

4.5 Implementation . . . 37

4.5.1 Computation of integrals . . . 37

4.5.2 Assembly of matrices . . . 38

5 A Semi-Lagrange Finite Element Method 39
5.1 Linearization 40
5.2 Derivation of the numerical scheme 41
5.3 Implementation 43

6 A Finite Element Method (2D) 47
6.1 Weak formulation 49
6.2 Approximation 50
6.3 Stabilization 51
6.4 Time Discretization 52
6.5 Solving the Optimization problem 52
6.6 Numerical computation of integrals 53
6.6.1 Mapping from the triangular reference element 54
6.6.2 Transformation of integrals 55
6.7 Implementation 56

7 Numerical Results 59
7.1 A “realistic” test case 59
7.2 Testing the convergence rate 60
7.2.1 FE method 61
7.2.2 FE-Semi-Lagrangian 62
7.3 Verification of convergence to the viscosity solution 63

8 Conclusion 67
8.1 Summary 67
8.2 Challenges 68
8.3 Future Work 68

List of Figures

4.1 Node placement on a generic element $K_i = [x_i, x_{i+1}] \in \mathcal{T}_h$, for linear, quadratic and cubic elements. Nodes displayed as black dots. 35

6.1 $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ triangular elements with nodes (degrees of freedom) displayed in black. 50

6.2 Mapping $\hat{F}(\hat{x}, \hat{y}) := (g(\hat{x}, \hat{y}), h(\hat{x}, \hat{y}))^\top$ from the reference element $\hat{K}$ (left) to a physical element K (right). 54

7.1 Numerical solution of the value function and the optimal control policy evaluated at $\tau = 1$. The discretization parameters are $\Delta x = x_{\max}/20, \Delta y = y_{\max}/20$ and $\Delta \tau = T/200$. 65

7.2 Numerical solution of the value function after 10 time steps, with $h = 0.04$ and $\Delta \tau = 0.0156$. When we remove the edge stabilization, spurious oscillations are observed in the solution. The oscillations starts in $x = 0$, where there is least diffusion. Similar observation are made with $\mathbb{P}_1$ elements. We found that the stabilization parameters $\gamma = 0.02$ and $\gamma = 0.5$ worked well for $\mathbb{P}_1$ and $\mathbb{P}_2$ elements respectively. 66

List of Tables

4.1 Gauss quadrature rules for the reference interval $(-1, 1)$. 37

6.1 Gauss quadrature rules for the reference triangle. 54

7.1 Verification of the convergence rate $\hat{\beta} = 2$ for the $p1$-FE method. with $\alpha = 0.7071$. The stabilization parameter is $\gamma = 0.5$ 62

7.2 Verification of the convergence rate $\hat{\beta} = 3$ for the $p2$-FE method. with $\alpha = 0.7937$. The stabilization parameter is $\gamma = 0.02$ 62

7.3 Verification of the convergence rate $\beta = 2$ for the $p1$-SLFE method with $\alpha = 0.7071$. No edge stabilization is used ($\gamma = 0$). 63

7.4 Verification of the convergence rate $\beta = 3$ for the $p2$-SLFE method with $\alpha = 0.7937$. No edge stabilization is used ($\gamma = 0$). 63

7.5 Verification of convergence to the viscosity solution for the $\mathbb{P}_1$-SLFE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the $\mathbb{P}_1$-SLFE-method and the finite difference method, respectively. No edge stabilization is used ($\gamma = 0$). 64

7.6 Verification of convergence to the viscosity solution for the $\mathbb{P}_2$-SLFE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the $\mathbb{P}_2$-SLFE-method and the finite difference method, respectively. No edge stabilization is used ($\gamma = 0$). 64

7.7 Verification of convergence to the viscosity solution for the $\mathbb{P}_1$-FE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the $\mathbb{P}_1$-FE-method and the finite difference method, respectively. The stabilization parameter is $\gamma = 0.5$. 64

7.8 Verification of convergence to the viscosity solution for the $\mathbb{P}_2$-FE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the FE-method and the finite difference method, respectively. The stabilization parameter is $\gamma = 0.02$. 64

Chapter 1

Introduction

The real option of storing natural gas in the presence of uncertain gas prices is important in planning-models for the production of natural gas and allows industries to exploit spot market variations and seasonal trends, [15]. As we will demonstrate, the real option of storing natural gas can be modeled as a stochastic control problem from which a Hamilton Jacobi Bellman (HJB) equation is derived. This thesis is concerned with the numerical solution of such models.

1.1 Previous Work

Thompson et al. [14] derive a natural gas storage model in the form of a HJB equation and propose a fully explicit finite difference scheme. Because of the hyperbolic nature of the HJB equation, the need for stabilization techniques to prevent numerical oscillations are addressed. Thompson et al. suggest a min-mod slope limiting method to cope with this problem.

Forsyth and Chen [19] present an implicit semi-Lagrange time stepping scheme combined with a finite difference method for the natural gas problem and show that the scheme converges to the viscosity solution of the HJB equation by proving that the scheme is consistent, stable and monotone. The semi-Lagrange time stepping method is widely used in the field of numerical weather predictions [13] and is known to be stabilizing for convection dominated problems.

1.2 The purpose of this thesis

- Provide modeling background and derive the HJB equation for the natural gas storage problem.

- Study numerical methods for solving the given HJB equation and propose new methods.

- Implement the methods in MATLAB and conduct numerical experiments.

1.3 Contribution

After providing a modeling background for the HJB equation, we study an upwind finite difference method which is shown to be consistent, stable and monotone provided a linear CFL-condition. Then two new methods for this problem are introduced:

(i) A semi-Lagrangian time stepping method in one spatial direction combined with a finite element method in the other spatial direction. The method is inspired from [19]. Another method that combines semi-Lagrangian time stepping with finite elements is given in [18], however, in [18] finite elements and semi-Lagrangian time stepping are applied in the same spatial direction.

(ii) A two dimensional finite element method combined with a finite difference discretization in time. To cope with the issue of spurious oscillations we have implemented the edge stabilization technique as given by Burman and Hansbo [4].

1.4 Outline

Chapter 2 Some basic ideas from the field of stochastic control theory are introduced. These ideas are applied in section 2.3 to derive a model for the valuation and optimal operation of a natural gas storage facility in the form of a Hamilton Jacobi Bellman equation.

Chapter 3 An upwind finite difference scheme is described and analyzed. The scheme is shown to be consistent, stable and monotone provided a linear CFL-condition.

Chapter 4 This chapter is to intended as a brief introduction to the finite element method which will be used in the chapter 5.

Chapter 5 A semi-Lagrange time stepping combined with a finite element method is presented. A fully implicit scheme is achieved via a linearization technique.

Chapter 6 We present a finite element method coupled with finite finite difference time stepping. Edge stabilization [4] is implemented to prevent numerical oscillations.

Chapter 7 Numerical experiments are conducted with a standard test case in the literature.

Chapter 8 We summarize the work and give some concluding remarks.

Chapter 2

Background

2.1 Stochastic control theory

Many real life phenomena, in which the random nature of certain quantities plays a significant role, can be successfully modeled as a *stochastic differential equation* (SDE). For instance, in finance, stochastic differential equations can be used to describe the evolution of stock prices or commodity prices [12].

In the field of Stochastic control theory, one studies how the state of a system, given as the solution of a system of SDE's, is influenced by some controlled parameter. The objective is typically to maximize (or minimize) some functional that depends on the the state of the system by choosing the optimal value for the controlled parameter. As described in [9, p. 27], a stochastic control problem is characterized by the following features;

- *State of the system*: We consider a system $\mathbf{X} = (X_1, X_2 \ldots, X_n)$ satisfying a system of SDE's. The state of the system at any time $s \geq 0$ is given by $\mathbf{X}(s)$.
- *Control*: The system $\mathbf{X}$ is influenced by a controlled process $\alpha : s \to A \subset \mathbb{R}$. The value of α at time s must be chosen by using only available information at time s. In addition, α must satisfy certain constraints to be *admissible.*
- *Performance criterion*: The objective is to maximize (or minimize) some functional $\mathcal{P}(X, \alpha)$ over all admissible controls. Hence, a stochastic control problem can be defined as finding an optimal control α^*, such that $\mathcal{P}(X, \alpha^*)$ is maximized (or minimized).

Let $t, T \in \mathbb{R}$, such that $0 \leq t < T$. Let $\mathbf{f} : \mathbb{R}^n \times A \to \mathbb{R}^n$ be a given function and let $\mathcal{A}$ represent the set of admissible control functions

$$\mathcal{A} = \{\alpha : [t, T] \to A \,|\, \alpha(\cdot) \,\text{admissible}\}.$$

The dynamics of the state $\mathbf{X}$ of the system is dependent on the control process

$\alpha(\cdot) \in \mathcal{A}$ and is given by the SDE

$$
\begin{aligned}
d\mathbf{X}(s) &= \mathbf{f}(\mathbf{X}(s), \alpha(s), s)ds + \sigma(\mathbf{X}(s), s)d\mathbf{W}(s), \qquad t < s < T && (2.1)\\
\mathbf{X}(t) &= \mathbf{x}, && (2.2)
\end{aligned}
$$

where $d\mathbf{W}(s)$ represents the standard increment of an n-dimensional Brownian motion and σ is assumed to be a known deterministic function. The point $(\mathbf{x}, t)$ is referred to as the initial state of the system.

A generic performance functional can be stated as

$$
\mathcal{P}[\alpha(\cdot)]_{\mathbf{x},t} = \mathrm{E}\left[\int_t^T r(\mathbf{X}(s), \alpha(s), s)\, ds \,\middle|\, \mathbf{X}(t) = \mathbf{x} \right], \qquad \forall \alpha \in \mathcal{A}. \tag{2.3}
$$

where r is some given function that typically represents cash flow or the instantaneous rate of change in some other desirable quantity that accumulates over the time horizon $[0, T]$. The requirements for a function $\alpha : T \to \mathbb{R}$ to be admissible is problem dependent, e.g. $\mathcal{A}$ can depend on the initial state of the system.

In this work, we will consider the following stochastic optimal control problem; For all initial states $(\mathbf{x}, t) \in \Omega \times [0, T]$, find $\alpha^* \in \mathcal{A}$ such that

$$
\mathcal{P}[\alpha^*(\cdot)]_{\mathbf{x},t} = \sup_{\alpha \in \mathcal{A}} \mathcal{P}[\alpha(\cdot)]_{\mathbf{x},t}, \tag{2.4}
$$

with $\Omega \subset \mathbb{R}^n$ representing all possible values for $\mathbf{x}$. The *value function* $v : \Omega \times [0, T] \to \mathbb{R}$ is as

$$
v(\mathbf{x}, t) = \sup_{\alpha \in \mathcal{A}} \mathcal{P}[\alpha^*(\cdot)]_{\mathbf{x},t}, \qquad \forall (\mathbf{x}, t) \in \Omega \times [0, T]. \tag{2.5}
$$

Under certain assumptions, one can show via *dynamic programming* that the value function satisfies the Hamilton Jacobi Bellman equation, which will be introduced in the next section.

2.2 Dynamic Programming Principle and the Hamilton Jacobi Bellman Equation

Bellman's principle of optimality, also referred to as the dynamic programming principle, see [2, p. 83], is stated as

Dynamic Programming Principle. *An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.*

With respect to the optimal control problem (2.4) defined in the previous section, the dynamic programming principle can be formulated mathematically as

$$
v(\mathbf{x}, t) = \sup_{\alpha \in \mathcal{A}} \mathrm{E}\left[\int_t^{t+\delta t} (r(\mathbf{X}(s), \alpha(s), s) + v(\mathbf{X}_\alpha(t + \delta t), t + \delta t))\, ds \right], \tag{2.6}
$$

where $\mathbf{X}_\alpha(t+\delta t)$ represents the state of the system $\mathbf{X}$ at time $t+\delta t$ and the subscript α indicates that we have applied the control $\alpha(\cdot)$ in the time interval $[t, t+\delta t]$.

It can be shown that the value function v satisfies the Hamilton Jacobi Bellman equation, stated as

$$-\frac{\partial v}{\partial t}(\mathbf{x}, t) - \sup_{a \in A}[\mathcal{L}^a v(\mathbf{x}, t) + r(\mathbf{x}, a, t)] = 0, \tag{2.7}$$

where $\mathcal{L}^a$ is an operator including partial derivatives up to second order depending on the control parameter $a \in A$. Equation (2.7) is derived from problem (2.4) using the dynamic programming principle (2.6) and assuming that the value function is sufficiently smooth [9]. We shall heuristically derive an instance of the Hamilton Jacobi Equation in section 2.3 relating to the optimal control of a natural gas storage facility.

2.3 Derivation of the Natural Gas Storage Model

We derive the Hamilton Bellman Jacobi equation for the valuation and optimal operation of a gas storage facility, following Thompson et al. [14]. Similar models can also be derived for the valuation and optimal operation of a hydro electrical power plant [8], [5].

For the purposes of this model we think of a natural gas storage facility as being essentially a storage unit from which gas can be injected and withdrawn at any time. In order to maximize profits the operators of the facility wish to, roughly speaking, sell gas when the price is high and buy gas when the price is low. We introduce the following variables

- X - the price per unit of gas,
- Y - the amount of working gas in storage.

It is assumed that the facility can only be operated within a finite time horizon. We let t denote the present time and we let T denote the end of the time horizon. For any time $s \in [t, T]$, the price of unit gas and the amount of gas in storage is expressed as $X(s)$ and $Y(s)$, respectively.

The policy chosen by the operators of the facility is represented by a *control function* $\alpha : [t, T] \to \mathbb{R}$. That is, for each time $s \in [t, T]$ the value of the control function $\alpha(s)$ represents the volume rate of gas being sold. We use the convention that if $\alpha > 0$, then gas is withdrawn from the facility and sold. On the other hand if $\alpha < 0$, then gas is being bought and injected into storage. For a control function $\alpha(\cdot)$ to be *admissible* we require that

$$a_{\min}(Y(s)) \leq \alpha(s) \leq a_{\max}(Y(s)), \qquad \forall s \in [t, T],$$

where $a_{\min}$ and $a_{\max}$ are known functions of Y representing the maximal injection rate and the maximal withdrawal rate respectively. In the following we let

$$A(y) = \{a \in \mathbb{R} : a_{\min}(y) \leq a \leq a_{\max}(y)\}, ,$$

and we let $\mathcal{A}(y)$ represent the collection of all admissible control functions given the initial level of gas $Y(t) = y$.

We assume that there is a leakage of gas represented by a deterministic function λ, perhaps dependent on the control policy α and the amount of gas in storage. The flow of gas out from storage, represented by the function f, is given by

$$f(Y(s), \alpha(s), s) = \lambda(Y(s), \alpha(s), s) + \alpha(s) \tag{2.8}$$

We are using the sign convention that if $f > 0$ then gas is flowing out from storage and if $f < 0$ then gas is flowing into storage. Consequently, the rate of change in the amount of gas in storage is given by $\frac{dY}{ds} = -f$. Let y represent the amount of working gas presently in storage at time t. Then Y satisfies the following ODE, (in infinitesimal form) :

$$\begin{aligned} dY(s) &= -f(Y(s), \alpha(s), s)\, ds \quad t < s < T \\ Y(t) &= y. \end{aligned} \tag{2.9}$$

The price per unit of gas X is assumed to be a stochastic process satisfying an SDE of the form

$$\begin{aligned} dX(s) &= \mu(X(s), s)ds + \sigma(X(s), s)dW(s), \quad t < s < T \\ X(t) &= x, \end{aligned} \tag{2.10}$$

where W represents a *Wiener process.*

DEFINITION. *A real valued stochastic process $W(t)$ is called a* Wiener process*, or* Brownian motion*, if*

1. $W(0) = 0$
2. *each sample path is continuous*
3. *$W(t)$ is $\mathrm{N}(0, t)$, (W is normally distributed with mean 0 and variance t)*
4. *W has independent increments*

The *revenue* of the gas storage facility is defined as the present value of the sum of all *cash flow* over the whole contract period. The cash flow, denoted r, is equal to the value of the gas currently being bought or sold minus the value of the gas currently being lost, that is

$$r(X(s), Y(s), \alpha(s), s) = \big(\alpha(s) - \lambda(Y(s), \alpha(s), s)\big)\, X(s), \qquad \forall s \in [t, T]. \tag{2.11}$$

The *present value* of this cash flow is assumed to be given by

$$\mathrm{e}^{-\rho(s-t)}(\alpha(s) - \lambda(Y(s), \alpha(s), s))\, X(s),$$

where ρ denotes the *risk free interest rate* [6, p. 115]. Consequently, the present value of the the sum of all cash flow over the whole contract period is equal to

$$\int_t^T \mathrm{e}^{-\rho(s-t)} r(X(s), Y(s), \alpha(s), s)\, ds.$$

The performance functional is given as

$$\mathcal{P}[\alpha(\cdot)]_{x,y,t} = \mathrm{E}\left[\int_t^T \mathrm{e}^{-\rho(s-t)} r(X(s), Y(s), \alpha(s), s)\, ds \,\middle|\, X(t) = x, Y(t) = y\right], \qquad \forall \alpha \in \mathcal{A}(y), \tag{2.12}$$

which associates with each admissible control $\alpha \in \mathcal{A}(y)$ an expected revenue $\mathcal{P}[\alpha]_{x,y,t}$, given the initial state $X(t) = x, Y(t) = y$.

Remark 2.1. *Note that with respect to generic performance functional (2.3), the performance functional given by the previous equation has an extra discount factor* $\mathrm{e}^{-\rho(s-t)}$.

We are interested in the following stochastic control problem for each initial state (x, y, t):

$$\text{find } \alpha \in \mathcal{A}(y)\text{: } \mathcal{P}[\alpha^*(\cdot)]_{x,y,t} = \sup_{\alpha \in \mathcal{A}(y)} \mathcal{P}[\alpha(\cdot)]_{x,y,t}. \tag{2.13}$$

Recall that the value function is given as

$$v(x, y, t) = \sup_{\alpha \in \mathcal{A}(y)} \mathcal{P}[\alpha(\cdot)]_{x,y,t}, \qquad \forall (x, y, t) \in \Omega \times [0, T]. \tag{2.14}$$

We will now proceed to derive heuristically a partial differential equation for the value function v, following [9, p. 43], specifically we obtain the Hamilton Jacobi Bellman equation (2.7). We will be needing the following well known result from the theory of stochastic differential equations, as given in [6, p. 78].

ITO'S FORMULA. *Let* $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ *represent an n-dimensional stochastic process such that*

$$dX_i = \mu_i(X(s), s) ds + \sum_{j=1}^{n} G_{ij}(X(s), s) dW_j \qquad \text{for } i = 1, \ldots, n.$$

with $\mu_i(X(s), s) \in \mathbb{L}^1(0, T)$ *and* $G_{ij}(X(s), s) \in \mathbb{L}^2(0, T)$.

If $u : \mathbb{R}^n \times [0, T] \to \mathbb{R}$ *is continuous and the partial derivatives* $\frac{\partial u}{\partial t}, \frac{\partial u}{\partial x_i}, \frac{\partial^2 u}{\partial x_i \partial x_j}$, $(i, j = 1, \ldots, n)$ *exist and are continuous, then*

$$d(u(X_1, \ldots, X_n, s)) = \frac{\partial u}{\partial t} ds + \sum_{i=1}^{n} \frac{\partial u}{\partial x_i} dX_i + \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial^2 u}{\partial x_i \partial x_j} \sum_{l=1}^{n} G_{il} G_{jl} ds.$$

□

Suppose that that the operators of the gas storage facility applies the constant control policy $\alpha \equiv a \in A(y)$ within the time interval $[t, t + \delta t]$. According to the dynamic programming principle (2.6), the value function satisfies the inequality

$$v(x,y,t) \geq \mathrm{E}\left[\int_t^{t+\delta t} \left(\mathrm{e}^{-\rho(s-t)} r(X,Y,a,s)\,ds + \mathrm{e}^{-\delta t} v(X(t+\delta t), Y(t+\delta t), t+\delta t)\right) ds\right]. \tag{2.15}$$

Provided that the value function v is a sufficiently smooth function of x, y and t, Ito's formula implies that

$$dv(X(s),Y(s),s) \;=\; \left(\frac{\partial v}{\partial t} + \mu\frac{\partial v}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 v}{\partial x^2} - f\,\frac{\partial v}{\partial y}\right) ds + \sigma\frac{\partial v}{\partial x} dW(s). \tag{2.16}$$

Remark 2.2. *For notational convenience we have omitted to write out that the partial derivatives $\frac{\partial v}{\partial t}$, $\frac{\partial v}{\partial x}$, $\frac{\partial^2 v}{\partial x^2}$, $\frac{\partial v}{\partial y}$ appearing on the right hand side of the last equation are evaluated at the point $(X(s),Y(s),s)$, the functions μ, σ are evaluated at $(X(s),s)$ and f is evaluated at $(X(s),Y(s),a,s)$.*

Inequality (2.15) can be rearranged as

$$0 \geq \mathrm{E}\left[\int_t^{t+\delta t} \mathrm{e}^{-\rho(s-t)} r\,ds + \mathrm{e}^{-\rho\delta t}\delta v - (1-\mathrm{e}^{-\rho\delta t})v\right]. \tag{2.17}$$

with

$$\delta v := v(X(t+\delta t), Y(t+\delta t), t+\delta t) - v(x,y,t).$$

From equation (2.16) we obtain

$$\delta v = \int_t^{t+\delta t} dv = \int_t^{t+\delta t} \left(\frac{\partial v}{\partial t} + \mu\frac{\partial v}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 v}{\partial x^2} - f\frac{\partial v}{\partial y}\right) ds + \int_t^{t+\delta t} \sigma\frac{\partial v}{\partial x} dW(s).$$

By substituting the expression for δv given by the previous equation into (2.17) and using that

$$\mathrm{E}\left[\int_t^{t+\delta t} \sigma\frac{\partial v}{\partial x} dW(s)\right] = 0,$$

we obtain

$$\begin{aligned} 0 \geq &\mathrm{E}\Bigg[\int_t^{t+\delta t} \mathrm{e}^{-\rho(s-t)} r\,ds \\ &+ \mathrm{e}^{-\rho\delta t}\int_t^{t+\delta t} \left(\frac{\partial v}{\partial t} + \mu\frac{\partial v}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 v}{\partial x^2} - f\frac{\partial v}{\partial y}\right) ds - (1-\mathrm{e}^{-\rho\delta t})v\Bigg]. \end{aligned}$$

Divide the previous inequality by δt and take the limit as δt goes to zero. Assuming that $\lim\limits_{\delta t\to 0}$ and $\mathrm{E}[\,\cdot\,]$ are interchangeable, we get

$$\begin{aligned} 0 \geq &\mathrm{E}\Bigg[\lim_{\delta t\to 0}\frac{1}{\delta t}\Bigg(\int_t^{t+\delta t} \mathrm{e}^{-\rho(s-t)} r\,ds \\ &+ \mathrm{e}^{-\rho\delta t}\int_t^{t+\delta t} \left(\frac{\partial v}{\partial t} + \mu\frac{\partial v}{\partial x} + \frac{1}{2}\sigma^2\frac{\partial^2 v}{\partial x^2} - f\frac{\partial v}{\partial y}\right) ds - (1-\mathrm{e}^{-\rho\delta t})v\Bigg)\Bigg]. \end{aligned}$$

Via the mean-value theorem, the previous equation implies that

$$0 \geq \left(\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} + \rho v \right)(x, y, t) + r(x, y, a, t) - f(y, a, s)\frac{\partial v}{\partial y}(x, y, t). \tag{2.18}$$

To obtain the last inequality we have also used the identity

$$\lim_{\delta t \to 0} \frac{1}{\delta t}(1 - \mathrm{e}^{-\rho \delta t}) = \rho.$$

Since (2.18) is true for any admissible $a \in A(y)$, we have

$$0 \geq \left(\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right)(x, y, t) + \sup_{a \in A(y)} \left(r - f\frac{\partial v}{\partial y} \right)(x, y, a, t). \tag{2.19}$$

On the other hand, if the operators of the gas storage facility applies the optimal policy α^* within the interval $[t, t + \delta t]$, it can be shown trough similar arguments that

$$0 = \left(\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right)(x, y, t) + r(x, y, a^*, t) - f(y, a^*, t)\frac{\partial v}{\partial y}(x, y, t), \tag{2.20}$$

with

$$a^* := \alpha^*(t).$$

Inequality (2.19) and equation (2.20) together imply that v should satisfy

$$0 = \left(\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right)(x, y, t) + \sup_{a \in A(y)} \left(r(x, y, a, t) - f(y, a, t)\frac{\partial v}{\partial y}(x, y, t) \right).$$

The previous equation is formulated backwards in time, via the change of variable $\tau = T - t$, we get

$$\left(\frac{\partial v}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} - \mu \frac{\partial v}{\partial x} + \rho v \right)(x, y, \tau) = \sup_{a \in A(y)} \left(r(x, y, a, \tau) - f(y, a, \tau)\frac{\partial v}{\partial y}(x, y, t) \right). \tag{2.21}$$

2.4 Important assumptions about the NGS model

2.4.1 Boundary

In reality there is no upper bound on the price variable x. However, in our numerical solution of (2.21) we can only consider a finite price range, so let $x_{\max}$ represent the maximal allowed price. The gas inventory variable y has an upper bound, represented by $y_{\max}$, equal to the maximum storage capacity. In this work we consider solving equation (2.21) on the domain $[0, x_{\max}] \times [0, y_{\max}] \times [0, T]$ and we refer to

$$\Omega = [0, x_{\max}] \times [0, y_{\max}],$$

as the *spatial* domain. For convenience we assume through out this work that the coefficients of (2.21) are scaled such that the spatial domain is equal to the unit square, that is

$$x_{\max} = y_{\max} = 1.$$

The gas price process is assumed to be *mean reverting.* Let $\mu_0 > 0$ and let $\bar{x} \in (0, 1)$. A simple example of a mean reverting stochastic process is stated

$$dX(s) = \mu_0(X(s) - \bar{x})\, ds + \sigma(X(s), s)\, dW(s),$$

which corresponds to setting

$$\mu(X(s), s) = \mu_0(X(s) - \bar{x}),$$

in equation (2.10). The process is called mean reverting because the gas price is constantly drawn to the mean reversion level $\bar{x} \in (0, 1)$ which corresponds to the long term average market price. When the price process is mean reverting, we have the following conditions on the boundaries $x = 0$ and $x = 1$

$$\begin{aligned} \mu|_{x=0} &\geq 0, \\ \mu|_{x=1} &\leq 0. \end{aligned} \tag{2.22}$$

For the gas inventory process it is safe to assume that gas can only be pumped into storage if the current level of gas is less than the maximum capacity. On the other hand, gas can only be pumped out of the facility if there is any gas left in inventory. This reasoning leads to the conditions

$$\begin{aligned} f|_{y=0} &\leq 0 \\ f|_{y=1} &\geq 0, \end{aligned} \tag{2.23}$$

where we recall that f represents the flow of gas out from storage, (2.8). Negative values for f corresponds to gas being pumped in to storage and positive values for f corresponds to gas being released.

In [14] and [19] it is argued without any real justification that

$$\frac{\partial^2 v}{\partial x^2} \to 0 \quad \text{as } x \to x_{\max}. \tag{2.24}$$

However, the boundary $x = x_{\max} := 1$ is assumed to be "far away" from the realistic price range, so this condition might not have a big impact on the solution. Condition (2.24) implies that the term $\sigma^2 \frac{\partial^2 v}{\partial x^2}$ in equation (2.21) goes to zero as

$x \to 1$. As suggested by [5] this can be implemented by replacing σ with $\hat{\sigma}$ such that

$$\begin{aligned} \hat{\sigma}(x) &= \sigma(x), && \text{if } x \in [0, 1-\epsilon], \\ \hat{\sigma}(x) &= 0, && \text{if } x \in [1-\epsilon, 1], \end{aligned}$$

where ϵ represents some small number. We will simply assume that $\sigma \to 0$ as $x \to x_{\max}$. It is also assumed that $\sigma \to 0$ as $x \to 0$. Consequently, equation (2.21) is nearly hyperbolic close to the boundaries $x = 0$ and $x = 1$, (because the term $\sigma^2 \frac{\partial^2 v}{\partial x^2}$ vanishes), and boundary conditions needs only be prescribed at the *inflow* part of the boundaries $x = 0$ and $x = 1$. However, condition (2.22) imply that the boundaries $x = 0$ and $x = 1$ are *outflow* boundaries. That is, the flow field given as

$$\mathbf{f} = (\mu, -f)^\top,$$

always points inwards to the domain. Hence, no boundary conditions are needed for equation (2.21) at $x \in \{0, 1\}$. Because the diffusive term in (2.21) has no component in the y-direction and the boundaries $y = 0$ and $y = 1$ are outflow boundaries by condition (2.23), no boundary conditions are needed at $y \in \{0, 1\}$.

For simplicity we will apply the initial condition $v(x, y, 0) = 0$, as suggested in [14]. Some alternative initial conditions are discussed in [19].

2.4.2 Bang-bang controls

The model can be split into two equations as follows

$$a^* = \arg \sup_{a \in A(y)} \left(r(x, y, a^*, \tau) - f(y, a^*, \tau) \frac{\partial v}{\partial y}(x, y, t) \right) \tag{2.25}$$

$$\frac{\partial v}{\partial \tau} = \left(\frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right)(x, y, \tau) - f(y, a^*, \tau) \frac{\partial v}{\partial y}(x, y, t) + r(x, y, a^*, \tau) \tag{2.26}$$

In [19] and [14] the chosen model for the gas leakage λ is of the form

$$\lambda(a) = \left\{ \begin{array}{ll} 0 & \text{if } a \geq 0 \\ k & \text{if } a < 0 \end{array} \right. .$$

With this choice of λ, we get

$$\begin{aligned} r(x, a) &= (a - \lambda(a))x, \\ f(a) &= (a + \lambda(a)), \end{aligned}$$

so the feedback control a^* can be found by solving two linear sub-problems, corresponding to $a \geq 0$ and $a < 0$ respectively, given as

$$\begin{aligned}
a_1^* &= \arg \sup_{a \in A(y)} \left(ax - a\frac{\partial v}{\partial y}(x, y, t) \right), && a \geq 0 \\
a_2^* &= \arg \sup_{a \in A(y)} \left((a-k)x - (a+k)\frac{\partial v}{\partial y}(x, y, t) \right) && a < 0
\end{aligned}$$

and then choose the best of these computed solutions:

$$a^* = \arg\sup\{(a_1^* - \lambda(a_1^*))x - (a_1^* + \lambda(a_1^*)), (a_2^* - \lambda(a_2^*))x - (a_2^* + \lambda(a_2^*))\}$$

Since both sub-problems are linear in a it is easy to verify that the optimal control only takes on values from the finite set $\{a_{\min}(y), 0, a_{\max}(y)\}$. Hence, the optimization problem is solved by simply comparing the possible values of the expression $r(x, a) - f(a)\frac{\partial v}{\partial y}$ for $a \in \{a_{\min}(y), 0, a_{\max}(y)\}$. That is,

$$a^*(x, y, \tau) = \arg\sup_{a \in \{a_{\min}, 0, a_{\max}\}} ((a - \lambda(a))x - (a + \lambda(a)).$$

The controls are said to be of *bang-bang* type.

Chapter 3

A Monotone Finite Difference Scheme

We present a first order semi implicit finite difference scheme for solving equation (2.21). In the spatial discretization we employ the standard first order *upwinding* technique, as described in [10, p. 284]. The time discretization is implicit in the price direction (x-direction) and explicit in the inventory direction (y-direction), leading to a linear CFL-condition.

Even if the following method is only first order accurate it has other desirable properties. The scheme is l_∞-stable, consistent and monotone. As noted in [19], these properties ensures that the numerical scheme converges to the *viscosity solution* of (2.21), which is the appropriate solution for stochastic control problems [9]. In addition, the finite difference method is generally easy to implement and vectorization of the method in MATLAB is straight forward.

We refer to [16] for an upwind finite volume method for the Hamilton Jacobi Bellman equation.

3.1 Grid

Let $(x_i)_{i=1}^{I}$, $(y_j)_{j=1}^{J}$ and $(\tau_n)_{n=0}^{N}$, be sets of *grid points* such that

$$0 = x_1 < x_1 < \cdots < x_{I-1} < x_I = x_{\max} := 1, \tag{3.1}$$

$$0 = y_1 < y_1 < \cdots < y_{J-1} < y_J = y_{\max} := 1, \tag{3.2}$$

and

$$0 = \tau_0 < \tau_1 < \cdots < \tau_{N-1} < \tau_N = T, \tag{3.3}$$

Fro simplicity, we assume that the grid points are uniformly spaced, that is

$$\Delta x = x_{i+1} - x_i, \qquad\qquad i = 1, \ldots, I,$$

$$\Delta y = y_{j+1} - y_j, \qquad\qquad j = 1, \dots, J,$$

and

$$\Delta \tau = \tau_{n+1} - \tau_n, \qquad\qquad n = 0, \dots, N.$$

for some $\Delta x > 0$, $\Delta y > 0$ and $\Delta \tau > 0$.

3.2 The upwinding technique

In order to introduce the upwind method, consider the following linear convection diffusion equation

$$-\sigma \frac{\partial^2 v}{\partial x^2} + b(x) \frac{\partial v}{\partial x} = 0. \tag{3.4}$$

Let $\tilde{v}$ represent the numerical approximation of v to be defined. According to the first order upwind method, the approximation $\tilde{v}_x$ of the convective term v_x is given by the following rule:

$$\begin{aligned} \tilde{v}_x &= \frac{\tilde{v}(x) - \tilde{v}(x - \Delta x)}{\Delta x}, \text{ if } b > 0, \\ \tilde{v}_x &= \frac{\tilde{v}(x + \Delta x) - \tilde{v}(x)}{\Delta x}, \text{ if } b < 0. \end{aligned} \tag{3.5}$$

Let

$$v_i := \tilde{v}(x_i), \qquad\qquad \text{for } i = 1, \dots, I,$$

define the operators $(\cdot)^+$ and $(\cdot)^-$ such that

$$a^+ = \max(a, 0), \qquad a^- = \min(a, 0), \qquad \forall a \in \mathbb{R},$$

and let the difference operators D_x^+ and D_x^- be given as

$$\mathrm{D}_x^+ \tilde{v}(x) = \frac{1}{\Delta x}(\tilde{v}(x + \Delta x) - \tilde{v}(x)), \qquad \mathrm{D}_x^- \tilde{v}(x) = \frac{1}{\Delta x}(\tilde{v}(x) - \tilde{v}(x - \Delta x)).$$

By using the standard central difference approach to approximate the diffusive term in equation (3.4) and the upwind method, given by (3.5), to approximate the convective terms, we end up with the following numerical scheme:

$$-\sigma \mathrm{D}_x^+(\mathrm{D}_x^- v_i) + b(x_i)^+ \mathrm{D}_x^- v_i + b(x_i)^- \mathrm{D}_x^+ v_i = 0, \qquad i = 1 \dots, I - 1.$$

Remark 3.1. *We observe that*

$$\mathrm{D}_x^+(\mathrm{D}_x^- v_i) = \frac{v_{i+1} - 2v_i + v_{i-1}}{\Delta x^2}.$$

3.3 Numerical scheme

In the previous section we introduced the upwind method for a one dimensional linear convection diffusion equation. We will now proceed to apply the method on equation (2.21), which we restate here for convenience:

$$\left(\frac{\partial v}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} - \mu \frac{\partial v}{\partial x} + \rho v\right)(x, y, \tau) = \\ \sup_{a \in A(y)} \left(r(x, y, a, \tau) - f(y, a, \tau)\frac{\partial v}{\partial y}(x, y, t)\right). \quad (3.6)$$

We assume, for simplicity, that σ and μ only depend on x. In what follows, we will use the notation

$$\mu_i = \mu(x_i), \qquad \sigma_i = \sigma(x_i),$$

and

$$f_j^{n+1}(a) = f(y_j, \tau_{n+1}, a), \qquad r_{ij}^{n+1}(a) = r(x_i, y_j, \tau_{n+1}, a).$$

Let the difference operators D_y^+ and D_y^- be defined such that

$$\mathrm{D}_y^+ v(x, y, \tau) = v(x, y + \Delta y, \tau) - v(x, y, \tau), \quad \mathrm{D}_y^- v(x, y, \tau) = v(x) - v(x, y - \Delta y, \tau).$$

By following the same discretization procedure as in the previous section for each direction x, y and τ, we obtain from equation (3.6) the numerical scheme:

$$\frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta \tau} - \sigma_i \mathrm{D}_x^+(\mathrm{D}_x^- v_{i,i}^{n+1}) - \mu_i^+ \mathrm{D}_x^+ v_{i,j}^{n+1} - \mu_i^- \mathrm{D}_x^- v_{i,j}^{n+1} = \\ \max_{a \in A(y_j)} \left(r_{ij}^n(a) - f_j^n(a)^+ \mathrm{D}^- v_{i,j}^n - f_j^n(a)^- \mathrm{D}_y^+ v_{i,j}^n\right), \quad (3.7)$$

for $i = 0, \ldots, I$, $j = 0, \ldots, J$ and $n = 0, \ldots, N - 1$.

Remark 3.2. *We have treated the convective term $f\frac{\partial v}{\partial y}$ and the term $r(x, y, a, \tau)$ in equation (3.6) explicitly, that is, in equation (3.7), these terms are evaluated at $\tau = \tau_n$ instead of $\tau = \tau_{n+1}$.*

Remark 3.3. *As we will show in the next subsection, no boundary conditions are needed for (3.7), provided that $\sigma(0) = \sigma(1) = 0$, $\mu(0) \geq 0$, $\mu(1) \leq 0$, $f_{|_{y=0}} \leq 0$ and $f_{|_{y=1}} \geq 0$.*

3.3.1 Boundary

At the boundaries $x = 1$ and $x = 0$, we apply the conditions

$$\sigma_1 = 0, \qquad \sigma_I = 0,$$

(recall section 2.4.1), and it is assumed that

$$\mu_1 \geq 0, \qquad\qquad \mu_I \leq 0,$$

so for $i = 1$ and $i = I$, respectively, the scheme given by (3.7) reads

$$\frac{v_{1,j}^{n+1} - v_{1,j}^{n}}{\Delta\tau} - \mu_1^+ \mathrm{D}_x^+ v_{1,j}^{n+1} = \max_{a \in A(y_j)} \left(r_{1,j}^n(a) - f_j^n(a)^+ \mathrm{D}^- v_{1,j}^n - f_j^n(a)^- \mathrm{D}_y^+ v_{1,j}^n \right), \tag{3.8}$$

and

$$\frac{v_{I,j}^{n+1} - v_{I,j}^{n}}{\Delta\tau} - \mu_0^- \mathrm{D}_x^- v_{I,j}^{n+1} = \max_{a \in A(y_j)} \left(r_{I,j}^n(a) - f_j^n(a)^+ \mathrm{D}^- v_{I,j}^n - f_j^n(a)^- \mathrm{D}_y^+ v_{I,j}^n \right). \tag{3.9}$$

At the boundaries $y = 0$ and $y = 1$, it is assumed that

$$f_0^n(a) \leq 0, \qquad\qquad f_J^n(a) \geq 0,$$

so for $j = 0$ and $j = J$, the scheme (3.7) reads

$$\begin{aligned} \frac{v_{i,1}^{n+1} - v_{i,0}^{n}}{\Delta\tau} - \sigma_i \mathrm{D}_x^+ (\mathrm{D}_x^- v_{i,1}^{n+1}) - \mu_i^+ \mathrm{D}_x^+ v_{i,1}^{n+1} - \mu_i^- \mathrm{D}_x^- v_{i,1}^{n+1} & \\ = \max_{a \in A(y_j)} \left(r_{i,1}^n(a) - f_j^n(a) \mathrm{D}_y^+ v_{i,1}^n \right) & \end{aligned} \tag{3.10}$$

and

$$\begin{aligned} \frac{v_{i,J}^{n+1} - v_{i,J}^{n}}{\Delta\tau} - \sigma_i \mathrm{D}_x^+ (\mathrm{D}_x^- v_{i,J}^{n+1}) - \mu_i^+ \mathrm{D}_x^+ v_{i,J}^{n+1} - \mu_i^- \mathrm{D}_x^- v_{i,J}^{n+1} & \\ = \max_{a \in A(y_j)} \left(r_{i,J}^n(a) - f_j^n(a) \mathrm{D}_y^- v_{i,J}^n \right) & \end{aligned} \tag{3.11}$$

3.4 Monotonicity

In this section we will show that, provided a linear CFL-condition (3.13), the numerical scheme (3.7) is *monotone*. That is, if u and v are solutions of (3.7), then

$$v^0 \geq u^0 \implies v^n \geq u^n \qquad\qquad \forall n > 0, \tag{3.12}$$

provided that

$$\Delta\tau \|f\|_\infty \leq \Delta y. \tag{3.13}$$

Remark 3.4. *The inequalities in (3.12) are to be intended component vice.*

The proof is by induction, suppose that $v^n \geq u^n$ and choose i^*, j^* such that

$$(i^*, j^*) = \arg\min_{i,j}(v_{i,j}^{n+1} - u_{i,j}^{n+1}). \tag{3.14}$$

The numerical scheme (3.7) can be rewritten as

$$v_{i,j}^{n+1} - \Delta\tau(\sigma_i \mathrm{D}_x^+(\mathrm{D}_x^- v_i) + \mu_i^+ \mathrm{D}_x^+ v_i + \mu_i^- \mathrm{D}_x^- v_i) = v_{i,j}^n + \tilde{\mathrm{H}}(v_{i,j}^n), \tag{3.15}$$

with

$$\tilde{\mathrm{H}}(v_{i,j}^n) = \max_{a \in A(y_j)} \left(r_{ij}^n(a) - f_j^n(a)^+ \mathrm{D}^- v_{i,j}^n - f_j^n(a)^- \mathrm{D}_y^+ v_{i,j}^n \right).$$

By writing out all the terms in equation (3.15), we get

$$v_{i,j}^{n+1} - \Delta\tau \left(\sigma_i \frac{v_{i+1,j}^{n+1} - 2v_{i,j}^{n+1} + v_{i-1,j}^{n+1}}{\Delta x^2} + \mu_i^+ \frac{v_{i+1,j}^{n+1} - v_{i,j}^{n+1}}{\Delta x} + \mu_i^- \frac{v_{i,j}^{n+1} - v_{i-1,j}^{n+1}}{\Delta x} \right)$$
$$= v_{i,j}^n + \tilde{\mathrm{H}}(v_{i,j}^n),$$

which can be rewritten as

$$v_{i,j}^{n+1} \frac{\Delta\tau}{\Delta x} \left(\frac{\Delta x}{\Delta \tau} + \frac{2\sigma_i}{\Delta x} + \mu_i^+ - \mu_i^- \right) - v_{i+1,j}^{n+1} \frac{\Delta\tau}{\Delta x} \left(\frac{\sigma_i}{\Delta x} + \mu_i^+ \right)$$
$$-v_{i-1,j}^{n+1} \frac{\Delta\tau}{\Delta x} \left(\frac{\sigma_i}{\Delta x} - \mu_i^- \right) = v_{i,j}^n + \tilde{\mathrm{H}}(v_{i,j}^n). \tag{3.16}$$

By rearranging the terms in the last equation, we get

$$v_{i,j}^{n+1} = c_1 \left(v_{i,j}^n + c_2 v_{i+1,j}^{n+1} + c_3 v_{i-1,j}^{n+1} \right) + c_1 \mathrm{H}(v_{i,j}^n) \tag{3.17}$$

with

$$c_1 = \frac{\frac{\Delta x}{\Delta \tau}}{\left(\frac{\Delta x}{\Delta \tau} + \frac{2\sigma_i}{\Delta x} + \mu_i^+ - \mu_i^- \right)}, \quad c_2 = \frac{\Delta\tau}{\Delta x} \left(\frac{\sigma_i}{\Delta x} + \mu_i^+ \right), \quad c_3 = \frac{\Delta\tau}{\Delta x} \left(\frac{\sigma_i}{\Delta x} - \mu_i^- \right), \tag{3.18}$$

and it can be easily checked that c_1, c_2 and c_2 are non-negative. We observe that $\tilde{\mathrm{H}}(v_{i,j}^n)$ can be written as

$$\tilde{\mathrm{H}}(v_{i,j}^n) = \max \left(r_{ij}^n(a) - \frac{1}{\Delta y} \left(v_{i,j}^n |f_j^n(a)| + v_{i,j-1}^n f_j^n(a)^+ - v_{i,j+1}^n f_j^n(a)^- \right) \right),$$

hence

$$\begin{aligned} \tilde{\mathrm{H}}(v_{i,j}^n) - \tilde{\mathrm{H}}(u_{i,j}^n) \geq & \min \left(\tilde{\mathrm{H}}(v_{i,j}^n) - \tilde{\mathrm{H}}(u_{i,j}^n) \right) \\ \geq & \frac{1}{\Delta y} \min \Big(- (v_{i,j}^n - u_{i,j}^n)|f_j^n(a)| + (v_{i,j}^n - u_{i,j-1}^n) f_j^n(a)^+ \\ & - (v_{i,j+1}^n - u_{i,j+1}^n) f_j^n(a)^- \Big). \end{aligned} \tag{3.19}$$

From the previous inequality and the fact that $v^n \geq u^n$, we get

$$\tilde{\mathrm{H}}(v_{i,j}^n) - \tilde{\mathrm{H}}(u_{i,j}^n) \geq -\frac{1}{\Delta y}(v_{i,j}^n - u_{i,j}^n)|f_j^n|_\infty, \tag{3.20}$$

with

$$|f_j^n|_\infty := \max_a(f_j^n(a)).$$

From equation (3.17) and inequality (3.20), it follows that

$$\begin{aligned} v_{i,j}^{n+1} - u_{i,j}^{n+1} =& \, c_1 \left(v_{i,j}^n - u_{i,j}^n + c_2(v_{i+1,j}^{n+1} - u_{i+1,j}^{n+1}) + c_3(v_{i-1,j}^{n+1} - u_{i-1,j}^{n+1})\right) \\ &+ c_1 \Delta\tau(\mathrm{H}(v_{i,j}^n) - \mathrm{H}(v_{i,j}^n)) \\ \geq& \, c_1 \left(v_{i,j}^n - u_{i,j}^n + c_2(v_{i+1,j}^{n+1} - u_{i+1,j}^{n+1}) + c_3(v_{i-1,j}^{n+1} - u_{i-1,j}^{n+1})\right) \\ &- c_1 \frac{\Delta\tau}{\Delta y}(v_{i,j}^n - u_{i,j}^n)\|f_j^n\|_\infty \\ =& \, c_1 \left(c_2(v_{i+1,j}^{n+1} - u_{i+1,j}^{n+1}) + c_3(v_{i-1,j}^{n+1} - u_{i-1,j}^{n+1})\right) \\ &+ c_1(v_{i,j}^n - u_{i,j}^n)(1 - \frac{\Delta\tau}{\Delta y}\|f_j^n\|_\infty), \end{aligned}$$

so provided that

$$\Delta\tau \|f_j^n\|_\infty \leq \Delta y, \tag{3.21}$$

the following inequality holds

$$v_{i,j}^{n+1} - u_{i,j}^{n+1} \geq c_1 c_2(v_{i+1,j}^{n+1} - u_{i+1,j}^{n+1}) + c_1 c_3(v_{i-1,j}^{n+1} - u_{i-1,j}^{n+1}).$$

The previous inequality together with (3.14) implies that

$$v_{i^*,j^*}^{n+1} - u_{i^*,j^*}^{n+1} \geq c_1 c_2(v_{i^*,j^*}^{n+1} - u_{i^*,j^*}^{n+1}) + c_1 c_3(v_{i^*,j^*}^{n+1} - u_{i^*,j^*}^{n+1}),$$

which can be rearranged into

$$(1 - c_1(c_2 + c_3))v_{i^*,j^*}^{n+1} \geq (1 - c_1(c_2 + c_3))u_{i^*,j^*}^{n+1}.$$

From the definition of c_1, c_2 and c_3, given by (3.18), it follows that

$$(1 - c_1(c_2 + c_3)) > 0,$$

this concludes the proof.

3.5 Stability

Let v represent a solution of (3.7). Provided that the CFL-condition (3.13) is satisfied, there exists $C > 0$, independent of $\Delta x, \Delta y$ and $\Delta\tau$, such that

$$\|v^n\|_\infty \leq C\left(\|v^0\|_\infty + \|r\|_\infty\right), \qquad \forall n > 0. \tag{3.22}$$

The proof is by induction. Let $(i,j) = \arg\max_{i,j}(v^n_{i,j})$, equation (3.16) implies that

$$v^{n+1}_{i,j} \leq v^n_{i,j} + \Delta\tau\tilde{\mathrm{H}}(v^n_{i,j}).$$

By writing out the term $\tilde{\mathrm{H}}(v^n_{i,j})$ in the last inequality, we get

$$\begin{aligned}
v^{n+1}_{i,j} &\leq v^n_{i,j} + \Delta\tau\max_a\left(r^n_{ij}(a) - \frac{1}{\Delta y}\left(v^n_{i,j}|f^n_j(a)| + v^n_{i,j-1}f^n_j(a)^+ - v^n_{i,j+1}f^n_j(a)^-\right)\right)\\
&= v^n_{i,j} + \Delta\tau r^n_{ij}(a^*) - \frac{\Delta\tau}{\Delta y}\left(v^n_{i,j}|f^n_j(a^*)| + v^n_{i,j-1}f^n_j(a^*)^+ - v^n_{i,j+1}f^n_j(a^*)^-\right)\\
&= v^n_{i,j}(1 - \frac{\Delta\tau}{\Delta y}|f^n_j(a^*)|) + \Delta\tau\left(v^n_{i,j-1}f^n_j(a^*)^+ - v^n_{i,j+1}f^n_j(a^*)^-\right) + \Delta\tau r^n_{ij}(a^*).
\end{aligned} \tag{3.23}$$

Suppose that (3.13) is satisfied, since $f^n_j(a^*)^+ \geq 0$, $-f^n_j(a^*)^- \geq 0$ and we have chosen i, j such that $v^{n+1}_{i,j} = \max_{i,j} v^{n+1}_{i,j}$, inequality (3.23) implies that

$$\max_{i,j}(v^{n+1}_{i,j}) \leq \Delta\tau\left(\|f^n\|_\infty \max_{i,j}(v^n_{i,j}) + \|r^n\|_\infty\right).$$

By choosing $(i,j) = \arg\min_{i,j}(v^n_{i,j})$ it follows by similar arguments that

$$\min_{i,j}(v^{n+1}_{i,j}) \geq \Delta\tau\left(\|f^n\|_\infty \min_{i,j}(v^n_{i,j}) + \|r^n\|_\infty\right).$$

The last two inequalities together implies that

$$\begin{aligned}
\|v^{n+1}\|_\infty &\leq \Delta\tau\left(\|f^n\|_\infty|v^n|_\infty + \|r^n\|_\infty\right)\\
&\leq \Delta y|v^n|_\infty + \Delta\tau\|r^n\|_\infty,
\end{aligned}$$

where we have used (3.13) to obtain the last inequality. From the last inequality it follows that

$$\begin{aligned}
\|v^{n+1}\|_\infty &\leq \Delta y(\Delta y\|v^{n-1}\|_\infty + \Delta\tau\|r^{n-1}\|_\infty) + \Delta\tau\|\tau^n\|_\infty\\
&= \Delta y^2\|v^{n-1}\|_\infty + \Delta y\Delta\tau\|r^{n-1}\|_\infty + \Delta\tau\|r^n\|_\infty\\
&\leq \Delta y^{k+1}\|v^0\|_\infty + \sum_{k=0}^{n}\Delta y^k\Delta\tau\|r^{n-k}\|,
\end{aligned}$$

since $\Delta y \leq y_{\max} := 1$, the last inequality implies that

$$\|v^{n+1}\|_\infty \leq C\left(\|v^0\|_\infty + \|r\|_\infty\right),$$

with $C = \max(T, 1)$. This concludes the proof.

3.6 Consistency

We prove that the scheme given by (3.7) is consistent, provided that the exact solution of (3.6), denoted $v(x, y, \tau)$, is sufficiently smooth.

Suppose that v is second order differentiable in x, first order differentiable in y and τ, then Taylor's theorem implies that

$$\begin{aligned}
&\frac{\partial v}{\partial \tau}(x_i, y_j, \tau_n) = \frac{v_{i,j}^n - v_{i,j}^{n-1}}{\Delta\tau} + \mathcal{O}(\Delta\tau), && \frac{\partial^2 v}{\partial x^2}(x_i, y_j, \tau_n) = \mathrm{D}_x^+(\mathrm{D}_x^- v_{i,j}^n) + \mathcal{O}(\Delta x^2),\\
&\frac{\partial v}{\partial x}(x_i, y_j, \tau_n) = \mathrm{D}_x^- v_{i,j}^n + \mathcal{O}(\Delta x), && \frac{\partial v}{\partial x}(x_i, y_j, \tau_n) = \mathrm{D}_x^+ v_{i,j}^n + \mathcal{O}(\Delta x),\\
&\frac{\partial v}{\partial y}(x_i, y_j, \tau_n) = \mathrm{D}_y^+ v_{i,j}^n + \mathcal{O}(\Delta y), && \frac{\partial v}{\partial y}(x_i, y_j, \tau_n) = \mathrm{D}_y^- v_{i,j}^n + \mathcal{O}(\Delta y).
\end{aligned} \tag{3.24}$$

Let the functional $\mathrm{H}(\cdot)$ be defined such that

$$\mathrm{H}(v)(x, y, \tau) = \sup_{a \in A(y)} \left(r(x, y, a, \tau) - f(y, a, \tau) \frac{\partial v}{\partial y}(x, y, t) \right).$$

Because v satisfies the equation

$$\left(\frac{\partial v}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} - \mu \frac{\partial v}{\partial x} + \rho v \right)(x, y, \tau) - \mathrm{H}(v)(x, y, \tau) = 0,$$

there holds

$$\int_{\tau_n}^{\tau_{n+1}} \left(\frac{\partial v}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} - \mu \frac{\partial v}{\partial x} + \rho v \right)(x, y, \tau)\, d\tau - \int_{\tau_n}^{\tau_{n+1}} \mathrm{H}(v)(x, y, \tau)\, d\tau = 0.$$

By approximating the two integrals in the last equation with the right end point rule and the left endpoint rule, respectively, we get

$$\begin{aligned}
v(x, y, \tau_{n+1}) - v(x, y, \tau_{n+1}) - \Delta\tau \left(\frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right)(x, y, \tau_{n+1})&\\
- \Delta\tau \mathrm{H}(v)(x, y, \tau_n) = \mathcal{O}(\Delta\tau^2)&,
\end{aligned}$$

which can be rewritten as

$$\begin{aligned}
\frac{v(x, y, \tau_{n+1}) - v(x, y, \tau_{n+1})}{\Delta\tau} - \left(\frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right)(x, y, \tau_{n+1})&\\
- \mathrm{H}(v)(x, y, \tau_n) = \mathcal{O}(\Delta\tau)&.
\end{aligned}$$

From the previous equation and (3.24), it follows that

$$
\begin{aligned}
e_{i,j}^{n+1} :=& \Bigg(\frac{v(x_i, y_j, \tau_{n+1}) - v(x_i, y_j, \tau_{n+1})}{\Delta\tau} - \left(\frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} + \mu \frac{\partial v}{\partial x} - \rho v \right) (x_i, y_j, \tau_{n+1}) \\
& \quad - \mathrm{H}(v)(x_i, y_j, \tau_n) \Bigg) - \Bigg(\frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta\tau} - \sigma_i \mathrm{D}_x^+ (\mathrm{D}_x^- v_{i,i}^{n+1}) - \mu_i^+ \mathrm{D}_x^+ v_{i,j}^{n+1} \\
& \quad - \mu_i^- \mathrm{D}_x^- v_{i,j}^{n+1} - \tilde{\mathrm{H}}(v_{i,j}^n) \Bigg) \\
=& \tilde{\mathrm{H}}(v_{i,j}^n) - \mathrm{H}(v)(x_i, y_j, \tau_n) + \mathcal{O}(\Delta x + \Delta\tau).
\end{aligned}
$$

We have

$$
\begin{aligned}
\left| \tilde{\mathrm{H}}(v_{i,j}^n) - \mathrm{H}(v)(x_i, y_j, \tau_n) \right| \leq& \max_{a \in A_j} \left| \tilde{\mathrm{H}}(v_{i,j}^n) - \mathrm{H}(v)(x_i, y_j, \tau_n) H \right| \\
\leq& \max_{a \in A_j} \Bigg| - f_j^n(a)^+ \mathrm{D}^- v_{i,j}^n - f_j^n(a)^- \mathrm{D}^+ v_{i,j}^n \\
& \quad + f_j^n(a) \frac{\partial v}{\partial y}(x_i, y_j, \tau_n) \Bigg| \\
=& \max_{a \in A_j} \left(|f_j^n(a)| \mathcal{O}(\Delta y) \right) \\
=& \mathcal{O}(\Delta y).
\end{aligned}
$$

This concludes the proof.

3.7 Implementation

The numerical scheme given by (3.7) can be rewritten in matrix form as follows

$$
\frac{1}{\Delta\tau}(\mathbf{v}^{n+1} - \mathbf{v}^n) + \mathrm{A}\mathbf{v}^{n+1} = \mathbf{r}^n + \mathrm{C}^n \mathbf{v}^n, \qquad n = 0, \dots, N-1, \tag{3.25}
$$

where $\mathbf{v}^0$ is a given initial solution. Before we define the vectors $\mathbf{v}^{n+1}, \mathbf{v}^n$ and $\mathbf{r}^n$ and the matrices A and C^n, appearing in the last equation, let

$$
m = I \cdot J,
$$

and define the function $\xi(x, y, \tau)$, such that

$$
\xi_{i,j}^n = \arg\sup \left(r_{ij}^n(a) - f_j^n(a)^+ \mathrm{D}^- v_{i,j}^n - f_j^n(a)^- \mathrm{D}_y^+ v_{i,j}^n \right). \tag{3.26}
$$

In equation (3.25) we have introduced the solution vector $\mathbf{v}^n \in \mathbb{R}^m$, for $n = 0, \dots, N$, given as

$$
\mathbf{v}^n = (v_{1,1}^n, \dots v_{I,1}^n, v_{1,2}^n, \dots v_{I,2}^n, \dots, v_{1,J}^n \dots v_{I,J}^n)^\top,
$$

the vector $\mathbf{r}^n \in \mathbb{R}^m$, for $n = 0, \dots, N$, given as

$$
\mathbf{r}^n = (r_{1,1}^{*,n}, \dots r_{I,1}^{*,n}, r_{1,2}^{*,n}, \dots r_{I,2}^{*,n}, \dots, r_{1,J}^{*,n} \dots r_{I,J}^{*,n})^\top, \quad r_{i,j}^{*,n} = r(x_i, y_j, \xi_{i,j}^n, \tau_n),
$$

the matrix $\mathrm{A} \in \mathbb{R}^{m \times m}$, defined as

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_1 & 0 & \dots & 0 \\ 0 & \mathrm{A}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{A}_J \end{bmatrix}, \tag{3.27}$$

where each block $\mathrm{A}_j \in \mathbb{R}^{I \times I}$, for $j = 1, \dots, J$, is given as

$$\mathrm{A}_j = -\frac{1}{\Delta x^2} \begin{bmatrix} -2\sigma_1 & \sigma_1 & & & \\ \sigma_2 & -2\sigma_2 & \sigma_2 & & \\ & \sigma_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \sigma_{I-1} \\ & & & \sigma_I & -2\sigma_I \end{bmatrix}$$

$$-\frac{1}{\Delta x} \begin{bmatrix} -\mu_1 & \mu_1 & & & & \\ -\mu_2^- & (\mu_2^- - \mu_2^+) & \mu_2^+ & & & \\ & -\mu_3^- & (\mu_3^- - \mu_3^+) & \mu_3^+ & & \\ & & & \ddots & \ddots & \\ & & & \ddots & \ddots & \mu_{I-1}^+ \\ & & & & -\mu_I & \mu_I \end{bmatrix}, \tag{3.28}$$

and the matrix $\mathrm{C}^n \in \mathbb{R}^{m \times m}$, for $n = 1, \dots, N$, defined such that

$$\mathrm{C}^n = \begin{bmatrix} \mathrm{C}_1^{n\top} & 0 & \dots & 0 \\ 0 & \mathrm{C}_2^{n\top} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathrm{C}_I^{n\top} \end{bmatrix},$$

where each block $\mathrm{C}_i^n \in \mathbb{R}^{J \times J}$, for $i = 1, \dots, I$, is given as

$$\mathrm{C}_i^n = \frac{1}{\Delta y} \begin{bmatrix} -f_1^{*,n} & f_1^{*,n} & & & & \\ -f_2^{*,n,+} & (f_2^{*,n,+} - f_2^{*,n,-}) & f_2^{*,n,-} & & & \\ & -f_3^{*,n,+} & (f_3^{*,n,+} - f_3^{*,n,-}) & f_2^{*,n,-} & & \\ & & & \ddots & \ddots & \\ & & & \ddots & \ddots & f_{J-1}^{n,*,-} \\ & & & & -f_J^{n,*} & f_J^{n,*} \end{bmatrix},$$

with

$$f_j^{*,n,+} = f(x_i, y_j, \tau_n, \xi_{i,j}^n)^+, \qquad f_j^{*,n,-} = f(x_i, y_j, \tau_n, \xi_{i,j}^n)^-.$$

In algorithm (1) we have tried to summarize the main flow of a MATLAB program that computes the solution of (3.25), with initial solution equal to zero for simplicity. We introduce the vector $\xi^n \in \mathbb{R}^m$, for $n = 0, \ldots, N$, given as

$$\xi^n = (\xi^n_{1,1}, \ldots \xi^n_{I,1}, \xi^n_{1,2}, \ldots \xi^n_{I,2}, \ldots, \xi^n_{1,J} \ldots \xi^n_{I,J})^\top,$$

and we let I_m represent the $m \times m$ identity matrix.

Algorithm 1 Algorithm for computing the solution of scheme (3.25)

```
$\mathbf{v}^0 \leftarrow \mathbf{0}$
for $n = 0, \ldots, N-1$ do
    $\xi^n \leftarrow$ ComputeFeedbackControl($\mathbf{v}^n$, $r$, $f$)
    $\mathbf{r}^n \leftarrow$ AssembleLoadVector($\xi^n$)
    $\mathrm{C}^n \leftarrow$ AssembleConvectionMatrix($\xi^n$, $f$)
    $\mathbf{v}^{n+1} \leftarrow (\mathrm{I}_m + \Delta\tau\mathrm{A}) \backslash (\mathbf{v}^n + \Delta\tau(\mathrm{C}^n\mathbf{v}^n + \mathbf{r}^n))$
end for
```

Remark 3.5. *The procedure* ComputeFeedbackControl($\mathbf{v}^n, r, f$) *in algorithm 1 solves the optimization problem given by (3.26), for $i = 1, \ldots, I$ and $j = 1, \ldots, J$, and returns the result as an $m \times 1$ array.*

Remark 3.6. *We have assumed for simplicity that the coefficients μ and σ needed to assemble the matrix* A*, see (3.28), does not depend on τ. However, if σ and/or μ should depend on τ, we simply add a line of code within the for-loop of algorithm and replace* A *with* A^{n+1} *at each time step, with* A^{n+1} *defined as in (3.27)- (3.28) with σ and μ evaluated at $\tau = \tau_{n+1}$.*

Chapter 4

The Finite Element method (1D)

We introduce the finite element method for a transient convection diffusion reaction problem in one space dimension. The method will be applied for solving (2.7) in the price direction (x-direction) in chapter 5. For the presentation of the finite element method it will be sufficient to consider a test problem of the form

$$\frac{\partial v}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} - \mu \frac{\partial v}{\partial x} = r, \qquad \text{in } (0,1) \times \mathbb{T}, \tag{4.1}$$

with boundary and initial conditions given as

$$\begin{aligned} \sigma \frac{\partial v}{\partial x} &= 0, && \text{on } \{0,1\}, && (4.2)\\ v &= 0, && \text{for } t = 0, && (4.3) \end{aligned}$$

with $\mu, \sigma, r \in L^2([0,1] \times \mathbb{T})$ being square integrable functions in x and τ.

4.1 Weak Formulation

Equation (4.1) can be rewritten as

$$\frac{\partial v}{\partial \tau} - \frac{1}{2}\frac{\partial}{\partial x}\left(\sigma^2 \frac{\partial v}{\partial x}\right) - \tilde{\mu} \frac{\partial v}{\partial x} = r, \qquad \text{in } (0,1) \times \mathbb{T}, \tag{4.4}$$

with

$$\tilde{\mu} = \mu - \sigma \frac{\partial \sigma}{\partial x}. \tag{4.5}$$

Let $w : (0,1) \to \mathbb{R}$ represent a *test function*, residing in some space V to be defined. Upon multiplying equation (4.4) by w and integrating the resulting equation with respect to x over the domain $(0,1)$, we get

$$\int_0^1 \left(\frac{\partial v}{\partial \tau} - \frac{1}{2} \frac{\partial}{\partial x} \left(\sigma^2 \frac{\partial v}{\partial x} \right) - \tilde{\mu} \frac{\partial v}{\partial x} \right) w \, dx = \int_0^1 rw \, dx.$$

Via integration by parts, the previous equation becomes

$$\int_0^1 \left(\frac{\partial v}{\partial \tau} w + \frac{1}{2} \sigma^2 \frac{\partial v}{\partial x} \frac{\partial w}{\partial x} - \tilde{\mu} \frac{\partial v}{\partial x} w \right) dx - \frac{1}{2} \Big|_0^1 \sigma^2 \frac{\partial v}{\partial x} w = \int_0^1 rw \, dx$$

The boundary condition (4.2) implies that the term $\big|_0^1 \sigma^2 \frac{\partial v}{\partial x} w = 0$ in the last equation. Let

$$V = \mathrm{H}^1(0,1) := \{ w : (0,1) \to \mathbb{R} : w \in \mathrm{L}^2(0,1) : w_x \in \mathrm{L}^2(0,1) \},$$

with w_x representing the derivative of w in the sense of distributions [10, p. 18]. The *weak formulation* of (4.1), see [10, p. 120], is stated

$$\forall t \in \mathbb{T} \text{ find } v(\tau) \in V : \int_0^1 \frac{\partial v}{\partial \tau} w \, dx + a(v,w) = R(w), \quad \forall w \in V, \tag{4.6}$$

in which the *bilinear form* $a(\cdot,\cdot)$ and the linear functional $R(\cdot)$, are given as

$$\begin{aligned} a(v,w) &= \int_0^1 \frac{1}{2} \sigma^2 \frac{\partial v}{\partial x} \frac{\partial w}{\partial x} \, dx - \int_0^1 \tilde{\mu} \frac{\partial v}{\partial x} w \, dx, & \forall v, w \in V, \\ R(w) &= \int_0^1 rw \, dx, & \forall w \in V. \end{aligned}$$

4.2 Approximation

Roughly speaking, the idea of the finite element method is to replace the space V by a finite dimensional approximation $V_h \subset H^1$ and then find the best solution of problem (4.6) in the space V_h. The space V_h is typically a space of picewise polynomials. We will now proceed to define a family of such spaces.

Let N be a positive integer and let $(x_i)_{i=0}^N \subset (0,1)$ be a collection of points called *vertices* such that

$$0 = x_0 < x_1 < \cdots < x_N < x_{N+1} = 1.$$

Let $\mathcal{T}_h$ represent a collection of the subintervals $K_i = (x_i, x_{i+1})$, with the subscript h of $\mathcal{T}_h$ representing the positive real number such that

$$\max_i (x_{i+1} - x_i) = h.$$

The subintervals $K_i = (x_i, x_{i+1}) \in \mathcal{T}_h$ are called *elements*. We define the following family of spaces

$$X_h^r = \{ v_h \in C^0([0,1]) : v_h|_K \in \mathbb{P}_r(K) \, \forall K \in \mathcal{T}_h \}.$$

with $\mathbb{P}_r(K)$ representing the space of polynomials with degree less than or equal to r on $K \subset \hat{\Omega}$.

Let $N_h = \dim(V_h)$ and let $(\varphi_i)_{i=1}^{N_h}$ be a basis for V_h such that

$$\varphi_i(X_j) = \delta_{ij},\ i, j = 1, \ldots, N_h, \tag{4.7}$$

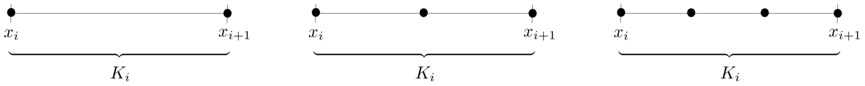
where the points $(X_j)_{j=1}^{N_h}$ are called *nodes*. Each basis function φ_i is uniquely determined by (4.7) if each element $K \in \mathcal{T}_h$ contains exactly $n_r = \dim(\mathbb{P}_r)$ nodes. For instance, in the case of linear elements, $(r = 1)$, we have

$$\varphi_i|_K \in \mathbb{P}_1(K),$$

which is equivalent to

$$\exists a_i, b_i \in \mathbb{R} : \varphi_i|_K = a_i x + b_i,$$

so we need $\dim(\mathbb{P}_1) = 2$ nodes in K to determine $\varphi_i|_K$ via (4.7). Consequently, for linear elements, the nodes coincides with the vertices $(x_i)_{i=1}^{N}$, which gives 2 nodes on each element. For higher order polynomial elements we need to add additional points, see figure 4.1.



(a) Linear element ($\mathbb{P}_1$). (b) Quadratic element ($\mathbb{P}_2$). (c) Cubic element ($\mathbb{P}_3$).

Figure 4.1: Node placement on a generic element $K_i = [x_i, x_{i+1}] \in \mathcal{T}_h$, for linear, quadratic and cubic elements. Nodes displayed as black dots.

For all $v_h \in V_h$ there exist coefficients $(v_i)_{i=1}^{N} \subset \mathbb{R}$, such that

$$v_h(x, \tau) = \sum_{i=1}^{N} v_i(\tau)\varphi_i(x). \tag{4.8}$$

Let $\mathbf{v}(\tau) = (v_1(\tau), v_2(\tau), \ldots, v_N(\tau))$. The approximation of problem (4.6) in the space $V_h = X_h^r$, is stated

$$\begin{aligned} &\forall \tau \in \mathbb{T} \text{ find } \mathbf{v}(\tau) \in \mathbb{R}^N : \\ &\sum_{i=1}^{N} \frac{\partial v_i}{\partial \tau} \int_0^1 \varphi_i \varphi_j + \sum_{i=1}^{N} v_i\, a(\varphi_i, \varphi_j) = R(\varphi_j), \quad \text{for } j = 1, 2, \ldots, N. \end{aligned} \tag{4.9}$$

Problem (4.9) is called *semi-discrete* as it is discretized in the variable x and continuous in τ.

4.3 Stabilization

It is well known that the standard finite element approximation (4.9) may be numerically unstable for convection dominated problems of the form (4.1) as numerical oscillations are produced . We have chosen to apply the edge stabilization method as described by Burman and Hansbo [4] and we shall further discuss the method in the two dimensional case in section 6.3, which is a more natural setting for the presentation of the method. In one space dimension [11], the edge stabilization method reduces to replacing the bilinear form $a(\cdot,\cdot)$ with $a_h(\cdot,\cdot)$ given as

$$a_h(v,w) = a(v,w) + s_h(v,w), \qquad \forall v,w \in V,$$

with

$$s_h(v,w) = \sum_{i=1}^{N_h} \gamma h^2 \left[\frac{\partial v}{\partial x}\right]_i \left[\frac{\partial w}{\partial x}\right]_i . \tag{4.10}$$

In the previous equation, the operator $[\,\cdot\,]_i$ evaluates the jump of a function across the vertice x_i, that is

$$\left[\frac{\partial v}{\partial x}\right]_i = \left.\frac{\partial v}{\partial x}\right|_{x_i^+} - \left.\frac{\partial v}{\partial x}\right|_{x_i^-},$$

and $\gamma \in (0,1)$ is a free parameter. With edge stabilization problem (4.9) becomes

$$\begin{aligned}&\forall \tau \in \mathbb{T} \text{ find } \mathbf{v}(\tau) \in \mathbb{R}^{N_h} : \\ &\sum_{i=1}^{N_h} \frac{\partial v_i}{\partial \tau} \int_0^1 \varphi_i \varphi_j + \sum_{i=1}^{N_h} v_i\, a(\varphi_i,\varphi_j) + \sum_{i=1}^{N_h} v_i\, s_h(\varphi_i,\varphi_j) = R(\varphi_j), \quad j = 1,\ldots,N_h.\end{aligned} \tag{4.11}$$

4.4 Matrix Formulation

Equation (4.11) can be rewritten in matrix from as follows

$$\mathrm{M}\mathbf{v}(\tau) + \mathrm{A}\mathbf{v}(\tau) = \mathbf{r}. \tag{4.12}$$

In the previous equation we have introduced the *mass matrix*

$$\mathrm{M} = [m_{i,j}], \qquad m_{i,j} = \int_0^1 \varphi_i \varphi_j, \tag{4.13}$$

the *stiffness matrix*

$$\mathrm{A} = [a_{i,j}], \qquad a_{i,j} = a_h(\varphi_i,\varphi_j), \tag{4.14}$$

and the *load vector*

$$\mathbf{r} = (r_1, r_2, \ldots, r_{N_h})^\top, \qquad r_j = R(\varphi_j). \tag{4.15}$$

nq	ξ_k	w_k
1-point rule	0	2
3-point rule	$(1/6, 1/6)$	$1/3$
	$(2/3, 1/6)$	$1/3$
	$(1/6, 2/3$	$1/3$
4-point rule	$(1/3, 1/3)$	$-27/48$
	$(1/5, 3/5)$	$25/48$
	$(1/5, 1/5)$	$25/48$
	$(3/5, 1/5)$	$25/48$

Table 4.1: Gauss quadrature rules for the reference interval $(-1, 1)$.

4.5 Implementation

In order to apply the finite element method, we need to write procedures for assembling the mass matrix M, the stiffness matrix A and the load vector $\mathbf{r}$, introduced in the previous section. We describe in the following subsections how this implemented in our code.

4.5.1 Computation of integrals

Consider an element $K = (x_i, x_{i+1}) \in \mathcal{T}$. The integral of a function f over K can be transformed to an integral over the *reference element* $\hat{K} = [-1, 1]$ via the linear mapping

$$F_{K_i}(\xi) = x_i + \frac{1}{2}(\xi + 1)(x_{i+1} - x_i).$$

That is, by taking $x = F_{K_i}(\xi)$, we obtain

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = \frac{|K_i|}{2} \int_{-1}^{1} \hat{f}(\xi)\,d\xi, \tag{4.16}$$

with

$$\hat{f}(\xi) := f(F_{K_i}(\xi)), \qquad |K_i| = x_{i+1} - x_i.$$

The last integral can be computed via some numerical quadrature rule, that is

$$\int_{-1}^{1} \hat{f}(\xi) \approx \sum_{q=1}^{N_q} w_k \hat{f}(\xi_k),$$

where $(w_1, \ldots, w_{N_q})$ and $(\xi_1, \ldots, \xi_{N_q})$ are the quadrature weights and the quadrature points respectively, see table 4.1.

4.5.2 Assembly of matrices

We shall now describe the algorithms for assembling the mass matrix M, the stiffness matrix A and the load vector $\mathbf{r}$. The entry M_{ij} of the mass matrix can be split into several terms corresponding to each interval $(x_i, x_{i+1}) \in \mathcal{T}_h$ as follows

$$m_{i,j} = \sum_{i=0}^{N} \int_{x_i}^{x_{i+1}} \varphi_i \varphi_j.$$

Clearly, only the terms where both φ_i and φ_j have support in K are non-zero in the previous equation, hence

$$m_{i,j} = \int_{x_{i-1}}^{x_i} \varphi_i \varphi_j \, dx + \int_{x_i}^{x_{i+1}} \varphi_i \varphi_j \, dx.$$

The integrals in the last equation can be computed analytically or via the method described in the previous subsection. In our implementation we have computed all integrals by numerical quadrature. Note that the N_q-point quadrature rules in table 4.1, ($N_q = 1, 3, 4$), are exact for $2N_q - 1$ polynomials. In algorithm 2 we have written a basic procedure to assemble the mass matrix by looping over all the elements in $\mathcal{T}_h$.

Algorithm 2 Assembly of the mass matrix

```
for $(x_i, x_{i+1}) \in \mathcal{T}_h$ do
    $m_{ii} \leftarrow m_{ii} + \int_{x_i}^{x_{i+1}} \varphi_i \varphi_i \, dx$
    $m_{i,i+1} \leftarrow m_{i,i+1} + \int_{x_i}^{x_{i+1}} \varphi_i \varphi_{i+1} \, dx$
    $m_{i+1,i+1} \leftarrow m_{i+1,i+1} + \int_{x_i}^{x_{i+1}} \varphi_{i+1} \varphi_{i+1} \, dx$
    $m_{i+1,i} \leftarrow m_{i+1,i} + \int_{x_i}^{x_{i+1}} \varphi_{i+1} \varphi_i \, dx$
end for
```

Chapter 5

A Semi-Lagrange Finite Element Method

We observe that equation (2.7) can be formulated equivalently as

$$\frac{\partial v}{\partial \tau} + \mathcal{L}v(x,y,\tau) + f(y,a^*,\tau)\frac{\partial v}{\partial y}(x,y,\tau) = r(x,y,a^*,\tau), \tag{5.1}$$

$$a^*(x,y,\tau) = \operatorname*{arg\,sup}_{a \in A(y)} \left(-f(y,a,\tau)\frac{\partial v}{\partial y}(x,y,\tau) + r(x,y,a,\tau) \right), \tag{5.2}$$

with the differential operator $\mathcal{L}$ defined such that

$$\mathcal{L}v(x,y,\tau) := -\frac{1}{2}\sigma^2(x)\frac{\partial^2 v}{\partial x^2}(x,y,\tau) - \mu(x)\frac{\partial v}{\partial x}(x,y,\tau) + \rho\, v(x,y,\tau).$$

Remark 5.1. *In general σ and μ can be functions of x and τ, however, to make the presentation of the method more readable we assume throughout this chapter that σ and μ only depend on x.*

We shall deal with equations (5.1) and (5.2) seperately in the following sections. In section 5.1 we describe how to solve the optimization problem (5.2), in the succseding section a numerical scheme is developed for equation (5.1). We introduce the set of discrete points $(\tau_n)_{n=0}^N$ in the τ-direction, defined such that

$$0 = \tau_0 \leq \tau_1 \leq \cdots \leq \tau_N = T.$$

For convenience we assume that the points are uniformly spaced such that

$$\Delta\tau = \tau_{n+1} - \tau_n \qquad\qquad n = 1, \ldots, N-1.$$

Similarly, let $(y_j)_{j=1}^J$ represent a set of uniformly spaced points in the y-direction, such that

$$0 = y_1 \leq y_2 \leq \cdots \leq y_J = 1, \qquad\qquad \Delta y = y_{j+1} - y_j.$$

The partition of the x-direction into a set of elements $\mathcal{T}_h = (K_i)_{i=1}^{N_h}$ is defined as in the previous section.

5.1 Linearization

In the following arguments we assume for simplicity that the function f only depends on a. We claim that the *feedback control* a^*, given by (5.2), can be consistently approximated as

$$a^*(x,y,\tau) = \operatorname*{arg\,sup}_{a\in A(y)} \Big(v(x, y - \Delta\tau f(a), \tau - \Delta\tau) + \Delta\tau\, r(x,y,a,\tau) \Big) + \mathcal{O}(\Delta\tau), \tag{5.3}$$

assuming that the value function v is sufficiently smooth.

Let $a \in A$, let $\tau' \in [0,T]$ and let $y_a(\tau)$ represent a path in the (y,τ)-plane such that

$$y_a(\tau) = y + (\tau - \tau')f(a).$$

Evaluating $a^*(\cdot)$ given by (5.2), at the point (x,y,τ'), we have

$$\begin{aligned} a^*(x,y,\tau') &= \operatorname*{arg\,sup}_{a\in A(y)} \left(-f(a)\frac{\partial v}{\partial y}(x,y,\tau') + r(x,a) \right) \\ &= \operatorname*{arg\,sup}_{a\in A(y)} \left(-\frac{\partial v}{\partial \tau}(x,y,\tau') - f(a)\frac{\partial v}{\partial y}(x,y,\tau') + r(x,a) \right) \\ &= \operatorname*{arg\,sup}_{a\in A(y)} \left(-\frac{dv}{d\tau}(x,y_a(\tau'),\tau') + r(x,a) \right). \end{aligned} \tag{5.4}$$

To obtain second of the previous equalities we have used the fact that $\frac{\partial v}{\partial \tau}(x,y,\tau')$ is independent of a. The succeeding equality follows from the identity

$$\frac{dv}{d\tau}(x,y_a(\tau'),\tau') = \frac{\partial v}{\partial \tau}(x,y_a(\tau'),\tau') + f(a)\frac{\partial v}{\partial y}(x,y_a(\tau'),\tau'), \tag{5.5}$$

and the fact that $y_a(\tau') = y$. If the value function v is first order continuous in y and τ, there holds

$$\frac{dv}{d\tau}(x,y,\tau') = \frac{v(x,y,\tau') - v(x,y-\Delta\tau f(a), \tau' - \Delta\tau)}{\Delta\tau} + \mathcal{O}(f(a)\Delta\tau + \Delta\tau).$$

By taking $\tau' = \tau_{n+1}$ and substituting the expression for $\frac{dv}{d\tau}(x,y,\tau')$ given by the previous equation into (5.4), we get

$$\begin{aligned} a^*(x,y,\tau_{n+1}) =& \operatorname*{arg\,sup}_{a\in A(y)} \Big(-\frac{v(x,y,\tau_{n+1}) - v(x,y-\Delta\tau f(a),\tau_n)}{\Delta\tau} + r(x,y,a,\tau) \\ & + \mathcal{O}(f(a)\Delta\tau + \Delta\tau) \Big) \\ =& \operatorname*{arg\,sup}_{a\in A(y)} \Big(v(x,y-\Delta\tau f(a),\tau_n) + \Delta\tau\, r(x,y,a,\tau) \Big) + |f|_\infty \mathcal{O}(\Delta\tau) \\ =& \operatorname*{arg\,sup}_{a\in A(y)} \Big(v(x,y-\Delta\tau f(a),\tau_n) + \Delta\tau\, r(x,y,a,\tau) \Big) + \mathcal{O}(\Delta\tau). \end{aligned}$$

The last equality follows from the fact that multiplying the expression inside the supremum with $\Delta\tau > 0$ does not change the optimal point of the supremum and the fact that $v(x, y, \tau_{n+1})$ is independent of a. In the previous equation we require that the point $y - \Delta\tau f(a)$ remain inside the domain. This condition can be enforced by replacing $A(y)$ with $\tilde{A}(y, \Delta\tau)$, given as

$$\tilde{A}(y, \Delta\tau) = \{a \in A(y) : 0 \leq y - \Delta\tau f(a) \leq 1\}. \tag{5.6}$$

We have $\tilde{A}(y, \Delta\tau) \subset A(y) \, \forall \Delta\tau > 0$ and $\tilde{A}(y, 0) = A(y)$.

5.2 Derivation of the numerical scheme

In this section we develop a numerical discretization of the equation

$$\frac{\partial v}{\partial \tau}(x, y, \tau) + f(y, a^*, \tau)\frac{\partial v}{\partial y}(x, y, \tau) + \mathcal{L}v(x, y, \tau) = r(x, y, a^*, \tau), \tag{5.7}$$

assuming that the feedback control a^* can be approximated trough solutions obtained from previous time steps, e.g. (5.3). The idea is to combine the terms $\frac{\partial v}{\partial \tau} - f\,\frac{\partial v}{\partial y}$ in equation (5.7) into a *material derivative* $\frac{dv}{d\tau}$, as shown in the previous section. The resulting equation can be solved by the finite element method introduced in chapter 4 and a suitable time integration method. Chen and Forsyth [19] present a similar method using a monotone finite difference method in the price direction. T.Ware [18] present a finite element, finite difference semi-Lagrangian method, however, in this method the semi-Lagrange method is used in the price direction.

Let $x \in [0, 1]$ and let $Y(\tau)$ represent a path in the (y, τ)-plane satisfying

$$\begin{aligned} \frac{dY}{d\tau}(\tau) &= f(a^*(x, Y(\tau), \tau)), \\ Y(\tau_{n+1}) &= y_j, \end{aligned} \tag{5.8}$$

where y_j represents a grid point. The chain rule states that

$$\frac{dv}{d\tau}(x, Y(\tau), \tau) = \frac{\partial v}{\partial \tau}(x, Y(\tau), \tau) + f(a^*)\frac{\partial v}{\partial y}(x, Y(\tau), \tau).$$

By substituting the identity given by the previous equation into (5.7), we get

$$\frac{dv}{d\tau}(x, Y(\tau), \tau) + \mathcal{L}v(x, Y(\tau), \tau) = r(x, y, a^*, \tau). \tag{5.9}$$

For all $n \geq 0$ let $Y_d(x; y_j, \tau_{n+1})$ represent the departure point at $\tau = \tau_n$ of the path given by (5.8), that is

$$Y_d(x; y_j, \tau_{n+1}) = y_j - \int_{\tau_n}^{\tau_{n+1}} f(a^*(x, Y(\tau), \tau))\, d\tau.$$

The value of $Y_d(x; y_j, \tau_{n+1})$ can be approximated via the right end point rule as follows

$$Y_d(x; y_j, \tau_{n+1}) \approx y_j - \Delta\tau f(\xi_j^{n+1}(x)), \tag{5.10}$$

where $\xi_j^{n+1}(x)$ represents an approximation of the feedback control $a^*(x, y_j, \tau_{n+1})$, depending only on solutions from previous time steps. More precisely, let v_h represent a numerical approximation of the value function, then via (5.3), we have

$$\xi_j^{n+1}(x) = \operatorname*{arg\,sup}_{a \in A(y_j)} \Big(v_h(x, y_j - \Delta\tau f(a), \tau_n) + \Delta\tau\, r(x, y_j, a, \tau_{n+1}) \Big).$$

The term $\frac{dv}{d\tau}$ in equation (5.9) can be approximated with the first order backward difference method as follows

$$\frac{dv}{d\tau}(x, Y(\tau), \tau)\Big|_{\tau=\tau_{n+1}} \approx \frac{v(x, y_j, \tau_{n+1}) - v(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)}{\Delta\tau}.$$

By substituting the approximation given by the previous equation into equation (5.9) evaluated at $\tau = \tau_{n+1}$, we get

$$\frac{v(x, y_j, \tau_{n+1}) - v(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)}{\Delta\tau} + \mathcal{L}v(x, y_j, \tau_{n+1}) = r(x, y_j, \xi_j^{n+1}(x), \tau_{n+1}). \tag{5.11}$$

We will now proceed with the discretizaztion of equation (5.11) in the price direction (x-direction), for which we will apply the finite element method introduced in the previous chapter. Upon multiplying (5.11) with a test function w and integrating the resulting equation with respect to x, we get

$$\int_0^1 \frac{v(x, y_j, \tau_{n+1}) - v(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)}{\Delta\tau} w(x)\, dx - \int_0^1 \mathcal{L}v(x, y_j, \tau_{n+1}) w(x)\, dx = \int_0^1 r(x, y_j, \xi_j^{n+1}(x), \tau_{n+1}) w(x)\, dx. \tag{5.12}$$

For notational convenience let $v_j^n(x) := v(x, y_j, \tau_{n+1})$ represent the continuous solution in x to be found for each pair $(j, n) \in \mathcal{I} = \{1, \ldots, J\} \times \{0, \ldots, N\}$. With $V = \mathrm{H}^1$, the weak formulation of (5.11) is stated: for all $(j, n) \in \mathcal{I}$ find $v_j^{n+1}(x) \in V$ such that

$$\int_0^1 \frac{v_j^{n+1}(x) - v(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)}{\Delta\tau} w(x)\, dx + a_h(v_j^{n+1}, w) = R(w; y_j, \tau_{n+1}), \quad \forall w \in V. \tag{5.13}$$

with

$$a_h(v, w) = \int_0^1 \left(\frac{1}{2}\sigma^2 \frac{\partial v}{\partial x}\frac{\partial w}{\partial x} - \tilde{\mu}\frac{\partial v}{\partial x} w + \rho v w \right) dx + s_h(v, w), \tag{5.14}$$

and

$$R(w; y_j, \tau_{n+1}) = \int_0^1 r(x, y_j, \xi_j^{n+1}(x), \tau_{n+1})\, w\, dx. \tag{5.15}$$

Remark 5.2. *Recall that the function $\tilde{\mu}$ appearing in equation (5.14) is given by (4.5) and that the stabilization term $s_h(v, w)$ is given by (4.10).*

As described in the previous chapter, an approximate solution of (5.13) is sought in the finite dimensional subspace $V_h = X_h^r \subset V$ with $\dim(V_h) = N_h$. Let $(\varphi_i)_{i=1}^{N_h}$ represent the Lagrangian basis for V_h. The numerical approximation of $v_j^n(x) := v(x, y_j, \tau_{n+1})$, denoted $v_h(x, y_j, \tau_{n+1})$ for each discrete point (y_j, τ_{n+1}) in the (y, τ)-plane, is expressed as a linear combination of the basis functions $(\varphi_i)_{i=1}^{N_h}$. That is,

$$v_h(x, y_j, \tau_{n+1}) = \sum_{i=1}^{N_h} v_{i,j}^{n+1} \varphi_i, \qquad \text{for } n = 0, \dots, N_\tau - 1,\ j = 1, \dots, J,$$

where the vector of unknown coefficients $\mathbf{v}_j^{n+1} = (v_{1,j}^{n+1}, \dots, v_{N_h,j}^{n+1})$, for each $(j, n) \in \mathcal{I}$, solves the following problem: find $\mathbf{v}_j^{n+1} \in \mathbb{R}^{N_h}$ such that

$$\begin{aligned} \sum_{i=1}^{N} v_{i,j}^{n+1} \int_0^1 \varphi_i \varphi_s\, dx + \Delta\tau \sum_{i=1}^{N} v_{i,j}^{n+1} a_h(\varphi_i, \varphi_s) &= \Delta\tau R(\varphi_s; y_j, \tau_{n+1}) \\ &\quad - \int_0^1 v_h(x, Y_d(x; y_j, \tau_{n+1}), \tau_n) \varphi_s\, dx, \quad \text{for } s = 1, \dots, N_h. \end{aligned} \tag{5.16}$$

The initial solution $v_h(x, y, \tau_0) = v_0$ is a given function. We have assumed that the boundary conditions satisfy (4.2).

Remark 5.3. *The point $Y_d(x; y_j, \tau_{n+1})$ equation (5.16) does in general not correspond to a grid point, so the value of $v_h(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)$, appearing on the right hand side of the previous equation, cannot be directly computed. We will therefore resort to an interpolation method to compute $v_h(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)$. In our implementation we have used linear interpolation.*

5.3 Implementation

Equation (5.16) can be written more compactly in matrix form. For each $(j, n) \in \{1, \dots, J\} \times \{1, \dots, N_\tau\}$, the vector of unknown coefficients $\mathbf{v}_j$ is given by

$$\mathrm{M}\mathbf{v}_j^{n+1} + \Delta\tau \mathrm{A}\mathbf{v}_j^{n+1} = \mathbf{L}_j^{n+1}, \tag{5.17}$$

where M and A represents the mass matrix and the stiffness matrix respectively, recall section 4.2. The vector $\mathbf{L}_j^{n+1} = ([\mathbf{L}_j^{n+1}]_1, [\mathbf{L}_j^{n+1}]_2, \dots, [\mathbf{L}_j^{n+1}]_{N_h})$ on the right hand of the last equation is defined such that

$$[\mathbf{L}_j^{n+1}]_s = \Delta\tau R(\varphi_s; y_j, \tau_{n+1}) - \int_0^1 v_h(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)\varphi_s \, dx, \quad s = 1, \ldots, N_h. \tag{5.18}$$

By writing out the definition for $R(\varphi_s; y_j, \tau_{n+1})$ in the previous equation, the explicit expression for $[\mathbf{L}_j^{n+1}]_s$ is

$$[\mathbf{L}_j^{n+1}]_s = \int_0^1 r(x, y_j, \xi_j^{n+1}(x), \tau_{n+1}) \, \varphi_s \, dx - \int_0^1 v_h(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)\varphi_s \, dx, \tag{5.19}$$

As indicated by our notation, the load vector $\mathbf{L}_j^{n+1}$ needs to be calculated once for each time step and each grid point in the y-direction. In order to approximate the integrals that defines $\mathbf{L}_j^{n+1}$, we need to evaluate $v_h(x, Y_d(x; y_j, \tau_{n+1}), \tau_n)$ and $r(x, y_j, \xi_j^{n+1}(x), \tau_{n+1})$ at some carefully chosen points. To maximize accuracy and efficiency, these points are chosen as the Gaussian quadrature points. In the following we let $(x_q)_{q=1}^{N_q}$ represent the collection of quadrature points for all the elements in the partition $\mathcal{T}_h$.

In algorithm 3 we have tried to summarize the main flow of the MATLAB program that computes the solution of (5.17). The first line of the most inner for-loop of algorithm 3 computes the approximation of the feedback control for each quadrature point x_q. This approximation is then used to calculate the departure point Y_d in the following line. In the next line of code the interpolated value of the solution at (x_q, Y_d, τ_n) is computed. This interpolated value is needed for the computation of the load vector, (via Gaussian quadrature), as explained above.

Algorithm 3 Algorithm for computing the solution of the Semi Lagrange Finite Element scheme (SLFE) given by (5.16)

```
M ← AssembleMassMatrix( )
A ← AssembleStiffnessMatrix($\sigma$, $\mu$, $\rho$)
$v_h^0(x,y) = v_0(x,y)$                                  ▷ initialize solution
for $n = 0, \ldots, N_\tau - 1$ do
    for $j = 1, \ldots, J$ do
        for $q = 1, \ldots, N_q$ do
            $\xi_j^n(x_q)$ ← ComputeFeedbackControl($\mathbf{v}^n$, $r$, $f$)
            $Y_d \leftarrow y_j - \Delta\tau f(\xi_j^{n+1}(x_q))$
            $v_h^n(x_q, Y_d)$ ← InterpolateSolution($v_h^n$, $x_q$, $Y_d$)
        end for
        $\mathbf{L}_j^{n+1}$ ← AssembleLoadVector($v_h^n$, $\xi_j^{n+1}$)
        $\mathbf{v}_j^{n+1} \leftarrow (\mathrm{M} + \Delta\tau \mathrm{A}) \backslash \mathbf{L}_j^{n+1}$
    end for
end for
```

Remark 5.4. *In algorithm 3 we have assumed that σ, μ are functions of x and that ρ is a contant. If for instance σ and μ depend on τ then the stifness matrix must be assembled once per time step.*

Chapter 6

A Finite Element Method (2D)

The idea of the semi Lagrange method introduced in the previous chapter works well when the equation of interest is only first order in y and τ. In this chapter we allow for the possibility of having diffusion also in the y-direction. Let

$$\mathbf{x} = (x, y)^\top,$$

we consider the equation

$$\frac{\partial v}{\partial \tau}(\mathbf{x}, \tau) - \operatorname{div}(\mathcal{D}\nabla v)(\mathbf{x}, t) + \rho v(\mathbf{x}, t) + \sup_{a \in A}[\mathbf{f}(\mathbf{x}, a, \tau) \cdot \nabla v(\mathbf{x}, \tau) + r(\mathbf{x}, a, \tau)] = 0. \tag{6.1}$$

In the previous equation we have introduced the diffusion matrix $\mathcal{D}$ and the vector valued function $\mathbf{f}$, given respectively as

$$\mathcal{D}(\mathbf{x}, \tau) = \frac{1}{2}\begin{pmatrix} \sigma_1(\mathbf{x}, \tau)^2 & 0 \\ 0 & \sigma_2(\mathbf{x}, \tau)^2 \end{pmatrix},$$

and

$$\mathbf{f}(\mathbf{x}, a, \tau) = (f_1(\mathbf{x}, a, \tau), f_2(\mathbf{x}, a, \tau))^\top,$$

where σ_1, σ_2, f_1 and f_2 are given square integrable functions.

Equation (2.21) relating to the natural gas storage model in section (2.3) can be recovered from (6.1) by choosing $\sigma_2 \equiv 0$, $f_1 = -\mu + \sigma$ and $f_2 = a + \lambda(a)$. On the other hand, equation (6.1) with $\sigma_2 > 0$ could be derived from the natural gas storage model by adding a *white noise* term to the leakage function $\lambda(\cdot)$, which might be reasonable if the leakage of gas depends on several unknown factors. Therefore, it might be reasonable to consider the gas inventory as a stochastic process of the form

$$\begin{aligned} dY(s) &= f_2(Y(s), \alpha(s), s)\, ds + \sigma_2(Y(s), s)\, dW(s), && t < s \leq T, \\ Y(t) &= y. \end{aligned}$$

Assuming that the price process is given as

$$\begin{aligned} dX(s) &= f_1(X(s), s)\, ds + \sigma_1(X(s), s)\, dW(s), && t < s \leq T, \\ X(t) &= x, \end{aligned}$$

then Ito's formula states that

$$dv(X(s), Y(s), s) = \left(\frac{\partial v}{\partial t} + \mathbf{f} \cdot \nabla v + \frac{1}{2} \mathrm{div}(\mathcal{D} \nabla v) \right) ds + \mathcal{D} \nabla v\, dW(s), \qquad \forall s[t, T].$$

and it can be shown by the same reasoning as in section (2.3) that the value function satisfies (6.1), assuming that the value function is sufficiently smooth.

As in the previous chapter, before we do anything to equation (6.1), we split (6.1) into to parts:

$$\frac{\partial v}{\partial \tau} - \mathrm{div}(\mathcal{D} \nabla v) + \rho v + \mathbf{f}(\mathbf{x}, a^*, \tau) \cdot \nabla v = r(\mathbf{x}, a^*, \tau), \tag{6.2}$$

$$a^*(\mathbf{x}, \tau) = \arg \sup_{a \in A(\mathbf{x})} \left(-\mathbf{f}(\mathbf{x}, a, \tau) \nabla v + r(\mathbf{x}, a^*, \tau) \right), \tag{6.3}$$

and we assume that the feedback control $a^*(\mathbf{x}, \tau)$ can be approximated using only solutions obtained from previous time steps. In sections 6.1 to 6.4 we develop a numerical scheme for (6.2). Problem (6.3) is treated in section 6.5. The boundary and initial conditions of (6.2) are assumed, for simplicity, to be given as

$$(\mathcal{D} \nabla v) \cdot \mathbf{n} = 0, \qquad \text{on } \partial\Omega, \tag{6.4}$$

$$v = 0, \qquad \text{for } \tau = 0. \tag{6.5}$$

Remark 6.1. *Consider the original problem given by equation (2.21), such that the diffusion matrix is given as*

$$\mathcal{D} = \begin{pmatrix} \sigma^2(x, \tau) & 0 \\ 0 & 0 \end{pmatrix}.$$

The boundary condition (6.4) is a relaxation of the condition $\sigma_{|x=1} = 0$, (and different from the condition $v_{xx} = 0$) discussed in section 2.4.1. However, we can choose to have the boundary $x = 1$ as far away as we like. That is, we choose $x_{\max}$ to be "far away" from the realistic price range and then scale the equation such that $x_{\max} = 1$. Consequently, the boundary condition (6.4) should not make much difference.

6.1 Weak formulation

Let V represent a space of test functions to be defined. Upon multiplying equation (6.2) with $w \in V$ and integrating the resulting equation over the spatial domain, we get

$$\int_\Omega \left(\frac{\partial v}{\partial \tau} - \operatorname{div}(\mathcal{D}\nabla v) + \rho v + \mathbf{f} \cdot \nabla v \right) w \, d\Omega = \int_\Omega rw \, d\Omega. \tag{6.6}$$

If $\mathbf{F} : \Omega \to \mathbb{R}^2$ is a sufficiently regular vector function, the divergence theorem states that

$$\int_\Omega \operatorname{div}(\mathbf{F}) \, d\Omega = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \, d\gamma. \tag{6.7}$$

Using the divergence theorem, the following relation can be obtained

$$-\int_\Omega \operatorname{div}(\mathcal{D}\nabla v) w \, d\Omega = \int_\Omega (\mathcal{D}\nabla v) \cdot \nabla w \, d\Omega - \int_{\partial\Omega} (\mathcal{D}\nabla v) \cdot \mathbf{n} \, d\gamma.$$

By substituting the identity given by the previous equation into (6.6), we get

$$\int_\Omega \left(\frac{\partial v}{\partial \tau} + \operatorname{div}(\mathcal{D}\nabla v) + \rho v + \mathbf{f} \cdot \nabla v \right) w \, d\Omega - \int_{\partial\Omega} (\mathcal{D}\nabla v) \cdot \mathbf{n} \, d\gamma = \int_\Omega rw \, d\Omega. \tag{6.8}$$

The boundary condition (6.4) implies that

$$\int_{\partial\Omega} (\mathcal{D}\nabla v) \cdot \mathbf{n} \, d\gamma = 0.$$

Let

$$V = \mathrm{H}^1(\Omega) := \{v : \Omega \to \mathbb{R} \text{ s.t. } v \in \mathrm{L}^2(\Omega), \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in \mathrm{L}^2\},$$

the weak formulation of (6.2) is stated

$$\forall \tau \in \mathbb{T} \text{ find } v \in V : \int_\Omega \frac{\partial v}{\partial \tau} w + b(v, w; a^*, \tau) = R(w; a^*, \tau), \qquad \forall w \in V, \tag{6.9}$$

with

$$b(v, w\,; a^*, \tau) = \int_\Omega (\nabla v \cdot \mathcal{D}(\mathbf{x}, \tau) \nabla w + (\mathbf{f}(\mathbf{x}, a^*(\mathbf{x}, \tau), \tau) \cdot \nabla v) w + \rho v w) \, dx, \tag{6.10}$$

$$R(w\,; a^*, \tau) = \int_\Omega r(\mathbf{x}, a^*(\mathbf{x}, \tau), \tau), \tau) w \, dx. \tag{6.11}$$

Remark 6.2. *For notational convenience we have omitted to write out that the functions v and w with respective gradients ∇v and ∇w are evaluated at $\mathbf{x}$ in the last two equations.*

6.2 Approximation

Let $\mathcal{T}_h$ represent a triangulation of the domain Ω, depending on the positive parameter h, with the following properties:

- $\mathcal{T}_h$ is a collection of triangles such that $\underset{K\in\mathcal{T}_h}{\cup} K = \bar{\Omega}$.
- Each element $K \in \mathcal{T}_h$ is made up of three straight lines such that K is closed; $K = \bar{K}$, and K has a non empty interior; $\overset{\circ}{K} \neq \emptyset$.
- If $K_1, K_2 \in \mathcal{T}_h$ are distinct elements then $\overset{\circ}{K}_1 \cap \overset{\circ}{K}_2 = \emptyset$.
- If $K_1, K_2 \in \mathcal{T}_h$ are distinct elements then $K_1 \cap K_2$ is either empty, a common vertex or a common side.
- $h = \max_{K\in\mathcal{T}_h} \operatorname{diam}(K)$, with $\operatorname{diam}(K) := \max(\|\mathbf{x}_1 - \mathbf{x}_2\| : \mathbf{x}_1, \mathbf{x}_2 \in K)$.

We define the following family of subspaces of V

$$X_h^r = \{v \in C^0(\Omega) : v|_K \in \mathbb{P}_r(K)\, \forall K \in \mathcal{T}_h\}.$$

with $\mathbb{P}_r(K)$ representing the space of polynomials with degree less than or equal to r on $K \subset \Omega$. Take $V_h = X_h^r$, problem (6.9) is approximated as

$$\forall t \in \mathbb{T} \text{ find } v_h(t) \in V_h : \int_\Omega \frac{\partial v_h}{\partial t} w_h + b(v_h, w_h) = R(w_h), \qquad \forall w_h \in V_h. \tag{6.12}$$

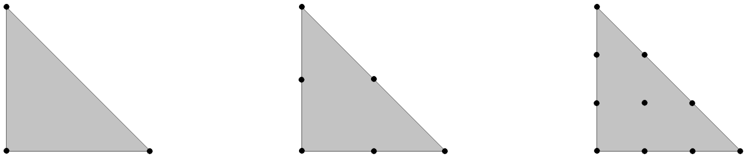
Let $N_h = \dim(V_h)$ and let $\{\varphi_i\}_{i=1}^{N_h}$ represent a basis for V_h, such that

$$\varphi_i(\mathbf{N}_j) = \delta_{ij}, \qquad\qquad i, j = 1, \ldots, N_h, \tag{6.13}$$

where the points $\{\mathbf{N}_j\}_{j=1}^{N_h} \subset \bar{\Omega}$ are reffered to as *nodes*. In order to ensure that the basis functions $\{\varphi_i\}_{i=1}^{N_h}$ are uniquely defined by equation (6.13), each element $K \in \mathcal{T}_h$ must contain exactly $n_r = \dim(\mathbb{P}_r)$ nodes. In general we have

$$n_r = (r+1)(r+2)/2.$$

For linear elements, $(r = 1)$, each node has $(1+1)(1+2)/2 = 3$ nodes which are typically located at the coners of each triangle, see figure 6.1.



(a) Linear Element ($\mathbb{P}_1$) (b) Quadratic Element ($\mathbb{P}_2$) (c) Cubic Element ($\mathbb{P}_3$)

Figure 6.1: $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ triangular elements with nodes (degrees of freedom) displayed in black.

Let the time dependent coefficients $(v_1(\tau), v_2(\tau), \ldots, v_{N_h}(\tau)) \in \mathbb{R}^{N_h}$ be such that

$$v_h(\mathbf{x}, \tau) = \sum_{i=1}^{N_h} v_i(\tau)\varphi_i(\mathbf{x}) \qquad \forall \tau \in \mathbb{T}. \tag{6.14}$$

By choosing $w_h = \varphi_j$ in (6.12) we see that $\mathbf{v}(\tau) = (v_1(\tau), v_2(\tau), \ldots, v_{N_h}(\tau))$ is the solution of the following problem:

$$\begin{gathered} \forall \tau \in \mathbb{T} \text{ find } \mathbf{v}(\tau) = (v_1(\tau), v_2(\tau), \ldots, v_{N_h}(\tau)) \in \mathbb{R}^{N_h} : \\ \sum_{i=1}^{N_h} \frac{\partial v_i(\tau)}{\partial \tau} \int_\Omega \varphi_i \varphi_j + \sum_{i=1}^{N_h} v_i(\tau) b(\varphi_i, \varphi_j; a^*, \tau) = R(\varphi_j; a^*, \tau), \quad j = 1, \ldots, N_h. \end{gathered} \tag{6.15}$$

6.3 Stabilization

It is well known that convection dominated equations, such as (6.2), may results in numerically oscillating solutions when solved by the standard finite element method, [4]. Many stabilization methods have been proposed to cope with this problem, see [1]. We have chosen to apply the *edge stabilization* method described in [4], also known as the continuous interior penalty method. This stabilization method works well with transient problems as it does depend on time derivatives or source terms, hence the resulting semi-discretization can be solved using standard finite difference techniques [3]. The bilinear form $b(\cdot, \cdot; a^*, \tau)$ is replaced by $b_h(\cdot, \cdot; a^*, \tau)$ given as

$$b_h(v, w; a^*, \tau) = b(v, w, a^*, \tau) + s_h(v, w), \qquad \forall v, w \in V_h,$$

with

$$s_h(v, w) = \sum_{E \in \mathcal{E}_h} \int_E \gamma h^2 \big[\nabla v \cdot \mathbf{n}\big]_E \big[\nabla w \cdot \mathbf{n}\big]_E, \qquad \forall v, w \in V_h.$$

In the previous equation $\mathcal{E}_h$ represents the collection of edges in $\mathcal{T}_h$, the vector $\mathbf{n}$ is the unit normal to E (with arbitrary sign) and the operator $[\,\cdot\,]_E$ is defined as the jump across the edge E

$$[\nabla v \cdot \mathbf{n}]_E = \nabla v \cdot \mathbf{n}\big|_{E^+} - \nabla v \cdot \mathbf{n}\big|_{E^-}.$$

The edge stabilization method introduces extra diffusion where the gradient of the numerical solution has high jumps and vanishes in areas where the solution is sufficiently smooth, that is if $[\nabla v \cdot \mathbf{n}]_E = 0$. We observe that if $v \in \mathrm{H}^2$, such that v has continuous first derivatives almost everywhere, then

$$s_h(v, w) = 0, \qquad \forall w \in V.$$

With edge stabilization problem (6.15) becomes

$$\begin{aligned}&\forall \tau \in \mathbb{T} \text{ find } \mathbf{v}(\tau) \in \mathbb{R}^N :\\ &\sum_{i=1}^{N} \frac{\partial v_i}{\partial \tau}(\tau) \int_0^1 \varphi_i \varphi_j + \sum_{i=1}^{N} v_i(\tau)\, b_h(\varphi_i, \varphi_j; a^*, \tau) = R(\varphi_j; a^*, \tau), \text{ for } j = 1, \ldots, N.\end{aligned} \tag{6.16}$$

6.4 Time Discretization

To obtain a fully discretized scheme it remains to discretize (6.16) in time. If we apply the backward Euler method, we get

$$\sum_{i=1}^{N} \frac{v_i^{n+1} - v_i^n}{\Delta \tau} \int_0^1 \varphi_i \varphi_j + \sum_{i=1}^{N} v_i(\tau)\, b_h(\varphi_i, \varphi_j; a^*, \tau^{n+1}) = R(\varphi_j; a^*, \tau^{n+1})$$

Let $\mathbf{v}^n := \mathbf{v}(\tau_n)$, then $\mathbf{v}^n$ for $n = 1, \ldots, N_\tau$ solves the following linear system

$$\frac{1}{\Delta \tau} \mathrm{M}(\mathbf{v}^{n+1} - \mathbf{v}^n) + \mathrm{A}^{n+1} \mathbf{v}^{n+1} = \mathbf{r}^{n+1}, \tag{6.17}$$

in which we have introduced the *mass matrix*

$$\mathrm{M} = [m_{i,j}], \qquad m_{i,j} = \int_\Omega \varphi_i \varphi_j,$$

the *stiffness matrix*

$$\mathrm{A}^{n+1} = [b_{i,j}^{n+1}], \qquad b_{i,j}^{n+1} = b_h(\varphi_i, \varphi_j, a^*|_{\tau=\tau_{n+1}}, \tau_{n+1}),$$

and the *load vector*

$$\mathbf{r}^{n+1} = (r_1^{n+1}, r_2^{n+1}, \ldots, r_{N_h}^{n+1})^\top, \qquad r_j^{n+1} = R(\varphi_j, a^*|_{\tau=\tau_{n+1}}, \tau_{n+1}).$$

We observe that the stiffness matrix A^{n+1} and the load vector $\mathbf{r}^{n+1}$ are dependent on τ_{n+1} and $a^*(\mathbf{x}, \tau_{n+1})$, therefore it must be assembled once per time step.

6.5 Solving the Optimization problem

In the following arguments we assume for simplicity that $\mathbf{f}$ only depends on a. To approximate the feedback control $a^*(\mathbf{x}, \tau)$, one possibility is to use the linearization technique described in section 5.1. That is,

$$a^*(\mathbf{x}, \tau_{n+1}) = \arg\sup_{a \in A(\mathbf{x})} \Big(v_h(\mathbf{x} - \Delta\tau\, \mathbf{f}(a) + \Delta\tau\, r(\mathbf{x}, a, \tau_{n+1}) \Big) + \mathcal{O}(\Delta \tau). \tag{6.18}$$

Remark 6.3. *Note that since the finite element approximation is continuous in* $\mathbf{x}$*, the expression* $v_h(\mathbf{x} - \Delta\tau\, \mathbf{f}(a)$ *in the previous equation makes sense even if the point* $\mathbf{x} - \Delta\tau \mathbf{f}(a)$ *does not coincide with a node coordinate.*

In the previous equation we require, as in section 5.1, that

$$\mathbf{x} - \Delta\tau\, \mathbf{f}(a) \in \Omega, \tag{6.19}$$

in other words the point $\mathbf{x} - \Delta\tau\, \mathbf{f}(a)$ cannot go outside the domain. This condition is enforced by replacing $A(\mathbf{x})$ with $\tilde{A}(\mathbf{x}, \Delta\tau)$, given as

$$\tilde{A}(\mathbf{x}, \Delta\tau) = \{a \in A(\mathbf{x}) : \mathbf{x} - \Delta\tau\, \mathbf{f}(a) \in \Omega\}. \tag{6.20}$$

In the preceding sections of this chapter we tried to emphasis in our notation that $b(\cdot\,,\cdot\,; a^*, \tau)$, given as

$$b(\varphi_i, \varphi_j; a^*, \tau) = \int_\Omega (\nabla\varphi_i \cdot \mathcal{D}\nabla\varphi_j + (\mathbf{f}(a^*) \cdot \nabla\varphi_i)\varphi_j + \rho\varphi_i\varphi_j)\, d\mathbf{x}, \\ \forall\varphi_i, \varphi_j \in V_h, \tag{6.21}$$

and $R(\cdot\,; a^*, \tau)$, given as

$$R(\varphi_j; a^*, \tau) = \int_\Omega r(\mathbf{x}, a^*(\mathbf{x}, \tau), \tau)\varphi_j, \qquad \forall\varphi_j \in V_h, \tag{6.22}$$

are dependent on the feedback control $a^*(\mathbf{x}, \tau)$ and τ. So in order to approximate the integrals appearing in the last two equations, we must evaluate the feedback control at some carefully chosen points. To maximize accuracy and efficiency, these points are chosen via the Gaussian quadrature method.

6.6 Numerical computation of integrals

When implementing the finite element method it is convenient to compute integrals using some numerical quadrature rule. Suppose that we want to integrate the function f over the triangle element K. A generic quadrature rule is stated

$$\int_K f(\mathbf{x}) \approx \frac{1}{2}\sum_{k=1}^{nq} f(\mathbf{x}_k)w_k. \tag{6.23}$$

The numbers $(w_k)_{k=1}^{nq}$ and $(\mathbf{x}_k)_{k=1}^{nq}$ are called the quadrature weights and quadrature points respectively. In general, the quadrature weights and quadrature points are different for each triangle $K \in \mathcal{T}$. Therefore, quadrature rules are typically given on the *reference element* $\hat{K}$ having corner coordinates $(0, 0), (0, 1)$ and $(1, 0)$. We report in table 6.1, Gaussian quadrature rules up to order 6, corresponding to the reference element. The value $\frac{1}{2}$ appearing on the right hand side of equation (6.23) is equal to the area of the reference element $\hat{K}$.

nq	$(\hat{x},\hat{y})$	w
1-point rule	$(1/3, 1/3)$	1
3-point rule	$(1/6, 1/6)$	$1/3$
	$(2/3, 1/6)$	$1/3$
	$(1/6, 2/3$	$1/3$
4-point rule	$(1/3, 1/3)$	$-27/48$
	$(1/5, 3/5)$	$25/48$
	$(1/5, 1/5)$	$25/48$
	$(3/5, 1/5)$	$25/48$

Table 6.1: Gauss quadrature rules for the reference triangle.

6.6.1 Mapping from the triangular reference element

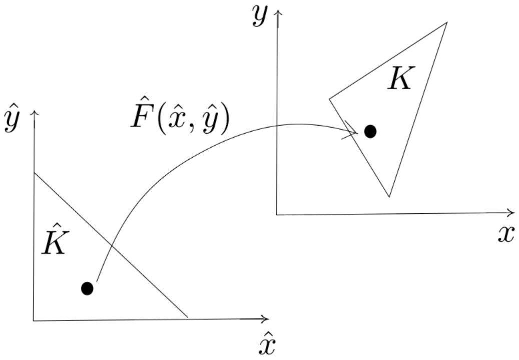


Figure 6.2: Mapping $\hat{F}(\hat{x},\hat{y}) := (g(\hat{x},\hat{y}), h(\hat{x},\hat{y}))^\top$ from the reference element $\hat{K}$ (left) to a physical element K (right).

Mapping coordinates from the reference element $\hat{K}$ to the *physical element*, see figure, can be achieved via the $\mathbb{P}_1$ shape functions $(\varphi_i)_{i=1}^3$ provided that the triangles have straight boundaries. Define the transformation $x = g(\hat{x},\hat{y})$, $y = h(\hat{x},\hat{y})$, such that

$$g(\hat{x},\hat{y}) = \sum_{j=1}^{3} X_j \hat{\varphi}_j(\hat{x},\hat{y}), \qquad h(\hat{x},\hat{y}) = \sum_{j=1}^{3} Y_j \hat{\varphi}_j(\hat{x},\hat{y}). \tag{6.24}$$

The Jacobian matrix J is defined

$$J = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix}$$

Via (6.24), the explicit entries of J are given by

$$J_{11} = \sum_{j=1}^{3} X_j \frac{\hat{\varphi_j}}{\partial \hat{x}}, \qquad J_{12} = \sum_{j=1}^{3} X_j \frac{\hat{\varphi_j}}{\partial \hat{y}},$$
$$J_{21} = \sum_{j=1}^{3} Y_j \frac{\hat{\varphi_j}}{\partial \hat{x}}, \qquad J_{23} = \sum_{j=1}^{3} Y_j \frac{\hat{\varphi_j}}{\partial \hat{y}},$$

The derivatives of the $p1$ hat functions are

$$\frac{\hat{\varphi_1}}{\partial \hat{x}} = -1, \qquad \frac{\hat{\varphi_1}}{\partial \hat{y}} = -1,$$
$$\frac{\hat{\varphi_2}}{\partial \hat{x}} = 1, \qquad \frac{\hat{\varphi_2}}{\partial \hat{y}} = 0,$$
$$\frac{\hat{\varphi_3}}{\partial \hat{x}} = 0, \qquad \frac{\hat{\varphi_3}}{\partial \hat{y}} = 1,$$

we obtain

$$J = \begin{bmatrix} X_2 - X_1 & X_3 - X_1 \\ Y_2 - Y_1 & Y_3 - Y_1 \end{bmatrix}.$$

6.6.2 Transformation of integrals

The Jacobian matrix, defined in the previous section, can be used to transform the integral of a function f over any triangular element $K \in \mathcal{T}$ to an integral over the reference element $\hat{K}$. Once the integral domain is transformed to the reference element, we can apply the quadrature rules from section 6.6. We have

$$\int_K f(x,y)\, d\Omega = \int_{\hat{K}} \hat{f}(\hat{x},\hat{y})|J|\, d\hat{\Omega}, \tag{6.25}$$

where $|J|$ denotes the determinant of J and

$$\hat{f}(\hat{x},\hat{y}) = f(g(\hat{x},\hat{y}), h(\hat{x},\hat{y})).$$

We will frequently be evaluating integrals that involves the gradient operator $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$. The chain rule gives

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial x} + \frac{\partial f}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial x}, \qquad \frac{\partial f}{\partial y} = \frac{\partial f}{\partial \hat{x}}\frac{\partial \hat{x}}{\partial y} + \frac{\partial f}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial y}.$$

The previous pair of equations is equivalent to the following relation [10]

$$\nabla f = J^{-T}\hat{\nabla} f, \tag{6.26}$$

where J^{-T} denotes the inverse transpose of the Jacobian matrix and $\hat{\nabla} = (\frac{\partial}{\partial \hat{x}}, \frac{\partial}{\partial \hat{y}})$ is the gradient operator with respect to the reference coordinates $(\hat{x},\hat{y})$. The explicit expression for J^{-T} is given by

$$J^{-T} = \frac{1}{|J|}\begin{bmatrix} J_{22} & -J_{21} \\ -J_{12} & J_{11} \end{bmatrix}.$$

Suppose that we want to evaluate the integral

$$I = \int_K \nabla v(x,y) \cdot \nabla u(x,y)\, d\Omega.$$

Let

$$f(x,y) := \nabla v(x,y) \cdot \nabla u(x,y),$$

via (6.25) we have

$$\begin{aligned} \int_K \nabla v(x,y) \cdot \nabla u(x,y)\, d\Omega &= \int_K f(x,y)\, d\Omega \\ &= \int_{\hat{K}} \hat{f}(\hat{x},\hat{y})\, |J| d\hat{\Omega} \\ &= \int_{\hat{K}} \nabla \hat{v}(\hat{x},\hat{y}) \cdot \nabla \hat{u}(\hat{x},\hat{y}) |J|\, d\hat{x} d\hat{y}. \end{aligned}$$

Using relation (6.26) we have

$$\nabla \hat{v}(\hat{x},\hat{y}) = J^{-T} \hat{\nabla} \hat{v}(\hat{x},\hat{y}), \qquad \nabla \hat{u}(\hat{x},\hat{y}) = J^{-T} \hat{\nabla} \hat{u}(\hat{x},\hat{y}).$$

Assuming that the derivatives $\frac{\partial \hat{v}}{\partial \hat{x}}, \frac{\partial \hat{v}}{\partial \hat{y}}, \frac{\partial \hat{w}}{\partial \hat{x}}, \frac{\partial \hat{w}}{\partial \hat{y}}$ can be computed, we can apply one of the quadrature rules from table 6.1 to compute

$$\begin{aligned} I \approx \sum_{k=1}^{nq} w_k \Bigg[&\left(J_{22} \frac{\partial \hat{v}}{\partial \hat{x}}(\hat{x}_k,\hat{y}_k) - J_{21} \frac{\partial \hat{v}}{\partial \hat{y}}(\hat{x}_k,\hat{y}_k) \right) \left(J_{22} \frac{\partial \hat{u}}{\partial \hat{x}}(\hat{x}_k,\hat{y}_k) - J_{21} \frac{\partial \hat{u}}{\partial \hat{y}}(\hat{x}_k,\hat{y}_k) \right) \\ &+ \left(-J_{12} \frac{\partial \hat{v}}{\partial \hat{x}}(\hat{x}_k,\hat{y}_k) + J_{11} \frac{\partial \hat{v}}{\partial \hat{y}}(\hat{x}_k,\hat{y}_k) \right) \left(-J_{12} \frac{\partial \hat{u}}{\partial \hat{x}}(\hat{x}_k,\hat{y}_k) + J_{11} \frac{\partial \hat{u}}{\partial \hat{y}}(\hat{x}_k,\hat{y}_k) \right) \Bigg]. \end{aligned}$$

6.7 Implementation

We recall that the fully discrete scheme for solving (6.2) is given as

$$\frac{1}{\Delta\tau} \mathrm{M}(\mathbf{v}^{n+1} - \mathbf{v}^n) + \mathrm{A}^{n+1} \mathbf{v}^{n+1} = \mathbf{r}^{n+1}. \tag{6.27}$$

One of the main tasks of our finite element program is to assemble the matrices $\mathrm{M}, \mathrm{A}^{n+1}$ and the vector $\mathbf{r}^{n+1}$ appearing in the previous equation. We have have obtained a reasonably efficient programs in MATLAB by vectorization of all matrix assembly procedures. For a guide to efficient implementation of the finite element method in MATLAB, we refer to [7].

In algorithm 4 we have tried to summarize the main flow of the program for solving the full problem (6.1). In the first line of code the mass matrix is assembled. Since the mass matrix is the same for each time step it can be assembled before the time iteration starts. In the next line of code the solution vector is initialized. The first line of code inside the time loop represents the procedure for approximating the feedback control which we denote by ξ^{n+1}. The feedback control ξ^{n+1} is approximated by using the numerical solution from the previous time step as decribed in section 6.5. In the next two lines we have written procedures for assembly of the stiffness matrix A^{n+1} and the load vector r^{n+1} that takes the approximation ξ^{n+1} as one of its input.

Algorithm 4 Algorithm for computing the solution of scheme (6.27)

```
$\mathrm{M} \leftarrow$ AssembleMassMatrix( )
$\mathbf{v}^0 \leftarrow \mathbf{0}$
for $n = 1, \ldots, N_\tau$ do
    $\xi^{n+1} \leftarrow$ ComputeFeedbackControl($\mathbf{v}^n$)
    $\mathrm{A}^{n+1} \leftarrow$ AssembleStiffnessMatrix($\xi^{n+1}$)
    $\mathbf{r}^{n+1} \leftarrow$ AssembleLoadVector($\xi^{n+1}$)
    $\mathbf{v}^{n+1} \leftarrow (\mathrm{M} + \Delta\tau\mathrm{A}^{n+1})\backslash\mathbf{r}^{n+1}$
end for
```

Chapter 7

Numerical Results

In the previous chapters we have provided a presentation of three different numerical methods for solving the Hamilton Jacobi Bellman equation related to the optimal operation of a natural gas storage facility. In this chapter we will try out the methods on some test problems. We will refer to the semi-Lagrange finite element method introduced in chapter 5 as the SLFE - method and we will refer to the finite element method introduced in chapter 6 as the FE - method.

7.1 A "realistic" test case

We consider the case described in [14] and [19] relating to the Texas based Stratton Ridge salt cavern gas storage facility. For a derivation of the model we refer to [14], for computational purposes, the model is summarized below:

- $v(x, y, 0) = 0$, initial condition
- $T = 1$, time horizon
- $x_{\max} = 12$, maximum price
- $y_{\max} = 2000$, maximum storage capacity
- $\mu(x) = 2.38(x - 6)$, drift term in the gas price process (mean reverting)
- $\sigma(x) = 0.59x$, volatility of gas price
- $\rho = 0.1$, the risk free interest rate
- $a_{\min}(y) = \sqrt{\frac{1}{y+25000} - \frac{1}{500}}$, maximal injection rate of gas
- $a_{\max}(y) = 2041.4\sqrt{y}$, maximal withdrawal rate of gas
- $\lambda(a) = \left\{ \begin{array}{ll} 0 & \text{if } a \geq 0 \\ -365 \cdot 1.7 & \text{if } a < 0 \end{array} \right.$, leakage of gas

- $r(x, a) = 1000(a - \lambda(a))$, cash flow
- $f(a) = a + \lambda(a)$, gas flow

The resulting Hamilton Belmann Jacobi equation is stated

$$\frac{\partial v}{\partial \tau} - \frac{1}{2}\sigma^2 \frac{\partial^2 v}{\partial x^2} - \mu \frac{\partial v}{\partial x} + \rho v - \sup_{a \in A(y)} \left(1000(a - \lambda(a))x - (a + \lambda(a)) \frac{\partial v}{\partial y} \right) = 0. \tag{7.1}$$

Figure (7.1) display the numerical solution of the value function and the optimal control respectively, obtained with the semi-Lagrangian finite element method (SLFE) introduced in chapter 5 using $\mathbb{P}_1$-elements. For a given amount of gas in storage we observe that the optimal policy depends on the price of gas such as one would expect. That is, when the price is "high" the optimal policy is to sell and when the price is "low" the optimal policy is to buy. We also observe that the policy of doing nothing, ($a^* = 0$), is optimal when the price of gas is close to average market price ($x = 6$). We have also performed the above test case with the finite element method (FE) presented in chapter 6 for $\mathbb{P}_1$ and $\mathbb{P}_2$ elements and produced similar results. However, according to our experiments, the FE method does not always work properly without the edge stabilization technique and the solution may develop spurious oscillations, see figure 7.2. The SLFE method seems to run fine without edge stabilization according to experiments.

Remark 7.1. *As described in section 2.4.1, equation (7.1) can be scaled such that the spatial domain $\Omega = [0, x_{\max}] \times [0, y_{\max}]$ is equal to the unit square. However, in figure 7.1 we have plotted the numerical solution on the actual domain. In all the remaining experiments of this chapter, the spatial domain is scaled into the unit square.*

Remark 7.2. *We have observed, trough experiments with the code, that the amount of stabilization required depends on the precision of the quadrature rule used to evaluate integrals. In general we see that higher precision quadrature leads to less need for stabilization.*

Remark 7.3. *According to our observations, the $\mathbb{P}_1 - \mathrm{FE}$ method seems to be more robust than the $\mathbb{P}_2 - \mathrm{FE}$ method. For instance, we have observed that the $\mathbb{P}_2$-FE method may become unstable if the time step is very small compared to the mesh size.*

Remark 7.4. *For the SLFE method and finite difference method we have enforced the condition $\sigma \to 0$ at the boundary $x = x_{\max}$, as discussed in section 2.4.1. This is implemented by simply setting σ equal to zero in the interval $[x_{\max} - \Delta x, x_{\max}]$. For the FE-method we have simply used the natural boundary condition (6.4).*

7.2 Testing the convergence rate

In this section we try to compute the convergence rate of our method with respect to the test case described in the previous section. However, these results are just estimates and may be inaccurate if the numerical error is not within the asymptotic region.

The convergence rate of a numerical scheme can be approximated by successively decreasing the discretization parameters and comparing subsequent solutions. We will now describe the general methodology with respect to the finite element method presented in chapter 6. Suppose that the error of the numerical solution u_h compared to the exact solution u is on the form

$$\|u - u_h\| \sim \Delta\tau + h^{\beta},$$

where $\|\cdot\|$ is some suitable norm and β is the convergence rate to be found, we have implicitly assumed in the previous equation that the backward Euler method gives first order convergence in τ. Let h_0 represent the size of the initial triangulation and consider a sequence of triangulations $(\mathcal{T}_{h_k})_{k=1}^{\infty}$ such that $h_{k+1} = \alpha h_k$ with $\alpha \in (0,1)$. Take $\Delta\tau_k \sim (h_k)^{\beta}$ and let $\tilde{u}_k$ represent the numerical solution corresponding to the dicretization parameters $(h_k, \Delta\tau_k)$. If the sequence $\tilde{u}_k$ converges smoothly towards the solution u with a rate equal to β, there is a number $C > 0$ independent of h_k such that

$$\|\tilde{u}_k - u\| = C(h_k)^{\beta}. \tag{7.2}$$

The last equation implies that

$$\frac{\|\tilde{u}_{k-1} - u\|}{\|\tilde{u}_k - u\|} = \left(\frac{h_{k-1}}{h_k}\right)^{\beta} = \alpha^{\beta-1},$$

so that

$$\beta = \frac{1}{\ln\alpha}\ln\frac{\|\tilde{u}_{h_{k-1}} - u\|}{\|\tilde{u}_{h_k} - u\|}.$$

We assume that the following approximation is valid as h_k goes to zero

$$\|\tilde{u}_{k-1} - \tilde{u}_k\| \approx \|\tilde{u}_{k-1} - u\|,$$

such that

$$\beta \approx \frac{1}{\ln\alpha}\ln\frac{\|\tilde{u}_{h_{k-1}} - \tilde{u}_k\|}{\|\tilde{u}_{h_k} - \tilde{u}_{k+1}\|}. \tag{7.3}$$

Alternatively we could use the approximation

$$\beta \approx \frac{1}{\ln\alpha}\ln\frac{\|\tilde{u}_{h_{k-1}} - \tilde{u}_0\|}{\|\tilde{u}_{h_k} - \tilde{u}_0\|},$$

where $\tilde{u}_0$ represents a numerical solution corresponding to a very fine discretization.

7.2.1 FE method

In tables 7.1 we have used (7.3) to approximate β. The error is defined as the difference between subsequent solutions measured in the $L_2 - norm$ measured at the last time step $\tau = 0.25$.

$\Delta\tau$	h	error $\times 10^6$	relative error	rate (β)
0.1250	0.0500	5.6203	2.3257	5.9879
0.0625	0.0354	0.7055	0.0980	2.1639
0.0312	0.0250	0.3333	0.0424	1.9976
0.0156	0.0177	0.1668	0.0206	2.4592
0.0078	0.0125	0.0711	0.0087	
0.0039	0.0088			

Table 7.1: Verification of the convergence rate $\hat{\beta} = 2$ for the $p1$-FE method. with $\alpha = 0.7071$. The stabilization parameter is $\gamma = 0.5$

$\Delta\tau$	h	error $\times 10^6$	relative error	rate (β)
0.1250	0.0500	1.5656	0.2774	3.4525
0.0625	0.0397	0.7051	0.0979	3.2370
0.0312	0.0315	0.3338	0.0425	3.0100
0.0156	0.0250	0.1665	0.0206	3.6672
0.0078	0.0198	0.0714	0.0087	
0.0039	0.0157			

Table 7.2: Verification of the convergence rate $\hat{\beta} = 3$ for the $p2$-FE method. with $\alpha = 0.7937$. The stabilization parameter is $\gamma = 0.02$

7.2.2 FE-Semi-Lagrangian

Since we have used a linear scheme in the y and τ direction and the finite element method only in the price direction (x-direction), we assume that the error is on the form

$$\|u_{\Delta x,\Delta y,\Delta\tau} - u\| = \Delta x^{\beta} + \Delta\tau + \Delta y,$$

where Δx now represents size of $\mathcal{T}_h$ defined in section 4.2, i.e. $\Delta x = h$. Consider the sequence $(\Delta x_k)_{k=1}^{\infty}$ such that $\Delta x_k = \alpha\Delta x_{k-1}$, take $\Delta\tau \sim \Delta x^{\beta}$, $\Delta y \sim \Delta x^{\beta}$ and let $\tilde{u}_k$ represent the solution corresponding to Δx_k. We have

$$\|\tilde{u}_k - u\| = C(\Delta x_k)^{\beta}, \tag{7.4}$$

and as in the previous section

$$\beta \approx \frac{1}{\ln\alpha}\ln\frac{\|\tilde{u}_{k-1} - \tilde{u}_k\|}{\|\tilde{u}_k - \tilde{u}_{k+1}\|}.$$

We have used the following norm to measure the error:

$$\|\tilde{u} - u\| = \max_{1\le j\le J}\sqrt{\int_0^1 (\tilde{u}(x,y_j,T) - u(x,y_j,T))^2\,dx},$$

with $T = 0.25$.

Δx	Δy	$\Delta \tau$	error	relative error	rate (β)
0.0833	0.2000	0.2000	5.7630	0.1154	1.0657
0.0589	0.1000	0.1000	2.0515	0.0791	1.7141
0.0417	0.0500	0.0500	0.7127	0.0437	1.8863
0.0295	0.0250	0.0250	0.2743	0.0227	1.9119
0.0208	0.0125	0.0125	0.1401	0.0117	
0.0147	0.0063	0.0063			

Table 7.3: Verification of the convergence rate $\beta = 2$ for the $p1$-SLFE method with $\alpha = 0.7071$. No edge stabilization is used ($\gamma = 0$).

Δx	Δy	$\Delta \tau$	error 10^6	relative error	rate (β)
0.0833	0.2000	0.2000	2.8867	0.1152	1.5972
0.0661	0.1000	0.1000	1.9946	0.0790	2.5648
0.0525	0.0500	0.0500	1.1019	0.0437	2.8203
0.0417	0.0250	0.0250	0.5730	0.0228	2.9015
0.0331	0.0125	0.0125	0.2952	0.0117	

Table 7.4: Verification of the convergence rate $\beta = 3$ for the $p2$-SLFE method with $\alpha = 0.7937$. No edge stabilization is used ($\gamma = 0$).

7.3 Verification of convergence to the viscosity solution

Because the finite difference method presented in chapter 3 is monotone, stable and point vise consistent, it is guaranteed to converge to the viscosity solution of (7.1), [19]. We can use this knowledge to verify that the SLFE method and the FE-method also converges correctly. This verification is conducted by comparing subsequent solutions of the finite difference method with the two other methods. If the numerical methods converge to the same solution we should observe that the difference between the solutions decreases as the discretization parameters are decreased. The results from experiments with the SLFE method and the FE-method with $\mathbb{P}_1$ and $\mathbb{P}_2$ elements are shown in tables 7.5, 7.6, 7.7, and 7.8. The discretization parameters of the methods are halved at each iteration and we observe that the difference between the solutions, measured in the $\|\cdot\|_2$-norm, decreases approximately with a factor of 0.5 for each iteration, expect for the $\mathbb{P}_2$-FE method where the rate seems to decrease slightly. The difference in the solutions are computed at the last time step, $\tau = 0.25$.

Remark 7.5. *The boundary conditions at $x = 1$ for the FE method is slightly different from the finite difference method and the SLFE method, see remark 7.4, so we only measure the difference in the solutions on a subset $\tilde{\Omega} = [1, 0.8] \times [0, 1] \subset \Omega$ of the domain, in tables 7.7 and 7.8.*

Δx	Δy	$\Delta \tau$	$\|\tilde{v}_k - \hat{v}_k\|_2 \times 10^5$	$\|\tilde{v}_k - \hat{v}_k\|_2 / \|\tilde{v}_k\|_2$
0.2500	0.2500	0.0217	3.6270	0.0385
0.1250	0.1250	0.0109	0.9193	0.0105
0.0625	0.0625	0.0055	0.3695	0.0044
0.0312	0.0312	0.0027	0.1565	0.0019
0.0156	0.0156	0.0014	0.0941	0.0011

Table 7.5: Verification of convergence to the viscosity solution for the $\mathbb{P}_1$-SLFE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the $\mathbb{P}_1$-SLFE-method and the finite difference method, respectively. No edge stabilization is used ($\gamma = 0$).

Δx	Δy	$\Delta \tau$	$\|\tilde{v}_k - \hat{v}_k\|_2 \times 10^5$	$\|\tilde{v}_k - \hat{v}_k\|_2 / \|\tilde{v}_k\|_2$
0.2500	0.2500	0.0217	2.6209	0.0276
0.1250	0.1250	0.0109	2.6088	0.0292
0.0625	0.0625	0.0055	1.1006	0.0129
0.0312	0.0312	0.0027	0.5215	0.0062
0.0156	0.0156	0.0014	0.2569	0.0031

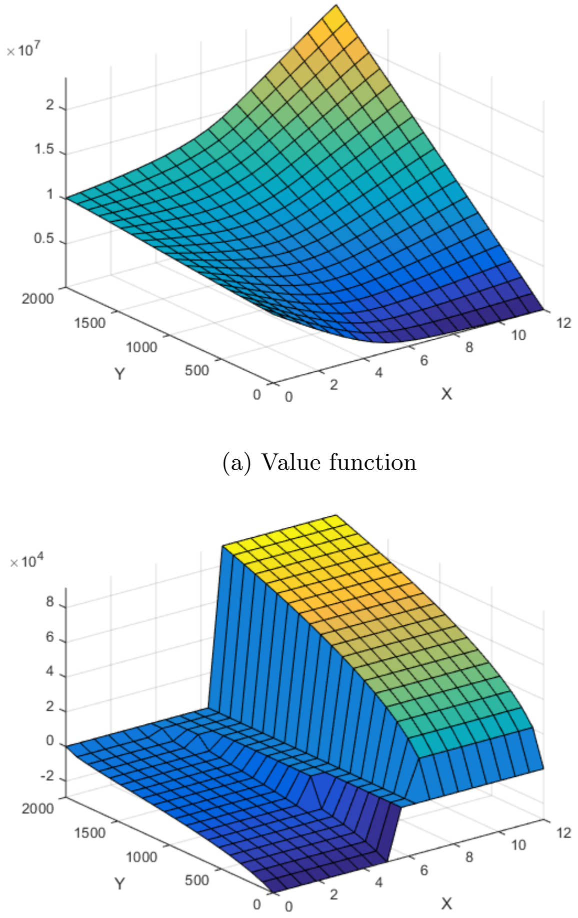
Table 7.6: Verification of convergence to the viscosity solution for the $\mathbb{P}_2$-SLFE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the $\mathbb{P}_2$-SLFE-method and the finite difference method, respectively. No edge stabilization is used ($\gamma = 0$).

h	Δx	Δy	$\Delta \tau$	$\|\tilde{v}_k - \hat{v}_k\|_2 \times 10^6$	$\|\tilde{v}_k - \hat{v}_k\|_2 / \|\tilde{v}_k\|_2$
0.2500	0.2500	0.2500	0.0217	5.7630	0.5089
0.1250	0.1250	0.1250	0.0109	2.0515	0.2546
0.0625	0.0625	0.0625	0.0055	0.7127	0.1001
0.0312	0.0312	0.0312	0.0027	0.2743	0.0404
0.0156	0.0156	0.0156	0.0014	0.1401	0.0208

Table 7.7: Verification of convergence to the viscosity solution for the $\mathbb{P}_1$-FE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the $\mathbb{P}_1$-FE-method and the finite difference method, respectively. The stabilization parameter is $\gamma = 0.5$.

h	Δx	Δy	$\Delta \tau$	$\|\tilde{v}_k - \hat{v}_k\|_2 \times 10^5$	$\|\tilde{v}_k - \hat{v}_k\|_2 / \|\tilde{v}_k\|_2$
0.2500	0.2500	0.2500	0.0217	6.1515	0.0976
0.1250	0.1250	0.1250	0.0109	2.4646	0.0387
0.0625	0.0625	0.0625	0.0055	1.3265	0.0203
0.0312	0.0312	0.0312	0.0027	0.8325	0.0127
0.0156	0.0156	0.0156	0.0014	0.6745	0.0102

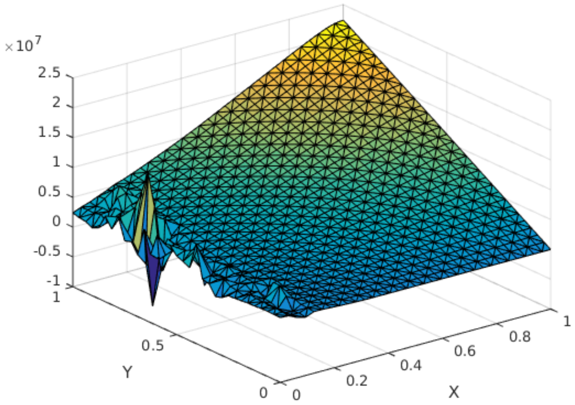
Table 7.8: Verification of convergence to the viscosity solution for the $\mathbb{P}_2$-FE method; $\tilde{v}_k$ and $\hat{v}_k$ represent the solutions of the FE-method and the finite difference method, respectively. The stabilization parameter is $\gamma = 0.02$.



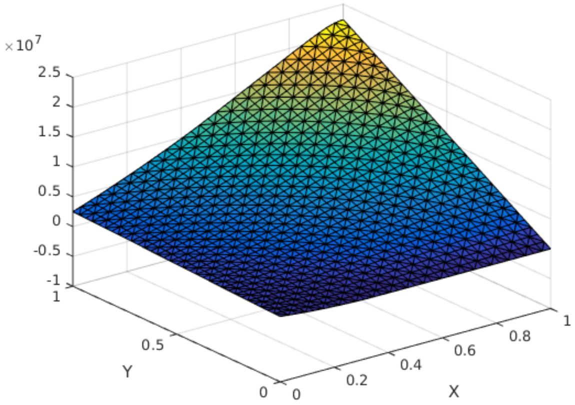
(a) Value function

(b) Optimal control

Figure 7.1: Numerical solution of the value function and the optimal control policy evaluated at $\tau = 1$. The discretization parameters are $\Delta x = x_{\max}/20, \Delta y = y_{\max}/20$ and $\Delta\tau = T/200$.



(a) FEM $\mathbb{P}_2$ with no stabilization $\gamma = 0$.



(b) FEM $\mathbb{P}_2$ with stabilization $\gamma = 0.02$

Figure 7.2: Numerical solution of the value function after 10 time steps, with $h = 0.04$ and $\Delta\tau = 0.0156$. When we remove the edge stabilization, spurious oscillations are observed in the solution. The oscillations starts in $x = 0$, where there is least diffusion. Similar observation are made with $\mathbb{P}_1$ elements. We found that the stabilization parameters $\gamma = 0.02$ and $\gamma = 0.5$ worked well for $\mathbb{P}_1$ and $\mathbb{P}_2$ elements respectively.

Chapter 8

Conclusion

8.1 Summary

This thesis has described three separate numerical methods for solving numerically the Hamilton Jacobi Bellman equation (2.21) relating to the valuation and optimal operation of a natural gas storage:

(i) A semi implicit upwind finite difference method was introduced in chapter 3. The method was shown to be consistent, monotone and $\|\cdot\|_\infty$-stable provided a linear CFL condition (3.13). Even if this method is only first order accurate it is attractive from a practical point of view as it is very easy to implement, numerical oscillations cannot occur due to the upwind technique and the numerical solution converges to the viscosity solution of 2.21, [19].

(ii) A semi-Lagrange finite element method based on the method developed by Forsyth and Chen [19] was decribed in chapter 5. A linearization technique, described in section 5.1 was used to obtain a fully implicit method. We discussed the possibility of adding stabilization in the price direction in section 4.3, however, in our numerical experiments this was not necessary.

(iii) A finite element method based on a triangulation of the domain in the x and y directions and a backward finite difference discretization in time was presented in chapter 6. To prevent numerical oscillations, the method was stabilized with the edge stabilization technique, described in [4]. Numerical results indicate that the edge stabilization method successfully prevent numerical oscillations, see figure 7.2.

All the methods were successfully implemented in MATLAB and in chapter 7 we have conducted numerical experiments with respect to a standard test case in he literature, see [14], [19], [17]. The numerical results provided in tables 7.5, 7.6, 7.7, and 7.8, indicate that the three methods converge to the same solution for the given test case.

8.2 Challenges

A substantial amount of the work in this thesis has gone to the implementation of the numerical schemes in MATLAB. In particular, the implementation of the finite element method presented in chapter 6 required a lot of work, as we have implemented the possibility of higher order elements and vectorization of the matrix assembly procedures. When implementation of the edge stabilization method described in section 6.3, it is necessary to obtain the indices of the elements that are connected to a given edge. This information is in general not provided in a standard mesh data structure. For this, we found that the triangulation class in MATLAB was particularly useful.

8.3 Future Work

- The test case in section 7 is very simple. It is possible to obtain a more realistic model by incorporating seasonal effects in the model and price jumps, as suggested in [19] and [14].
- We have restricted the dicretization of the temporal variable to first order methods. More accurate time stepping techniques such as the theta method or Runge-Kutta schemes could be tried out.
- Apart from the upwind finite difference method presented in chapter 3 this thesis has been very practical in nature and it remains to properly analyze theoretically the methods that were introduced in chapters 5 and 6.
- The features of equation (2.21) are very different in each spatial direction. It could therefore be reasonable to apply a fractional step method in time that divides the differential operator into two parts corresponding to each spatial direction. In this way, the two resulting sub problems can be discretized independently with specialized methods.
- One of the advantages with the finite element method is the possibility of locally refining the grid in areas where the discretization error is estimated to be large. Grid refinement can take place subdividing triangles into smaller triangles or increasing the degree of the basis functions in some elements. We did not have time to try out this these techniques.

Bibliography

[1] Matthias Augustin, Alfonso Caiazzo, André Fiebach, Jürgen Fuhrmann, Volker John, Alexander Linke, and Rudolf Umla. An assessment of discretizations for convection-dominated convection–diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 200(47):3395–3409, 2011.

[2] Richard Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[3] Erik Burman and Miguel A Fernández. Finite element methods with symmetric stabilization for the transient convection–diffusion–reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 198(33):2508–2519, 2009.

[4] Erik Burman and Peter Hansbo. Edge stabilization for Galerkin approximations of convection–diffusion–reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(15):1437–1453, 2004.

[5] Zhuliang Chen and Peter A Forsyth. Pricing hydroelectric power plants with-/without operational restrictions: a stochastic control approach. *Nonlinear Models in Mathematical Finance. Nova Science Publishers, To appear*, 1(9), 2008.

[6] Lawrence C Evans. An introduction to stochastic differential equations version 1.2. *Lecture Notes, UC Berkeley*, 2001.

[7] Stefan Funken, Dirk Praetorius, and Philipp Wissgott. Efficient implementation of adaptive P1-FEM in MATLAB. *Computational Methods in Applied Mathematics Comput. Methods Appl. Math.*, 11(4):460–490, 2011.

[8] Matt Davison Henning Rasmussen Matt Thompson. Valuation and Optimal Operation of Electric Power Plants in Competitive Markets. *Operations Research*, 52:546–562, 2004.

[9] Huyên Pham. *Continuous-time Stochastic Control and Optimization with Financial Applications.* Stochastic Modelling and Applied Probability. Springer, 2009.

[10] A. Quarteroni. *Numerical Models for Differential Problems.* MS&A. Springer, 2010.

[11] Friedhelm Schieweck. On the role of boundary conditions for CIP stabilization of higher order finite elements. *Electronic Transactions on Numerical Analysis*, 32:1–16, 2008.

[12] Eduardo S Schwartz. The stochastic behavior of commodity prices: Implications for valuation and hedging. *The Journal of Finance*, 52(3):923–973, 1997.

[13] Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models-a review. *Monthly weather review*, 119(9):2206–2223, 1991.

[14] Matt Thompson, Matt Davison, and Henning Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics (NRL)*, 56(3):226–238, 2009.

[15] Asgeir Tomasgard, Frode Rømo, Marte Fodstad, and Kjetil Midthun. Optimization models for the natural gas value chain. In *Geometric modelling, numerical simulation, and optimization*, pages 521–558. Springer, 2007.

[16] Song Wang, Les S Jennings, and Kok Lay Teo. Numerical solution of Hamilton-Jacobi-Bellman equations by an upwind finite volume method. *Journal of Global Optimization*, 27(2-3):177–192, 2003.

[17] Antony Ware. Accurate semi-lagrangian time stepping for stochastic optimal control problems with application to the valuation of natural gas storage. *SIAM Journal on Financial Mathematics*, 4(1):427–451, 2013.

[18] Tony Ware. Swing options in a mean-reverting world. In *Presentation at: Stochastic Calculus and its applications to Quantitative Finance and Electrical Engineering, a conference in honour of Robert Elliott, Calgary*, 2005.

[19] Peter A. Forsyth Zhuliang Chen. A Semi-Lagrangian Approach for Natural Gas Storage Valuation and Optimal Operations. *SIAM J. Sci. Comput.*, 30:339–368, 2007.