COMPUTING THE EFFECTIVE HAMILTONIAN IN THE MAJDA-SOUGANIDIS MODEL

COMPUTING THE EFFECTIVE HAMILTONIAN IN THE MAJDA-SOUGANIDIS MODEL

By:

MIRELA CARA, B.Sc.

A Thesis
Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Science

McMaster University

MASTER OF SCIENCE (2005)

Department of Mathematics and Statistics

McMaster University

Hamilton, Ontario

TITLE: Computing the Effective Hamiltonian in the Majda-Souganidis Model

AUTHOR: Mirela Cara

SUPERVISOR: Doctor Agnès Tourin

NUMBER OF PAGES: iii, 89

Abstract

In premixed turbulent combustion, the normal speed of propagation of the flame front is enhanced by the turbulent velocity field. This project will focus on the method of computing the normal speed of propagation of the flame front in the Majda-Souganidis model of turbulent combustion. Solving this problem involves computing the eigenvalue of a nonlinear *cell problem.* Discussed in this thesis is a new, simple and direct numerical method for approximating the eigenvalue, also called the *effective Hamiltonian.*

Acknowledgements

I would like to offer special thanks to my supervisor Dr. Agnès Tourin for her very helpful suggestions, immense patience and for introducing me to the subject of Nonlinear Partial Differential Equations. Your encouragement and support will always be remembered.

I am also very grateful to my best friend Linda Vrbova, for continuing to be a positive influence in my life and in my education. You and I have always been in a pursuit for knowledge. I hope that we can continue in that direction and that we will have the opportunity to share with each other what we learn for many more years to come.

Finally I wish to thank Bogdan Traicu, who has helped me focus on matters of the present, rather than the uncontrollable past or future. Your realistic view has helped me remain motivated during the preparation of this thesis.

Contents

Introduction

Front propagation in turbulent flow dynamics is a problem of significance in combustion science. Reaction-diffusion-advection equations provide the description of propagation of fronts with applications in physics phenomena such as combustion and biological sciences. It is easier to predict the front speed in the laminar case, where the advecting fluid velocity involves spatio-temporal scales which are small when compared to the flame front thickness. However, the problem of predicting the front speed enhancement becomes more difficult when the advecting fluid flow field has two separated scales both larger than the front thickness. In this case, the two-scale velocity field is comprised of a large scale flow and an intermediate scale flow due to turbulence. An asymptotic model has been developed by Majda and Souganidis [34] in order to predict the front speed enhancement. Their model, used in this project, applies only to the case of premixed flame combustion, where turbulence does contribute to the location of the front by enhancement [35].

The mathematical theory of the Majda-Souganidis model introduces a

method for computing the speed enhancement by minimizing a function of an averaged (effective) Hamiltonian. The *effective Hamiltonian* ($\overline{H}(P)$) is the eigenvalue of a nonlinear, stationary Hamilton-Jacobi equation, also called the *cell problem.* In this case explicit formulas for the front speed cannot be deduced anymore and the problem must be solved numerically.

Some numerical methods for solving Hamilton-Jacobi equations with no eigenvalues [37] or by solving a system of Conservation Laws [26], are already known. In this project, we present a new, direct and simple numerical method for computing the normal speed of propagation of the flame front for the Majda-Souganidis model of turbulent premixed combustion.

This project is organized as follows: in chapter 1 we discuss the theory behind the derivation of the formula for the normal speed of propagation of the front. In chapter 2 we analyze the solutions of the stationary nonlinear *cell problem* using the theory of viscosity solutions [3]. Also in this chapter we will introduce the relationship between the solutions of the *cell problem* and the solutions of an evolution problem. It will be made clear why we will solve the evolution problem in order to obtain the eigenvalue and eigenfunction of the *cell problem.* In chapter 3 we concentrate on the numerical schemes implemented, respectively, the first order monotone finite difference (FOMFD) scheme and the second order Essentially-Non-Oscillatory (ENO) scheme. We also present the numerical results and explain how they follow the theoretical aspects of chapters 1 and 2, as well as how they compare to the results obtained by a different method by Bourlioux and Khouider [9].

Chapter 1

Equations for the Propagation of the Front

1.1 Introduction

This chapter starts by introducing the reaction-diffusion-advection equation for the temperature field. Majda and Souganidis [34] derived their mathematical model of turbulent combustion for premixed flames from the reaction-diffusion-advection equation. In this chapter, we will closely follow their work in order to describe how in the limit $\varepsilon \to 0$ the reaction-diffusion-advection equation is related to a *variational inequality* problem. We will also show how the solution of the variational problem is related to the *cell problem* and the *effective Hamiltonian.* At the end of the chapter a brief account of the derivation of the normal speed of propagation of the flame front will be given.

1.2 The General Front Equation

In the physical process of turbulent combustion, the propagation of the flame front is governed by an advection-diffusion-reaction equation for temperature [16],[34]. This is an initial value problem of the form:

$$\begin{cases} T_t^\varepsilon + V \cdot DT^\varepsilon - \varepsilon k D^2 T^\varepsilon - \varepsilon^{-1} f(T^\varepsilon) = 0; \quad x \in \mathbb{R}^N \times (0, \infty) \\ \\ T^\varepsilon(x, 0) = T_0 \quad \text{for} \quad x \in \mathbb{R}^N \times \{0\}. \end{cases} \tag{1.2.1}$$

Here we have a flame front thickness of order $\varepsilon \ll 1$, a weak diffusion term of order ε and also a fast reaction term of order ε^{-1}. The temperature T has been normalized such that $0 \leq T_0 \leq 1$, where $T = 0$ represents the unburnt temperature (in an unstable region ahead of the front), while $T = 1$ is the temperature on the burnt side (in a region of equilibrium behind the front). The nonlinear reaction rate $f(T) = -\overline{K}T(1-T)$ is assumed to be of Kolmogorov-Petrovski-Piskunov (KPP) type:

$$\begin{aligned} &f(T) > 0 \quad \text{for} \quad T \in \mathbb{R} \setminus (0,1) \quad \text{and} \quad f(T) < 0 \quad \text{for} \quad T \in (0,1) \\ &f'(0) = \inf_{T>0}(\tfrac{f(T)}{T}) < 0. \end{aligned} \tag{1.2.2}$$

In the general case, we assume that the advecting velocity field $V(x, t, y, \tau) =$

$\overline{v}(x,t)+\lambda v(y,\tau)$ is periodic in the unit cubes in $\mathbb{R}^N\times\mathbb{R}$ for each fixed (x,t), and that it is a bounded, Lipschitz continuous function on $\mathbb{R}^N\times\mathbb{R}^+\times\mathbb{R}^N\times\mathbb{R}^+$. By [34] taking into consideration assumption (1.2.2) and the above mentioned restrictions on the velocity field, one can derive the effective equation for the propagation of the flame front.

Before describing (see section 1.2.2) the asymptotic behaviour of the system (1.2.1), we relate the variables T and Z by $T^\varepsilon = exp(\varepsilon^{-1}Z^\varepsilon)$. After this change of variables, equation (1.2.1) reduces to:

$$\begin{cases} Z_t^\varepsilon + V\cdot DZ^\varepsilon - \varepsilon k D^2 Z^\varepsilon - k|DZ^\varepsilon|^2 + (T^\varepsilon)^{-1} f(T^\varepsilon) = 0 \quad \text{in} \quad \mathbb{R}^N\times(0,\infty) \\ \\ Z^\varepsilon = \varepsilon \log T_0 \quad \text{in} \quad \text{int}\Omega\times\{t=0\} \\ \\ Z^\varepsilon(x,t)\to\ -\infty \quad \text{as} \quad T\to 0 \quad \text{and} \quad x\in\mathbb{R}^N\setminus\overline{\Omega} \end{cases} \tag{1.2.3}$$

where $\Omega = spt(T_0)$ is a compact set.

1.2.1 The Homogenization Procedure

Majda and Souganidis [34] developed a nonlinear averaging (homogenization) theory in order to obtain the evolution of the effective flame front in equation (1.2.1) in the limit $\varepsilon\to 0$. In this section, we present the homogenization procedure and derive formally the *cell problem* and the effective front equation by assuming that all the functions involved are smooth.

Derivation of the Cell Problem

The *cell problem* arises in the formal expansion of the averaging problem. The method involves finding the limit as $\varepsilon \to 0$ of an *approximate cell problem.* We introduce the homogenization problem by writing:

$$Z^\varepsilon(x,t,\frac{x}{\varepsilon^\alpha},\frac{t}{\varepsilon^\alpha}) = Z(x,t) + \varepsilon^\alpha w\left(\frac{x}{\varepsilon^\alpha},\frac{t}{\varepsilon^\alpha}\right) + O(\varepsilon^{2\alpha}) \tag{1.2.4}$$

and denote $(\frac{x}{\varepsilon^\alpha},\frac{t}{\varepsilon^\alpha}) \mapsto (y,\tau)$, where $\alpha \in (0,1)$. Then we can write:

$$\begin{aligned} Z_t^\varepsilon &= Z_t + w_\tau^\varepsilon \\ D_x Z^\varepsilon &= D_x Z + D_y w^\varepsilon \\ D_x^2 Z^\varepsilon &= D_x^2 Z + \varepsilon^{-\alpha} D_y^2 w^\varepsilon \end{aligned} \tag{1.2.5}$$

Substituting the above relations (1.2.5) into equation (1.2.3) we obtain:

$$\begin{cases} Z_t + w_\tau^\varepsilon + V\cdot(D_x Z + D_y w^\varepsilon) - \varepsilon k(D_x^2 Z + \varepsilon^{-\alpha} D_y^2 w^\varepsilon) - \\ k|D_x Z + D_y w^\varepsilon|^2 + (T^\varepsilon)^{-1} f(T^\varepsilon) = 0 \quad \text{in} \quad \mathbb{R}^N \times (0,\infty) \\ \\ Z^\varepsilon(x,0,\frac{x}{\varepsilon^\alpha},0) = Z(x,0) + \varepsilon^\alpha w\left(\frac{x}{\varepsilon^\alpha},0\right) + O(\varepsilon^{2\alpha}) \quad \text{for} \quad x \in \mathbb{R}^N. \end{cases} \tag{1.2.6}$$

In this formal expansion we can identify the *approximate cell problem*:

$$\begin{cases} w^\varepsilon_\tau + V \cdot (D_x Z + D_y w^\varepsilon) - \varepsilon k(D^2_x Z + \varepsilon^{-\alpha} D^2_y w^\varepsilon) - \\ k|D_x Z + D_y w^\varepsilon|^2 = -H^\varepsilon(\frac{x}{\varepsilon^\alpha}, \frac{t}{\varepsilon^\alpha}, D_x Z) \\ \\ w^\varepsilon \quad \text{is periodic in} \quad [0,1]^N \times [0,1]. \end{cases} \tag{1.2.7}$$

We let $P = D_x Z$ and $k = 1$. It will be shown later that in the limit $\varepsilon \to 0$ there exists a unique $H^\varepsilon(P, \frac{x}{\varepsilon^\alpha}, \frac{t}{\varepsilon^\alpha}) \to \overline{H}(P, x, t)$ and a smooth function $w^\varepsilon \to w$ satisfying the *cell problem*:

$$w_\tau - |P + D_y w|^2 + V \cdot (P + D_y w) = -\overline{H}(x, t, P). \tag{1.2.8}$$

In chapter 2, we will focus only on the case where the solution of the *cell problem* w is not be dependant on $\tau = \frac{t}{\varepsilon^\alpha}$ and P is a constant vector in $\mathbb{R}^N$. In this particular case, the *cell problem* reduces to:

$$-|P + D_y w|^2 + V \cdot (P + D_y w) = -\overline{H}(P). \tag{1.2.9}$$

and contains the *effective Hamiltonian* $(\overline{H}(P))$.

Derivation of the Effective Flame Front Equation

Majda and Souganidis [34] have proven that in the limit $\varepsilon \to 0$, the effective equation for the propagation of the flame front can be derived from equation (1.2.6). It is given by the following *variational inequality*:

$$\max(Z_t - \overline{H}(x,t,D_xZ) + f'(0), Z) = 0 \quad \text{in} \quad \mathbb{R}^N \times (0,\infty) \tag{1.2.10}$$

$$Z(x,0) = \begin{cases} 0 & : \quad x \in \Omega \\ -\infty & : \quad x \in \mathbb{R}^N \backslash \overline{\Omega} \end{cases} \tag{1.2.11}$$

where the set $\Gamma_t = \partial\{x \in \mathbb{R}^N : Z(x,t) < 0\}$ describes the location of the flame front for $t > 0$. Majda and Souganidis [34] showed that there are velocity fields V for which the effective flame front described by the *variational inequality* do not evolve according to Huygen's Principle. In our project we will confine ourselves to velocity fields $V = \overline{v} + \lambda v$ with constant mean velocity $\overline{v}$ and incompressible periodic velocity field v with zero mean (i.e. div $v = 0$ and $\langle v \rangle = 0$). In this case, the evolving effective front will be described by an evolution equation that is consistent with Huygen's Principle and the location of the effective front depends only on the initial location and the normal velocity to the front. More precisely, the *variational inequality* is equivalent to the following Hamilton-Jacobi equation [16], [34]:

$$u_t - F(Du) = 0 \tag{1.2.12}$$

where $u(x,t) = 0$ is the zero level set and equation (1.2.12) is the level set equation, also called the *geometric partial differential equation.* Here, the speed function $F(Du) = F(\mathbf{n})$ is the normal speed of propagation of the effective flame front. In section 1.3 it will be shown how an expression for

the normal speed of propagation was derived.

The above arguments can be made rigorous by using the theory of viscosity solutions. This is what we are planning to do in the next section.

1.2.2 Main Theorem of the Effective Flame Front

We start by introducing the main theorem for the flame front propagation. We focus on the general case, where the velocity field V is dependent on (x,t). In this situation, the effective flame front is described by the *variational inequality* (1.2.10). We will sketch the proof of the theorem by using the rigorous proof provided by Majda and Souganidis [34], [35]. In the rest of the project, solutions will be interpreted in the viscosity sense. An introduction to the basic theory of viscosity solutions is given in section 2.2 and we will refer to it when necessary.

Theorem 1.2.1 *Let T^ϵ be the solution of (1.2.1). Then as $\varepsilon \to 0$: $T^\epsilon \to 0$ locally uniformly in $\{Z < 0\}$ and $T^\epsilon \to 1$ locally uniformly in int$\{Z = 0\}$, where $Z \in C(\mathbb{R}^N \times [0,\infty))$ is the unique viscosity solution of the* variational inequality problem*:*

$$\max(Z_t - \overline{H}(x,t,D_xZ) + f'(0), Z) = 0 \quad in \quad \mathbb{R}^N \times (0,\infty) \tag{1.2.13}$$

$$Z(x,0) = \begin{cases} 0 & : \quad x \in \Omega \\ -\infty & : \quad x \in \mathbb{R}^N \backslash \overline{\Omega}. \end{cases} \tag{1.2.14}$$

Proof Theorem 1.2.1

By the Maximum Principle

$$\sup_{\varepsilon>0} \|Z^\varepsilon\|_{L^\infty(\mathbb{R}^N\times[0,\infty))} < \infty$$

the following definitions are given [34] for the half-relaxed limits:

Definition 1.2.2 *For each* $(x,t) \in \mathbb{R}^N \times (0,\infty)$, *the upper semicontinuous envelope* Z^* *and lower semicontinuous envelope* Z_* *of* Z^ε *are:*

$$Z_*(x,t) = \liminf_{(y,s)\to(x,t),\varepsilon\to 0} Z^\varepsilon(y,s),$$

$$Z^*(x,t) = \limsup_{(y,s)\to(x,t),\varepsilon\to 0} Z^\varepsilon(y,s)$$

Proposition 1.2.3

$$\max(Z_t^* - \overline{H}(x,t,D_xZ^*) + f'(0), Z^*) \le 0 \quad in \quad \mathbb{R}^N \times (0,\infty) \tag{1.2.15}$$

$$Z^*(x,0) = \begin{cases} 0 & : \quad x \in \Omega \\ -\infty & : \quad x \in \mathbb{R}^N \backslash \overline{\Omega}. \end{cases} \tag{1.2.16}$$

Proof

1. Since $T^\varepsilon = exp(\varepsilon^{-1}Z^\varepsilon)$ and $0 \le T^\varepsilon \le 1$ on $\mathbb{R}^N \times (0,\infty)$, we obtain:

$$Z^* \le 0 \quad \text{for} \quad \mathbb{R}^N \times (0,\infty). \tag{1.2.17}$$

We want to prove:

$$Z_t^* - \overline{H}(x,t,D_xZ^*) + f'(0) \le 0 \quad \text{in} \quad \mathbb{R}^N \times (0,\infty) \tag{1.2.18}$$

Therefore, we can fix a smooth test function ϕ and assume:

$$Z^* - \phi \quad \text{has a local maximum at a point} \quad (x_0, t_0) \in \mathbb{R}^N \times (0, \infty). \tag{1.2.19}$$

2. We introduce the *approximate cell problem* at (x_0, t_0):

$$\begin{cases} w_\tau^\varepsilon + V \cdot (D_x\phi + D_y w^\varepsilon) - \varepsilon k(D_x^2\phi + \varepsilon^{-\alpha} D_y^2 w^\varepsilon) - \\ \qquad k|D_x\phi + D_y w^\varepsilon|^2 = -H^\varepsilon(x_0, t_0, D_x\phi) \\ \\ \qquad w^\varepsilon \quad \text{is periodic in} \quad [0,1]^N \times [0,1] \end{cases} \tag{1.2.20}$$

where w^ε is chosen to be a solution of (1.2.20). Here the derivative of ϕ is evaluated at (x_0, t_0) and the derivative of w is evaluated at $(\frac{x_0}{\varepsilon^\alpha}, \frac{t_0}{\varepsilon^\alpha})$. The existence of a unique $\overline{H}^\varepsilon$ and some smooth w^ε satisfying (1.2.20) will be proven in section 2.3 for our case of interest. It can be shown as in [34], [35] that the solution is bounded:

$$\sup_\varepsilon (|w^\varepsilon| + |D_y w^\varepsilon|) < \infty \quad \text{in} \quad \mathbb{R}^N \times (0, \infty) \tag{1.2.21}$$

and

$$H^\varepsilon(x_0, t_0, P) \to \overline{H}(x_0, t_0, P). \tag{1.2.22}$$

3. By definition of Z^*, (1.2.3) and (1.2.21) we know that there exists a subsequence $\varepsilon_n \to 0$ and $(x_{\varepsilon_n}, t_{\varepsilon_n})$ such that:

$$\begin{aligned} Z^{\varepsilon_n}(x_{\varepsilon_n}, t_{\varepsilon_n}) &\to Z^*(x_0, t_0) \\ (x_{\varepsilon_n}, t_{\varepsilon_n}) &\to (x_0, t_0) \quad \text{as} \quad n \to \infty \end{aligned} \tag{1.2.23}$$

Then we fix a smooth, perturbed test function: $\phi^{\varepsilon_n}(x, t, \frac{x}{\varepsilon^\alpha}, \frac{t}{\varepsilon^\alpha}) = \phi(x,t) + \varepsilon_n^\alpha w^{\varepsilon_n}$ and assume:

$$Z^\varepsilon - \phi^{\varepsilon_n} \quad \text{has a local maximum at} \quad (x_{\varepsilon_n}, t_{\varepsilon_n}). \tag{1.2.24}$$

Applying the Maximum Principle and using (1.2.2), we obtain from (1.2.3):

$$\phi_t + w_s^{\varepsilon_n} - \varepsilon k(D_x^2\phi + \varepsilon_n^{-\alpha} D_y^2 w^{\varepsilon_n}) - k|D_x\phi + D_y w^{\varepsilon_n}|^2 + V \cdot (D_x\phi + D_y w^{\varepsilon_n}) + f'(0) \leq 0 \tag{1.2.25}$$

In (1.2.25), the derivatives of ϕ are evaluated at $(x_{\varepsilon_n}, t_{\varepsilon_n})$, the derivatives of w are evaluated at $(\varepsilon_n^{-1} x_{\varepsilon n}, \varepsilon_n^{-1} t_{\varepsilon n})$. Due to the boundedness of w^{ε_n}, property (1.2.22) and the assumed continuity of V, we obtain:

$$\phi_t(x_0, t_0) - \overline{H}(x_0, t_0, D_x\phi(x_0, t_0)) + f'(0) \leq o(1). \tag{1.2.26}$$

Letting $n \to \infty$ we obtain the inequality (1.2.15).

4. To verify (1.2.16), we fix $\mu > 0$ and choose $\rho \in C^\infty(\mathbb{R}^N)$. We let $\rho = 0$ on $\overline{\Omega}$, $\rho > 0$ on $\mathbb{R}^N \setminus \overline{\Omega}$ with $0 \leq \rho \leq 1$. We claim that:

$$\min(Z_t^* - \overline{H}(x,t,DZ^*) + f'(0), Z^* + \mu\rho) \leq 0 \quad \text{in} \quad \mathbb{R}^N \times \{0\}. \tag{1.2.27}$$

We choose a smooth test function ϕ and assume again that:

$$Z^* - \phi \quad \text{has a local maximum at a point} \quad (x_0, t_0) \in \mathbb{R}^N \times \{0\} \tag{1.2.28}$$

then either:

$$Z^*(x_0, 0) \leq -\mu\rho(x_0, 0) \tag{1.2.29}$$

or

$$\phi_t(x_0, 0) - \overline{H}(x_0, t_0, D\phi(x_0, t_0) + f'(0) \leq 0. \tag{1.2.30}$$

If $x_0 \in \overline{\Omega}$ then (1.2.29) is satisfied. Now if we assume $x_0 \in \mathbb{R}^N \setminus \overline{\Omega}$ and $Z^*(x_0, 0) > -\mu\rho(x_0) > -\infty$. Since $Z^{\varepsilon_n}(x_0, 0) = -\infty$ for all x near x_0, the points $(x_{\varepsilon_n}, t_{\varepsilon_n})$ are in $\mathbb{R}^N \times (0, \infty)$. We then repeat the arguments leading

to (1.2.26) and can show that (1.2.30) is satisfied.

5. If $x_0 \in \Omega$ then $Z^\varepsilon(x_0, 0) \to 0$ as $\varepsilon \to 0$ and therefore $Z^* = 0$ on $\Omega \times \{0\}$. To show that $Z^*(x, 0) = -\infty$ on $\mathbb{R}^N \setminus \overline{\Omega} \times \{0\}$, we assume $x_0 \in \mathbb{R}^N \setminus \overline{\Omega}$ and assume $Z^*(x_0, 0) > -\infty$. We can fix $\alpha > 0$ and define:

$$\phi^\alpha(x, t) = \alpha^{-1}|x - x_0|^2 + \lambda t_\alpha \tag{1.2.31}$$

Since Z^* is upper semi-continuous, $Z^* - \phi^\alpha$ has a maximum at $(x_\alpha, t_\alpha) \in \mathbb{R}^N \times [0, \infty)$. Then we have:

$$-\alpha^{-1}|x_\alpha - x_0|^2 \geq Z^*(x_\alpha, t_\alpha) - (\alpha^{-1}|x_\alpha - x_0|^2 + \lambda t_\alpha) \geq Z^*(x_0, 0) > -\infty \tag{1.2.32}$$

If $t_\alpha > 0$, we obtain:

$$\phi_t^\alpha(x_\alpha, t_\alpha) - \overline{H}(x_\alpha, t_\alpha, D\phi^\alpha(x_\alpha, t_\alpha)) + f'(0) \leq 0 \tag{1.2.33}$$

Therefore:

$$\lambda - \overline{H}(x_\alpha, t_\alpha, D\phi^\alpha(x_\alpha, t_\alpha)) + f'(0) \leq 0 \tag{1.2.34}$$

But by (1.2.32) $\lambda = \lambda(\alpha)$ is large enough such that we obtain a contradiction in (1.2.34). Therefore, $t_\alpha = 0$. If $Z^*(x_0, 0) > -\mu\rho(x_0)$ then (1.2.32)

implies $Z^*(x_\alpha, 0) > -\mu\rho(x_\alpha)$ for ρ small enough. However, by (1.2.27), we obtain (1.2.34) and $Z^*(x_0, 0) \leq -\mu\rho(x_0)$. Since $\rho > 0$ and $\mu > 0$ is chosen arbitrarily, we can see that (1.2.34) is not true. Therefore $Z^*(x_0, 0) = -\infty$ in $\mathbb{R}^N \setminus \overline{\Omega}$.

Proposition 1.2.4

$$\max(Z_{*t} - \overline{H}(x, t, DZ_*) + f'(0), Z_*) \geq 0 \quad in \quad \mathbb{R}^N \times (0, \infty) \tag{1.2.35}$$

$$Z_*(x, 0) = \begin{cases} 0 & : \quad x \in \Omega \\ -\infty & : \quad x \in \mathbb{R}^N \setminus \overline{\Omega} \end{cases} \tag{1.2.36}$$

Proof

The proof of this inequality is analogous to the proof of Proposition 1.2.3 and is explained in detail by Majda and Souganidis [34].

Consequences of Proposition 1.2.3 and Proposition 1.2.4

By the coercivity assumption of $\overline{H}(P)$ and the above propositions we can show the following uniqueness result of the solution to the *variational inequality* (1.2.13) and (1.2.14):

$$Z^* = Z_* = Z \quad \text{in} \quad \mathbb{R}^N \times (0, \infty) \tag{1.2.37}$$

Uniqueness of the solution can be proven by standard viscosity solution theory as will be shown in chapter 2. From Theorem 2.2.8 and the results of the propositions, we conclude that Z^* is a viscosity supersolution and Z_* is

a subsolution of (1.2.13) and (1.2.14) such that $Z_* \leq Z^*$ on $\partial\Omega_0$, where Ω_0 is a compact subset of $\mathbb{R}^N$. Then we have:

$$Z_* \leq Z^* \quad \text{in} \quad \overline{\Omega}_0. \tag{1.2.38}$$

By Corollary 2.2.4, equation (1.2.13) and (1.2.14) has at most one continuous viscosity solution $Z = Z^* = Z_*$ in $\mathbb{R}^N \times (0,\infty)$. In view of Definition 1.2.2 and the uniqueness of Z, we can conclude that $Z^\varepsilon \to Z$ on compact subsets of $\mathbb{R}^N \times (0,\infty)$. Using the change of variables relation, $T^\varepsilon = exp(\varepsilon^{-1}(Z + o(1)))$, it was shown in detail by Majda and Souganidis [34] that as $\varepsilon \to 0$:

$$T^\varepsilon \to 0 \quad \text{uniformly on compact subsets of} \quad \{Z < 0\}$$

and

$$T^\varepsilon \to 1 \quad \text{uniformly on compact subsets of} \quad int\{Z < 0\}.$$

1.3 Normal Speed of Propagation of the Front

As mentioned in section 1.2.1, we are interested in deriving the normal speed of propagation of the effective flame front which appears in the Hamilton-Jacobi equation of the type:

$$\begin{cases} u_t = F(Du) \quad \text{in} \quad \mathbb{R}^N \times (0,\infty) \\ \\ u(x,0) = u_0 \quad \text{on} \quad \mathbb{R}^N \end{cases} \tag{1.3.1}$$

where $F : \mathbb{R}^N \to \mathbb{R}$ is such that:

$$F(\lambda P) = \lambda F(P) \quad \text{for} \quad \lambda > 0. \tag{1.3.2}$$

In other words, we are interested in the dynamics of the front along a ray, the characteristic lines along which the gradient of the solution is constant. In our case $F(Du) = F(\mathbf{n})$ represents the normal velocity to the effective front and was derived by Majda and Souganidis [34]. It is given by:

$$F(\mathbf{n}) = \max_{\overline{H}^*(q)=f'(0)} (\mathbf{n} \cdot q) = inf_{\lambda>0}\{\lambda f'(0) + \lambda \overline{H}(\frac{\mathbf{n}}{\lambda})\} \tag{1.3.3}$$

where $\mathbf{n} = (\cos\theta, \sin\theta)$ denotes the unit exterior normal to the flame front, $\cdot$ represents the inner product, $\lambda = \frac{1}{r} > 0$, $f(T) = \overline{K}T(1-T)$ is the KPP reaction rate function and $\overline{K} > 0$ is the reaction rate constant.

Lemma 1.3.1 *Let $\overline{H} : \mathbb{R}^N \to \mathbb{R}$ be convex in P and assume $\overline{H}(P) \to \infty$ as $|P| \to \infty$. Then for all $p \in \mathbb{R}^N$ and all $a \geq inf_{P\in\mathbb{R}^N}\overline{H}^*(P)$:*

$$\max_{\overline{H}^*(P)=a} (P \cdot p) = inf_{\lambda>0}\{a\lambda + \lambda \overline{H}(\frac{p}{\lambda})\} \tag{1.3.4}$$

where $\overline{H}^*(P) = sup\{P \cdot Q - \overline{H}(Q)\}$ is the convex conjugate of $\overline{H}$.

Proof

By the definition of $\overline{H}^*$ we obtain:

$$a \geq P \cdot Q - \overline{H}(Q). \tag{1.3.5}$$

Let $Q = \frac{p}{\lambda}$:

$$a \geq P \cdot \frac{p}{\lambda} - \overline{H}(\frac{p}{\lambda}). \tag{1.3.6}$$

Then we have:

$$\max_{\overline{H}^*(P)=a} (p \cdot P) \leq \inf_{\lambda>0}\{a\lambda + \lambda \overline{H}(\frac{p}{\lambda})\}. \tag{1.3.7}$$

To prove the opposite inequality, we assume that $\overline{H}$ is smooth and $\lambda^* > 0$ is chosen such that:

$$\lambda^* a + \lambda^* \overline{H}(\frac{p}{\lambda^*}) = \inf_{\lambda>0}\{\lambda a + \lambda \overline{H}(\frac{p}{\lambda})\} \tag{1.3.8}$$

Here we let $a = \overline{H}^*(D\overline{H}(\frac{p}{\lambda}))$. By relations from convex analysis between $\overline{H}$ and $\overline{H}^*$ we obtain:

$$\max_{\overline{H}(P)=a}(p \cdot P) \geq (p, D\overline{H}(\frac{p}{\lambda_*})) = \lambda^* a + \lambda^* \overline{H}(\frac{p}{\lambda^*}) = \inf_{\lambda>0}\{\lambda a + \lambda \overline{H}(\frac{p}{\lambda})\}. \tag{1.3.9}$$

Chapter 2

Solutions of the Cell Problem

2.1 Introduction

This chapter will summarize the basics of the theory of viscosity solutions [1], [3], [10], [11], [12], [14], [15] . In chapter 1 we discussed how the *cell problem* was derived by a homogenization procedure developed by [32], and referred to by many other authors [18], [19], [33]. Here we will discuss the existence and uniqueness of the *effective Hamiltonian* ($\overline{H}(P)$) as well as the existence of viscosity solutions to the *cell problem.* We will study the asymptotic behaviour of the viscosity solution of the *evolution problem* (see 2.4.1 below) and show that this solution converges to the viscosity solution of (1.2.13) for large time [8]. We will conclude the chapter by proposing a new method of solving the *cell problem* by solving instead the Cauchy problem (see 2.5.8 below), which will be solved numerically in chapter 3.

2.2 Theory of Viscosity Solutions

This section introduces basic definitions and properties of viscosity solutions for the Hamilton-Jacobi Equation of the form:

$$H(x, Du(x)) = 0 \quad \text{in} \quad \mathbb{R}^N \tag{2.2.1}$$

The Hamiltonian $H : \mathbb{R}^N \to \mathbb{R}$ is given. The unknown function is $u : \mathbb{R}^N \to \mathbb{R}$ and $Du = (\frac{\partial u}{\partial x_1}, ..., \frac{\partial u}{\partial x_N})$.

We define the super-differential of a function u at the point x:

$$D^+u(x) = \{p \in \mathbb{R}^N : \limsup_{y \to x, y \in \Omega} \frac{u(y) - u(x) - p \cdot (y - x)}{|y - x|} \leq 0\} \tag{2.2.2}$$

and the sub-differential:

$$D^-u(x) = \{p \in \mathbb{R}^N : \limsup_{y \to x, y \in \Omega} \frac{u(y) - u(x) - p \cdot (y - x)}{|y - x|} \geq 0\} \tag{2.2.3}$$

The super- and sub-differentials have the following properties:

(i) $D^+u(x), D^-u(x)$ are closed, convex, possibly empty subsets of $\mathbb{R}^N$

(ii) If u is differentiable at x, then $D^+u(x) = D^-u(x) = Du(x)$

(iii) If for some x, both $D^+u(x)$ and $D^-u(x)$ are nonempty, then $D^+u(x) = D^-u(x) = Du(x)$.

Two equivalent definitions for viscosity solutions are introduced next.

Definition 2.2.1 *Given:*

$$H(x, Du(x)) = 0 \quad \textit{in} \quad \Omega \tag{2.2.4}$$

where Ω is an open, bounded domain in $\mathbb{R}^N$.

(i) $u \in C(\Omega)$ is a viscosity subsolution of (2.2.4) if:

$$H(x,p) \leq 0 \tag{2.2.5}$$

for all $x \in \Omega$ and for all $p \in D^+u(x)$

(ii) $u \in C(\Omega)$ is a viscosity supersolution of (2.2.4) if:

$$H(x,p) \geq 0 \tag{2.2.6}$$

for all $x \in \Omega$ and for all $p \in D^-u(x)$

(iii) u is a viscosity solution of (2.2.4) if it is a viscosity subsolution and supersolution of (2.2.4).

Definition 2.2.2 *A function $u \in C(\Omega)$ is a viscosity subsolution of (2.2.4) if at any local maximum point $x_0 \in \Omega$ of $u - \phi$ and for every smooth function $\phi \in C^1(\Omega)$:*

$$H(x_0, D\phi(x_0)) \leq 0. \tag{2.2.7}$$

A function $u \in C(\Omega)$ is a viscosity supersolution of (2.2.4) if at any local minimum point $x_0 \in \Omega$ of $u - \phi$ and for every smooth function $\phi \in C^1(\Omega)$

$$H(x_0, D\phi(x_0)) \geq 0. \tag{2.2.8}$$

Finally u is a viscosity solution of (2.2.4) if it is both a viscosity subsolution and supersolution of (2.2.4).

Theorem 2.2.3 *Definitions (2.2.1) and (2.2.2) are equivalent.*

Proof

For $u \in C(\Omega)$, the proof would consist of the following facts:

(i) $p \in D^+u$ iff there exists $\phi \in C^1(\Omega)$ such that $D\phi(x) = p$ and x is a local maximum point of $u - \phi$.

(ii) $p \in D^+u$ iff there exists $\phi \in C^1(\Omega)$ such that $D\phi(x) = p$ and x is a local minimum point of $u - \phi$.

Properties of Viscosity Solutions

Proposition 2.2.4 *Let $u \in C(\Omega)$ be a viscosity solution of (2.2.4). If u is differentiable, then u is also a classical solution of (2.2.4).*

Proof

Let $x \in \Omega$. Since u is differentiable , we have $D^+u(x) = D^-u(x) = Du(x)$ and by Definition (2.2.1):

$$0 \leq H(x, Du(x)) \leq 0.$$

Therefore,

$$H(x, Du(x)) = 0 \quad \text{in} \quad \Omega.$$

Proposition 2.2.5 *Let $u \in C(\Omega)$ be a viscosity solution of (2.2.4). If u is Lipschitz continuous in Ω, then:*

$$H(x, Du(x)) = 0 \quad \textit{almost everywhere in} \quad \Omega$$

Proof

By the Rademacher theorem, a Lipschitz continuous function is differentiable almost everywhere. The result follows by the fact that at every point of differentiability of u, the equation holds in the classical sense and (2.2.4) is satisfied.

Next we introduce a proposition that proves the stability of viscosity solutions.

Proposition 2.2.6 *Let $u^\varepsilon \in C(\Omega)$ be a viscosity solution of:*

$$H^\varepsilon(x, Du^\varepsilon(x)) = 0 \quad in \quad \Omega.$$

If $H^\varepsilon \to H$ locally uniformly in $\mathbb{R}^N \times \Omega$ and $u^\varepsilon \to u$, then u is a viscosity solution of (2.2.4).

Proof

The proof uses the following lemma [3], which we present here without proof.

Lemma 2.2.7 *Let $u \in C(\Omega)$ and let x_0 be a strict local maximum point of u in $\overline{D}(x_0, r)$ for some $r > 0$. Let $u^\varepsilon \in C(\Omega)$ converge locally uniformly to u in Ω. Then there exists a sequence $x_\varepsilon \to x_0$ such that*

$$u^\varepsilon(x_\varepsilon) \geq u^\varepsilon(x) \quad in \quad \overline{D}(x_0, r).$$

Theorem 2.2.8 *Let H satisfy Assumption 2.3.1 (see section 2.3) and let $u \in C(\overline{\Omega})$ be a viscosity subsolution of (2.2.4) and $v \in C(\overline{\Omega})$ be a viscosity supersolution of (2.2.4) such that $u \leq v$ on $\partial\Omega$. Then:*

$$u \leq v \quad in \quad \overline{\Omega}. \tag{2.2.9}$$

Corollary 2.2.9 *Equation (2.2.4) has at most one continuous viscosity solution in $\overline{\Omega}$.*

2.3 The Eigenvalue and Eigenfunction of the Cell Problem

In order to introduce the main theorem concerning the solutions of the *cell problem*, we make some assumptions on the Hamiltonian: $H(x, Du) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$:

Assumption 2.3.1 *There exists a continuous nondecreasing modulus function $m : [0,\infty) \to [0,\infty)$ such that $m(0^+) = 0$ and for all $x, y \in \mathbb{R}^N$ and $p \in \mathbb{R}^N$:*

$$|H(x,p) - H(y,p)| \leq m|x-y|(1+|p|).$$

Assumption 2.3.2 *$H(x,p) \to \infty$ as $|p| \to \infty$ uniformly for $x, p \in \mathbb{R}^N$.*

Assumption 2.3.3 *$H(x,P)$ is continuous in $\mathbb{R}^N \times \mathbb{R}^N$ for all $x,p \in \mathbb{R}^N$ and $\mathbb{Z}^N$-periodic, for $L \in \mathbb{Z}^N$ such that:*

$$H(x+L,p) = H(x,p).$$

For the periodic case, the following theorem applies:

Theorem 2.3.4 *Let $H : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ satisfy Assumption 2.3.1, 2.3.2, 2.3.3. Then for each $P \in \mathbb{R}^N$ there exists a unique constant $\overline{H}(P)$ and a periodic viscosity solution $w \in BUC(\mathbb{R}^N) \bigcap C^{0,1}(\mathbb{R}^N)$ of the* cell problem*:*

$$H(x, P+Dw) = \overline{H}(P) \tag{2.3.1}$$

where

$$|P+Dw|^2 - V \cdot (P+Dw) = H(x, P+Dw) \quad in \quad \mathbb{R}^N. \tag{2.3.2}$$

Remark: By Definition 2.2.2, if we understand w to solve (2.3.2) in the viscosity sense, then there is a smooth function ϕ such that if $w - \phi$ has a local maximum/minimum at a point $x_0 \in \Omega$. Then we know:

$$H(x_0, P + D\phi(x_0)) \leq (\geq) \overline{H}(P). \tag{2.3.3}$$

Sketch of Proof

Throughout this proof we will assume that V is bounded and Lipschitz continuous on $\mathbb{R}^N$. Here we present only key arguments in the proof of the theorem, working closely with the properties of viscosity solutions summarized in the previous section and the proofs presented in [33], [34].

We consider for each $\delta > 0$ the approximating equation:

$$\delta w^\delta + |P + Dw^\delta|^2 - V \cdot (P + Dw^\delta) = 0 \quad \text{in} \quad \mathbb{R}^N \times \mathbb{R}. \tag{2.3.4}$$

By Assumption 2.3.1, 2.3.2, 2.3.3, equation (2.3.2) has a unique, periodic solution $w^\delta \in BUC(\mathbb{R}^N) \bigcap C^{0,1}(\mathbb{R}^N)$ such that:

$$\sup_{0<\delta<1} (\|\delta w^\delta\|_\infty + \|Dw^\delta\|_\infty) < \infty. \tag{2.3.5}$$

Since w^δ's are bounded and periodic, by the stability property (Proposition 2.2.6) of viscosity solutions we can pass the limit in equation (2.3.4). We see that $w^\delta \to w$, where w is a bounded, Lipschitz continuous solution of (2.3.4). Moreover, $\delta w \to -\overline{H}(P)$ and we can write $\overline{H}(P) = lim_{\delta \to 0} H^\delta = \lim_{\delta \to 0}(-\delta w^\delta)$ uniformly on $\mathbb{R}^N$.

Since there exists at most one bounded viscosity solution of the *cell problem* (Proposition 2.2.6, Corollary 2.2.9), uniqueness of $\overline{H}(P)$ can be shown making use of arguments from the classical theory of viscosity solutions [3], [12].

2.4 The Evolution Equation

Barles and Souganidis [8] examined the asymptotic behaviour of the viscosity solution of the first order Hamilton-Jacobi evolution equation:

$$\begin{cases} u_t + H(x, Du) = 0 \quad \text{in} \quad \mathbb{R}^N \times (0, \infty) \\ \qquad u_0 = g \quad \text{on} \quad \mathbb{R}^N \times \{0\}. \end{cases} \tag{2.4.1}$$

Throughout the proof they assumed that $H(x, Du)$ and u_0 are periodic in x for all $x, p \in \mathbb{R}^N$ and $z \in \mathbb{Z}^N$, meaning:

$$H(x+z,p) = H(x,p) \quad \text{and} \quad u_0(x+z) = u_0. \tag{2.4.2}$$

Existence of solutions to (2.4.1) can be proven by the method of Perron [3], [15] whose rigorous proofs we omit.

Next, we will show how Assumption 2.3.1 can be modified to apply to the evolution equation appearing in (2.4.1). We follow the work presented by Barles [3] and denote:

$$u_t + H(x, Du) = \widetilde{H}(x, Du). \tag{2.4.3}$$

Suppose $\widetilde{H}$ is locally Lipschitz continuous and write:

$$|\frac{\partial \widetilde{H}}{\partial x}| \leq C(1 + |Du|). \tag{2.4.4}$$

Since $\widetilde{H} = 0$, we have $u_t = -H(x, Du)$ and

$$|\frac{\partial \widetilde{H}}{\partial x}| \leq C(1 + |Du| + |H(x, Du)|). \tag{2.4.5}$$

Rewriting 2.4.5 using the modulus function, we obtain the equivalent assumption for the evolution equation (2.4.1):

Assumption 2.4.1

$$|H(x, Du) - H(y, Du)| \leq m((|x-y|)(1 + |Du| + Q(x, y, Du))). \tag{2.4.6}$$

where

$$Q(x, y, Du) = \max(|H(x, Du)|, |H(y, Du)|).$$

The following theorem concerns itself with uniqueness of viscosity solutions of the initial value problem (2.4.1).

Theorem 2.4.2 *Under Assumption 2.4.1, there exists at most one viscosity solution u of (2.4.1)*

The proof of the above theorem uses standard uniqueness arguments from the theory of viscosity solutions [3], [17]. The formal proof will be omitted in this project. We are ready now to present an extension of Definition 2.2.2, that applies to equation (2.4.1):

Definition 2.4.3 *The function $u \in BUC(\mathbb{R}^N \times (0,\infty))$ is called a viscosity solution of the initial value problem (2.4.1) if:*
(i) $u_0 = g$ *on* $\mathbb{R}^N \times t = 0$
(ii) For each $\phi \in C^\infty(\mathbb{R}^N \times (0,\infty))$

$$\begin{cases} If \quad u-\phi \quad \text{has a local maximum at a point} \quad (x_0,t_0) \in \mathbb{R}^N \times (0,\infty) \\ \qquad\qquad \text{then} \quad \phi_t(x_0,t_0) - H(x_0, D\phi(x_0,t_0) \leq 0 \end{cases}$$

and

$$\begin{cases} If \quad u-\phi \quad \text{has a local minimum at a point} \quad (x_0,t_0) \in \mathbb{R}^N \times (0,\infty) \\ \qquad\qquad \text{then} \quad \phi_t(x_0,t_0) - H(x_0, D\phi(x_0,t_0) \geq 0. \end{cases}$$

One of the consequences of Theorem 2.4.2, is that it justifies the use of the relations presented in Definition 2.4.3(ii) as the basis for the theory of viscosity solutions of (2.4.1).

The existence of a constant $\overline{H}(P)$ depending only on H, such that $u+\overline{H}t$ remains bounded as $t \to \infty$, was presented in detail in [32]. This was done under the assumption that $H(x,P)$ is coercive:

$$H(x,P) \to \infty \quad \text{when} \quad |P| \to \infty \quad \text{uniformly in} \in \mathbb{R}^N. \tag{2.4.7}$$

Barles and Souganidis [8] showed that as $t \to \infty$:

$$u(\cdot,t) + \overline{H}(P)t \to w(\cdot) \quad \text{in} \quad C(\mathbb{R}^N) \tag{2.4.8}$$

where w is a viscosity solution of the *cell problem* (2.3.2). For their proof, they assumed without loss of generality, that $\overline{H}(P) = 0$. We introduce the formal theorem presented in their work:

Theorem 2.4.4 *If Assumptions 2.3.1, 2.3.2 hold and* $u \in BUC(\mathbb{R}^N \times (0,\infty))$ *is a* $\mathbb{Z}^N$*-periodic in* x *solution of (2.4.1), then there exists a* $\mathbb{Z}^N$*-periodic* $w \in BUC(\mathbb{R}^N)$ *such that:*

$$\begin{cases} H(x,Dw) = 0 \quad \textit{in} \quad \mathbb{R}^N \\ u(x,t) \to w(x) \quad \textit{uniformly in} \quad \mathbb{R}^N \quad \textit{as} \quad t \to \infty \end{cases} \tag{2.4.9}$$

Sketch of Proof

1. Since $u(x,t)$ is compact in $BUC(\mathbb{R}^N)$ for all $x \in \mathbb{R}^N$ and $t > 0$ and since the function $u(x,t)$ is periodic in x for all t, we can consider a subsequence $u(\cdot,t_n)$ with $t_n \to \infty$ converging uniformly in $\mathbb{R}^N$. Then by the Maximum Principle for viscosity solutions, we have for any $n,q \in \mathbb{N}$:

$$\|u(x,t+t_n) - u(x,t+t_q)\|_\infty \leq \|u(x,t_n - u(x,t_q)\|_\infty$$

By this inequality $(u(x,t+t_n))_n$ is a Cauchy sequence and therefore converges uniformly to $u_\infty \in BUC(\mathbb{R}^N \times (0,\infty)$ as $n \to \infty$.

2. Since u is uniformly continuous, it can be shown that for all $0 \leq t \leq \tau$ and for all $x \in \mathbb{R}^N$:

$$u_\infty(x,t) - u_\infty(x,\tau) \leq 0$$

3. We assume that u converges uniformly in x for $x \in \Omega$ as $t \to \infty$. Therefore we can say that u_∞ is constant in time on Ω.

5. The stability property of viscosity solutions applied to $u(x,t+t_n)$ yields that u_∞ is a solution of:

$$(u_\infty)_t + H(x, Du_\infty) = 0 \quad \text{in} \quad \mathbb{R}^N \times (0,\infty)$$

Since u_∞ is increasing linearly with respect to t for all $x \in \mathbb{R}^N$ we obtain:

$$H(x, Du_\infty) \leq 0 \quad \text{in} \quad \mathbb{R}^N \quad \text{for all} \quad t > 0$$

Also by the stability property:

$$H(0, Du_\infty) \leq 0 \quad \text{in} \quad \mathbb{R}^N$$

6. By the uniform convergence of u_n to u_∞:

$$-o_n(1) + u_\infty(x,t) \leq u(x,t+t_n) \leq u_\infty(x,t) + o_n(1)$$

Since u_∞ is increasing in t, we conclude that $u_\infty(x,t) \to w(x)$ uniformly in x as $t \to \infty$. Therefore taking the lim sup, lim inf of the above inequality as $t \to \infty$:

$$-o_n(1) + w(x) \leq \liminf_{y \to x} u(y,t) \leq \limsup_{y \to x} u(y,t) \leq w(x) + o_n(1)$$

As $n \to \infty$:

$$w = \limsup u = \liminf u \quad \text{in} \quad \mathbb{R}^N,$$

which proves that $u(x,t) \to w(x)$ uniformly as $t \to \infty$.

7. By the stability property of viscosity solutions w is a viscosity solution of $H(x, Dw) = 0$ in $\mathbb{R}^N$.

2.5 Proposed Method of Solution of the Cell Problem

In practice we assumed without loss of generality that the cell problem has a solution w which is biperiodic in the unit square domain: $\Omega = [0,1] \times [0.1]$. We restate the stationary Hamilton-Jacobi equation:

$$H((x,y), P + D_{x,y}w) = \overline{H}(P) \quad \text{where} \quad (x,y) \in \Omega, \tag{2.5.1}$$

which in our case is represented by the specific *cell problem*:

$$|P + D_{x,y}w|^2 - V(x,y) \cdot (P + D_{x,y}w) = \overline{H}(P) \quad \text{for} \quad (x,y) \in \Omega. \tag{2.5.2}$$

In the above equation P is a constant vector in $\mathbb{R}^2$, V is the velocity field periodic in both x and y with two separate scales, $D_{x,y} = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ represents the spatial gradient, $|\cdot|$ is the standard Euclidean norm in $\mathbb{R}^2$ and $\cdot$ is the Euclidean inner product in $\mathbb{R}^2$. In (2.5.2) we want to find for a given $P \in \mathbb{R}^2$ the pair $(w, \overline{H}(P))$.

The velocity field $V : \Omega \to \mathbb{R}^2$ depends on two separated scales, a constant large-scale velocity and a small-scale turbulent velocity field:

$$V = \overline{v} + \lambda v = \overline{\lambda}(\cos\overline{\theta}, \sin\overline{\theta}) + \lambda(\Psi_y, -\Psi_x) \tag{2.5.3}$$

where $\overline{v} \in \mathbb{R}^2$ is the constant mean flow speed, $\overline{\lambda}$ is the amplitude of the mean flow and $\overline{\theta}$ is its direction. The small scale velocity field v is assumed to be periodic, of zero mean (i.e. div$v = 0$) and unit amplitude. The other parameter describing the turbulent velocity field is λ, also called the turbulent intensity. In all of our experiments, the turbulent velocity field v is derived from the Childress-Soward stream periodic function:

$$\Psi(x, y) = \sin(2\pi x)\sin(2\pi y) + \delta\cos(2\pi x)\cos(2\pi y), \tag{2.5.4}$$

where the parameter $\delta = 1$ represents simple shear at 45 degrees, $\delta = 0.5$ represents a combination of eddies and shear layers and $\delta = 0$ represents a periodic array of eddies.

Solving (2.5.2) involves finding the viscosity solution of the equation coupled with the unknown constant $\overline{H}(P)$. By theorem 2.3.4 we know that there exists a unique eigenvalue $\overline{H}(P)$ for any given P, and strictly convex Hamiltonians:

$$\frac{\overline{H}(P)}{|P|} \to +\infty \text{ as } |P| \to +\infty. \tag{2.5.5}$$

After computing the *effective Hamiltonian* we can use it in the computation of the enhancement in the normal speed of propagation ($F_e(\mathbf{n})$) of the front due to turbulence. First we calculate $F(\mathbf{n})$, the normal speed of propagation due to turbulence by referring to (1.3.3). The expression simplifies

for practical purposes to:

$$F(\mathbf{n}) = \min_{r>0} \frac{\overline{H}(r\mathbf{n}) + f'(0)}{r}, \tag{2.5.6}$$

where $f'(0) = \frac{1}{4}$ has been chosen such that the laminar speed $F_L = 2(f'(0))^{\frac{1}{2}} = 1$. The laminar speed can be easily deduced from the cell problem by setting the solution $w = 0$. One can see that when there is no turbulence, $v = 0$ the speed of the normal front speed is simply $1 - \overline{\lambda}\mathbf{n}$. Then $F_e(\mathbf{n})$ is defined by:

$$F_e(\mathbf{n}) = F(\mathbf{n}) + \overline{\lambda}\mathbf{n} - 1 \tag{2.5.7}$$

We avoid the difficulties of solving (2.5.2) and solve instead the following Cauchy problem:

$$u_t + H((x,y), P + D_{x,y}u) = 0 \quad \text{for} \quad (x,y) \in \Omega, \tag{2.5.8}$$

which in our case is of the form:

$$u_t + |P + D_{x,y}u|^2 - V(x,y) \cdot (P + D_{x,y}u) = 0 \quad \text{for} \quad (x,y) \in \Omega, \tag{2.5.9}$$

which after expanding, simplifies to:

$$u_t + |P|^2 + 2P \cdot D_{x,y}u + |D_{x,y}u|^2 - V \cdot P - V \cdot D_{x,y}u = 0. \tag{2.5.10}$$

By the choice of a periodic velocity field, the solution u is necessarily periodic. It is also shown in [8] that u grows at most linearly in time. From section 2.4 we know that the periodic solution u of the Cauchy problem is related to the periodic solution w of the *cell problem* through the following asymptotic expression:

$$u + \overline{H}(P)t \to w \quad \text{as} \quad t \to \infty \tag{2.5.11}$$

Chapter 3

Numerical Algorithms: The First Order and ENO Schemes

3.1 Introduction

In this chapter we solve the Cauchy problem (2.5.10) numerically. We use a First Order Monotone Finite Difference (FOMFD) scheme and a second order extension of this numerical approximation, the Essentially Non-Oscillatory (ENO) scheme, developed by Osher and Shu [37]. The FOMFD scheme, converges to the viscosity solution of problem (2.5.10) and therefore to the desired eigenvalue [7]. No proof of convergence is available for the ENO schemes. For both schemes, the numerical approximations of the eigenvalue were used in computing the normal speed of propagation $F(\mathbf{n})$ and ultimately the enhancement due to turbulence of the normal speed $F_e(\mathbf{n})$.

3.2 Numerical Scheme Construction

We start by approximating numerically all the parameters of the problem (2.5.10), which we restate here:

$$u_t + |P|^2 + 2P \cdot D_{x,y}u + |D_{x,y}u|^2 - V \cdot P - V \cdot D_{x,y}u = 0. \tag{3.2.1}$$

These approximations will be used in both the FOMFD scheme and in the ENO scheme. We set $P = r\mathbf{n}$, where $r > 0$ is a real number and $\mathbf{n} = (\cos\theta, \sin\theta)$ denotes the unit exterior normal to the front. For convenience we assume that $\theta \in [0, \frac{\pi}{2}]$. Each mesh point of our domain of interest is denoted by $(i\Delta x, j\Delta y, k\Delta t)$, where $i, j \in \{0, 1, 2...N\}$, $\Delta x = \Delta y = \frac{1}{N}$ and $k \in \{0, 1, ...\}$. We let $u^k_{i,j} = u(i\Delta x, j\Delta y, k\Delta t)$ denote the approximation of u at that mesh point.

Next, we are defining the velocity field V. The mean flow field is a constant vector represented by $\overline{v} = \overline{\lambda}(\cos\overline{\theta}, \sin\overline{\theta})$. For the turbulent velocity field v we let $v_{i,j} = v(i\Delta x, j\Delta y)$, where $v_{1,i,j} = v_1(i\Delta x, j\Delta y)$ and $v_{2,i,j} = v_1(i\Delta x, j\Delta y)$.

We use well known definitions for the forward and backward finite differences of the partial derivatives of u:

$$\begin{aligned} D^+_x u^k_{i,j} &= \frac{u^k_{i+1,j} - u^k_{i,j}}{\Delta x}, \; D^-_x u^k_{i,j} = \frac{u^k_{i,j} - u^k_{i-1,j}}{\Delta x} \\ D^+_x u^k_{i,j} &= \frac{u^k_{i,j+1} - u^k_{i,j}}{\Delta y}, \; D^-_x u^k_{i,j} = \frac{u^k_{i,j} - u^k_{i,j-1}}{\Delta y} \end{aligned} \tag{3.2.2}$$

Since our solution u is required to be biperiodic (periodic in both x and y), both the FOMFD and the ENO scheme have to satisfy the periodic boundary

conditions:

$$u^k_{i+N,j} = u^k_{i,j}, \quad u^k_{i,j+N} = u^k_{i,j}. \tag{3.2.3}$$

The main approximation of our problem, is the estimate for the *effective Hamiltonian*:

$$\overline{H}(r\mathbf{n}) \approx -\frac{u^{k+1}_{i,j} - u^k_{i,j}}{\Delta t} \quad \text{for } k \text{ large} \tag{3.2.4}$$

3.3 Approximation of the Nonlinear Term

We need to take a closer look at the approximation of the nonlinear quadratic term $|D_{x,y}u|^2$ appearing in (2.5.10). This was derived previously by Rouy and Tourin [39] in a Shape-from-Shading problem. The numerical scheme was derived from the Dynamic Programming Principle (DPP). To summarize their findings [39], we can use the example of an Eikonal equation in $\mathbb{R}^2$ with Dirichlet boundary conditions:

$$\begin{cases} |Du(x)| = 1 \quad \text{in} \quad \Omega \\ \\ u(x) = 0 \quad \text{on} \quad \partial\Omega \end{cases} \tag{3.3.1}$$

Here $x \in \mathbb{R}^2$ and we assume that Ω is an open bounded domain in $\mathbb{R}^2$. The Eikonal equation can be rewritten as:

$$\sup_{|\alpha|\leq 1} \{-\alpha \cdot Du(x) - 1\} = 0 \quad \text{in} \quad \Omega \tag{3.3.2}$$

This equation corresponds to the optimal control problem in which the state variable $X(t) \in \overline{\Omega}$ evolves according to the following ordinary differential

equation:

$$\begin{cases} \frac{dX}{dt}(t) = \alpha(t) \\ X(0) = x \end{cases} \tag{3.3.3}$$

where α is a measurable control function with values in $\overline{\Omega}$. The cost to be minimized, depends on the initial position x and the control α:

$$J(x, \alpha) = \int_0^{\Delta t} 1 dx = \Delta t \tag{3.3.4}$$

where $\Delta t = \min\{t \geq 0 : X_{x,\alpha}(t) \in \partial\Omega\}$ is the first time at which the trajectory hits the boundary. We can restrict ourselves to a constant control α such that:

$$X_{x,\alpha}(\Delta t) = x - \alpha \Delta t. \tag{3.3.5}$$

For our application, we can choose the rectangular subset of $\mathbb{R}^2$, $\overline{\Omega} = [0, 1]^2$. If the initial point x belongs to the interior of Ω, we can choose Δt sufficiently small in order to make the trajectory stay inside Ω. Then the semi-discretized DPP is:

$$u(x) = \inf_{\alpha \in S} \left\{ \int_0^{\Delta t} 1 dt + u(X_{x,\alpha}(\Delta t)) \right\} \tag{3.3.6}$$

where $S = \{\alpha : \mathbb{R}^+ \to \mathbb{R}^2 \quad \text{measurable function s.t.} \quad |\alpha| \leq 1\}$ is the set of admissable controls. After substituting (3.3.5) into (3.3.6), dividing by Δt and transferring the infimum to the left-hand side, we obtain:

$$\sup_{|\alpha| \leq 1} \lim_{\Delta t \to 0} \frac{u(x) - u(x - \alpha \Delta t)}{\Delta t} = 1. \tag{3.3.7}$$

We continue by fully discretizing DPP. We assume that the neighbours of the mesh point (x_i, y_j), meaning (x_{i-1}, y_j), (x_{i+1}, y_j), (x_i, y_{j-1}), (x_i, y_{j+1}) belong

to the closure of Ω. Then, starting from a mesh point (x_i, y_j), we don't know in which of the triangles $\{(x_i, y_j), (x_{i-1}, y_j), (x_i, y_{j+1})\}$, $\{(x_i, y_j), (x_{i-1}, y_j), (x_i, y_{j-1})\}$, $\{(x_i, y_j), (x_i, y_{j-1}), (x_{i+1}, y_j)\}$, $\{(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1})\}$ the optimal control will lead the state. Each of these cases has an appropriate monotone approximation to $D_{x,y}u$. After performing a full discretization, we can solve explicitly the following optimal control problem:

$$\begin{aligned}\max\{\sup_{\alpha_1^2+\alpha_2^2\leq 1}((\alpha_1 D_x^- u_{i,j} + \alpha_2 D_y^- u_{i,j} - 1),\\ (\alpha_1 D_x^- u_{i,j} + \alpha_2 D_y^+ u_{i,j} - 1),\\ (\alpha_1 D_x^+ u_{i,j} + \alpha_2 D_y^- u_{i,j} - 1),\\ (\alpha_1 D_x^+ u_{i,j} + \alpha_2 D_y^+ u_{i,j} - 1))\} = 0\end{aligned}$$

in order to determine the appropriate finite differences. The solution to this optimization problem, provides the following approximation for problem (3.3.1):

$$\sqrt{(\max((D_x^+ u_{i,j})^-, (D_x^- u_{i,j})^+)^2 + (\max((D_y^+ u_{i,j})^-, (D_y^- u_{i,j})^+)^2} = 1 \quad (3.3.8)$$

where $f^+ = \max(f, 0)$, $f^- = \max(-f, 0)$. The scheme is monotone as it can be shown that the left hand side of (3.3.8) is nondecreasing with respect to $u_{i,j}$, $u_{i,j+1}$, $u_{i-1,j}$, $u_{i+1,j}$ and $u_{i,j-1}$. Finally we use this result to approximate the nonlinear spatial gradient term as:

$$|Du_{i,j}|^2 \approx \max((D_x^+ u_{i,j})^-, (D_x^- u_{i,j})^+)^2 + (\max((D_y^+ u_{i,j})^-, (D_y^- u_{i,j})^+)^2. \quad (3.3.9)$$

3.4 Numerical Algorithm: First Order Monotone Finite Difference (FOMFD) Scheme

The first numerical scheme implemented inside Ω is a first order monotone and explicit approximation. The scheme is explicit and therefore it is easy to compute at each time step $k+1$, the value of $(u_{i,j})_{i,j}^{k+1}$ from the value of the approximation at $(u_{i,j})_{i,j}^{k}$ as follows:

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \Delta t L(D_x^+ u_{i,j}^k, D_x^- u_{i,j}^k, D_y^+ u_{i,j}^k, D_y^- u_{i,j}^k) \tag{3.4.1}$$

where,

$$\begin{aligned} L(D_x^+ u_{i,j}^k, D_x^- u_{i,j}^k, D_y^+ u_{i,j}^k, D_y^- u_{i,j}^k) = & -|Du_{i,j}^k|^2 \\ & + (\overline{v} + \lambda v_{i,j}) \cdot ((D_x^+ u_{i,j}^k, D_y^+ u_{i,j}^k) + r\mathbf{n}) \\ & - r^2 - 2r\mathbf{n} \cdot (D_x^- u_{i,j}^k, D_x^- u_{i,j}^k). \end{aligned}$$

The only term which requires a different numerical approximation is $(\overline{v} + \lambda v) \cdot D_{(x,y)} u$. Therefore, we obtain four cases for (3.4.1) depending on the sign of $v + \overline{v}$.

Case 1: If (i,j) is such that $\overline{\lambda}\cos\overline{\theta} + \lambda v_{1,i,j} \geq 0$ and $\overline{\lambda}\sin\overline{\theta} + \lambda v_{2,i,j} \geq 0$

$$\begin{aligned} u_{i,j}^{k+1} = u_{i,j}^k + \Delta t \{ & -(\max\{(D_x^- u_{i,j}^k)^+, (D_x^+ u_{i,j}^k)^-\})^2 \\ & - (\max\{(D_y^- u_{i,j}^k)^+, (D_y^+ u_{i,j}^k)^-\})^2 \\ & + (\overline{v} + \lambda v_{i,j}) \cdot ((D_x^+ u_{i,j}^k, D_y^+ u_{i,j}^k) + rn) - r^2 - 2rn \cdot (D_x^- u_{i,j}^k, D_y^- u_{i,j}^k)\} \end{aligned}$$

Case 2: If (i,j) is such that $\overline{\lambda}\cos\overline{\theta}+\lambda v_{1,i,j}<0$ and $\overline{\lambda}\sin\overline{\theta}+\lambda v_{2,i,j}<0$

$$\begin{aligned}u_{i,j}^{k+1} = u_{i,j}^{k} + \Delta t\{&-(\max\{(D_x^- u_{i,j}^k)^+, (D_x^+ u_{i,j}^k)^-\})^2\\ &-(\max\{(D_y^- u_{i,j}^k)^+, (D_y^+ u_{i,j}^k)^-\})^2\\ &+(\overline{v}+\lambda v_{i,j})\cdot(D_x^- u_{i,j}^k, D_y^- u_{i,j}^k)+rn)-r^2-2rn\cdot(D_x^- u_{i,j}^k, D_y^- u_{i,j}^k)\}\end{aligned}$$

Case 3: If (i,j) is such that $\overline{\lambda}\cos\overline{\theta}+\lambda v_{1,i,j}\geq 0$ and $\overline{\lambda}\sin\overline{\theta}+\lambda v_{2,i,j}<0$

$$\begin{aligned}u_{i,j}^{k+1} = u_{i,j}^{k} + \Delta t\{&-(\max\{(D_x^- u_{i,j}^k)^+, (D_x^+ u_{i,j}^k)^-\})^2\\ &-(\max\{(D_y^- u_{i,j}^k)^+, (D_y^+ u_{i,j}^k)^-\})^2\\ &+(\overline{v}+\lambda v_{i,j})\cdot((D_x^+ u_{i,j}^k, D_y^- u_{i,j}^k)+rn)-r^2-2rn\cdot(D_x^- u_{i,j}^k, D_y^- u_{i,j}^k)\}\end{aligned}$$

Case 4: If (i,j) is such that $\overline{\lambda}\cos\overline{\theta}+\lambda v_{1,i,j}<0$ and $\overline{\lambda}\sin\overline{\theta}+\lambda v_{2,i,j}\geq 0$

$$\begin{aligned}u_{i,j}^{k+1} = u_{i,j}^{k} + \Delta t\{&-(\max\{(D_x^- u_{i,j}^k)^+, (D_x^+ u_{i,j}^k)^-\})^2\\ &-(\max\{(D_y^- u_{i,j}^k)^+, (D_y^+ u_{i,j}^k)^-\})^2\\ &+(\overline{v}+\lambda v_{i,j})\cdot(D_x^- u_{i,j}^k, D_y^+ u_{i,j}^k)+rn)-r^2-2rn\cdot(D_x^- u_{i,j}^k, D_y^- u_{i,j}^k)\}.\end{aligned}$$

To ensure the monotonicity and stability of the scheme, we choose at every time step k, an adaptive time step Δt_{CFL} calculated by differentiating the right side of (3.4.1) with respect to $u_{i,j}^k$. The Courant-Friedrichs-Lewy (CFL) condition is as follows:

$$\Delta t_{CFL} \leq \min_{i,j}\left\{\frac{\Delta x}{A}\right\} \tag{3.4.2}$$

where

$$\begin{aligned}A =&2(\max\{(D_x^- u_{i,j}^k)^+, (D_x^+ u_{i,j}^k)^-\}+\max\{(D_y^- u_{i,j}^k)^+, (D_y^+ u_{i,j}^k)^-\})\\ &+|\overline{\lambda}\cos\overline{\theta}+\lambda v_{1,i,j}|+|\overline{\lambda}\sin\overline{\theta}+\lambda v_{2,i,j}|+2r(\cos\theta+\sin\theta).\end{aligned}$$

The algorithm is summarized as follows:

(i) Step $k = 0$: we choose $u^0 = u^0_{i,j}$, $\overline{H}^0 = \overline{H}^0_{i,j}$ for all $(i,j) \in \Omega$.

(ii) Step $k+1$: for each i, j from 1 to N, we calculate Δt_{CFL} and perform the iterative step

$$u^{k+1}_{i,j} = u^k_{i,j} + \Delta t_{CFL} L(D^+_x u^k_{i,j}, D^-_x u^k_{i,j}, D^+_y u^k_{i,j}, D^-_y u^k_{i,j})$$

and also

$$\overline{H}^{k+1}_{i,j} = -\frac{u^{k+1}_{i,j} - u^k_{i,j}}{\Delta t}.$$

Each iteration yields a collection of eigenvalues $\overline{H}(r\mathbf{n})$ indiced by (i,j). We stop when the L^∞ norm of the difference between two successive eigenvalues falls below a given threshold. The eigenvalue is then taken to be the average over all pairs (i,j). The algorithm is applied to a number of values of r and the minimization is then performed using a simple line search.

3.4.1 Numerical Solutions: FOMFD

To obtain results that can be compared to the ones obtained by Bourlioux and Khouider [9] we used the same parameters. We first choose a velocity field V with zero mean flow ($\overline{\lambda} = 0$, $\overline{\theta} = 0$). We also assume that $\theta = \frac{\pi}{4}$ which yields $\mathbf{n} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$. The intermediate velocity field v was obtained as discussed previously in equations (2.5.3) and (2.5.4). We also successively set the turbulence intensity to $\lambda = 0.4, 1.6$ and 6.4.

Parameter $\delta = 1$

In this case the normal to the flame front is aligned with the shearing direction described by the parameter δ. We want to check that the numerical solutions are consistent to the theoretical results presented in chapter 3 as well as the numerical results obtained through another method by Bourlioux and Khouider [9]. For the case $\delta = 1$, the velocity field v is given by:

$$v(x,y) = \frac{1}{\sqrt{2}}(\sin 2\pi(x-y), \sin 2\pi(x-y)). \tag{3.4.3}$$

As shown by Barles and Souganidis [8], the solution u grows at most linearly in time. Therefore, this numerical method is stable if the approximated time derivative of the solution converges before u can reach large values. In all our experiments it was observed that u stays roughly of the same order of magnitude as the initial condition or about 10 times larger at most.

To illustrate this convergence in time of the time derivative and therefore of the approximated *effective Hamiltonian*, we present in Figure 3.4.1 the approximation of the eigenvalue versus the number of iterations for $N = 16, \Delta t = 0.005, \lambda = 1.6$ and $r = 0.5$.

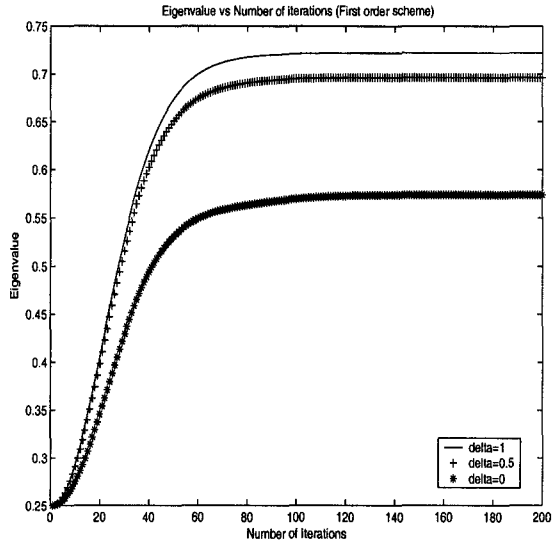


Figure 3.4.1: Convergence of the time derivative to the eigenvalue $\overline{H}(P)$.

As desired, the scheme shows fast convergence to the eigenvalue. With the time increment Δt at 200 iterations, the time $t = 1$ and the L^∞ norm of the difference between two successive time derivatives was approximately $6.7e{-}8$.

To show the rate of convergence of the scheme to the desired eigenvalue, we show in Figure 3.4.2. the logarithm of the L^∞ norm of the difference between two successive time derivatives for $\delta = 1$, $\lambda = 1.6$, $N = 16$. For this test, r was set successively to $r = 0.1, 0.5$ and 0.7. In this case, the time interval is between $[0.2, 1.5]$. One can check that at $t = 1.5$, the L^∞ norm of the difference between two successive time derivatives is $1.75e - 11$.

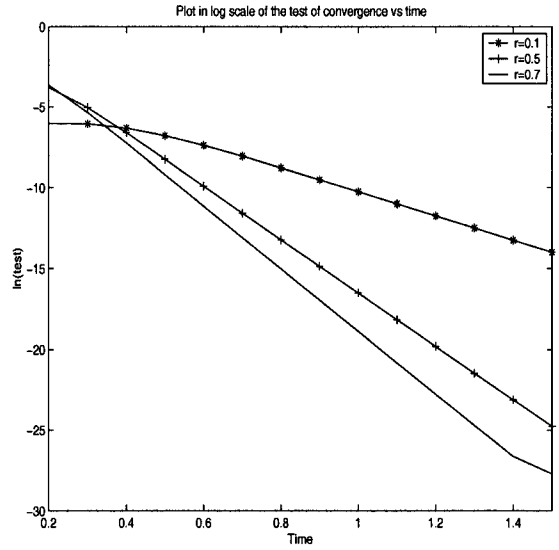


Figure 3.4.2: Convergence rate of the algorithm to $\overline{H}(P)$.

To test some properties of the *effective Hamiltonian*, we plot $\overline{H}(r\mathbf{n})$ as a function of the parameter r when $\lambda = 1.6$ and $\delta = 1$ for 3 different numbers of grid points, $N = 16, 32$ and 64. The result is presented in Figure 3.4.3 and we can conclude that $\overline{H}(P)$ is convex and coercive.

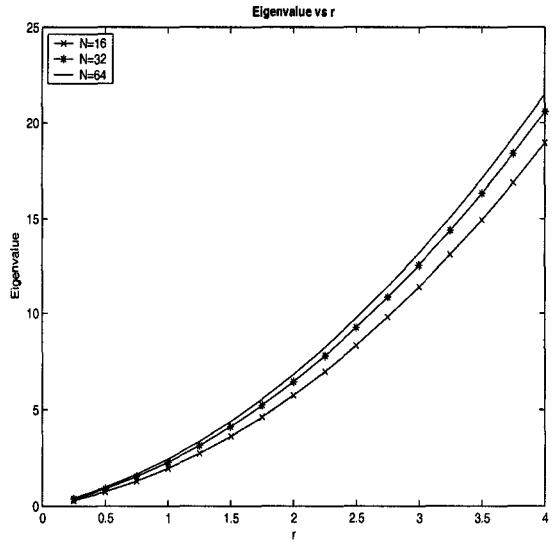


Figure 3.4.3: Convexity and coercivity property of $\overline{H}(P)$.

In Figure 3.4.4 we present the graph of the function $G(r)$ to be minimized. The same set of parameters as in Figure 3.4.3. was used.

$$G(r) = \frac{\overline{H}(r\mathbf{n}) + \frac{1}{4}}{r}$$

The result of this figure helps identify r for which the normal speed of propagation $F(\mathbf{n})$ should be computed. We observe that the minimum point of $G(r)$ is located at $r = 0.5$. This result is consistent with the result obtained by the explicit calculations by Majda and Souganidis [34].

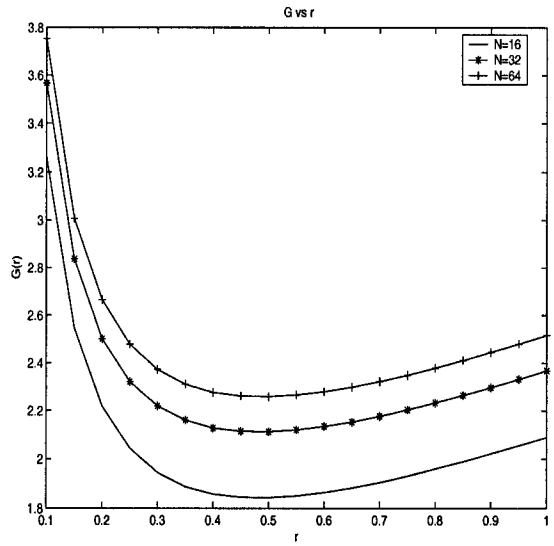


Figure 3.4.4: $G(r)$ as a function of r.

Parameters $\delta = 0$ and $\delta = 0.5$

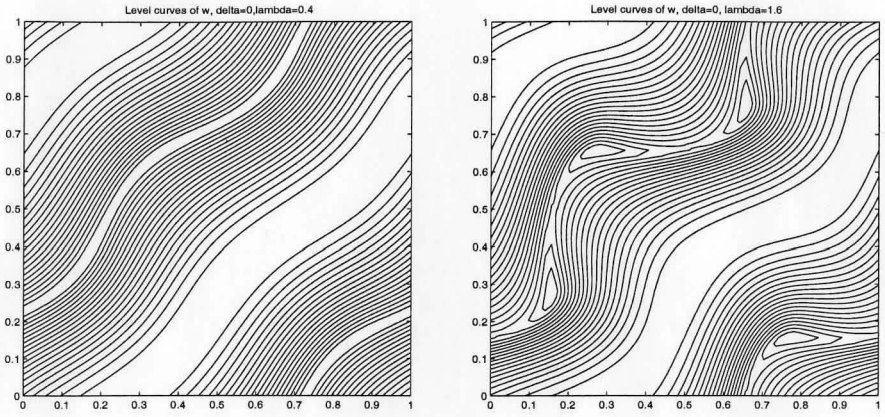
For $\delta = 0.5$, the normalized v is given by:

$$\begin{aligned} v(x,y) = &\frac{2}{\sqrt{5}}(\sin 2\pi x \cos 2\pi y - \frac{1}{2}\cos 2\pi x \sin 2\pi y, \\ &\frac{1}{2}\sin 2\pi x \cos 2\pi y - \cos 2\pi x \sin 2\pi y) \end{aligned} \tag{3.4.4}$$

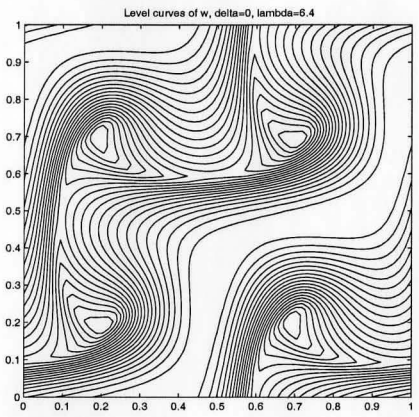
and for $\delta = 0$:

$$v(x,y) = (\sin 2\pi x \cos 2\pi y, -\cos 2\pi x \sin 2\pi y). \tag{3.4.5}$$

In Figure 3.4.5 we show the level curves of the solution w of the *cell problem* computed here only for $\delta = 0$, when the flame front corresponds to an array of eddies.



(a) (b)



(c)

Figure 3.4.5: (a) Level curves of w for δ=0, $\lambda = 0.4$, (b) Level curves of w for δ=0, $\lambda = 1.6$, (c) Level curves of w for δ=0, $\lambda = 6.4$

For each (i, j) and for large k:

$$w_{i,j}^{k+1} = u_{i,j}^{k+1} + \overline{H}_{i,j}^{k} * (k\Delta t)$$

The method of obtaining the level curves follows from the theory presented in chapter 3. The grid contains 64×64 points and $\lambda = 0.4, 1.6$ and 6.4. One can see that the front becomes wrinkled as the turbulent intensity λ is increased.

Finally, we tested the behaviour of the eigenvalue in relation to angle θ (see $\mathbf{n} = (\cos\theta, \sin\theta)$) and displayed the result in Figure 3.4.6. The parameters used in this computation are $N = 16$, $\lambda = 1.6$, $r = 0.5$ and θ was allowed to vary in the interval $[0.015, 1.5]$.

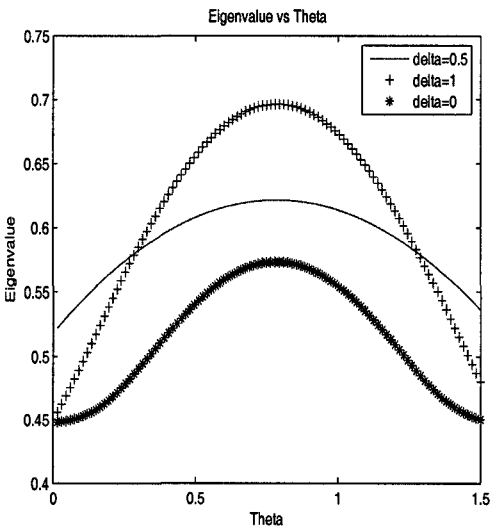


Figure 3.4.6: Relationship between $\overline{H}(P)$ and angle θ (FOMFD)

3.5 Numerical Algorithm: The Essentially Non-Oscillatory (ENO) Scheme

The solution to (2.5.10) may develop discontinuities in the gradient even if the initial condition is a smooth function. The ENO scheme is a more efficient finite difference method which is expected to yield more accurate results when compared to the first order schemes [37],[38]. Higher order schemes with second or higher rates of convergence are appealing because they would provide more accurate results in the regions where the solution is smooth. In general their convergence cannot be proven as they are not strictly monotone and a theory for ENO schemes is still unavailable. Here we construct an ENO scheme of order two in space and time, based on the first-order monotone approximation. The truncation error is of order two and our approximations are expected to be second order accurate [37]. Here we assumed that the function u of interest is smooth in a region around $u_{i,j}$. The ENO scheme uses a second order approximation to the first derivative with respect to x, y, t by Taylor expansion. Time accuracy is obtained by a Total-Variation-Diminishing (TVD) Runge-Kutta type time discretization [37], [38]. TVD Runge-Kutta type time discretization is used in order to maintain TVD condition $TV(u^{n+1}) \leq TV(u^n)$, where $TV(u) = \sum_j |u_{j+1} - u_j|$, while achieving higher order accuracy in time with a different time step restriction $\Delta t \leq C_{CFL}\Delta t_1$. To deduce the CFL coefficient (C_{CFL}), one starts from a general form for explicit r-order Runge-Kutta

method:

$$\begin{aligned} u^{(m)} &= u^{(0)} + \Delta t \sum_{l=0}^{m-1} c_{m,l} L(u^{(l)}) \qquad m = 1, ..., r \\ u^{(0)} &= u^k, \quad u^{(r)} = u^{k+1}. \end{aligned} \tag{3.5.1}$$

For $l = 0, ..., m-1$, we introduce at each stage, $\alpha_{m,l}$ satisfying $\alpha_{m,l} \geq 0$ and $\sum_{l=0}^{m-1} \alpha_{m,l} = 1$. We can rewrite (3.5.1):

$$\begin{aligned} u^{(m)} &= \sum_{l=0}^{m-1} [\alpha_{m,l} u^{(l)} + \beta_{m,l} \Delta t L(u^{(l)})] \quad m = 1, ..., r \\ \beta_{m,l} &= c_{m,l} - \sum_{s=l+1}^{m-1} c_{s,l} \alpha_{s,m}. \end{aligned} \tag{3.5.2}$$

By rewriting (3.5.2) as:

$$u^{(m)} = \sum_{l=0}^{m-1} \alpha_{m,l} \left[u^{(l)} + \frac{\beta_{m,l} \Delta t}{\alpha_{m,l}} L(u^{(l)}) \right] \quad m = 1, ..., r \tag{3.5.3}$$

it is easy to prove [23] that if $\beta_{m,l} \geq 0$, the Runge-Kutta method is TVD under the CFL condition:

$$\Delta t \leq \left(\min_{m,l} \frac{\alpha_{m,l}}{\beta_{m,l}} \right) \Delta t_1 \tag{3.5.4}$$

where $C_{CFL} = \min_{m,l} \frac{\alpha_{m,l}}{\beta_{m,l}}$. For schemes up to third order, $C_{CFL} = 1$ needs to be satisfied in order to obtain a TVD Runge-Kutta type time discretization [37]. The coefficients $\alpha_{m,l}$ and $\beta_{m,l}$, for $l = 0, ..., m-1$ and $m = 1, ..., r$, are deduced from the Taylor expansion.

In this project we deal with a second order scheme ($r = 2$). We obtain $u_{i,j}^{k+1}$ from $u_{i,j}^k$ by the following second order TVD Runge-Kutta procedure:

$$\begin{cases} u_{i,j}^{(1)} = \alpha_{1,0} u_{i,j}^{(0)} + \beta_{1,0} \Delta t L(D_x^+ u_{i,j}^{(0)}, D_x^- u_{i,j}^{(0)}, D_y^+ u_{i,j}^{(0)}, D_y^- u_{i,j}^{(0)}) \\ u_{i,j}^{(2)} = \alpha_{2,0} u_{i,j}^{(0)} + \alpha_{2,1} u_{i,j}^{(1)} \\ \qquad +\beta_{2,1} \Delta t L(D_x^+ u_{i,j}^{(1)}, D_x^- u_{i,j}^{(1)}, D_y^+ u_{i,j}^{(1)}, D_y^- u_{i,j}^{(1)}) \\ u_{i,j}^{(0)} = u_{i,j}^k, \quad u_{i,j}^{(2)} = u_{i,j}^{k+1} \\ C_{CFL} = 1. \end{cases} \tag{3.5.5}$$

We can choose $\beta_{1,0}$ and $\alpha_{2,1}$ as free parameters. The other coefficients are then:

$$\begin{cases} \alpha_{1,0} = 1 \\ \alpha_{2,0} = 1 - \alpha_{2,1} \\ \beta_{2,0} = 1 - \frac{1}{2\beta_{1,0}} - \alpha_{2,1}\beta_{1,0} \\ \beta_{2,1} = \frac{1}{2\beta_{1,0}} \end{cases} \tag{3.5.6}$$

The positive coefficients $\alpha_{1,0} = \beta_{1,0} = 1$, $\alpha_{2,0} = \alpha_{2,1} = \frac{1}{2}$, $\beta_{2,0} = 0$ and $\beta_{2,1} = \frac{1}{2}$ were deduced in [37], [38]. Then (3.5.5) is equivalent to the following procedure:

$$\begin{cases} u_{i,j}^{(1)} = u_{i,j}^{(0)} + \Delta t L(D_x^+ u_{i,j}^{(0)}, D_x^- u_{i,j}^{(0)}, D_y^+ u_{i,j}^{(0)}, D_y^- u_{i,j}^{(0)}) \\ u_{i,j}^{(2)} = u_{i,j}^{(0)} + \frac{1}{2} \Delta t L(D_x^+ u_{i,j}^{(0)}, D_x^- u_{i,j}^{(0)}, D_y^+ u_{i,j}^{(0)}, D_y^- u_{i,j}^{(0)}) \\ \qquad +\frac{1}{2} \Delta t L(D_x^+ u_{i,j}^{(1)}, D_x^- u_{i,j}^{(1)}, D_y^+ u_{i,j}^{(1)}, D_y^- u_{i,j}^{(1)}) \\ u_{i,j}^{(0)} = u_{i,j}^k, \quad u_{i,j}^{(2)} = u_{i,j}^{k+1} \\ C_{CFL} = 1. \end{cases} \tag{3.5.7}$$

Next we concern ourselves with the approximation of the gradient. Here the upwind and downwind spatial finite differences are of second order approximation. At each (i,j) the possible approximations for the gradient are deduced by Taylor expansion along the x direction (fixing j):

$$\begin{aligned}
D_x^+ u_{i,j}^k &= \frac{u_{i+1,j}^k - u_{i,j}^k}{\Delta x} - \frac{u_{i+1,j}^k + u_{i-1,j}^k - 2u_{i,j}^k}{\Delta x}\Delta x^2 \\
D_x^+ u_{i,j}^k &= \frac{u_{i+1,j}^k - u_{i,j}^k}{\Delta x} - \frac{u_{i+2,j}^k + u_{i,j}^k - 2u_{i+1,j}^k}{\Delta x}\Delta x^2 \\
D_x^- u_{i,j}^k &= \frac{u_{i,j}^k - u_{i-1,j}^k}{\Delta x} - \frac{u_{i+1,j}^k + u_{i-1,j}^k - 2u_{i,j}^k}{\Delta x}\Delta x^2 \\
D_x^- u_{i,j}^k &= \frac{u_{i,j}^k - u_{i-1,j}^k}{\Delta x} - \frac{u_{i-2,j}^k + u_{i,j}^k - 2u_{i-1,j}^k}{\Delta x}\Delta x^2
\end{aligned} \tag{3.5.8}$$

Similarly we approximate the gradients along the y direction to obtain $D_y^+ u_{i,j}$ and $D_y^- u_{i,j}$.

For stability (oscillation-free spatial gradients), we have used a min-mod limiter rather than a TVD limiter as for the stability of the time derivative approximations. For convenience only the x-dimensional differences will be used to outline this technique:

$$\begin{aligned}
&\text{min-mod}(u_{i+1,j}^k + u_{i-1,j}^k - 2u_{i,j}^k, u_{i+2,j}^k + u_{i,j}^k - 2u_{i+1,j}^k) = \\
&\quad \min(|u_{i+1,j}^k + u_{i-1,j}^k - 2u_{i,j}^k|, |u_{i+2,j}^k + u_{i,j}^l - 2u_{i+1,j}^k|)
\end{aligned} \tag{3.5.9}$$

and

$$\begin{aligned}\text{min-mod}(u^k_{i+1,j} + u^k_{i-1,j} - 2u^k_{i,j}, u^k_{i,j} + u^k_{i-2,j} - 2u^k_{i-1,j}) = \\ \min(|u^k_{i+1,j} + u^k_{i-1,j} - 2u^k_{i,j}|, |u^k_{i,j} + u^k_{i-2,j} - 2u^k_{i-1,j}|)\end{aligned} \tag{3.5.10}$$

is zero if the sign of the arguments are different or if their product is equal to zero and different than zero otherwise.

The algorithm is summarized as follows:

(i) Step $k = 0$: we choose $u^0 = u^0_{i,j}$, $\overline{H}^0 = \overline{H}^0_{i,j}$ for all $(i\Delta x, j\Delta y) \in \Omega$.

(ii) Step $k + 1$:

$$\begin{cases} u^*_{ij} = u^k_{ij} + \Delta t L(D^+_x u^k_{ij}, D^-_x u^k_{ij}, D^+_y u^k_{ij}, D^-_y u^k_{ij}) \\ u^{k+1}_{ij} = u^k_{ij} + \frac{1}{2}\Delta t L(D^+_x u^k_{ij}, D^-_x u^k_{ij}, D^+_y u^k_{ij}, D^-_y u^k_{ij}) \\ \qquad +\frac{1}{2}\Delta t L(D^+_x u^*_{ij}, D^-_x u^*_{ij}, D^+_y u^*_{ij}, D^-_y u^*_{ij}) \end{cases}$$

and

$$\overline{H}^{k+1}_{i,j} = -\frac{u^{k+1}_{i,j} - u^k_{i,j}}{\Delta t}.$$

The number of iterations performed is determined by a test of convergence performed for each (i, j) by taking the L^∞ of the difference between two successive eigenvalues.

The solution of w of the *cell problem* is semiconcave [18] and satsifies:

$$w(x + h, y + k) - 2w(x, y) + w(x - h, y - k) \leq C(h^2 + k^2)$$

for any (x, y) and h, k sufficiently small. Experimental results in [37] show that ENO schemes perform well when used for computing semiconcave functions. The semiconcavity of w prevented the ENO scheme from undergoing

large oscillations. A single computation for a specific r, 32×32 grid and $\delta = 1$ took less than a second.

3.5.1 Numerical Solutions: ENO

In most cases, the ENO algorithm converges as fast as the monotone scheme. This property of convergence of the algorithm is presented in Figure 3.5.1. As for the first order scheme, the parameters used for this convergence test are: $N = 16$, $\lambda = 1.6$, $\Delta t = 0.005$, $r = 0.5$, $\delta = 0, 0.5$ and 1. In some instances, the ENO scheme undergoes some small amplitude oscillations for a much longer time before converging to the eigenvalue. The stability of the algorithm does not seem affected.

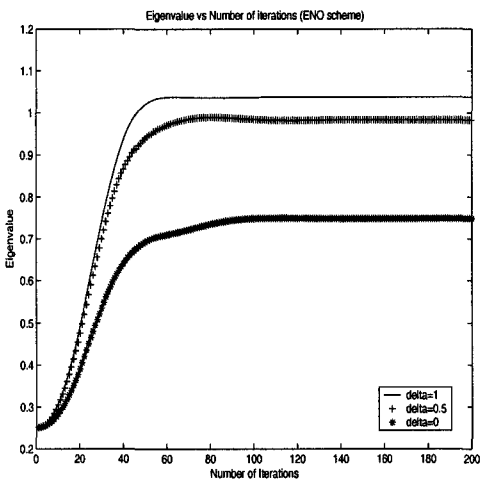
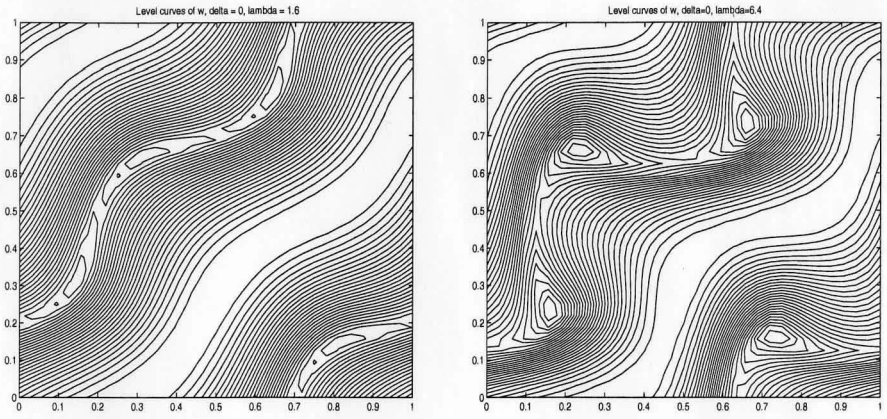


Figure 3.5.1: Convergence of the time derivative to $\overline{H}(P)$ (ENO)

In Figure 3.5.2 we present again the level set of the solution w to the *cell problem* using the same parameters as for FOMFD. The level sets are similar to the ones obtained for FOMFD.



(a) (b)

Figure 3.5.2: (a) Level curves of w for $\delta = 0$ and $\lambda = 1.6$

(b) Level curves of w for $\delta = 0$ and $\lambda = 6.4$

In Figure 3.5.3 we analyze the behaviour of the eigenvalue by varying angle θ. We used parameters $N = 16$, $\lambda = 1.6$, $r = 0.5$, $\Delta t = 0.005$ and angle θ between $[0.015, 1.5]$.

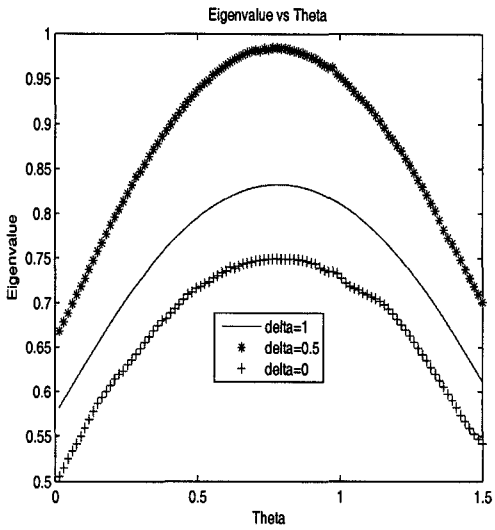


Figure 3.5.3: Relationship between $\overline{H}(P)$ and angle θ (ENO)

3.6 Summary of Results: The Speed Enhancement

The final goal of the project was the computation of the normal speed enhancement $F_e(\mathbf{n})$ using the values of the *effective Hamiltonian.* For all of our experiments we assumed that $\theta = \frac{\pi}{4}$ such that $\mathbf{n} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$. We compute $F_e(\mathbf{n})$ for both the ENO and FOMFD schemes. We compare our results with the explicit solution $F_e(\mathbf{n}) = \lambda$ [9], [34], and compute the error. Referring to Table 3.6.1, we observe that for small values of λ, the error is roughly of the same order as the truncation error. The FOMFD scheme is more

accurate for smaller values of λ than for larger $\lambda = 6.4$. For large λ, the accuracy of the FOMFD scheme falls between $\sqrt{\Delta x}$ and Δx. The accuracy of the ENO scheme is slightly smaller, yet still of order two. When λ is not too large, we obtain better rates than the ones predicted by the theory for Hamilton-Jacobi-Bellman equations [6].

Table 3.6.1: Numerical results for $F_e(\mathbf{n})$ for $\delta = 1$

λ	Reference F_e	Grid	F_e order 1	Error	F_e order 2	Error
1.6	1.6	16×16	0.94420	0.65580	1.57425	0.02575
		32×32	1.25746	0.34254	1.59755	0.00245
		64×64	1.42743	0.17257	1.59965	0.00035
		128×128	1.51434	0.08566	1.59995	0.00005
0.4	0.4	16×16	0.21666	0.18334	0.39328	0.00672
		32×32	0.30561	0.09439	0.39936	0.00064
		64×64	0.35303	0.04697	0.39990	0.00010
6.4	6.4	16×16	2.95339	3.44661	6.25465	0.14535
		32×32	4.47727	1.92273	6.38496	0.01504
		64×64	5.41928	0.98072	6.39778	0.00222

We also collected some results for both $\delta = 0$ and $\delta = 0.5$ presented in Table 3.6.2 and 3.6.3. As there are no explicit theoretical speed enhancement bounds available (for $\delta = 0$ and 0.5), we estimated $F_e(\mathbf{n})$ by extrapolation

based on the second-order results. For $\delta = 0.5$, we find $F_e(\mathbf{n}) = 1.50520$ for $\lambda = 1.6$ and $F_e(\mathbf{n}) = 5.95802$ for $\lambda = 6.4$ whereas, for $\delta = 0$, we obtain $F_e(\mathbf{n}) = 1.06683$ for $\lambda = 1.6$ and $F_e(\mathbf{n}) = 3.75604$ for $\lambda = 6.4$. The errors were computed using these estimates. For $\lambda = 1.6$, for both $\delta = 0.5$ and $\delta = 0$, the computed $F_e(\mathbf{n})$ are consistent with those obtained by Bourlioux and Khouider [9].

Table 3.6.2: Numerical results for $F_e(\mathbf{n})$ for $\delta = 0.5$

δ	λ	Grid	F_e order 1	Error	F_e order 2	Error
0.5	1.6	16×16	0.89277	0.61243	1.46740	0.03780
		32×32	1.18680	0.31840	1.49616	0.00904
		64×64	1.34530	0.15990	1.50335	0.00185
		128×128	1.42604	0.07916	1.50477	0.00083
	6.4	16×16	2.66041	3.29761	5.51902	0.43900
		32×32	4.10810	1.84992	5.88770	0.07032
		64×64	5.05144	0.90658	5.94322	0.01480
		128×128	5.51295	0.44607	5.95545	0.00357

Table 3.6.3: Numerical results for $F_e(\mathbf{n})$ for $\delta = 0$

δ	λ	Grid	F_e order 1	Error	F_e order 2	Error
0	1.6	16×16	0.64681	0.42002	0.99433	0.07250
		32×32	0.85037	0.21646	1.04925	0.01758
		64×64	0.96046	0.10637	1.06232	0.00451
		128×128	1.01446	0.05237	1.06569	0.00114
	6.4	16×16	2.00674	1.74930	2.67844	1.07760
		32×32	2.72345	1.03259	3.59100	0.16504
		64×64	3.22917	0.52687	3.70981	0.05623
		128×128	3.50145	0.25459	3.74399	0.01205

In Table 3.6.4 we report the results obtained in the presence of a non-zero mean flow. In this case, the error was calculated using again a reference $F_e(\mathbf{n}) = 1.59751$.

Table 3.6.4: Numerical results for $F_e(\mathbf{n})$ for $\overline{\theta} = \frac{\pi}{2}$ and $\overline{\lambda} = 0.1$

δ	λ	Grid	F_e order 2	Error
1	1.6	16×16	1.57252	0.02499
		32×32	1.59513	0.00238
		64×64	1.59716	0.00035

Summary

In this project, we showed that the *effective Hamiltonian* arising in the Majda-Souganidis model of premixed turbulent combustion can be computed directly through a new method based on the theory of viscosity solutions and on the recent theoretical work about the long time asymptotic behaviour of the solutions of Hamilton-Jacobi [8]. We have successfully implemented two numerical schemes, the First Order Monotone Finite Difference (FOMFD) scheme and the Essentially Non-Oscillatory (ENO) scheme. We have compared our numerical results for the enhancement in the normal speed of propagation of the flame front with those obtained by an alternate method [9] and showed that they are consistent with their results. The algorithms used in this thesis have been implemented in MATLAB on a single user workstation with Linux. The codes have not been optimized but showed reasonable efficiency. The first order scheme showed a more stable convergence to the desired eigenvalue of the *cell problem*, when compared to the second order scheme. However, the accuracy of the monotone scheme (FOMFD) is decreasing as the turbulent intensity increases. Moreover, the error of our

experimental results has decreased as the mesh was refined. Our algorithms provided a successful new and direct method for computing the averaged Hamiltonian by putting in practice the theory of viscosity solutions.

In chapter 1 we introduced the equations describing the propagation of the front, as well as the *cell problem* and the *effective Hamiltonian.* The relationship between the normal speed of propagation and the *effective Hamiltonian* was also introduced.

In chapter 2 and 3 we presented the relationship between the solutions of a time-dependent Hamilton-Jacobi equation and the solutions to the *cell problem* in both a theoretical and practical way. The numerical results of chapter 3, were consistent with the theory and showed that the eigenvalue of the *cell problem* is indeed given by the time-derivative of a time-dependent evolution equation [8].

Appendix: Numerical Codes for Chapter 3

MATLAB code used to generate the eigenvalue for the FOMFD scheme ($\delta = 0.5$, $\theta = \frac{\pi}{4}$). Note that similar MATLAB scripts were used to compute the eigenvalue for $\delta = 0, 1$ and for an arbitrary θ.

```
%*****************************************************************
% Mirela Cara
% M.Sc. Project
% File name: FOMFDdeltahalf.m
% Boundary Conditions: biperiodic
%*****************************************************************
function [eigvaluem]=FOMFDdeltahalf(N,lambda,r)
dx=1/N;
newsol=zeros(N+1,N+1);
oldsol=zeros(N+1,N+1);
oldeigvalue=zeros(N+1,N+1);
eigvalue=zeros(N+1,N+1); dt=1000; k=0;
test=1.;

for i=1:N+1
    for j=1:N+1

        V1(i,j)=sin(2*pi*(i-1)*dx)*cos(2*pi*(j-1)*dx)-...
```

```
                1/2)*cos(2*pi*(i-1)*dx)*sin(2*pi*(j-1)*dx);

        V2(i,j)=(1/2)*sin(2*pi*(i-1)*dx)*cos(2*pi*(j-1)*dx)-...
                cos(2*pi*(i-1)*dx)*sin(2*pi*(j-1)*dx);

    end
end

 c=max(max(V1.^2+V2.^2));
 v1=1/sqrt(c)*V1;
 v2=1/sqrt(c)*V2;

while k<1000
        oldeigvalue=eigvalue;
        k=k+1;
%*************************************************************************
% CFL condition calculation
%*************************************************************************
        for i=1:N+1
            for j=1:N+1

        if v1(i,j)>=0 & v2(i,j)>=0

        if i>1
            backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
        else
            backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
        end

        if j>1
            backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
        else
            backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
        end

        if i<N+1
            forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
        else
```

```
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

    condition=dx/(2*(max(max(-forwx,0),max(backx,0)))+...
            2*(max(max(-forwy,0),max(backy,0)))+...
            lambda*v1(i,j)+lambda*v2(i,j)+2*sqrt(2)*r);

end

if v1(i,j)>=0 & v2(i,j)<0

if i>1
    backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
else
    backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
end

if j>1
    backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
else
    backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
end

if i<N+1
    forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
else
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
```

```
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

condition=dx/(2*(max(max(-forwx,0),max(backx,0)))+...
        2*(max(max(-forwy,0),max(backy,0)))-...
        lambda*v2(i,j)+lambda*v1(i,j)+2*sqrt(2)*r);
end

if v1(i,j)<0 & v2(i,j)>=0

if i>1
    backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
else
    backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
end

if j>1
    backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
else
    backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
end

if i<N+1
    forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
else
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

condition=dx/(2*(max(max(-forwx,0),max(backx,0)))+...
        2*(max(max(-forwy,0),max(backy,0)))+...
        lambda*v2(i,j)-lambda*v1(i,j)+2*sqrt(2)*r);
```

```
end

if v1(i,j)<0 & v2(i,j)<0

if i>1
    backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
else
    backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
end

if j>1
    backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
else
    backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
end

if i<N+1
    forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
else
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

condition=dx/(2*(max(max(-forwx,0),max(backx,0)))+...
        2*(max(max(-forwy,0),max(backy,0)))-...
        lambda*v2(i,j)-lambda*v1(i,j)+2*sqrt(2)*r);

end

if condition<dt
    dt=condition;
end
```

```
            end
        end
%*************************************************************************
% Iteration
%*************************************************************************

    for i=1:N+1
        for j=1:N+1

        if v1(i,j)>=0 & v2(i,j)>=0

        if i>1
            backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
        else
            backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
        end

        if j>1
            backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
        else
            backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
        end

        if i<N+1
            forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
        else
            forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
        end

        if j<N+1
            forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
        else
            forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
        end

        newsol(i,j)=oldsol(i,j)+dt*(-(max(max(-forwx,0),max(backx,0)))^2-...
        (max(max(-forwy,0),max(backy,0)))^2+lambda*v1(i,j)*forwx+...
        lambda*v2(i,j)*forwy-sqrt(2)*r*backx-sqrt(2)*r*backy+...
```

```
lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-...
r^2);

eigvalue(i,j)=-(newsol(i,j)-oldsol(i,j))/dt;

end

if v1(i,j)>=0 & v2(i,j)<0

if i>1
    backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
else
    backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
end

if j>1
    backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
else
    backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
end

if i<N+1
    forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
else
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

newsol(i,j)=oldsol(i,j)+dt*(-(max(max(-forwx,0),max(backx,0)))^2-...
(max(max(-forwy,0),max(backy,0)))^2+lambda*v1(i,j)*forwx+...
lambda*v2(i,j)*backy-sqrt(2)*r*backx-sqrt(2)*r*backy+...
lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-...
r^2);
```

```
eigvalue(i,j)=-(newsol(i,j)-oldsol(i,j))/dt;

end

if v1(i,j)<0 & v2(i,j)>=0

if i>1
    backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
else
    backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
end

if j>1
    backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
else
    backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
end

if i<N+1
    forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
else
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

newsol(i,j)=oldsol(i,j)+dt*(-(max(max(-forwx,0),max(backx,0)))^2-...
(max(max(-forwy,0),max(backy,0)))^2+lambda*v1(i,j)*backx+...
lambda*v2(i,j)*forwy-sqrt(2)*r*backx-sqrt(2)*r*backy+...
lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-...
r^2);

eigvalue(i,j)=-(newsol(i,j)-oldsol(i,j))/dt;
```

```
end

if v1(i,j)<0 & v2(i,j)<0

if i>1
    backx=(oldsol(i,j)-oldsol(i-1,j))/dx;
else
    backx=(oldsol(i,j)-oldsol(i-1+N,j))/dx;
end

if j>1
    backy=(oldsol(i,j)-oldsol(i,j-1))/dx;
else
    backy=(oldsol(i,j)-oldsol(i,j-1+N))/dx;
end

if i<N+1
    forwx=(oldsol(i+1,j)-oldsol(i,j))/dx;
else
    forwx=(oldsol(i+1-N,j)-oldsol(i,j))/dx;
end

if j<N+1
    forwy=(oldsol(i,j+1)-oldsol(i,j))/dx;
else
    forwy=(oldsol(i,j+1-N)-oldsol(i,j))/dx;
end

newsol(i,j)=oldsol(i,j)+dt*(-(max(max(-forwx,0),max(backx,0)))^2-...
(max(max(-forwy,0),max(backy,0)))^2+lambda*v1(i,j)*backx+...
lambda*v2(i,j)*backy-sqrt(2)*r*backx-sqrt(2)*r*backy+...
lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-...
r^2);

eigvalue(i,j)=-(newsol(i,j)-oldsol(i,j))/dt;

end
```

```
    end
end

test=max(abs(eigvalue-oldeigvalue)); oldsol=newsol;
eigvaluem(k)=mean(mean(eigvalue));

end
%*****************************************************************************
% END
%*****************************************************************************

%*****************************************************************************
% LEVEL CURVES
% Figure 3.4.5
%*****************************************************************************
x=(0:N)*dx;
y=(0:N)*dx;
newsol=newsol-eigvaluem(k)*k*dt;
contour(x,y,newsol,30)
```

MATLAB code used to generate the eigenvalue for the ENO scheme ($\delta = 0.5$ and $\theta = \frac{\pi}{4}$). Note that similar MATLAB scripts were used to generate the eigenvalue for $\delta = 0, 1$ and for an arbitrary θ.

```
%*********************************************************************
% Mirela Cara
% M.Sc. Project
% File name: ENOdeltahalf.m
% Boundary Conditions: biperiodic
%*********************************************************************
function [eigvalue]=ENOdeltahalf(N,lambda,r,dt)
dx=1/N;
new=zeros(N+1,N+1);
old=zeros(N+1,N+1);
newh=zeros(N+1,N+1);
eigvalue=zeros(N+1,N+1);
oldeigvalue=zeros(N+1,N+1);
k=0;
test=1.;

for i=1:N+1
    for j=1:N+1
        v1(i,j)=sin(2*pi*(i-1)*dx)*cos(2*pi*(j-1)*dx)-...
                0.5*cos(2*pi*(i-1)*dx)*sin(2*pi*(j-1)*dx);
        v2(i,j)=0.5*sin(2*pi*(i-1)*dx)*cos(2*pi*(j-1)*dx)-...
                cos(2*pi*(i-1)*dx)*sin(2*pi*(j-1)*dx);
    end
end

c=max(max(v1.^2+v2.^2));

for i=1:N+1
    for j=1:N+1
        v1(i,j)=(1/sqrt(c))*v1(i,j);
        v2(i,j)=(1/sqrt(c))*v2(i,j);
    end
end
```

```
while k<100
    oldeigvalue=eigvalue;
    k=k+1;
%*********************************************************************
% first iteration
%*********************************************************************
for i=1:N+1
    for j=1:N+1
        if i>1
             downonex=old(i-1,j);
        else
             downonex=old(i-1+N,j);
        end

        if i>2
             downtwox=old(i-2,j);
        else
             downtwox=old(i-2+N,j);
        end

        if i<N+1
             uponex=old(i+1,j);
        else
             uponex=old(i+1-N,j);
        end

        if i<N
             uptwox=old(i+2,j);
        else
             uptwox=old(i+2-N,j);
        end

        h=(uponex+downonex-2*old(i,j))/dx;
        r1=(old(i,j)+downtwox-2*downonex)/dx;

            if(r1*h<=0)
```

```
        r1=0;
        h=0;
    end

backx_1=(old(i,j)-downonex)/dx+0.5*h;
backx_2=(old(i,j)-downonex)/dx+0.5*r1;

a=(uponex+downonex-2*old(i,j))/dx;
b=(uptwox+old(i,j)-2*uponex)/dx;

    if(a*b<=0)
        a=0;
        b=0;
    end

forwx_1=(uponex-old(i,j))/dx-0.5*a;
forwx_2=(uponex-old(i,j))/dx-0.5*b;

if j>1
     downoney=old(i,j-1);
else
     downoney=old(i,j-1+N);
end

if j>2
     downtwoy=old(i,j-2);
else
     downtwoy=old(i,j-2+N);
end

if j<N+1
     uponey=old(i,j+1);
else
     uponey=old(i,j+1-N);
end

if j<N
     uptwoy=old(i,j+2);
```

```
else
     uptwoy=old(i,j+2-N);
end

m=(uponey+downoney-2*old(i,j))/dx;
o=(old(i,j)+downtwoy-2*downoney)/dx;

    if(m*o<=0)
        m=0;
        o=0;
    end

backy_1=(old(i,j)-downoney)/dx+0.5*m;
backy_2=(old(i,j)-downoney)/dx+0.5*o;

d=(uponey+downoney-2*old(i,j))/dx;
g=(uptwoy+old(i,j)-2*uponey)/dx;

     if(d*g<=0)
         d=0;
         g=0;
     end

forwy_1=(uponey-old(i,j))/dx-0.5*d;
forwy_2=(uponey-old(i,j))/dx-0.5*g;

if abs(a)>=abs(b)
    up=forwx_2;
else
    up=forwx_1;
end

if abs(d)>=abs(g)
    vp=forwy_2;
else
    vp=forwy_1;
end
```

```
if abs(h)>abs(r1)
    um=backx_2;
else
    um=backx_1;
end

if abs(m)>=abs(o)
    vm=backy_2;
else
    vm=backy_1;
end

if (v1(i,j)>=0) & (v2(i,j)>=0)

  newh(i,j)=old(i,j)+dt*(-(max(max(-up,0),max(um,0)))^2-...
  (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*up+...
  lambda*v2(i,j)*vp-sqrt(2)*r*um-sqrt(2)*r*vm+...
  lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

elseif (v1(i,j)>=0) & (v2(i,j)<0)

  newh(i,j)=old(i,j)+dt*(-(max(max(-up,0),max(um,0)))^2-...
  (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*up+...
  lambda*v2(i,j)*vm-sqrt(2)*r*um-sqrt(2)*r*vm+...
  lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

elseif (v1(i,j)<0) & (v2(i,j)>=0)

  newh(i,j)=old(i,j)+dt*(-(max(max(-up,0),max(um,0)))^2-...
  (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*um+...
  lambda*v2(i,j)*vp-sqrt(2)*r*um-sqrt(2)*r*vm+...
  lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

else

  newh(i,j)=old(i,j)+dt*(-(max(max(-up,0),max(um,0)))^2-...
  (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*um+...
  lambda*v2(i,j)*vm-sqrt(2)*r*um-sqrt(2)*r*vm+...
```

```
          lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);
        end
    end
end
%*************************************************************************
% second iteration
%*************************************************************************
for i=1:N+1
   for j=1:N+1
        if i>1
            downonex=newh(i-1,j);
        else
            downonex=newh(i-1+N,j);
        end
        if i>2
            downtwox=newh(i-2,j);
        else
            downtwox=newh(i-2+N,j);
        end
        if i<N+1
            uponex=newh(i+1,j);
        else
            uponex=newh(i+1-N,j);
        end
        if i<N
            uptwox=newh(i+2,j);
        else
            uptwox=newh(i+2-N,j);
        end

        h=(uponex+downonex-2*newh(i,j))/dx;
        r1=(newh(i,j)+downtwox-2*downonex)/dx;

           if(h*r1<=0)
               h=0;
               r1=0;
           end
```

```
backx_1=(newh(i,j)-downonex)/dx+0.5*h;
backx_2=(newh(i,j)-downonex)/dx+0.5*r1;

a=(uponex+downonex-2*newh(i,j))/dx;
b=(uptwox+newh(i,j)-2*uponex)/dx;

   if(a*b<=0)
       a=0;
       b=0;
   end

forwx_1=(uponex-newh(i,j))/dx-0.5*a;
forwx_2=(uponex-newh(i,j))/dx-0.5*b;

if j>1
    downoney=newh(i,j-1);
else
    downoney=newh(i,j-1+N);
end
if j>2
    downtwoy=newh(i,j-2);
else
    downtwoy=newh(i,j-2+N);
end
if j<N+1
    uponey=newh(i,j+1);
else
    uponey=newh(i,j+1-N);
end
if j<N
    uptwoy=newh(i,j+2);
else
    uptwoy=newh(i,j+2-N);
end

m=(uponey+downoney-2*newh(i,j))/dx;
o=(newh(i,j)+downtwoy-2*downoney)/dx;
```

```
if(r<0.5)
    if(m*o<=0)
        m=0;
        o=0;
    end
end

backy_1=(newh(i,j)-downoney)/dx+0.5*m;
backy_2=(newh(i,j)-downoney)/dx+0.5*o;

d=(uponey+downoney-2*newh(i,j))/dx;
g=(uptwoy+newh(i,j)-2*uponey)/dx;

if(r<0.5)
    if(d*g<=0)
        d=0;
        g=0;
    end
end

forwy_1=(uponey-newh(i,j))/dx-0.5*d;
forwy_2=(uponey-newh(i,j))/dx-0.5*g;

if abs(a)>=abs(b)
   up=forwx_2;
else
   up=forwx_1;
end

if abs(d)>=abs(g)
   vp=forwy_2;
else
   vp=forwy_1;
end

if abs(h)>abs(r1)
   um=backx_2;
else
```

```
    um=backx_1;
 end

 if abs(m)>=abs(o)
    vm=backy_2;
 else
    vm=backy_1;
 end

 if (v1(i,j)>=0) & (v2(i,j)>=0)

    new(i,j)=0.5*old(i,j)+0.5*newh(i,j)+...
    0.5*dt*(-(max(max(-up,0),max(um,0)))^2-...
    (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*up+...
    lambda*v2(i,j)*vp-sqrt(2)*r*um-sqrt(2)*r*vm+...
    lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

elseif (v1(i,j)>=0) & (v2(i,j)<0)

    new(i,j)=0.5*old(i,j)+0.5*newh(i,j)+...
    0.5*dt*(-(max(max(-up,0),max(um,0)))^2-...
    (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*up+...
    lambda*v2(i,j)*vm-sqrt(2)*r*um-sqrt(2)*r*vm+...
    lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

elseif (v1(i,j)<0) & (v2(i,j)>=0)

    new(i,j)=0.5*old(i,j)+0.5*newh(i,j)+...
    0.5*dt*(-(max(max(-up,0),max(um,0)))^2-...
    (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*um+...
    lambda*v2(i,j)*vp-sqrt(2)*r*um-sqrt(2)*r*vm+...
    lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

else

    new(i,j)=0.5*old(i,j)+0.5*newh(i,j)+...
    0.5*dt*(-(max(max(-up,0),max(um,0)))^2-...
    (max(max(-vp,0),max(vm,0)))^2+lambda*v1(i,j)*um+...
```

```
            lambda*v2(i,j)*vm-sqrt(2)*r*um-sqrt(2)*r*vm+...
            lambda*r*(sqrt(2)/2)*v1(i,j)+lambda*r*(sqrt(2)/2)*v2(i,j)-r^2);

        end
            eigvalue(i,j)=-(new(i,j)-old(i,j))/dt;
      end
   end

        test=max(max(abs(eigvalue-oldeigvalue)));
        old=new;
        eigvaluem(k)=mean(mean(eigvalue));
   end
%*************************************************************************
% END
%*************************************************************************

%*************************************************************************
% LEVEL CURVES
% Figure 3.5.2
%*************************************************************************
x=(0:N)*dx;
y=(0:N)*dx;
new=new-eigvaluem(k)*k*dt;
mesh(x,y,new)
contour(x,y,new,20)
```

MATLAB codes used to generate Figures 3.4.1, 3.4.3, 3.4.4, 3.5.1

```
%*************************************************************************
% Mirela Cara
% M.Sc. Project
%*************************************************************************

%*************************************************************************
% CONVERGENCE OF TIME DERIVATIVE TO THE EIGENVALUE:
% EIGENVALUE VS NUMBER OF ITERATIONS
% Figures 3.4.1, 3.5.1
% Note: Example of MATLAB code presented is for the ENO scheme but is
% similar to the MATLAB code for the FOMFD scheme
%*************************************************************************
clear all [eigvaluem1]=ENOdeltaone(32,1.6,0.5,0.002);
[eigvaluem2]=ENOdeltahalf(32,1.6,0.5,0.002);
[eigvaluem3]=ENOdeltazero(32,1.6,0.5,0.002);
% eigenvalue vs k (N=32, lambda=1.6, r=0.5, dt=0.002, delta=0, 0.5, 1)
k=(1:1000);
plot(k,eigvaluem1, '-')
hold on
plot(k,eigvaluem2,'+')
hold on
plot(k,eigvaluem3,'*')

%*************************************************************************
% CONVEXITY AND COERCIVITY OF THE EIGENVALUE
% Figure 3.4.3
% Note: Example of MATLAB code presented is for the FOMFD scheme
%*************************************************************************
clear all
r=(2:14)*0.05;
for i=1:13
eig1(i)=mean(mean(FOMFDdeltaone(16,1.6,r(i)))');
eig2(i)=mean(mean(FOMFDdeltaone(32,1.6,r(i)))');
eig3(i)=mean(mean(FOMFDdeltaone(64,1.6,r(i)))');
% eigenvalue vs r (delta=1, lambda=1.6, N=16, 32, 64)
plot(r,eig1,'*')
```

```
hold on
plot(r,eig2,'.')
hold on plot(r,eig3,'-')

%*************************************************************************
% G(r) vs r
% Figure 3.4.4
% Note: Example of MATLAB code presented is for the FOMFD scheme
%*************************************************************************
clear all
r=(2:14)*0.05;
for i=1:13
eig1(i)=mean(mean(FOMFDdeltaone(16,1.6,r(i)))');
eig2(i)=mean(mean(FOMFDdeltaone(32,1.6,r(i)))');
eig3(i)=mean(mean(FOMFDdeltaone(64,1.6,r(i)))');
G1(i)=(eig1(i)+0.25)/r(i);
G2(i)=(eig2(i)+0.25)/r(i);
G3(i)=(eig3(i)+0.25)/r(i);
% G(r) vs r (delta=1, lambda=1.6, N=16, 32, 64)
plot(r,G1,'-')
hold on
plot(r,G2,'*')
hold on
plot(r,G3,'|')
%*************************************************************************
% RELATIONSHIP BETWEEN EIGENVALUE AND THETA
% Figures 3.4.6 and 3.5.3
% Note: Example of MATLAB code is for the FOMFD scheme
%*************************************************************************
clear all for i=1:100 teta(i)=0.015*i;
[eig_1(i)]=mean(mean(FOMFDdeltaoneangle(16,1.6,0.5,teta(i)))');
[eig_2(i)]=mean(mean(FOMFDdeltahalfangle(16,1.6,0.5,teta(i)))');
[eig_3(i)]=mean(mean(FOMFDDdeltazeroangle(16,1.6,0.5,teta(i)))');
end plot(teta,eig_1,'-')hold on plot (teta,eig_2,'+') hold on
plot(teta,eig_3,'*')
```

Bibliography

[1] M. Bardi, M. G. Crandall, L. C. Evans, H. M. Soner and P.E. Souganidis, *Viscosity Solutions and Applications*, Springer-Verlag, 1997.

[2] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhäuser, Boston, 1997.

[3] G. Barles, *Solutions de viscosité des équations de Hamilton-Jacobi*, Mathematics and Applications, Springer-Verlag, 1994.

[4] G. Barles, *The convergence of approximation Schemes for Parabolic Equations Arising in Finance Theory*, Cambridge University Press, 1997.

[5] G. Barles, C. Daher and M. Romano, *Convergence of Approximation Schemes for Parabolic Equations Arising in Finance Theory*, Math. Model. Numer. Anal., **5**, (1995), 125–143.

[6] G. Barles and E.R. Jakobsen, *On the convergence for approximation schemes for the Hamilton-Jacobi-Bellman equation*, Math. Model. Numer. Anal., **36**, (2002), 33–54.

[7] G. Barles and P. E. Souganidis, *Convergence of Approximation Schemes for Fully Nonlinear Second Order Equations*,Asymptot. Anal., **4**, (1991), 271–283.

[8] G. Barles and P. E. Souganidis, *On the large time behavior of solutions of Hamilton-Jacobi equations*, Siam J. Math. Anal., **31**, (2000), 925–939.

[9] A. Bourlioux and B. Khouider, *Computing the effective Hamiltonian in the Majda-Souganidis model of turbulent premixed flames*, Siam J. Numer. Anal., **40**, (2002), 1330–1353.

[10] I. Capuzzo-Dolcetta and P. -L. Lions, *Hamilton-Jacobi Equations with State Constraints*, Trans. Amer. Math. Soc., **318**, (1990), 643–683.

[11] I. Capuzzo Dolcetta and P. -L. Lions, *Viscosity Solutions and Applications*, Springer-Verlag, 1997.

[12] M. G. Crandall, L. C. Evans and P. -L. Lions, *Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations*, Trans. Amer. Math. Soc., **282**, (1984), 487–502.

[13] M. G. Crandall, H. Ishii and P. -L. Lions, *User's Guide to Viscosity Solutions of Second Order Partial Differential Equations*, Bull. Amer. Math. Soc., **27**, (1992), 1–67.

[14] M. G. Crandall and P.-L. Lions, *Viscosity Solutions of Hamilton-Jacobi Equations*, Trans. Amer. Math. Soc., **277**, (1983), 1–42.

[15] M. G. Crandall and P. L. Lions, *Two Approximations of Solutions of Hamilton-Jacobi Equations*, Math. Comp., **43**, (1984), 1–19.

[16] P. F. Embid, A. J. Majda and P. E. Souganidis, *Comparison of Turbulent Flame Speeds from Complete Averaging and the G-equation*, Phys. Fluids, **7**, #8, (1995), 2052–2060.

[17] L. C. Evans, *Partial Differential Equations: Graduate Studies in Mathematics v.19*, American Mathematical Society, (1998).

[18] L. C. Evans and D. Gomes, *Effective Hamiltonians and averaging for Hamiltonian dynamics I*, Arch. Rational Mech. Anal., **157**, (2001), 1–33.

[19] L. C. Evans and D. Gomes, *Effective Hamiltonians and averaging for Hamiltonian dynamics II*, Arch. Rational Mech. Anal., **161**, (2002), 271–305.

[20] M. Falcone, P. Lanucara and A. Seghini, *A splitting algorithm for Hamilton-Jacobi-Bellman equations*, Appl. Numer. Math, **15**, (1994), 207–218.

[21] W. H. Fleming and P. -L. Lions, *Stochastic Differential Systems, Stochastic Control Theory and Applications*, Springer-Verlag, 1988.

[22] D. A. Gomes and A. Oberman *Computing the effective Hamiltonian: A variational approach to homogenization*, Siam J. Control Optim., **43** #3, (2004), 792-812.

[23] S. Gottlieb and C. -W. Shu, *Total Variation Diminishing Runge-Kutta Schemes*, Mathematics of Computation, **67** #221, (1998), 73-85.

[24] E. R.Jakobsen, K. H. Karlsen and N. H. Risebro, *On the convergence rate of operator splitting for Hamilton-Jacobi equations with source terms*, Siam J. Numer. Anal., **39**, (2001), 499–518.

[25] K. H. Karlsen and N. H. Risebro, *An operator splitting method for nonlinear convection-diffusion equations*, Numer. Math., **77**, (1997), 365–382.

[26] B. Khouider, A. Bourlioux and A. J. Majda, *Parametrizing the burning speed enhancement by small scale periodic flows I*, Combust. Theory Model., **5**, (2001), 295–318.

[27] P. -L. Lions, *Generalized solutions of Hamilton-Jacobi equations*, Pitman, Boston, 1982.

[28] P. -L. Lions, *Optimal Control of Diffusion Processes and Hamilton-Jacobi-Bellman Equations I: The Dynamic Programming Principle and Applications*, Comm. Partial Diff. Equations., **8**, (1983), 1101–1174.

[29] P. -L. Lions, *Optimal Control of Diffusion Processes and Hamilton-Jacobi-Bellman Equations II: Viscosity Solutions and Uniqueness*, Comm. Partial Diff. Equations, **8**, (1983), 1229–1276.

[30] P. -L. Lions, *Optimal Control of Diffusion Processes and Hamilton-Jacobi-Bellman Equations III: Regularity of the Optimal Cost Function*, Nonlinear Partial Differential Equations and Their Applications, Pitman Advanced Publishing, Boston, 1983.

[31] P. -L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, Siam J. Numer. Anal., **16**, (1979), 964–979.

[32] P. -L. Lions, G. Papanicolaou and S.R.S. Varadhan, *Homogenization of Hamilton-Jacobi equations*, Unpublished, 1988.

[33] P.-L. Lions and P. Souganidis, *Correctors for the Homogenization of Hamilton-Jacobi Equations in the Stationary Ergodic Setting*, Communications on Pure and Applied Mathematics, **LVI**, (2003), 1501-1524.

[34] A. J. Majda and P. E. Souganidis, *Large scale front dynamics for turbulent reaction diffusion equations with separated velocity scales*, Nonlinearity, **7**, (1994), 1–30.

[35] A. J. Majda and P. E. Souganidis, *The effect of Turbulence on Mixing in Prototype Reaction-Diffusion Systems* Communications on Pure and Applied Mathematics, **LIII**, (2000), 1284-1304

[36] K. W. Morton and D. F. Mayers, *Numerical Solution of Partial Differential Equation, An Introduction*, 1994, Cambridge University Press.

[37] S. Osher and C. -W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, Siam J. Numer. Anal., **28**, (1991), 907-922.

[38] S. Osher and C. -W. Shu, *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes*, J. Comp. Physics, **77**, (1988), 439-471.

[39] E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading* Siam J. Numer. Anal., **29**, 3, (1992), 867–884.

[40] H. M. Soner, *Optimal Control with State-Space Constraint I.* Siam J. Control Optim., **24**, (1986), 552-561.

[41] H. M. Soner, *Optimal Control with State-Space Constraint II.* Siam J. Control Optim., **24**, (1986), 1110-1122.

[42] P. E. Souganidis, *Approximation schemes for viscosity solutions of Hamilton-Jacobi equations*, J. Diff. Eq., **59**, (1985), 1-43.

[43] P. E. Souganidis, *Max-min representations and product formulas for the viscosity solutions of Hamilton-Jacobi equations with applications to differential games*, Nonlinear Anal., **9**, (1985), 217-257.

[44] M. Sun, *Alternating directions algorithms for solving Hamilton-Jacobi-Bellman equations*, Applied Math. Optim., **34**, (1996), 267-277.