# Blow-up in the Nonlinear Schrodinger Equation Using an Adaptive Mesh Method

Ashley Twigger

Supervisor: Professor Michael J. Baines

# Contents

# Abstract

In this dissertation we give a brief overview of moving mesh methods, including one based upon moving mesh PDEs and one based on relative conservation. Then we describe the blow-up problems that we are interested in applying the relative conservation method to. Later on we analyse our results comparing them to existing results.

## Acknowledgements

Firstly I would like to thank the entire department for being so supportive and understanding. In particular my supervisor Mike Baines who has gone above and beyond what I would expect of a supervisor, and Sue Davis who throughout the year has been there when I needed her. I would also like to thank the EPSRC for allowing me the opportunity to be on this course, without the funding that they provided it would not of been possible for me to further my education.

## Declaration

I con rm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Ashley Twigger

# Chapter 1

# Introduction

Blow-up has increasingly become a major phenomena in the evolution of nonlinear equations. Physical problems when modelled may develop singularities in a finite amount of time $T$ ($T < 1$). Combustion in chemicals, chemotaxis in cellular aggregates, or the formation of shocks in the inviscid Burgers equation are examples of blow-up in the solution of a model [3]. These singularities can represent a change in the properties of the model such as ignition of a heated gas mixture.

A class of problems that displays this feature is the semilinear parabolic equations. These are used in the description of blow-up in the temperature of a reacting medium such as burning gas. These have the form

$$u_t = u_{xx} + f(u) \tag{1.1}$$

with boundary conditions

$$u(0; t) = u(1; t)$$

tive numerical method must be used that evolves the spatial mesh as the singularity develops. The singularity develops in a fairly simple manner, often independent of local structures in the initial conditions.It is conjectured in [5] that the growth of $u(x;t)$ near the blow-up time $T$ is described by

$$\max|u(x;t)| \;/\; (T - t)^{-} \qquad > 0$$

We will be looking at the equation when $f(u) = u^p$ $(p > 1)$ which is Fisher's equation and $f(u) = e^u$ which is the Kassoy problem in addition to the nonlinear Schrodinger equation which we will discuss later. These problems are great for testing out numerical methods as the formation of the singularities is typical of a wide range of PDEs (partial di erential equations). Also, a lot is known about the underlying analytic structure of the solutions for $t$ close to $T$ and $x$ close to $x^*$. Thus they make excellent problems for testing performance and accuracy. If the numerical method faithfully follows the underlying asymptotic structure we can assume that it does the same for more complicated problems where we do not know the underlying structure.

Most of the work that precedes [3

which is invariant under this scaling. Such scaling invariance and corresponding self-similar solutions can be found in various equations describing blow-up, with very similar scalings used for the Kassoy problem and nonlinear Schrödinger equation.

Here are a few that we will be looking at in this dissertation.

Fisher's equation

$$u_t = u_{xx} + u^p \qquad (p > 1)$$

The Kassoy problem

$$u_t = u_{xx} + e^u$$

The nonlinear Schrödinger equation

$$i\,\psi_t + \psi^2\,\psi + |\psi|^2\,\psi = 0$$

The first two are semilinear parabolic equations and the final one is a hyperbolic PDE.

# Chapter 2

# Moving grids

When a singularity forms it gains height and loses width at increasingly smaller time scales when approaching time $T$. This isolated spike could be missed by a fixed mesh method over time, as the spike could fall between mesh points. So an adaptive mesh method should be used to overcome this. There are three main types of adaptivity:

(1) h-refinement is static and refines the mesh by adding nodes to make the mesh finer in places shown in figure 2.1, but practically this is not a viable method to use in this case as it becomes more and more computationally expensive as the problem develops and the singularity loses width.

(2) p-refinement is also static and uses higher order polynomials to represent the solution more accurately (figure 2.2). It has high rates of convergence and accuracy compared to h-refinement but a polynomial

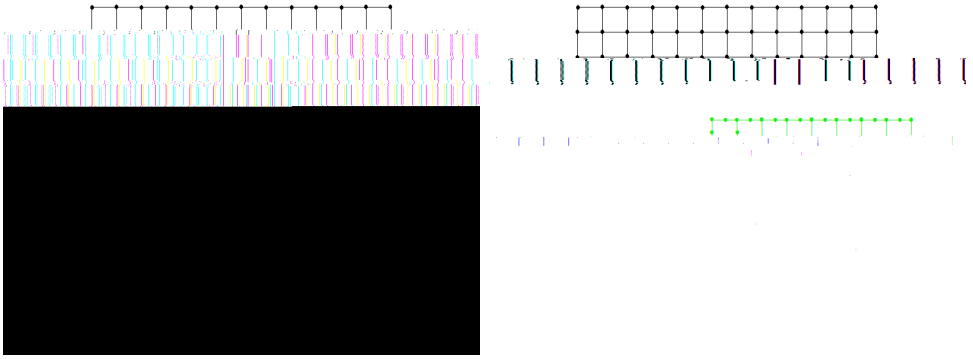

Figure 2.1: Left: original mesh. Right: post h-re nement, re ned edges in red.

will never be able to fully model the blow-up if it falls between nodes.

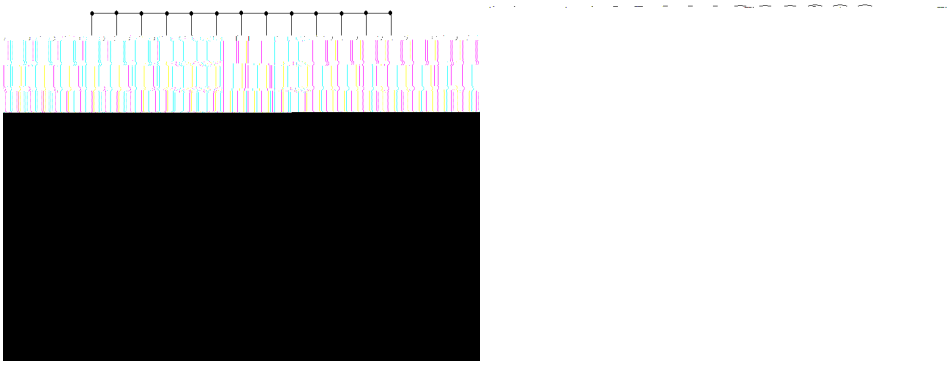

Figure 2.2: A sketch of p-re nement where higher order polynomials are used between nodes.

(3) The methods that both ourselves and Budd use are based on r-re nement. These are moving mesh methods that uses a xed number of nodes and

redistribute them to keep track of main features according to a certain criteria which is set ( gure 2.3). This has the advantage that it can keep track of the singularity right up to blow-up time $T$ without being expensive to compute, but it has the drawback that away from the blow-up point the solution can be poorly resolved due to few nodes remaining close.

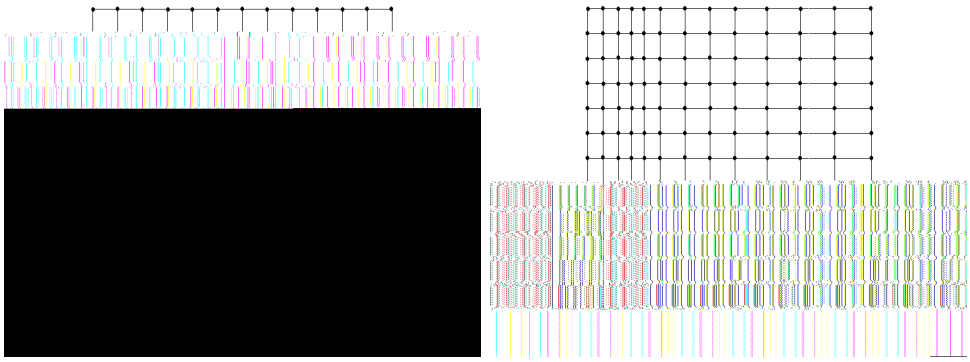

Figure 2.3: r-re nement: the nodes are redistributed give ner resolution in places.

Now we describe the r-re nement based moving mesh methods.

## 2.1 Moving mesh methods

### 2.1.1 Moving mesh PDEs

To solve PDEs such as (1.1) Budd proposes in [3] that by using a moving mesh PDE method in which $u(x;t)$ is discretised in the spatial variable to give the solution $u_i(t)$ on a moving mesh $x_i(t)$, $i = 0;:::;N$. The boundary conditions of (1.1) dictate that $u_0(t) = u_N(t) = 0$, $x_0 = 0$ and $x_N = 1$. The mesh $x_i(t)$ is de ned by the mesh transformation

$$x(\ ;t) : [0;1] \ ! \ [0;1];$$

where $x$ is the physical coordinate and is the computational coordinates.

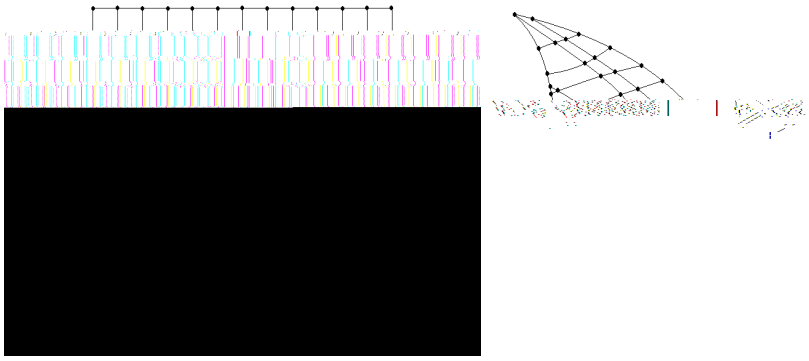

Figure 2.4: Sketch of transformation from computational space (left) to physical space (right)

As can be seen in gure 2.4 a simple mesh in computational space can be

used to describe a complicated physical space.

$$x_i(t) = x(\ ; t) = x\ \ \frac{i}{N}; t\ \ . \tag{2.1}$$

The constraint $\frac{@x}{@} > 0$ ensures that mesh crossing does not occur.

The moving mesh PDE (MMPDE) approach [2] requires a new PDE to solve $x(\ ; t)$ known as the moving mesh PDE which is solved simultaneously with the original PDE to nd $u(x; t)$.

The process used to determine $x(\ ; t)$ is the equidistribution of a positive monitor function $M(u)$. The equidistribution principle takes some measure of something such as error, density or a function and places the nodes of a mesh so that the contributions between the nodes are distributed equally to give a smooth solution.

$$\int_0^{x_i(t)} M\ \mathrm{dx} = \frac{i}{N} \int_0^1 M\ \mathrm{dx} = \int_0^1 M\ \mathrm{dx} \tag{2.2}$$

since $M$ is distributed equally between nodes (2.2) holds. Di erentiating (2.2) gives

$$\frac{@}{@}\ \ M\frac{@x}{@}\ \ = 0; \qquad x(0; t) = 0; \qquad x(1; t) = 1 \tag{2.3}$$

If (2.3) holds then a coordinate transformation is said to be equidistributed. Budd et al [2] found that for his method it is more convenient not to strictly enforce equidistribution but instead to solve for an MMPDE which tends towards an equidistributed solution. That way he states a simple initial mesh

and a smoothing approach was used.

Out of the various MMPDEs proposed in [4], Budd uses the two labelled MMPDE4 and MMPDE6, which are respectively

$$\frac{@}{@} \quad M \frac{@\underline{x}}{@} \quad = \quad \frac{@}{@} \quad M \frac{@x}{@} \tag{2.4}$$

and

$$\frac{@^2\underline{x}}{@^{\ 2}} = \quad \frac{@}{@} \quad M \frac{@x}{@} \tag{2.5}$$

where $\underline{x}$ denotes $\frac{@\underline{x}}{@t}$ , and is a small parameter used to relax the mesh to increase resolution away from the blow-up.

tends to zero on the left hand side of both (2.4) and (2.5) as $t$ tends to $T$ therefore the MMPDEs head towards an equidistributed state (2.3). This relaxes the need to enforce an exact equidistribution at the start allowing the use of a simple initial mesh such as a uniform one. Also this relaxation increases resolution further away from blow-up giving a better approximarion to the exact solution in the region.

If the MMPDE method is used the MMPDE must be invariant under the scaling (1.5), which can be achieved by using a suitable parameter and monitor function $M(u)$ [2]. For Fisher's equation the MMPDEs remain invariant

$$u_t = \underline{u} + u_x \underline{x} \qquad u_{x_i} = u_x x_i$$

and discretised by a central nite di erence into the quasi-Lagrangian form

$$\begin{cases} \underline{u} \quad \frac{u}{x}\underline{x} = \frac{1}{x} \quad \frac{u}{x} \quad + u \end{cases}$$

$(\underline{x}_{i+1}$

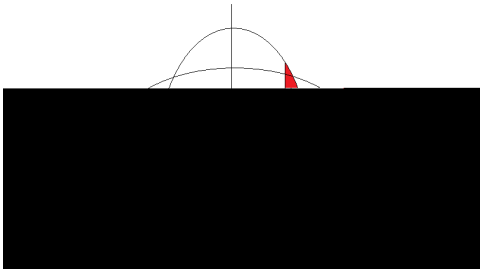

Figure 2.5: Sketch showing the same cell represented in blue then red and containing the same relative "mass" in both as the problem has evolved

time step.

Now we look at the general outline to the method that has been used.

### Method

The conservative method begins with an initialisation process. $t$ is the time step used throughout the computations and if xed then it is chosen here, whereas if a variable $t$ is being used then the constants $s$ which is a small scalar, and $T$ an estimate to the blow-up time are chosen now, later determining $t$ at each time step.

$$x_i(0) = \frac{i}{N}L \qquad \text{for } i = 0;:::;N; \qquad L > 0:$$

This creates a uniformed mesh of $N + 1$ nodes in the region $[0; L]$. We then apply the initial condition

$$u(x;0) = f(x_i) \qquad \text{for } i = 0;:::;N$$

to the mesh points. If the mass of the region changes during evolution then we de ne in terms of the monitor function $M(u)$ as follows.

$$= \int_{x_0}^{x_N} M(u)\,dx:$$

This is a normalising variable used in the next part of the initialisation process

$$c_i = \frac{1}{-} \int_{x_0}^{x_i} M(u)\,dx \qquad \text{for } i = 1;:::;N \quad 1; \tag{2.10}$$

which remain constant for all $t$ due to the $\frac{1}{-}$ in front of the integral.

This is the beginning of the loop, where the method starts in earnest. It determines the velocities of and each mesh point individually. From equation (2.10) - is given by

$$- = \int_{x_0}^{x_N} \frac{@M(u)}{@t}\,dx$$

also from (2.10) using Leibnitz' rule

$$\underline{x}_i = \frac{1}{u_i} \int_{x_0}^{x_i} \frac{@M(u)}{@t}\,dx + c_i - \qquad \text{for } i = 1;:::;N \quad 1: \tag{2.11}$$

The singularity at $x_0$ is an attractor, all the mesh points $x_i$ that are not xed

have a negative velocity heading towards $x_0$ at a quicker rate the closer they are. This monotonic decrease in velocity ($\underline{x}_{i+1} < \underline{x}_i$) insures that no node crossing occurs during the evolution of the problem. By using an Euler time stepping equation it is then possible to approximate and the mesh points at the next time step by

$$(t + \quad t) = \quad (t) + \quad t -$$

and

$$x_i(t + \quad t) = x_i(t) + \quad t\mathrm{t}$$

$$u(x;t) = 0 \text{ at } x = 0 \text{ and } x = 1$$

to

$$u_x(x;t) = 0 \text{ at } x = 0 \text{ and } u(x;t) = 0 \text{ at } x = 0.5.$$

In addition the initial condition has to be modified so that it is translated left from

$$u(x;0) = 20\sin(\pi x) \tag{3.1}$$

to

$$u(x;0) = 20\sin\left(\pi x + \frac{1}{2}\right) \tag{3.2}$$

For the equation we have used the monitor function

$$M(u) = u^{p-1} \tag{3.3}$$

which remains invariant under evolution, thus the rescaling (1.5) holds for all time before the blow-up time $T$.

As can be seen in figure 3.1 for the initial condition (3.2) the solution shows convergence, as the number of nodes is increased and the fixed time step is reduced the blow-up time approaches a time of $T \approx 0.08244$ which is close

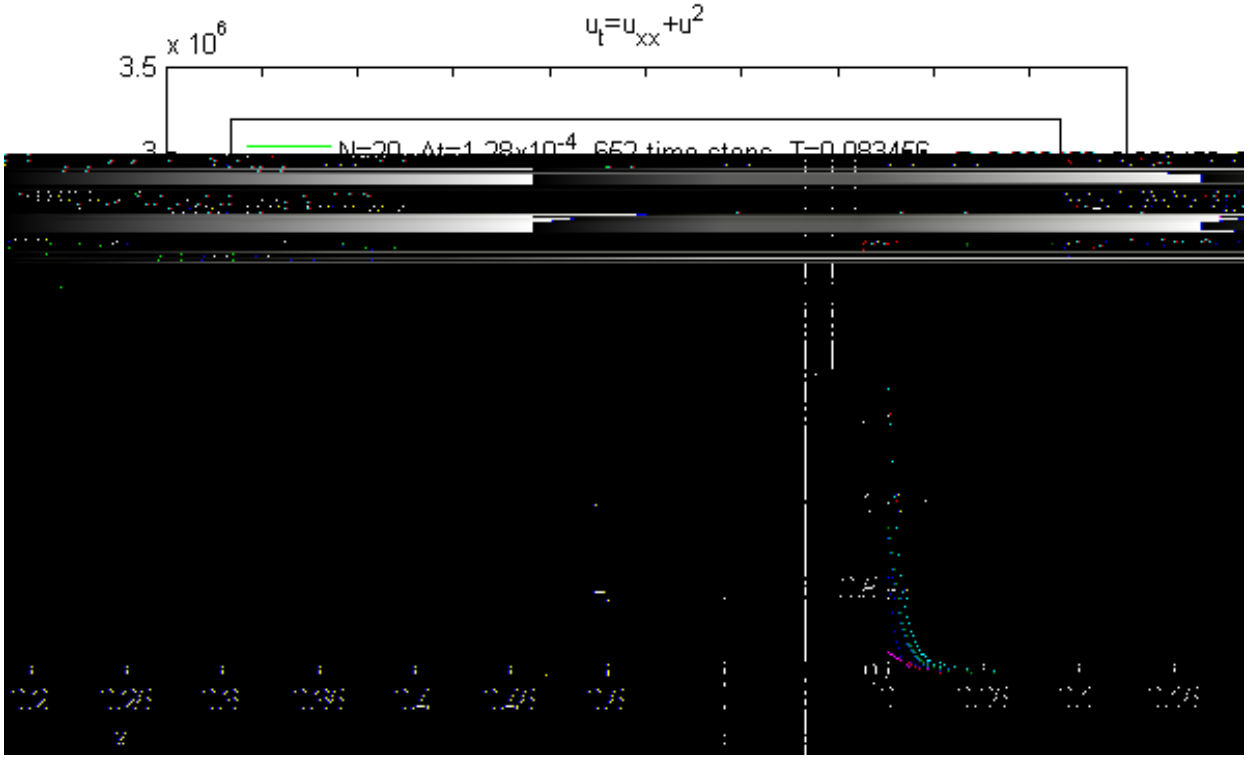


Figure 3.1: Convergence of the solution for increasing numbers of nodes.

to the blow-up time $T = 0.08237$ that Budd found in [2] with his method.

Since we know that the blow-up occurs at $u_0$ we can divide the entire region by this to get a solution that remains between zero and one for all time. Figure 3.2 shows the normalised evolution from the initial state for a few time steps close to blow-up. The normalised solution is converging towards a delta function this shows that it is only $u_0$ that is blowing up creating an isolated spike.

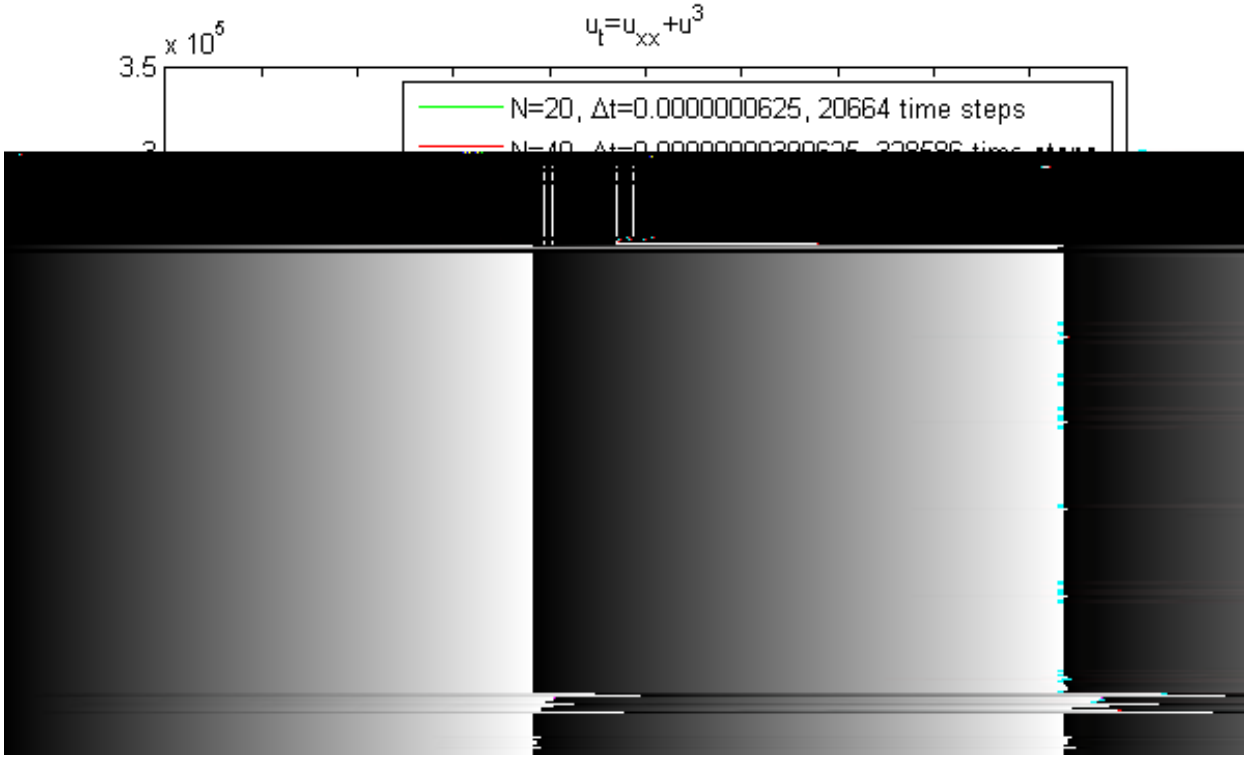


Figure 3.3: Blow-up in Fisher's equation for $u^3$.

## 3.2 The Kassoy problem

The Kassoy problem is much like the Fisher's equation except that the forcing function is now $e^u$ instead of $u^p$.

The monitor function we use throughout the numerical results for the Kassoy problem is

$$M(u) = e^u \tag{3.4}$$

which remains invariant under the scaling (1.5).

After much experimentation with the same initial and boundary conditions as for the Fisher's equation

There is nothing keeping us from using a variable time step that adheres to a certain set criteria, thus we can use an increasingly small time step as we

## 3.3 The nonlinear Schrodinger equation

Assuming radial symmetry the nonlinear Schrodinger equation is

$$i\frac{\partial}{\partial t} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + |\psi|^2\psi = 0 \tag{3.8}$$

where

$$\psi = u + iv$$

and we use the initial condition

$$u(r,0) = \begin{cases} 6\sqrt{2}e^{-r^2} & \text{if } 0 \leq r < 5 \\ 0 & \text{if } r \geq 5 \end{cases} \tag{3.9}$$

and boundary conditions

$$u_r(0,t) = 0 \text{ and } u(5,t) = 0 \tag{3.10}$$

using the monitor function

$$M(\psi) = |\psi|^2$$

$u(r,t)$ in fact tends towards zero as $r$ tends to in nity but since it is not possible to compute an in nite region we can truncate it at $r = 5$ in the boundary conditions because the rate at which it tends towards zero is of $O(e^{-r^2})$. Furthermore it should be noted that there is no need for the normalisation factor in the nonlinear Schrodinger equation numerical method since the total "mass" of a cell is invariant throughout the evolution.

To solve we seperate the problem into a real part and an imaginary part, solving seperately we can then find a solution for at each time step using much the same method as in the general case. The biggest change is that – is zero, erasing a term in (2.11), losing the need to calculate – and (2.10) entirely.

Approaching the nonlinear Schrödinger equation using this method created results that clearly did not describe the blow-up. This seemed to be because of the way the mesh was being redistributed, from (2.11)

$$M(\ _i)\underline{r}_i = \int_{r_0}^{r_i} \frac{@M(\ )}{@t} r\ \mathrm{d}r + c_i\ |_{\{z\}=0}; \tag{3.11}$$

Our monitor function in this case is

$$M(\ _i) = j\ _i j^2 = u$$

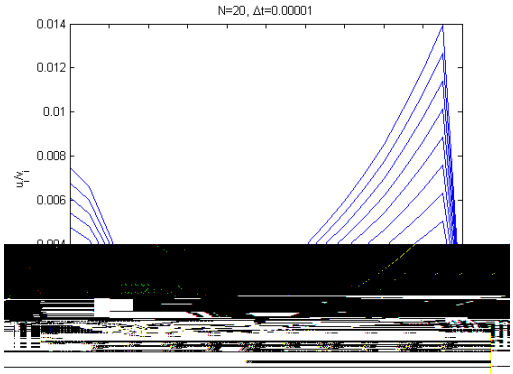


Figure 3.6: Plot of $\frac{v_i}{u_i}$ against $r_i$

From gure 3.6

# Bibliography

[1] J. Bebernes and S. Bricher, *Final time blowup pro les for semilinear parabolic equations via center manifold theory*