TRINITY COLLEGE DUBLIN
COLÁISTE NA TRÍONÓIDE
THE UNIVERSITY OF DUBLIN

Numerical Analysis of nonlinear fourth-order Differential Equations

by

Patrick Sinnott

Supervised by Prof. Paschalis Karageorgis

A final year research project submitted in partial fulfillment of the requirements for the degree of

Bachelors of Arts Moderatorship in Theoretical Physics

School of Mathematics

Trinity College Dublin

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at: http://www.tcd.ie/calendar

I declare that the assignment being submitted represents my own work and has not been taken from the work of others save where appropriately referenced in the body of the assignment.

Signed: Patrick Sinnott
Date: 08/04/2021

Abstract

The main objective of this project is to establish a method for numerically modelling solutions of a given fourth-order differential equation, then using this model to analyse various solutions of this equation and establish their properties.

The existence of global solutions to the equation is established first and then theorems are presented which predict the behaviour of these solutions. The same kind of analysis is then applied to solutions which blow-up as we approach infinity as well as rapidly increasing and decreasing solutions which approach infinity in finite time.

Finite-Difference methods are then derived, most importantly the fourth-order Runge Kutta algorithm which allows us to calculate solutions accurate to the order 10^{-11}

This RK4 method was applied to our given equation which demonstrated the properties of solutions which were predicted in our qualitative analysis. Global solutions to our equation were shown to display oscillatory behaviour, uncontrollable blow-up solutions of the equation were shown to take the form of increasingly wide oscillations.

Our algorithm was then used to analyse the physical problem of travelling waves in a suspension bridge first suggested by Walter and Mckenna (1990) and the results are consistent with behaviour recorded by observers of the Golden Gate Bridge in San Francisco validating this proposed model.

Contents

Chapter 1

Acknowledgements

Most importantly, Dr. Paschalis Karageorgis, His constant knowledge and support made this project a joy to complete. I couldn't have asked for a better supervisor

Dr. Michael Peardon, who helped with the Numerical Analysis.

My parents, who have given me every opportunity to excel with unwavering support.

My housemates: Steve, Mac and Lee, who have turned what should have been a terrible year into one of my fondest memories

The Bass brothers: Cathal, Liam and Conor, For motivating me to be the best I can be.

And all of my friends and peers in Theoretical Physics, without whom I never would have made it this far.

Chapter 2

Introduction

In this project, we consider the equation:

$$w''''(s) + kw''(s) + f(w(s)) = 0 \quad (s \in R) \tag{2.1}$$

where $k \in R$ and f is a locally Lipschitz function.

This equation arises in many contexts. When k is negative (2.1) is known as the extended Fisher-Kolmogorov equation which is a reaction–diffusion system that can be used to model population growth and wave propagation. When k is positive (2.1) is referred to as Swift-Hohenberg equation which is derived from the equations for thermal convection. This equation serves as a model of pattern formation in many physical, chemical or biological systems.

Of particular interest is the case where this equation can be used to simulate travelling waves in a suspension bridge. This model was first proposed in (McKenna, P.J. and Walter, W., 1990.) [2] and it occurs when the nonlinearity $f(w(s))$ is taken as the piecewise function $(s^+ - 1)$ where

$$s^+ = \begin{cases} s, & s \geq 0 \\ 0, & s < 0 \end{cases}$$

In this model, we take the constant $k \equiv c^2$ where c is the speed of the travelling wave on the bridge and our nonlinearity $f(w(s))$ simulates the effect that the cables hold the bridge up but the constant force of gravity holds it down.

Our primary focus is to find an accurate method of numerically modelling solutions of this equation. Using this model would enable us to verify theoretical predictions about the solutions of our equation and further analyse their behaviour.

This equation gives rise to a variety of solutions. We will consider the different solutions which arise from varying the following parameters: the value of the constant k, different non-linear functions $f(w(s))$ as well as varying initial values for $w(s))$ and its higher order derivatives. In order to model our equation we will need to specify these parameters. We will begin our discussion by attempting to determine which choices for these parameters should lead to global solutions

of equation (2.1), A global solution is one which is defined for every real-value of s. When we determine the existence of these well-behaved solutions we can study their properties. We will also analyse the existence and behaviour of other types of solutions such as ones that increase or decrease to infinity in finite time or solutions that blow up ie. approach infinity as $t \to \infty$.

We will then take a detour to study the Finite-Difference methods which will be used in the approximation of our equation. The Euler Method is the simplest way of approximating an ODE, and forms the basis of our more advanced methods. The Euler Method is accurate to order h (with h being the step-size), so for this reason we will expand our analysis to the more advanced and accurate fourth order Runge-Kutta method which is accurate to order h^4.

We will then attempt to reconcile our Qualitative analysis of the behaviour of our solutions with the numerical analysis methods that we have derived. If we can demonstrate the properties of the solutions that we noted in our mathematical theory then we know that our numerical methods are functioning correctly.

The algorithm should then be fit to analyse the more interesting physical problem of the Suspension Bridge.

We should then begin to analyse the accuracy of our approximation of the solutions to verify the validity of our results. This can be achieved by first calculating the global discretization error for a number of different step sizes, then by the conservation of a pre-defined energy function.

2.1 Energy

We can define an Energy Functional associated to (2.1) which will be useful in our analysis.

This function should be constant for any w that solves (2.1).

$$\mathcal{E}(s) := \frac{1}{2}w''(s)^2 - \frac{k}{2}w'(s)^2 - w'(s)w'''(s) - F(w(s)) \tag{2.2}$$

where $F(w(s)) = \int f(w(s))\,ds$. We can show that this new energy functional is constant for a solution w of (2.1) quite readily:

$$\mathcal{E}'(s) = w'''(s)w''(s) + w''''(s)w'(s) - (w'(s)w'''(s))' = 0 \quad \Longrightarrow \quad \mathcal{E}(s) = C \tag{2.3}$$

It will also be useful to define a second energy function which is constant at critical points for w, namely $w'(s_1) = w'(s_2) = 0$

$$E(s) := \mathcal{E}(s) + w'(s)w'''(s)E(s) := \frac{1}{2}w''(s)^2 - \frac{k}{2}w'(s)^2 - F(w(s)) \tag{2.4}$$

To show the property that this energy functional is constant at the critical points, we can differentiate it and then do a definite integral by parts between two critical points and show that the given value is zero.

$$E'(s) = w'''(s)w''(s) - kw''(s)w'(s) - f(w(s))w'(s)$$

$$
\begin{aligned}
E\left(s_2\right)-E\left(s_1\right) &= \int_{s_1}^{s_2} E'(s)ds \\
&= \int_{s_1}^{s_2} \left(w'''(s)w''(s) - kw''(s)w'(s) - f(w(s))w'(s)\right) ds \\
&= -\int_{s_1}^{s_2} \left(w''''(s) + kw''(s) + f(w(s))\right) w'(s)ds = 0
\end{aligned}
$$

The properties of these defined energies will be useful at many points during our analysis.

Chapter 3

Qualitative Analysis of Solutions

3.1 Existence and behaviour of Global Solutions

This section will focus on analysing the different solutions to equation (2.1). For ODEs in general, the solutions need not exist at all times and some can blow up in finite time. Our first task will be to determine if it is possible to find any solution to (2.1) which is global, and then to study the behaviour of these solutions if they exist. A global solution is one which is defined for every real-value of s. To achieve this, we will need to impose further restrictions on f. We will require the following conditions:

$$f \in \mathrm{Lip}_{\mathrm{loc}}(R), \quad f(s)s > 0 \quad \text{for every } s \in R\backslash\{0\} \tag{3.1}$$

$$\limsup_{t\to+\infty} \frac{f(s)}{s} < +\infty \quad \text{or} \quad \limsup_{t\to-\infty} \frac{f(s)}{s} < +\infty \tag{3.2}$$

The two assumptions above allow us to narrow down the nonlinearities $f(w(s))$ which will be well-behaved and give us a continuous global solution of our ODE. Examples of functions which obey these conditions would be: $f(w(s)) = e^w - 1$ or $f(w(s)) = \arctan(w)$. Under the two assumptions above, it can be shown that local solutions to (2.1) can be extended to the whole real line.

Theorem 1: *Let $k \in R$ and assume that f satisfies (3.1) and (3.2). Then any local solution to (2.1) exists for all $s \in R$*

The proof of this theorem is discussed in (Berchio, Ferrero, Gazzola and Karageorgis, 2011) [1] but in this project we will take it to be true. This theorem confirms the conditions for existence of global solutions to (2.1). We will examine the properties of these global solutions.

Theorem 2: *Let $k \geq 0$ and f satisfy (3.1). If w is a global solution to (2.1), then*

$$\liminf_{s\to+\infty} w(s) \leq 0 \leq \limsup_{s\to+\infty} w(s) \tag{3.3}$$

so that if $\lim_{s\to+\infty} w(s)$ exists then

$$\lim_{s\to+\infty} w(s) = 0 \tag{3.4}$$

Furthermore, if $w \not\equiv 0$ then $w(s)$ changes sign infinitely many times as $s \to +\infty$ and the same result holds for $s \to -\infty$

This theorem demonstrates that any global solution we find for this equation, with $k \geq 0$ and f satisfying (3.1) will demonstrate oscillatory behaviour.
Our theorems in the next section will show that a similar phenomenon may not occur when $k < 0$.

Proof of Theorem 2

To prove this statement, it is sufficient to show it holds for $s \to +\infty$ as when this is done we can remark that (2.1) is invariant under the change of variables $s \mapsto -s$. which then extends the proof to $s \to -\infty$
We begin by proving (3.3) (The weaker statement)

Assume for the purpose of contradiction that

$$\liminf_{s \to +\infty} w(s) \in (0, +\infty] \tag{3.5}$$

Then there must exist $\sigma \in R$ and $\gamma > 0$ such that $f(w(s)) \geq \gamma$ for all $s \geq \sigma$. Hence,

$$[w''(s) + kw(s)]'' = w''''(s) + kw''(s) \leq -\gamma \quad \forall s \geq \sigma \tag{3.6}$$

We now have a negative upper bound for the second derivative of the map $s \mapsto w''(s) + kw(s)$. This means that the second derivative is strictly decreasing, which then implies the first derivative is eventually decreasing which shows

$$\lim_{s \to +\infty} (w''(s) + kw(s)) = -\infty \tag{3.7}$$

By (3.5), we can straightforwardly imply that $w''(s) \to -\infty$ as $s \to +\infty$ which demonstrates a contradiction to (3.5).

We can then assume that

$$\limsup_{s \to +\infty} w(s) \in [-\infty, 0), \tag{3.8}$$

and by the same logic we conclude that $w''(s) \to +\infty$ as $s \to +\infty$ which thereby reaches a contradiction.

Now we set out to prove (3.4) and that $w(s)$ changes sign infinitely many times as $s \to +\infty$. We must first prove it for the case of $k > 0$ and then for the case where $k = 0$
$\underline{k > 0}$:
Assume for the sake of contradiction that $w(s)$ is eventually nonnegative. This corresponds to

$$w(s) \geq 0 \quad \forall s \geq 0 \tag{3.9}$$

In the case of (3.9) we can use (2.1) to deduce that

$$[w'''(s) + kw'(s)]' = w''''(s) + kw''(s) = -f(w(s)) \leq 0 \quad \forall s \geq 0 \tag{3.10}$$

such that $s \mapsto w'''(s) + kw'(s)$ is nonincreasing and two cases may occur: either its limit for $s \to +\infty$ is strictly negative or it is nonnegative. This gives us the following alternatives:

$$(i)\ w'''(s) + kw'(s) < 0 \quad \forall s \geq 0 \quad (ii)\ w'''(s) + kw'(s)0 \quad \forall s \geq 0. \tag{3.11}$$

If we can show that both (i) and (ii) lead to a contradiction, the proof will be complete.

Case (i):
If case (i) was to occur, then

$$[w''(s) + kw(s)]' = w'''(s) + kw'(s) \to -K \in [-\infty, 0) \tag{3.12}$$

proving that $w''(s) + kw(s) \to -\infty$ which shows by (3.9) that $w''(s) \to -\infty$ and this is a contradiction of (3.9).

Case (ii):
If case (ii) is to occur, then:

$$[w''(s) + kw(s)]' = w'''(s) + kw'(s) \geq 0 \tag{3.13}$$

so that

$$s \mapsto w''(s) + kw(s) \quad \text{is nondecreasing} \tag{3.14}$$

and we see that the following limit exists

$$\lim_{s \to +\infty} [w''(s) + kw(s)] = \ell \in (-\infty, +\infty] \tag{3.15}$$

We can rule out the case where $\ell \leq 0$ as if this holds then we evaluate that $w''(s) + kw(s) \leq 0$ for $s \geq 0$ so that by (3.9) also $w''(s) \leq 0$ for $s \geq 0$. $w(s)$ obtains a strictly positive limit. From (3.15) and our assumption on l we can infer that $s \mapsto w''(s)$ has a strictly negative limit, contradicting (3.9).

Now we can disprove the case where

$$\ell \in (0, +\infty) \tag{3.16}$$

For the sake of contradiction we will assume (3.16). From (3.9) and (3.3) two subcases may exist: either

$$\lim_{s \to +\infty} w(s) = 0 \tag{3.17}$$

or

$$0 = \liminf_{s \to +\infty} w(s) < \limsup_{s \to +\infty} w(s) \tag{3.18}$$

(3.17) can be dealt with first as if we combine it with (3.15) and (3.16) we can notice that it infers $w''(s) \to \ell > 0$, which implies $w(s) \to +\infty$ and contradicts (3.17).

(3.18) implies that there exists two divergent sequences $\left\{s_m^j\right\}_{j\in N}$ and $\left\{s_M^J\right\}_{j\in N}$ of local minima and local maxima respectively for w such that

$$w'(s) \geq 0 \quad \forall s \in \left[s_m^j, s_M^j\right], \quad w'(s) \leq 0 \quad \forall s \in \left[s_M^j, s_m^{j+1}\right] \tag{3.19}$$

for all $j \in N$. Multiplying (3.15) by $w'(s)$ gives $w''(s)w'(s) + kw'(s)w(s) = (\ell + o(1))w'(s)$ as $s \to +\infty$, which if we integrate over $\left[s_m^j, s_M^j\right]$, we get

$$\frac{k}{2}\left[w\left(s_M^j\right)^2 - w\left(s_m^j\right)^2\right] = (\ell + o(1))\left[w\left(s_M^j\right) - w\left(s_m^j\right)\right] \text{ as } j \to +\infty \tag{3.20}$$

which finally leads to

$$\frac{k}{2}\left[w\left(s_M^j\right) + w\left(s_m^j\right)\right] = \ell + o(1) \quad \text{ as } j \to +\infty \tag{3.21}$$

by (3.5) and (3.3) we can deduce that

$$\liminf_{j\to+\infty} w\left(s_m^j\right) = \liminf_{s\to+\infty} w(s) = 0 \tag{3.22}$$

We will now replace our sequence $\left\{s_m^j\right\}_{j\in N}$ with one of its subsequences (still denoted in the same way) such that

$$\lim_{j\to+\infty} w\left(s_m^j\right) = 0 \tag{3.23}$$

We should consider also the corresponding subsequence for $\left\{s_M^j\right\}_{j\in N}$. Inserting these into (3.21) shows that

$$\lim_{j\to+\infty} w\left(s_M^j\right) = \frac{2\ell}{k} \tag{3.24}$$

We can then insert (3.23) and (3.24) into (3.15) to deduce that

$$\lim_{j\to+\infty} w''\left(s_m^j\right) = \ell, \quad \lim_{j\to+\infty} w''\left(s_M^j\right) = -\ell \tag{3.25}$$

We can now use the energy function that we defined in equation (2.4) and since it has the property that it is constant at its critical points we find that

$$\frac{w''\left(s_m^j\right)^2}{2} - F\left(w\left(s_m^j\right)\right) = E\left(s_m^j\right) = E\left(s_M^j\right) = \frac{w''\left(s_M^j\right)^2}{2} - F\left(w\left(s_M^j\right)\right) \tag{3.26}$$

which, when we let $j \to +\infty$ and use (3.23), (3.24) and (3.25) gives us

$$\frac{\ell^2}{2} - F(0) = \frac{\ell^2}{2} - F\left(\frac{2\ell}{k}\right) \tag{3.27}$$

by (3.1) we can deduce that $F(s) = F(0) = 0$ iff $s = 0$. From (3.27) this would imply that $l = 0$ which contradicts (3.16). Therefore we see that (3.18) leads to a contradiction.

We are now left only with the case where $\ell = +\infty$. And again we must consider the two subcases: (3.17) and (3.18).

In the case of (3.17), then (3.15) gives a contradiction with our assumption on ℓ.

If we have oscillations as in the case of (3.18), when we take (3.25) as $\ell = +\infty$ we will infer that $E\left(s_m^j\right) \to +\infty$ as $j \to +\infty$. But in our definition of the Energy functional we took that this function would be constant through the critical points which therefore leads to a contradiction. We have now finally shown that (3.9) cannot ever occur for $k > 0$ since each case and subcase has led to a contradiction.
A completely analogous proof can be completed for $w(s) \leq 0 \quad \forall s \geq 0$ by simply changing all signs involved. This completes this part of the proof

We must now consider the case where $k = 0$.

For this part of the proof, we will need the Lemma:
Lemma 3: *Let $k \in R$ and let $f : R \to R$ be a continuous function such that $f(0) = 0$. Let w be a global solution to (2.1) such that $\lim_{s\to+\infty} w(s) = 0$. Then:*

$$\lim_{s\to+\infty} w'(s) = \lim_{s\to+\infty} w''(s) = \lim_{s\to+\infty} w'''(s) = \lim_{s\to+\infty} w''''(s) = 0 \tag{3.28}$$

The same result holds for $-\infty$ in place of $+\infty$.
This Lemma is proven in the appendices.

$\underline{k = 0}$
Assume for contradiction that (3.9) holds. According to (3.3) then (3.17) and (3.18) may occur. From Lemma 3 we know that:

$$\lim_{s\to+\infty} w'(s) = \lim_{s\to+\infty} w''(s) = \lim_{s\to+\infty} w'''(s) = 0 \tag{3.29}$$

But on the other hand by (3.1) and (3.9) we get that

$$w''''(s) = -f(w(s)) \leq 0 \quad \forall s \geq 0 \tag{3.30}$$

This would imply that $s \mapsto w'''(s)$ is nonincreasing and, in turn by Lemma 3 that

$$w'''(s) \geq 0 \quad \forall s \geq 0 \tag{3.31}$$

The same argument may lead to

$$w''(s) \leq 0 \quad \text{and} \quad w'(s) \geq 0 \quad \forall s \geq 0 \tag{3.32}$$

and may finally give us

$$w(s) \leq 0 \quad \forall s \geq 0 \tag{3.33}$$

This now contradicts (3.9) with $w = 0$. Now we must consider the situation in (3.18) let $\left\{s_m^j\right\}_{j\in N}$ and $\left\{s_M^j\right\}_{j\in N}$ retain their earlier definitions and denote the increasing divergent sequence of local

minima and maxima respectively. By (3.9), we can deduce that $\lim_{s\to+\infty} w''(s)$ exists. Since $w''\left(s_m^j\right) \geq 0$ and $w''\left(s_M^i\right) \leq 0$ we can infer that

$$\lim_{s\to+\infty} w''(s) = 0 \tag{3.34}$$

However, by (3.3) and (3.18) we find that

$$\lim_{i\to+\infty} w\left(s_m^j\right) = 0 \quad \text{and} \quad \lim_{i\to+\infty} w\left(s_M^i\right) = \delta \in (0, +\infty] \tag{3.35}$$

Using the energy function we have defined in (2.4) and its property of being constant at the critical points, we find:

$$\frac{w''\left(s_m^j\right)^2}{2} - F\left(w\left(s_m^j\right)\right) = E\left(s_m^j\right) = E\left(s_M^i\right) = \frac{w''\left(s_M^i\right)^2}{2} - F\left(w\left(s_M^i\right)\right) \tag{3.36}$$

which by letting $i, j \to +\infty$ and using (3.34) we find that $F(\delta) = F(0)$. Since $\delta > 0$, we get a contradiction to (3.9) for the case where $k = 0$, if we reverse all the signs we can have a completely analogous argument for $w(s) \leq 0$.

Now we have confirmed the existence of global solutions to our equation and proven that they will display oscillatory behaviour under suitable conditions, namely: that our solution will change sign infinitely many times as it approaches infinity if $k \geq 0$ and $f \in \mathrm{Lip}_{\mathrm{loc}}(R), \quad f(s)s > 0 \quad$ for every $s \in R\backslash\{0\}$. We will investigate this when we begin the numerical analysis of our solutions. Next we will study the cases when $k < 0$, where we can observe conditions that lead to a rapidly increasing or decreasing solutions as well as solutions which blow-up.

3.2 Blow-up solutions

Theorem 4: *Let $k \in R$ and assume that f satisfies (3.1). If a local solution w to (2.1) blows up at some finite $s_0 \in R$, then*
$\liminf_{s\to s_0} w(s) = -\infty$ *and* $\limsup_{s\to s_0} w(s) = +\infty$

This suggests that uncontrollable behaviours of f take the form of increasingly wide oscillations. The proof of this theorem is discussed in (Berchio, Ferrero, Gazzola and Karageorgis, 2011)[1] We will investigate this property in a later section.

3.3 Rapidly Increasing or Decreasing Solutions

Theorem 5: *Let $k < 0$ and assume that f satisfies (3.1). In the case*

$$\sup_{t\in R} f(t) = M < +\infty \tag{3.37}$$

there exists a global solution w of (2.1) which is eventually positive, increasing and convex as $s \to +\infty$. In particular,

$$\lim_{s\to+\infty} w(s) = +\infty \tag{3.38}$$

In the case where f satisfies

$$\inf_{t\in R} f(t) = -M > -\infty \tag{3.39}$$

then there exists a global solution w of (2.1) which is eventually negative, decreasing and concave as $s \to +\infty$. In particular,

$$\lim_{s\to+\infty} w(s) = -\infty \tag{3.40}$$

Proof of Theorem 5:

We begin by assuming that (3.37) holds, and we can consider a solution w of (2.1) which satisfies the following initial conditions:

$$w(0) = 0, \quad w'(0) = 0, \quad w''(0) > \frac{M}{|k|} > 0, \quad w'''(0) = 0 \tag{3.41}$$

where M is defined as in (3.37). Since $k < 0$, (2.1) implies that $w''''(0) = -kw''(0) > M$. We can define

$$\bar{s} := \sup\{s > 0 : w''''(\sigma) > 0 \quad \text{for all} \quad \sigma \in (0, s)\} \in (0, +\infty]$$

By (3.37) and Theorem 1 we know that w is defined on the whole real line and we can claim that $\bar{s} = +\infty$. We can assume for the sake of contradiction that $\bar{s} < +\infty$, this implies

$$w''''(\bar{s}) = 0 \tag{3.42}$$

Since w''' is increasing on $(0, \bar{s}]$ and $w'''(0) = 0$, then w''' is positive on $(0, \bar{s}]$. This can be extended to show that w'', w' and w are positive on $(0, \bar{s}]$.
Therefore by (2.1) and (3.41):

$$w''''(\bar{s}) = |k|w''(\bar{s}) - f(w(\bar{s})) \geq |k|w''(0) - M > 0 \tag{3.43}$$

This contradicts (3.42). This proves that $\bar{s} = +\infty$. And in particular by the use of the above iterative scheme we have proved that w is positive, increasing and convex in $(0, +\infty)$. We can now see that (3.38) holds which completes the first part of the proof.
When we are instead considering (3.40), we can construct a completely analogous proof where we assume in (3.41) that $w''(0) < -M/|k| < 0$ and we reverse all signs.

This theorem helps us to predict the behaviour of solutions to our equation when $k < 0$. We have found conditions which lead to solutions which approach $-\infty$ and $+\infty$ in finite time, we will investigate these solutions later in the project.

Chapter 4

Finite-Difference Methods

4.1 Euler's Method

The Euler Method is a first-order numerical procedure for solving ordinary differential equations. It is the most basic explicit method for numerical integration of ordinary differential equations. Suppose we wish to solve the initial value problem:

$$\frac{dy}{dt} = f(y) \quad y(t_0) = y_0 \tag{4.1}$$

The Euler method can be derived in a number of ways. It can be done quite simply geometrically but we will derive it by method of Taylor Expansions as this can be more clearly extended to higher-order methods. Consider the Taylor expansion of the function y about t_0

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{1}{2}h^2 y''(t_0) + \frac{1}{3!}h^3 y'''(t_0) + \ldots + \frac{1}{n!}h^n y^{(n)}(t_0) \tag{4.2}$$

The Euler method is a first-order method so we can approximate this expansion by truncating at the first term.

$$y(t_0 + h) \sim y(t_0) + hy'(t_0) \tag{4.3}$$

This gives us an expression for the first point after t_0. We can continue this for every point along our function.

$$y_{n+1} = y_n + hf(y_n) \text{ where } f(y_n) = y'(t_n) \tag{4.4}$$

As you can see in Figure 4.1, the Euler Method takes the value of the function at a point and uses the derivative at that point to approximate a new y-value a distance of h away. This is a valid approximation for functions whose derivatives do not change greatly between steps, however if we have a rapidly changing function (which many we would like to consider are) the error quickly grows. The Euler Method is a first-order method so the global error of these solutions is proportional to h.

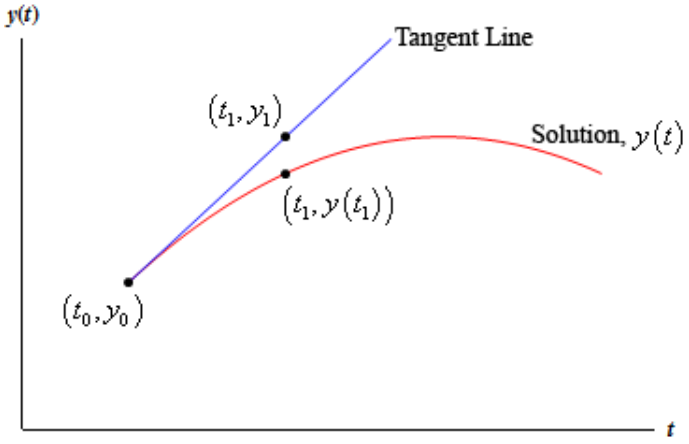

Figure 4.1: Euler Method approximating a function [3]

4.2 Runge Kutta Methods

Runge-Kutta methods follow a similar idea to the Euler method however they are based on using higher order terms of the Taylor series expansion. The derivation of the formulae for this method is very tedious and offers little intuition so I have relegated it to the appendices.

$$y_{n+1} = y_n + \frac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right), \quad \begin{aligned} k_1 &= f\left(t_n, y_n\right) \\ k_2 &= f\left(t_n + \tfrac{h}{2}, y_n + h\tfrac{k_1}{2}\right) \\ k_3 &= f\left(t_n + \tfrac{h}{2}, y_n + h\tfrac{k_2}{2}\right), \\ k_4 &= f\left(t_n + h, y_n + hk_3\right) \end{aligned} \tag{4.5}$$

Here y_{n+1} is the RK4 approximation of $y(t_{n+1})$, and the next value y_{n+1} is determined by the present value y_n plus the weighted average of four increments, where each increment is the product of the size of the interval, h, and an estimated slope/derivative specified by the differential equation. Figure 4.2 offers a graphical intuition of how each of these terms contributes to an accurate approximation.

- k_1 is the derivative at the beginning of the interval, it is exactly the same as the $f(y_n)$ term in the Euler Method.
- k_2 is the derivative at the midpoint of the interval, using y and k_1
- k_3 is the derivative at the midpoint, but now using y and k_2
- k_4 is the derivative at the end of the interval using y and k_3

So we have 4 different slope values where extra weight is added to the midpoint. Using these formulae we can establish a 4th order method of approximating solutions to our equation. This means that the global error of our approximation should be of order h^4. This is far more efficient than Euler's Method as it can produce a much better result in a small fraction of the number of steps.

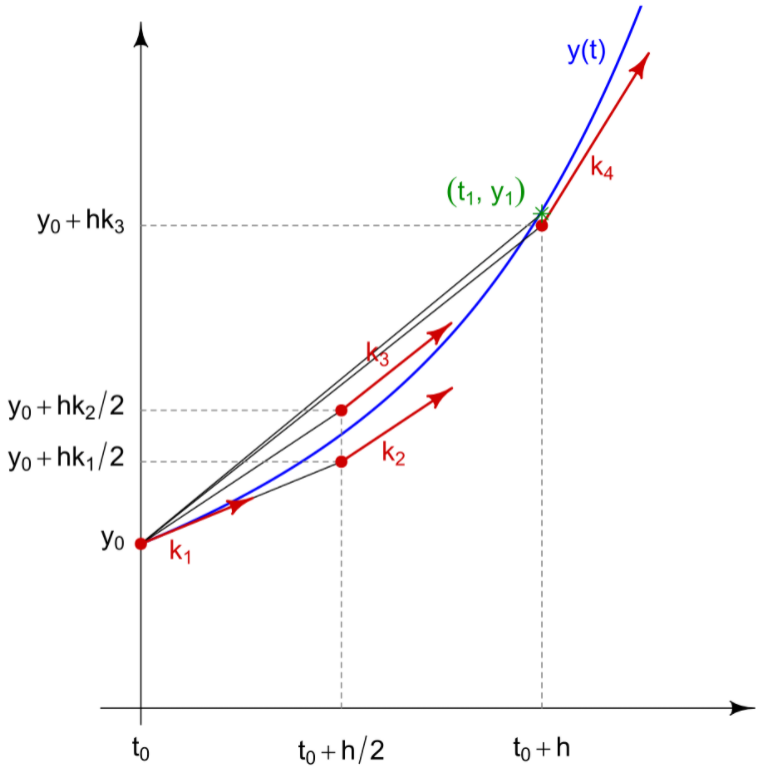

Figure 4.2: RK4 Method approximating a function [8]

4.3 Decomposing the Equation

We now have a suitable algorithm for modelling solutions of our equation but we are left with another issue. The RK4 algorithm is only suitable for first-order ODEs whereas we have in our case a fourth order ODE, this means we will have to decompose our fourth order equation in to four separate 1st order equations. Fortunately, this can be done rather simply.

$$\omega^{(n)}(s) \equiv x_n \quad \begin{array}{l} (1)\ \frac{d}{dt}(x_0) = x_1 \\ (2)\ \frac{d}{dt}(x_1) = x_2 \\ (3)\ \frac{d}{dt}(x_2) = x_3 \\ (4)\ \frac{d}{dt}(x_3) = \frac{d}{dt}\left[-kx_2 - f(x_0)\right] = x_4 \end{array} \tag{4.6}$$

As you can see, each of the derivatives of ω can simply be written in terms of each other to form the first three differential equations. Then for the last equation we simply rearrange our original fourth order equation.

With the equation decomposed it is rather simple to approximate solutions to our equation with the Runge-Kutta algorithm. The most important thing to take note of is that when these four ODEs are evaluated, they must be incremented simultaneously as opposed to one after the other to obtain a correct result. You can see how this is done in Figure 4.3

The C++ code uses a for-loop which allows all four of the first-order ODEs to be incremented

as a weighted average of four slopes in the interval simultaneously. It also outputs our values of $w(s)$ for each value of s which can then be inputted to Mathematica to model our solutions.

We will discuss the results of this in the next section.

Note: In my code I have used the variable t instead of s, they have completely equivalent meanings.

```
31      for (int i=0; i<n; i++)
32      {
33
34    a1 = h*x1;
35    b1 = h*x2;
36    c1 = h*x3;
37    k1 = h*f(x0,x1,x2,x3);
38
39     a2 = h*(x1+b1/2);
40     b2 = h*(x2+c1/2);
41     c2 = h*(x3+k1/2);
42     k2 = h*f(x0+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);
43
44    a3 = h*(x1+b2/2);
45    b3 = h*(x2+c2/2);
46    c3 = h*(x3+k2/2);
47    k3 = h*f(x0+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
48
49    a4 = h*(x1+b3);
50    b4 = h*(x2+c3);
51    c4 = h*(x3+k3);
52    k4 = h*f(x0+a3,x1+b3,x2+c3,x3+k3);
53
54  x0 += (1.0/6.0)*(a1+(2*a2)+(2*a3)+a4);
55  x1 += (1.0/6.0)*(b1+(2*b2)+(2*b3)+b4);
56  x2 += (1.0/6.0)*(c1+ (2.0*c2) + (2.0*c3) + c4);
57  x3 += (1.0/6.0)*(k1+(2.0*k2)+(2.0*k3)+k4);
58
59
60          t = t0 + h*double(i+1) ;
61          cout  << t << "" "\n" ;
62          cout  << x0 << "" "\n";
```

Figure 4.3: C++ code to implement the RK4 Algorithm

Chapter 5

Results and Numerical Analysis

5.1 Numerical Analysis of Oscillating Solutions

A good place to begin would be by seeing if we can demonstrate the results of Theorems 1 and 2 which relate to global solutions of our ODE. Theorem 1 states that if we have a real value of k and a nonlinearity f which satisfies (3.1) and (3.2), any local solution to (2.1) should exist for all real-values of s.

The results of this theorem are far too varied for us to exhaust all possibilities so we should simply attempt to demonstrate this result. Firstly we would like a value for k so I have arbitrarily chosen $k = 10$. We now need a nonlinearity $f(w(s))$ that satisfies (3.1) and (3.2). I have chosen $f(w(s)) = e^w - 1$ for this purpose. If we start our algorithm with all values of $w(s)$ and its derivatives initialised to 0, it will simply give a null result so I have set $w'(s) = 1$. The theorem suggests that the result should be a global solution.

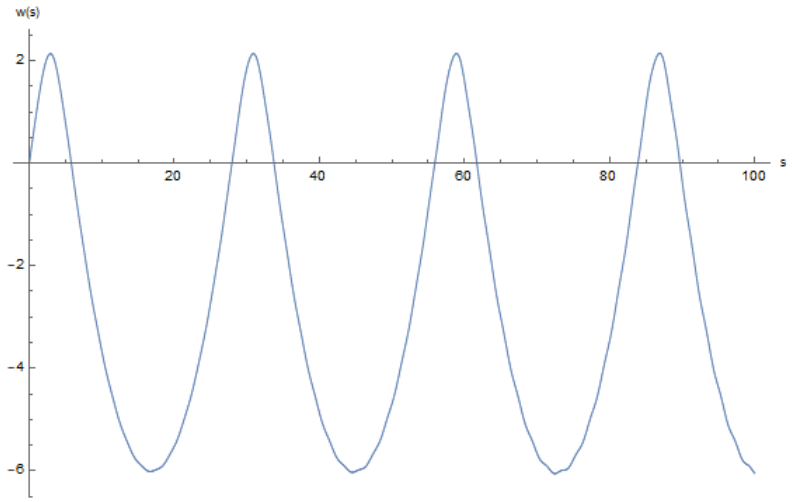

Figure 5.1: Oscillating solution for $f(w(s)) = e^w - 1$, k=10

I have cut off the solution in Figure 5.3 off at $s = 100$ for clarity but it is worth noting that this

oscillatory behaviour continues as $s \to \infty$ demonstrating the proposed property of the solution that is is defined for all real-values of s.

We should now turn our attention to Theorem 2, which states that if $k \geq 0$, f satisfies (3.1) and w is a global solution to (2.1) then $w(s)$ will change signs infinitely many times as $s \to \infty$. It should be clear that our solution in Figure (5.1) fits this description perfectly. It is a global oscillating solution which changes sign infinitely many times as s approaches infinity.
I then ran the algorithm with $k = 1.6$ and the nonlinearity $f(w(s)) = w^3$ which satisfies (3.1), and observed another global oscillating solution, further demonstrating the results proposed in Theorem 2.

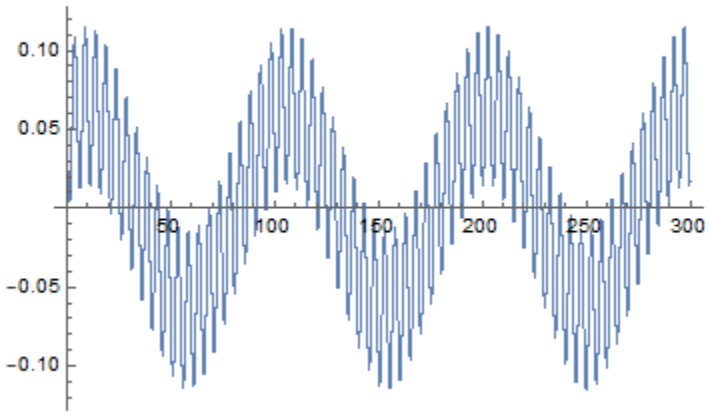

Figure 5.2: Oscillating solution for $f(w(s)) = w^3$, k=1.6

5.2 Numerical Analysis of Blow-up Solutions

There are many different blow up solutions which can occur for this equation, a precise condition for blow-up was determined in (Radu, Toundykov and Trageser, 2014)[7], which is again unfortunately beyond the scope of this project. However through experimentation a number of these have been reproduced (Figure (5.3) + Figure (5.4)) which agree with the statement in Theorem 4 that they consist of increasingly wide oscillations.

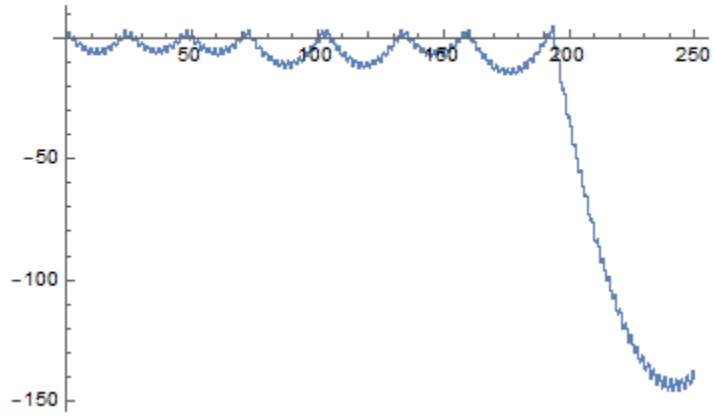

Figure 5.3: Blow-up solution for $f(w(s)) = e^w - 1$, k=8

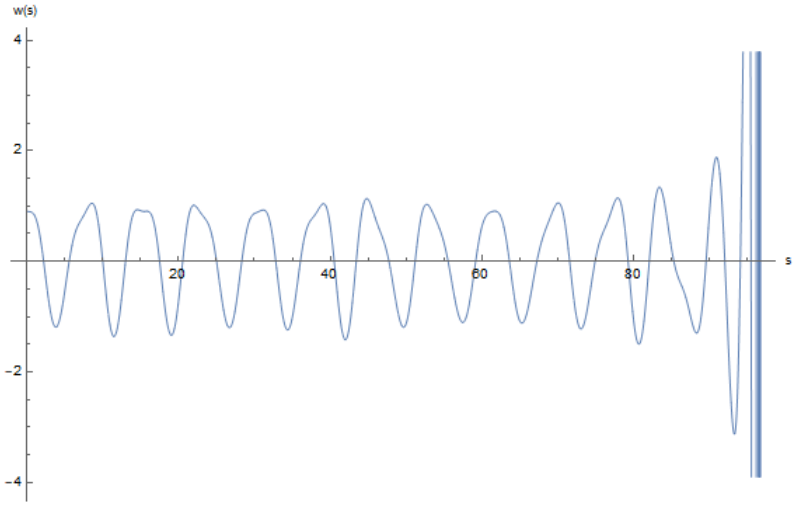

Figure 5.4: Blow-up solution for $f(w(s)) = w + w^3$, k=3.6

Both of these figures represent increasingly volatile oscillations which when they approach infinity become too large to calculate with our current algorithm. An idea for a future project could be to attempt to reliably study the behaviour of these blow-up solutions.

5.3 Numerical Analysis of Rapidly Increasing or Decreasing Solutions

We will now move on to investigating the proposals of Theorem 5.
Both cases of Theorem 5 have a value of $k < 0$, and a nonlinearity f which satisfies (2). I will arbitrarily choose a value of $k = -10$

Case 1: we want a function f such that $\sup_{t\in R} f(t) = M < +\infty$. For this purpose I will choose the function $f(w(s)) = \arctan(w)$.

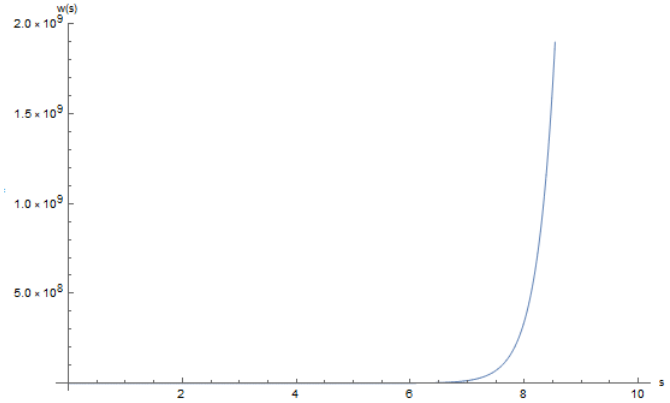

Figure 5.5: Rapidly increasing solution for $f(w(s)) = \arctan(w)$, k=-10

Figure 5.5 shows the expected solution predicted by Theorem 5. It is a global solution which is eventually positive, increasing and convex as $s \to +\infty$. And we can infer from the graph that $\lim_{s\to+\infty} w(s) = +\infty$ for this solution.

Case 2: in this case we are looking for a function f such that $\inf_{t\in R} f(t) = -M > -\infty$. For this purpose I will choose $f(w(s)) = \frac{w}{1+w^2}$

We can see in Figure 5.6 the behaviour of the solution predicted by Theorem 5. We have found a global solution which is eventually negative, decreasing and convex as $s \to +\infty$. We can infer from the graph that $\lim_{s\to+\infty} w(s) = -\infty$ for this solution.

It is again worth noting that these graphs have been cut off at a low value of s for clarity but the trajectories shown continue as s approaches infinity. It is also important to note that these trajectories were observed for almost every choice of initial values for $w(s)$ and its higher order derivatives.

We have found that our algorithm reproduces the expected results of the analytic solutions of the equation. Therefore we can move on to studying a more interesting problem.

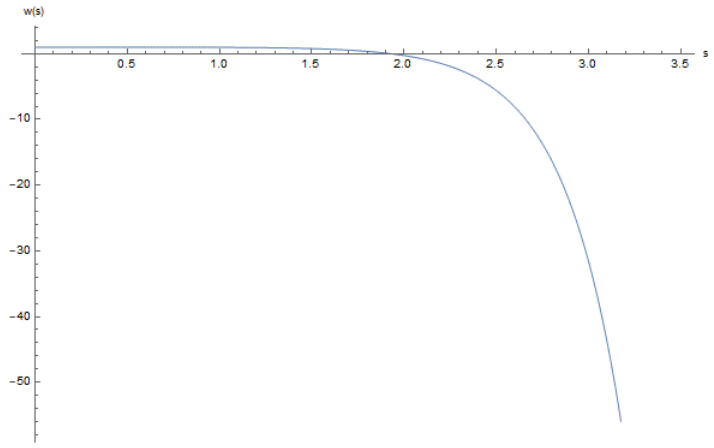

Figure 5.6: Rapidly Decreasing solution for $f(w(s)) = \frac{w}{1+w^2}$, k=-10

Chapter 6

Travelling Waves in a Suspension Bridge

We now move on to the physical problem most notably studied in (McKenna, P.J. and Walter, W., 1990.)[2] which is using our equation (2.1) to model a Suspension bridge which is supported by one cable on each side. In this case, we choose the idealized nonlinearity

$$f(w(s)) = (s^+ - 1) \ \ where \ \ s^+ = \begin{cases} s, & s \geq 0 \\ 0, & s < 0 \end{cases} \tag{6.1}$$

This f is chosen to simulate the effect that the cables hold the bridge up but the constant force of gravity holds it down.

For the purposes of this problem we will define our $k = c^2$ with c being the speed of the travelling wave. Our equation now takes the form

$$w''''(s) + c^2 w''(s) + \left(s^+ - 1\right) = 0, \quad s \in \left[-\frac{L}{2}, \frac{L}{2}\right] \tag{6.2}$$

where L is the length of the bridge

We propose that it is possible to find travelling wave solutions to this equation and that these travelling wave solutions simulate the motion of a suspension bridge.

This discussion was first motivated in (McKenna, P.J. and Walter, W., 1990.)[2] where they very notably have included a report from the Chief Engineer of the Golden Gate Bridge where he details his observations of the motion of the bridge during a severe wind storm.

He is noted as saying that "The suspended structure of the bridge was undulating vertically in a wavelike motion of considerable amplitude. ... The wave motion appeared to be a running wave similar to that made by cracking a whip."

In an attempt to simulate these results, it would be useful to note that we are looking for homoclinic solutions of our equation and to consider the physical restraints presented by the problem.

We should also take note of a proposition from (Berchio, Ferrero, Gazzola and Karageorgis, 2011)[1] which narrows down our parameter k. Since for this problem we are dealing with $k = c^2$ we would expect real values for c and therefore strictly positive k values but this theorem shows we are not missing any homoclinic solutions for negative values of k.

Proposition 6: *Let $k \leq 0$ and f satisfy (3.1). Then eqn (2.1) (and by extension eqn (6.2)) has no homoclinic solutions.*

Proof

This proposition can be proven by introducing a third energy function. Define:

$$H(s) := w'(s)w''(s) - w(s)w'''(s) - kw(s)w'(s) \tag{6.3}$$

then take its derivative and antiderivative

$$H'(s) = w''(s)^2 - kw'(s)^2 + w(s)f(w(s)) \tag{6.4}$$

$$G(s) := w'(s)^2 - w(s)w''(s) - \frac{k}{2}w(s)^2 \tag{6.5}$$

If $k \leq 0$ and (3.1) holds, by (6.4) we can deduce that $H'(s) \geq 0$ such that H is nondecreasing and G is convex. If w is a homoclinic solution then $\lim_{s\to\pm\infty} H(s) = 0$, see Lemma 3. Hence, $H = H' = 0$ and we see that no homoclinic solution exists for (2.1) or (6.2).

This paper also presents another proposition:

Proposition 7: *Assume that f satisfies (2), $f'(0) = 1$. and*

$$\lim_{t\to-\infty} \frac{F(t)}{t^2} = 0 \tag{6.6}$$

then there exists a homoclinic solution to (2.1) (and by extension (6.2)) for almost every $k \in (0, 2)$.

There is a very extensive proof of this in the literature which is beyond the scope of this project but we will use this proposition to inform our decision of a value for k.

From the above propositions it is clear that we should choose a value for k between 0 and 2. We should choose a value $L \in R < \infty$ to give a fixed bridge length. And at each end point $s = -L/2$ and $s = L/2$ our solution w should be very close to 0 to simulate the fixed end points of the bridge. We have hopefully now built up the parameters necessary to attempt to find a travelling wave solution such as that described by Peletier and Troy.

The only thing remaining is to decide the initial values required to find a physical solution. If we simply use random initial values for $w(s)$ and its higher order derivatives we will almost certainly get a solution that blows up or behaves unreasonably, therefore we must be careful about our initial conditions. This can be achieved by placing constraints on the initial conditions and then testing a large number of initial values to see which gives the most accurate result.

Our first constraint on the initial values is given by the fact that our homoclinic travelling wave solution will be an even function about the origin. This means that we can initialise

$$w'(s) = w'''(s) = 0 \tag{6.7}$$

Our energy function defined in (2.2) will allow us to initialise $w(s)$ and $w''(s)$. For homoclinic solutions of our equation, all derivatives of w must go to 0 as $s \to +\infty$. This energy is constant and its limit at infinity is zero, therefore the energy must be zero.

Using this information along with (6.7) we can impose some restrictions on the remaining initial conditions.

$$E(s) := \frac{1}{2}w''(s)^2 - F(w(s)) = 0 \tag{6.8}$$

$$w''(s) = \pm\sqrt{2F(w(s))} \tag{6.9}$$

From (6.9), we see that we have a strict restriction on $w''(s)$ which has a strict dependence on $w(s)$.

This means the only initial value we need to worry about inputting is $w(s)$. So we should test a large number of values of $w(s)$ and see which gives the solution closest to the origin at the endpoints. The code which shows exactly how this is done is included in the Appendices.

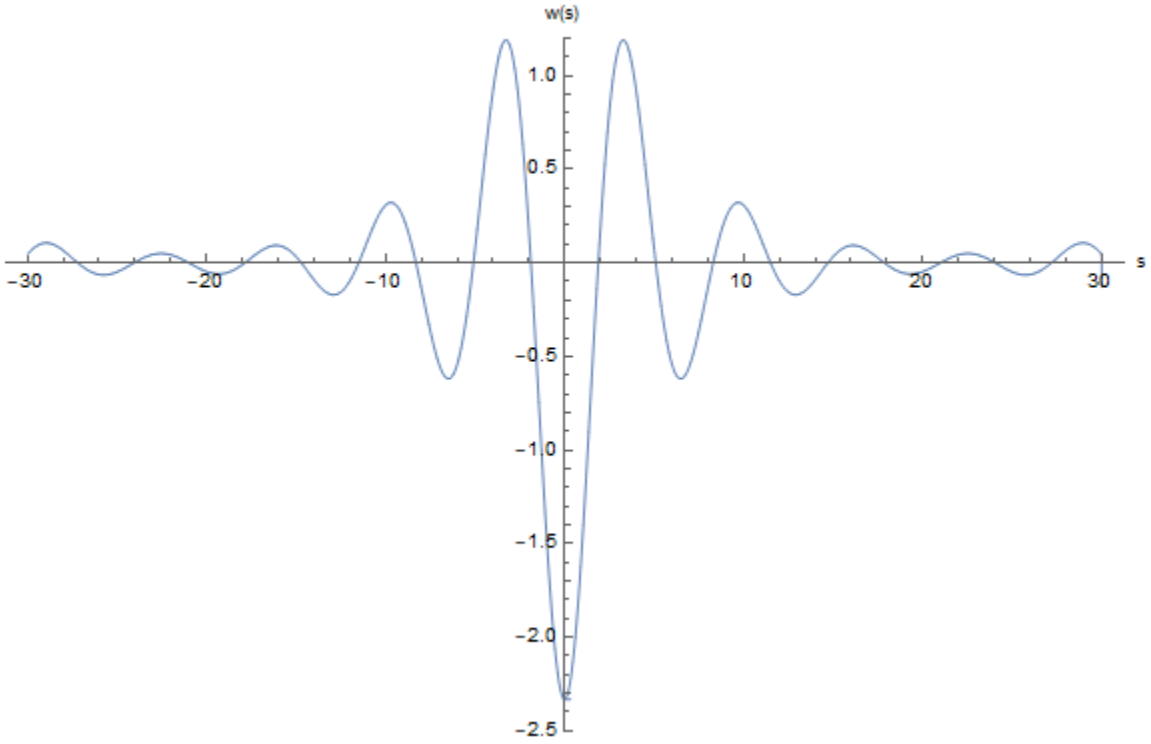

Figure 6.1: Physical solution for k=2

We can observe in Figure (6.1) the expected behaviour of the solution proposed by Peletier and Troy where the solution takes the form of a travelling wave. This also resembles the physical phenomenon noted by the engineer of the Golden Gate Bridge in his testimony.

By this we can see the justification for proposing equation (2.1) as a valid model for analysing the behaviour of travelling waves in a suspension bridge.

Chapter 7

Error Analysis

7.1 Deviation from Analytical Solution

In this section, we will study the error produced by our chosen finite-difference method of approximating solutions to our equation. There will always be an error involved with computer-generated solutions of a differential equation due to the fact that some kind of lattice must be used as it is impossible to generate truly infinitesimal distances. However, it is our goal to minimise these errors to an acceptable degree.

Theorem 8: *Let $\eta(s;h)$ be the approximate solution obtained by a one-step method of order p, to the solution $w(s)$ of the initial value problem:*

$$w' = f(s,w), \quad w(s_0) = w_0, \quad w_0 \in [a,b] \tag{7.1}$$

then $\eta(s;h)$ has an asymptotic expansion of the form

$$\eta(s;h) = w(s) + h^p e_p(s) + h^{p+1} e_{p+1}(s) + \cdots + h^N e_N(s) \tag{7.2}$$

which is valid for all $s \in [a,b]$ and all $h = h_n = (s - s_0)/n, n = 1, 2, \ldots$

Proof: Proof can be found in Stoer, Josef, and Roland Bulirsch. Introduction to numerical analysis. Vol. 12. Springer Science Business Media, 2013.[5]

Using this theorem we can calculate a value for $e(s;h)$ - The Global Discretization Error, discretization error is the error resulting from the fact that a function of a continuous variable is represented in the computer by a finite number of evaluations. Suppose that a method of order p has an asymptotic expansion of the form (7.2) so that:

$$e(s;h) = \eta(s;h) - w(s) = h^p e_p(s) + O\left(h^{p+1}\right) \tag{7.3}$$

Having found the approximate value $\eta(s;h)$ with stepsize h, one computes for the same s, but with a stepsize $h/2$, the approximation $\eta(s;h/2)$. For sufficiently small h [and $e_p(s) \neq 0$] one then has in first approximation

$$\eta(s;h) - w(s) \doteq e_p(s) \cdot h^p \tag{7.4}$$

$$\eta\left(s;\frac{h}{2}\right) - w(s) \doteq e_p(s) \cdot \left(\frac{h}{2}\right)^p \tag{7.5}$$

subtracting the second equation from the first gives

$$\eta(s;h) - \eta\left(s;\frac{h}{2}\right) \doteq e_p(s) \cdot \left(\frac{h}{2}\right)^p (2^p - 1) \tag{7.6}$$

$$e_p(s)\left(\frac{h}{2}\right)^p \doteq \frac{\eta(s;h) - \eta(s;h/2)}{2^p - 1} \tag{7.7}$$

Then one can substitute into (7.6) to find

$$\eta\left(s;\frac{h}{2}\right) - w(s) \doteq \frac{\eta(s;h) - \eta(s;h/2)}{2^p - 1} \tag{7.8}$$

This is a general formula for the Global Discretization error of a one-step method. We can fill in $p = 4$ for our 4th Order Runge Kutta Algorithm to find

$$\eta\left(s;\frac{h}{2}\right) - w(s) \doteq \frac{\eta(s;h) - \eta(s;h/2)}{15} \tag{7.9}$$

Using this result, I graphed this value for a number of different values for h. You can note in the figures how this error evolves over time for different step sizes.

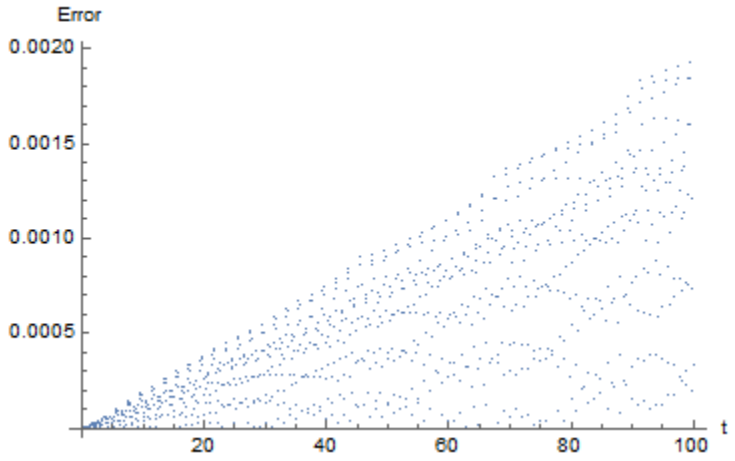

Figure 7.1: Error vs time for $h = 0.1$

In figure 7.1 we can see that a step size of $h = 0.1$ gives a maximum error of 0.002 which I would consider too high a margin. But in figure 7.2 we can see that for $h = 0.001$ we only get a maximum error of the order 10^{-11}. I would consider this an acceptable order of error and this is why for all of my simulations I used this step size.
It is also worth noting that these errors are all of roughly order h^4 which matches what we know about the Runge-Kutta algorithm.

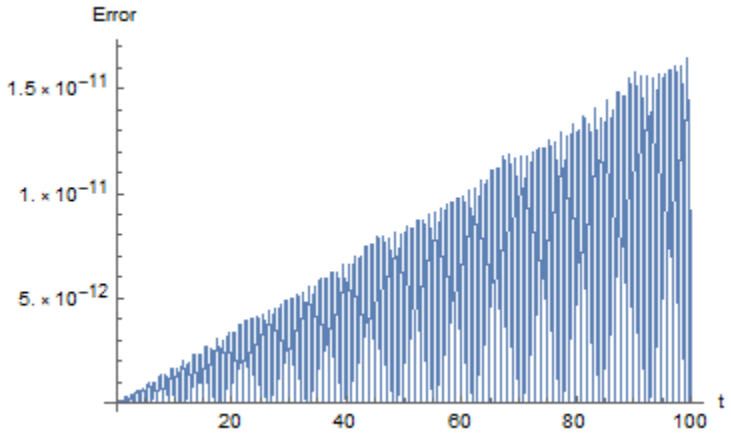

Figure 7.2: Error vs. time for $h = 0.001$

7.2 Conservation of Energy

Another useful way of checking the validity of our algorithm is to calculate our energy function (2.2) and see if it is conserved over time as we deduced earlier that it should be. After performing this calculation several times, it was seen that the energy was largely conserved over time, with the exception of small fluctuations which can be explained by the Discretization error discussed in the previous section. In Figure 7.3 you can see this result.

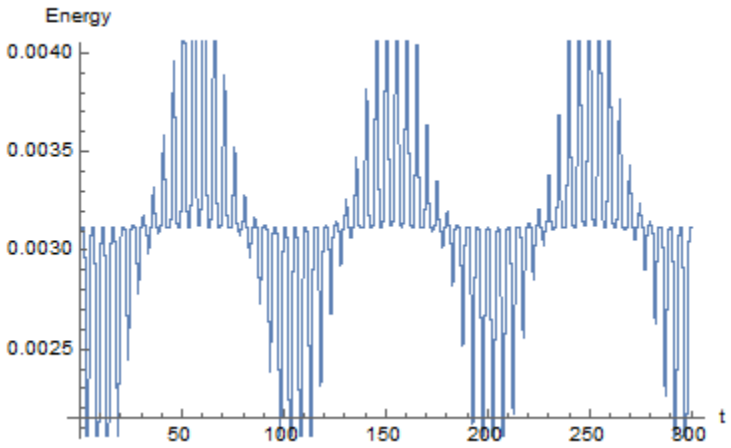

Figure 7.3: Energy vs. time

Chapter 8

Appendices

8.1 Appendix A: Proof of Lemma 4

Proof: We will prove the statement for $s \to +\infty$ but the proof for $s \to -\infty$ is completely analogous. C denotes a general constant and $\delta_i(s)$ denotes continuous functions such that $\delta_i(s) \to 0$ as $s \to +\infty$. Assuming that $w(s) \to 0$ as $s \to +\infty$ we can rewrite (2.1) as

$$\left(e^s \left[w'''(s) - w''(s) + (k+1)w'(s) - (k+1)w(s)\right]\right)' = \delta_1(s)e^s \tag{8.1}$$

where $\delta_1(s) = -f(w(s)) - (k+1)w(s)$. We know that

$$\forall \varepsilon > 0 \quad \exists \sigma > 0 \quad \text{s.t.} \quad |\delta_1(s)| < \varepsilon \quad \forall s > \sigma \tag{8.2}$$

Let $\varepsilon > 0$ be fixed and we can integrate (8.1) over $(0, s)$ for any $s > \sigma$ to obtain

$$e^s \left[w'''(s) - w''(s) + (k+1)w'(s) - (k+1)w(s)\right] = C + \int_0^\sigma \delta_1(t)e^t dt + \int_\sigma^s \delta_1(t)e^t dt \tag{8.3}$$

and from this we obtain

$$\left|w'''(s) - w''(s) + (k+1)w'(s) - (k+1)w(s)\right| Ce^{-s} + \varepsilon e^{-s} \int_\sigma^s e^t dt = Ce^{-s} + \varepsilon \quad \forall s > \sigma \tag{8.4}$$

Due to the arbitrariness of ϵ and letting $s \to +\infty$. It is proven that

$$w'''(s) - w''(s) + (k+1)w'(s) - (k+1)w(s) = \delta_2(s) \tag{8.5}$$

for a continuous function δ_2 which vanishes as $s \to +\infty$. We can rewrite this equation as
$\left(e^s \left[w''(s) - 2w'(s) + (k+3)w(s)\right]\right)' = \delta_3(s)e^s$
where $\delta_3(s) = \delta_2(s) + (2k+4)w(s)$.
We can continue this scheme as follows:

$$w''(s) - 2w'(s) + (k+3)w(s) = \delta_4(s) \tag{8.6}$$

$$\left(e^s \left[w'(s) - 3w(s)\right]\right)' = \delta_5(s)e^s \tag{8.7}$$

where $\delta_5(s) = \delta_4(s) - (k+6)w(s)$. We can once more utilise this scheme to show from (8.7)

$$w'(s) - 3w(s) \to 0 \quad \text{as} \quad s \to +\infty \tag{8.8}$$

Since $w(s) \to 0$ this implies $w'(s) \to 0$. Now in turn, these two limits coupled with (8.6) imply that $w''(s) \to 0$. In the same manner we can find that $w'''(s) \to 0$ from (8.5) and $w''''(s) \to 0$ from (2.1).

8.2 Appendix B: Derivation of RK4 Algorithm

In general, we can write a Runge-Kutta mathod of order 's' as:

$$y_{t+h} = y_t + h \cdot \sum_{i=1}^{s} a_i k_i + \mathcal{O}\left(h^{s+1}\right) \quad where \quad k_i = y_t + h \cdot \sum_{j=1}^{s} \beta_{ij} f\left(k_j, t_n + \alpha_i h\right)$$

We continue by using this formula for $s = 4$ as it is a fourth order method and we choose to evaluate k-values at the starting point, the midpoint and the end point of our interval $(t, t+h)$. Therefore:

$$\begin{array}{ll} \alpha_1 = 0 & \beta_{21} = \frac{1}{2} \\ \alpha_2 = \frac{1}{2} & \beta_{32} = \frac{1}{2} \\ \alpha_3 = \frac{1}{2} & \beta_{43} = 1 \\ \alpha_4 = 1 & \end{array}$$

with $\beta_{ij} = 0$ in all other cases. We may now define the following quantities:

$$y^1_{t+h} = y_t + hf\left(y_t, t\right)$$

$$y^2_{t+h} = y_t + hf\left(y^1_{t+h/2}, t + \tfrac{h}{2}\right)$$

$$y^3_{t+h} = y_t + hf\left(y^2_{t+h/2}, t + \tfrac{h}{2}\right)$$

where $\mathrm{y}^1_{t+h/2} = \frac{y_t + y^1_{t+h}}{2}$ and $y^2_{t+h/2} = \frac{y_t + y^2_{t+h}}{2}$ If we define:

$$\begin{aligned} k_1 &= f\left(y_t, t\right) \\ k_2 &= f\left(y^1_{t+h/2}, t + \frac{h}{2}\right) = f\left(y_t + \frac{h}{2}k_1, t + \frac{h}{2}\right) \\ k_3 &= f\left(y^2_{t+h/2}, t + \frac{h}{2}\right) = f\left(y_t + \frac{h}{2}k_2, t + \frac{h}{2}\right) \\ k_4 &= f\left(y^3_{t+h}, t + h\right) = f\left(y_t + hk_3, t + h\right) \end{aligned}$$

we can show that the following equalities hold up to $\mathcal{O}\left(h^2\right)$

$$\begin{aligned} k_2 = f\left(y^1_{t+h/2}, t + \tfrac{h}{2}\right) &= f\left(y_t + \tfrac{h}{2}k_1, t + \tfrac{h}{2}\right) \\ &= f\left(y_t, t\right) + \tfrac{h}{2}\tfrac{d}{dt} f\left(y_t, t\right) \end{aligned}$$

$$\begin{aligned} k_3 = f\left(y^2_{t+h/2}, t + \tfrac{h}{2}\right) &= f\left(y_t + \tfrac{h}{2} f\left(y_t + \tfrac{h}{2}k_1, t + \tfrac{h}{2}\right), t + \tfrac{h}{2}\right) \\ &= f\left(y_t, t\right) + \tfrac{h}{2}\tfrac{d}{dt}\left[f\left(y_t, t\right) + \tfrac{h}{2}\tfrac{d}{dt} f\left(y_t, t\right)\right] \end{aligned}$$

$$\begin{aligned} k_4 = f\left(y^3_{t+h}, t + h\right) &= f\left(y_t + hf\left(y_t + \tfrac{h}{2}k_2, t + \tfrac{h}{2}\right), t + h\right) \\ &= f\left(y_t + hf\left(y_t + \tfrac{h}{2} f\left(y_t + \tfrac{h}{2} f\left(y_t, t\right), t + \tfrac{h}{2}\right), t + \tfrac{h}{2}\right), t + h\right) \\ &= f\left(y_t, t\right) \dashv h\tfrac{d}{dt}\left[f\left(y_t, t\right) + \tfrac{h}{2}\tfrac{d}{dt}\left[f\left(y_t, t\right) + \tfrac{h}{2}\tfrac{d}{dt} f\left(y_t, t\right)\right]\right] \end{aligned}$$

We can now express the general formula using what we just derived:

$$
\begin{gathered}
y_{t+h} = \\
y_t + h\left\{a \cdot f\left(y_t, t\right) + b \cdot \left[f\left(y_t, t\right) + \frac{h}{2}\frac{d}{dt}f\left(y_t, t\right)\right] + \right. \\
+c \cdot \left[f\left(y_t, t\right) + \frac{h}{2}\frac{d}{dt}\left[f\left(y_t, t\right) + \frac{h}{2}\frac{d}{dt}f\left(y_t, t\right)\right]\right] + \\
\left. +d \cdot \left[f\left(y_t, t\right) + h\frac{d}{dt}\left[f\left(y_t, t\right) + \frac{h}{2}\frac{d}{dt}\left[f\left(y_t, t\right) + \frac{h}{2}\frac{d}{dt}f\left(y_t, t\right)\right]\right]\right]\right\} + \mathcal{O}\left(h^5\right)
\end{gathered}
$$

$$
\begin{gathered}
= y_t \dashv \cdot a \cdot hf_t \dashv \cdot b \cdot hf_t + b \cdot \frac{h^2}{2}\frac{df_t}{dt} \dashv c \cdot hf_t + c \cdot \frac{h^2}{2}\frac{df_t}{dt} + \\
+c \cdot \frac{h^3}{4}\frac{d^2f_t}{dt^2} + d \cdot hf_t + d \cdot h^2\frac{df_t}{dt} + d \cdot \frac{h^3}{2}\frac{d^2f_t}{dt^2} + d \cdot \frac{h^4}{4}\frac{d^3f_t}{dt^3} + \mathcal{O}\left(h^5\right)
\end{gathered}
$$

and then we can compare this with the Taylor Series of y_{t+h} around y_t

$$
\begin{aligned}
y_{t+h} &= y_t + h\dot{y}_t + \frac{h^2}{2}\ddot{y}_t + \frac{h^3}{6}y_t^{(3)} + \frac{h^4}{24}y_t^{(4)} + \mathcal{O}\left(h^5\right) = \\
&= y_t + hf\left(y_t, t\right) + \frac{h^2}{2}\frac{d}{dt}f\left(y_t, t\right) + \frac{h^3}{6}\frac{d^2}{dt^2}f\left(y_t, t\right) + \frac{h^4}{24}\frac{d^3}{dt^3}f\left(y_t, t\right)
\end{aligned}
$$

This gives us a system of constraints for the constant coefficients:

$$
\begin{aligned}
a + b + c + d &= 1 \\
\frac{1}{2}b + \frac{1}{2}c + d &= \frac{1}{2} \\
\frac{1}{4}c + \frac{1}{2}d &= \frac{1}{6} \\
\frac{1}{4}d &= \frac{1}{24}
\end{aligned}
$$

When solved gives $a = \frac{1}{6}, b = \frac{1}{3}, c = \frac{1}{3}, d = \frac{1}{6}$ which gives us the Runge-Kutta algorithm formulae as stated above:

$$
y_{n+1} = y_n + \frac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right)
$$

$$
\begin{aligned}
k_1 &= f\left(t_n, y_n\right) \\
k_2 &= f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right) \\
k_3 &= f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right) \\
k_4 &= f\left(t_n + h, y_n + hk_3\right)
\end{aligned}
$$

8.3 Appendix C: Complete C++ code for generating Initial Conditions

```
#include <string>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <vector>
#include <math.h>
#include <tgmath.h>
#include <cstdlib>
#include <cmath>
using namespace std;

"c here is the speed of the Travelling wave"

double c = "insert value for c here"
double k = c*c;

"f1 and f2 make up the two pieces of the piecewise nonlinearity s+"

double f1(double x0, double x1, double x2,double x3)
{
    return -k*x2 - x0;
}

double f2(double x0, double x1, double x2,double x3)
{
    return -k*x2 +1;
}

"This energy is use for checking the validity of our algorithm,
it should be a conserved quantity"

double energy(double x0, double x1, double x2, double x3)
{
  return 0.5*x2*x2-k*0.5*x1*x1-x1*x3+x0+0.5;
}
```

```
"Here we initialise our algorithmm, we must enter an initial and final time,
 a number of steps 'n' and step-size 'h' as well as give w(s) and all of its
 higher order derivatives an initial value"

int main()
{
    double t0=, t1=;
    double x0= ,x1=, x2=;
    double x3 =;
    double t=t0;
    int n=1000000;
    double h = (t1-t0)/double(n);
    double a1,a2,a3,a4,b1,b2,b3,b4,c1,c2,c3,c4,k1,k2,k3,k4;


   std:: ofstream outfile ("Plot.csv"); //Writing a CSV file for plotting
   outfile << "x" <<","<< "y" << std::endl; //Writes headings

  "Here we begin testing our initial conditions, this section will run the algorithm for a huge number
   of values for w(s) and w''(s) and the values which give the lowest result at the end points will be
   saved under 'result'"

  double min = 1000;
  double result = 0;
  double start = 0.0;
  x2 = sqrt(-2*x0 -1);


for ( int i=0;i<1000000;++i)
{
    x0 =  **- 0.0000001*i;
    start = x0;
    x2 = sqrt(-2*x0 -1);
    x1 = 0.0;
    x3 = 0.0;
     cout << "x0: " << std::setprecision(12)  << x0 << endl;

     for (int i=0; i<n; i++)
   {

 a1 = h*x1;
 b1 = h*x2;
 c1 = h*x3;
 if(x0 > -1){
 k1 = h*f1(x0,x1,x2,x3);
        }
 else {
  k1 = h*f2(x0,x1,x2,x3);
    }
```

```
 a2 = h*(x1+b1/2);
 b2 = h*(x2+c1/2);
 c2 = h*(x3+k1/2);
 if(x0 > -1){
 k2 = h*f1(x0+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);
                }
 else{
   k2 = h*f2(x0+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);
 }

 a3 = h*(x1+b2/2);
 b3 = h*(x2+c2/2);
 c3 = h*(x3+k2/2);
 if (x0 > -1){
 k3 = h*f1(x0+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
                }
 else{
  k3 = h*f2(x0+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
 }

 a4 = h*(x1+b3);
 b4 = h*(x2+c3);
 c4 = h*(x3+k3);
 if (x0 > -1){
 k4 = h*f1(x0+a3,x1+b3,x2+c3,x3+k3);
              }
else{
 k4 = h*f2(x0+a3,x1+b3,x2+c3,x3+k3);
}


 x0 += (1.0/6.0)*(a1+(2*a2)+(2*a3)+a4);
 x1 += (1.0/6.0)*(b1+(2*b2)+(2*b3)+b4);
 x2 += (1.0/6.0)*(c1+ (2.0*c2) + (2.0*c3) + c4);
 x3 += (1.0/6.0)*(k1+(2.0*k2)+(2.0*k3)+k4);

 t = t0 + h*double(i+1) ;
 }

if (std::abs(x0)<std::abs(min)){

  min = x0;
  result = start;
 cout << " min: " << std::setprecision(12) << min << endl;
                                              }
}

  "We can then reinitialise all of our variables but with 'result' as our new value for w(s).
   Then we can run our RK4 algorithm for positive 't' and record position values for plotting"

  double start2 = result;
  cout << "\n x0 = "<<result;

  t0=, t1=; //Start and end time
  x1=, x2=sqrt(-2*result-1);
  x3=;
  t=t0;
```

```
for (int i=0; i<n; i++)
   {

 a1 = h*x1;
 b1 = h*x2;
 c1 = h*x3;
 if(result > -1){
 k1 = h*f1(result,x1,x2,x3);
          }
 else {
  k1 = h*f2(result,x1,x2,x3);
    }

  a2 = h*(x1+b1/2);
  b2 = h*(x2+c1/2);
  c2 = h*(x3+k1/2);
  if(result > -1){
  k2 = h*f1(result+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);
            }
  else{
    k2 = h*f2(result+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);
  }

 a3 = h*(x1+b2/2);
 b3 = h*(x2+c2/2);
 c3 = h*(x3+k2/2);
 if (result > -1){
 k3 = h*f1(result+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
               }
 else{
   k3 = h*f2(result+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
  }


  a4 = h*(x1+b3);
  b4 = h*(x2+c3);
  c4 = h*(x3+k3);
  if (result > -1){
  k4 = h*f1(result+a3,x1+b3,x2+c3,x3+k3);
            }
 else{
  k4 = h*f2(result+a3,x1+b3,x2+c3,x3+k3);
 }

 result += (1.0/6.0)*(a1+(2*a2)+(2*a3)+a4);
 x1 += (1.0/6.0)*(b1+(2*b2)+(2*b3)+b4);
 x2 += (1.0/6.0)*(c1+ (2.0*c2) + (2.0*c3) + c4);
 x3 += (1.0/6.0)*(k1+(2.0*k2)+(2.0*k3)+k4);

 outfile << t <<","<< result<<std::endl;
          t = t0 + h*double(i+1) ;
   }
```

```
  "We can then do the same thing again, but instead this time we will run it backwards
  through time to find our position values for negative values of 't'."

    t0=, t1=; //Start and end time
    result=start2, x1= ;
    x3 =;
    x2 = sqrt(-2*result-1);


   for (int i=0; i<n; i++)
   {

  a1 = h*x1;
  b1 = h*x2;
  c1 = h*x3;
  if(result > -1){
  k1 = h*f1(result,x1,x2,x3);
       }
  else {
   k1 = h*f2(result,x1,x2,x3);
    }

   a2 = h*(x1+b1/2);
   b2 = h*(x2+c1/2);
   c2 = h*(x3+k1/2);
   if(result > -1){
   k2 = h*f1(result+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);
        }
   else{
     k2 = h*f2(result+a1/2,x1+b1/2,x2+c1/2,x3+k1/2);


  a3 = h*(x1+b2/2);
  b3 = h*(x2+c2/2);
  c3 = h*(x3+k2/2);
  if (result > -1){
  k3 = h*f1(result+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
          }
  else{
    k3 = h*f2(result+a2/2,x1+b2/2,x2+c2/2,x3+k2/2);
   }

  a4 = h*(x1+b3);
  b4 = h*(x2+c3);
  c4 = h*(x3+k3);
  if (result > -1){
  k4 = h*f1(result+a3,x1+b3,x2+c3,x3+k3);
       }
 else{
  k4 = h*f2(result+a3,x1+b3,x2+c3,x3+k3);
 }

 result += (1.0/6.0)*(a1+(2*a2)+(2*a3)+a4);
 x1 += (1.0/6.0)*(b1+(2*b2)+(2*b3)+b4);
 x2 += (1.0/6.0)*(c1+ (2.0*c2) + (2.0*c3) + c4);
 x3 += (1.0/6.0)*(k1+(2.0*k2)+(2.0*k3)+k4);

 outfile << t <<","<< result<<std::endl;
       t = t0 - h*double(i+1) ;
    }

    outfile.close();
    return 0;
}
```

Bibliography

[1] Berchio, E., Ferrero, A., Gazzola, F. and Karageorgis, P., 2011. Qualitative behavior of global solutions to some nonlinear fourth order differential equations. Journal of Differential Equations, 251(10), pp.2696-2727.

[2] McKenna, P. J., and W. Walter. "Travelling waves in a suspension bridge." SIAM Journal on Applied Mathematics 50.3 (1990): 703-715.

[3] Calc Workshop, accessed 15 November 2020, https://calcworkshop.com/first-order-differential-equations/eulers-method-table/.

[4] Gazzola, Filippo, and Raffaella Pavani. "Blow up oscillating solutions to some nonlinear fourth order differential equations." Nonlinear Analysis: Theory, Methods Applications 74.17 (2011): 6696-6711.

[5] Stoer, Josef, and Roland Bulirsch. Introduction to numerical analysis. Vol. 12. Springer Science Business Media, 2013.

[6] Michael Peardon - Practical Numerical solutions lecture notes, 2020.

[7] Radu, P., Toundykov, D. and Trageser, J., 2014. Finite time blow-up in nonlinear suspension bridge models. Journal of Differential Equations, 257(11), pp.4030-4063.

[8] HilberTraum, Runge-Kutta slopes, accessed 21 November 2020
https://commons.wikimedia.org/wiki/File:Runge-Kuttaslopes.svgfilelinks