Numerical Methods for the Observational Model

Author:

George Crowley (217889)

Supervisors: Prof Anotida Madzvamuse, Dr James Van Yperen and Dr Eduard Campillo-Funollet

School of Mathematical and Physical Sciences
University of Sussex

May 2022

Acknowledgements

I would like to extend my sincere thanks to Anotida for giving me the opportunity to work under him, without whom this project would not have been possible. I would also like to extend my deepest gratitude to James, who has consistently been the source of encouragement and excellent ideas behind this project, who also without, would not have made this project possible. I would also like to thank James for his commitment on previous projects in which gave insight to the work in this thesis, and that I wish him the best in his academic journey. I am unable to express my full gratitude to you both. I would also like to thank those who I have spoken to in passing, that have also suggested thoughtful ideas that have contributed to this work.

I would also like to give thanks to my friends and family, and especially my partner Helen for their continued love and support.

Contents

List of Figures

List of Abbreviations, Parameters and Function Spaces/Norms

SIR	Susceptible - Infectious - Removed
ODE	Ordinary Differential Equation
IVP	Initial Value Problem
BVP	Boundary Value Problem
FEM	Finite Element Method
EOC	Estimated Order of Convergence
FTC	Fundamental Theorem of Calculus
IBP	Integration by Parts
RK4	Runge-Kutta 4'th Order Method

β	Average transmission rate
γ	Average removal rate
N	Population size
r	Under-reporting parameter $\in (0, 1)$

For the following definitions, $\Omega \subset \mathbb{R}$, ($\Omega$ Bounded).

$$\mathcal{L}^2(\Omega) := \left\{ f : \Omega \to \mathbb{R} \;\middle|\; \int_\Omega f(x)^2\, dx < \infty \right\}$$

$$\|f\|_{\mathcal{L}^2(\Omega)} := \left(\int_\Omega f(x)^2\, dx \right)^{\frac{1}{2}}$$

$$\|f\|_{\mathcal{L}^\infty(\Omega)} := \sup_{x \in \Omega} |f(x)|$$

$$\mathcal{H}^1(\Omega) := \left\{ f : \Omega \to \mathbb{R} \;\middle|\; \|f\|^2_{\mathcal{L}^2(\Omega)} + \left\| \frac{df}{dx} \right\|^2_{\mathcal{L}^2(\Omega)} < \infty \right\}$$

$$\|f\|_{\mathcal{H}^1(\Omega)} := \left(\|f\|^2_{\mathcal{L}^2(\Omega)} + \left\| \frac{df}{dx} \right\|^2_{\mathcal{L}^2(\Omega)} \right)^{\frac{1}{2}}$$

$f'(x) = \left(\dfrac{df}{dx} \right)$ For the purposes of this dissertation, all true solutions we consider are assumed to be continuously differentiable as many times as needed.

1 Abstract

The human race is no stranger to the danger and impact of prevalent infectious diseases. Epidemiology, the study of diseases, is concerned with the spread of diseases and what will happen, in an attempt to deploy countermeasures in aid of mitigating further spread. Epidemiologists often model infectious diseases using 'compartmental models', where they compartment the population into smaller subsections, due to their simplicity and wide range of applications. The simplest and most well known of these compartmental models is the 'Susceptible - Infectious - Removed' equations, developed by Kermack and Mckendrick, perhaps better known as the 'SIR' equations. In order to solve the SIR equations, one must know the initial conditions for each of the compartments, in which any set of (positive) conditions can be prescribed. Moreover, any data collected (about infectious cases) showcases changes happening to the infectious compartment and not necessarily about its initial condition to start with, hence a new model needs be derived to interpret these initial conditions, given data. In this dissertation, we explore numerical methods to obtain solutions to the 'Observational model' [1], a second order nonlinear and nonlocal ODE derived by reformulating the SIR model in terms of the detected cases. By using the Observational model, we can re-interpret the data given as changes to the infectious compartment in conjunction with the necessary parameters associated, to solve for an initial condition to the infectious compartment in the SIR model. The motivation behind this is to see if we can find a more efficient time, cost and accuracy driven numerical method(s) than the method shown in [1] and deduce any further results about either method. In the first approach outlined in [1], we use an IVP approach to solving the Observational model, using point value guesses at $t = 0$. In the second approach, using a nonlinear finite elements scheme, we use an exponential type curve formed in conjunction with the data as an initial guess, and hence take different approaches to see which method produces the best results.

1.1 Motivation for a new model

The SIR equations are a set of three simultaneous first order ODEs given as follows;

$$\frac{dS}{dt} = -\beta \frac{I}{N} S, \quad S(0) = S_0, \tag{1.1.1}$$

$$\frac{dI}{dt} = \beta \frac{I}{N} S - \gamma I, \quad I(0) = I_0, \tag{1.1.2}$$

$$\frac{dR}{dt} = \gamma I, \quad R(0) = R_0. \tag{1.1.3}$$

Where β denotes the average transmission rate (i.e., average number of contacts multiplied by probability of transmission from Infectious $\rightarrow$ Susceptible), and γ denotes the average removal rate (i.e. how quickly you stop infecting others on average, e.g., by quarantining, recovering). N denotes the total population being considered (i.e a country or county). We can deduce from the SIR equations that

$$\frac{dS}{dt} + \frac{dI}{dt} + \frac{dR}{dt} = 0 \implies S(t) + I(t) + R(t) = C, \quad C \in \mathbb{R}.$$

This implies that the population is constant, and to find C, we look at the initial conditions at the start of the pandemic. Note that we typically set

$$S_0 + I_0 + R_0 = N \implies C = N.$$

In which we note that $I_0 \geqslant 1$, otherwise nobody is infected and this model does not make sense, and that the other initial conditions are non-negative. The important question to ask is, do we really know the initial conditions, or even how to find them at the start of an epidemic? What information can we gather to help us decide on the initial conditions or even the parameters β and γ?

In the approach taken by the SIR equations, one can formulate multiple simulations and parameters to try and model an epidemic using the SIR approach, however - in an ever changing environment, this does not always give good or even legible results. The need for a new model that can take information as it comes would lead to a more confident projection of the current spread. For example, the UK government provides statistical figures on a daily basis on the number of people currently infected with Coronavirus in the UK, along with some other hospital figures. Is there a way we can formulate a new model using this data we are given?

1.2 Derivation of the Observational model

By taking equations (1.1.1,1.1.2) and adding them together, we see that

$$\frac{dS}{dt} + \frac{dI}{dt} = -\gamma I, \tag{1.2.1}$$

and by taking the derivative with respect to t on both sides yields,

$$\frac{d^2S}{dt^2} + \frac{d^2I}{dt^2} = -\gamma \frac{dI}{dt}. \tag{1.2.2}$$

Furthermore, by differentiating 1.1.1, we obtain

$$\frac{d^2S}{dt^2} = -\frac{\beta}{N}\left(S\frac{dI}{dt} + I\frac{dS}{dt}\right). \tag{1.2.3}$$

By re-arranging equation 1.1.1, so that

$$S = -\frac{dS}{dt}\frac{N}{I\beta}, \tag{1.2.4}$$

and substituting equation 1.2.4 into 1.2.3 gives

$$\frac{d^2S}{dt^2} = -\frac{\beta}{N}\left(-\frac{dS}{dt}\frac{N}{I\beta}\frac{dI}{dt} + I\frac{dS}{dt}\right) = \frac{dS}{dt}\left(\frac{1}{I}\frac{dI}{dt} - \frac{I\beta}{N}\right). \tag{1.2.5}$$

Then by re-arranging equations 1.2.1 and 1.2.2, and inserting them into equation 1.2.5 gives

$$-\gamma\frac{dI}{dt} - \frac{d^2I}{dt^2} = \left(-\gamma I - \frac{dI}{dt}\right)\left(\frac{1}{I}\frac{dI}{dt} - \frac{\beta I}{N}\right),$$

which implies

$$\frac{d^2I}{dt^2} = \left(\gamma I + \frac{dI}{dt}\right)\left(\frac{1}{I}\frac{dI}{dt} - \frac{\beta I}{N}\right) - \gamma\frac{dI}{dt}.$$

Expanding what we have and rearrange we see

$$\frac{d^2I}{dt^2} = -\frac{\gamma\beta I^2}{N} + \frac{1}{I}\left(\frac{dI}{dt}\right)^2 - \frac{dI}{dt}\frac{\beta I}{N} - \cancelto{0}{\left(\gamma\frac{dI}{dt} - \frac{\gamma I}{I}\frac{dI}{dt}\right)},$$

which gives rise to the observational model

$$\frac{d^2I}{dt^2} = \frac{dI}{dt}\left(\frac{1}{I}\frac{dI}{dt} - \frac{\beta I}{N}\right) - \frac{\beta\gamma I^2}{N}, \tag{1.2.6}$$

a second order, nonlinear differential equation.

1.3 Boundary conditions

Normally with any second order differential equation, IVP conditions are prescribed with an initial condition for the function and its derivative, or in the case of a BVP, we prescribe some combination of either Dirichlet, Neumann or Robin conditions on the boundaries. As mentioned before, we are looking to incorporate statistical figures into the model in order to capture a more accurate picture of what is going on. Let X_m denote the number of observations of detected cases, then we formally define

$$X_m := r\gamma\int_{t_m}^{t_{m+1}} I(s)\,ds, \tag{1.3.1}$$

where $r \in (0,1)$ is an under-reporting parameter and (t_m, t_{m+1}) describes the time interval of the given data, for example - this could be days or weeks.

Intuitively, we assume that the given X_m only captures part of the picture of how many people are

infected, since for example, we know not everybody tests when they have symptoms, reports they have tested positive, or possibly because they are asymptomatic - hence the need for an under-reporting parameter. In this dissertation, we do not discuss how one would go about estimating r, but only give examples with fixed values.

We note that these conditions are not exactly our typical Dirichlet boundary conditions, they are in fact nonlocal boundary conditions, since they aren't defined for a single point in time, but rather for an interval of time.

1.4 The well-posedness of the Observational model

Before one can look for solutions, it is sensible to ask given X_0 and X_1, does a solution exist? Furthermore, is it unique? The complexity of the nonlinearity, coupled with the nonlocal boundary conditions makes this question extremely difficult to answer. Given $X_0, X_1 > 0$ with sensible parameters, we have existence and uniqueness of a solution $\forall t > 0$. For the interested reader, it is very much recommended for a deeper insight into the analysis of the observational model and well-posedness that you read [1].

To make the problem slightly easier to digest and implement in one of our numerical methods, we make the following substitution. Let $\beta_\epsilon := \beta r^{-1} N^{-1}$ and let $z(t) := \ln(r\gamma I(t))$, the Observational model (1.2.6) is equivalent to

$$\frac{d^2 z}{dt^2} = -\beta_\epsilon e^z \left(\frac{1}{\gamma} \frac{dz}{dt} + 1 \right), \tag{1.4.1}$$

with the nonlocal boundary conditions equivalent to

$$X_m = \int_{t_m}^{t_{m+1}} e^{z(s)}\, ds. \tag{1.4.2}$$

Proof. With regards to the nonlocal boundary conditions (1.4.2), it is clear to see

$$X_m = \int_{t_m}^{t_{m+1}} e^{z(s)}\, ds = \int_{t_m}^{t_{m+1}} e^{\ln(r\gamma I(s))}\, ds = r\gamma \int_{t_m}^{t_{m+1}} I(s)\, ds.$$

In order to prove the change for the Observational model (1.2.6), we first make the following observations.

$$z(t) = \ln(r\gamma I(t)) \iff I(t) = \frac{e^{z(t)}}{r\gamma}, \tag{1.4.3}$$

$$\therefore \frac{d}{dt}\left(I(t)\right) = \frac{d}{dt}\left(\frac{e^{z(t)}}{r\gamma}\right) = \frac{e^{z(t)}}{r\gamma}\frac{dz}{dt}, \tag{1.4.4}$$

$$\implies \frac{d^2}{dt^2}\left(I(t)\right) = \frac{d}{dt}\left(\frac{e^{z(t)}}{r\gamma}\frac{dz}{dt}\right) = \frac{e^{z(t)}}{r\gamma}\left(\left(\frac{dz}{dt}\right)^2 + \left(\frac{d^2 z}{dt^2}\right)\right). \tag{1.4.5}$$

Substituting equations 1.4.3,1.4.4 and 1.4.5 into equation 1.2.6;

$$\frac{e^{z(t)}}{r\gamma}\left(\left(\frac{dz}{dt}\right)^2 + \left(\frac{d^2 z}{dt^2}\right)\right) = \frac{e^{z(t)}}{r\gamma}\frac{dz}{dt}\left(\frac{e^{z(t)}}{r\gamma}\frac{dz}{dt}\frac{r\gamma}{e^{z(t)}} - \frac{\beta}{N}\frac{e^{z(t)}}{r\gamma}\right) - \frac{\beta\gamma}{N}\frac{e^{2z(t)}}{r^2\gamma^2}.$$

Since $e^{z(t)} \neq 0$, dividing by $e^{z(t)}$, multiplying by $r\gamma$ and simplifying like terms means we can re write this as

$$\left(\frac{dz}{dt}\right)^2 + \left(\frac{d^2 z}{dt^2}\right) = \left(\frac{dz}{dt}\right)^2 - \frac{\beta}{r\gamma N}e^{z(t)}\frac{dz}{dt} - \frac{\beta}{rN}e^{z(t)},$$

$$\implies \frac{d^2 z}{dt^2} = -\frac{\beta}{rN}e^{z(t)}\left(\frac{dz}{dt}\frac{1}{\gamma} + 1\right).$$

Finally, substituting $\beta_\epsilon = \beta r^{-1} N^{-1}$ gives the result.

□

1.5 Abstract of numerical methods

In order to solve the Observational model numerically, we first require some preliminaries. In the first method, we use the shooting method and pose the problem as a root finding problem. The second method involves converting 1.4.1 or 1.2.6 into a variational problem in which we make use of Lagrange multipliers in order to reformulate a new constrained equation, where we apply the Euler Lagrange equation and produce a finite element formulation of what is left. We now pose the problem we wish to solve; given X_0, X_1, r, N, γ and β ($\implies \beta_\epsilon$), find $z(t)$ such that

$$\frac{d^2z}{dt^2} = -\beta_\epsilon e^z \left(\frac{1}{\gamma}\frac{dz}{dt} + 1\right), \tag{1.5.1}$$

$$X_0 = \int_0^1 e^{z(s)}\, ds \quad | \quad X_1 = \int_1^2 e^{z(s)}\, ds. \tag{1.5.2}$$

Or equivalently, find $I(t)$ such that

$$\frac{d^2I}{dt^2} = \frac{dI}{dt}\left(\frac{1}{I}\frac{dI}{dt} - \frac{\beta I}{N}\right) - \frac{\beta\gamma I^2}{N}, \tag{1.5.3}$$

$$X_0 = r\gamma \int_0^1 I(s)\, ds \quad | \quad X_1 = r\gamma \int_1^2 I(s)\, ds. \tag{1.5.4}$$

2 The Shooting Method

Numerically solving a second order ODE is not that difficult given some initial conditions (IVP). It is well documented in standard literature [2, 3] of the multiple methods available for us to use. However, when we look to solve boundary value problems, the standard approach changes slightly. As we will shortly see, the shooting method is an excellent choice for solving BVP ODE's, due to its high flex-ability in terms of implementation. Of-course, when we consider moving away from linear problems and into the realm of nonlinear problems, the need for root finders quickly becomes apparent. Therefore, whilst solving anything more than a second order linear ODE, the shooting method becomes just one of the building blocks used to solve these problems. In the approach for solving the Observational model with the shooting method, we take the approach of using a root finder, but not in the way described above. Going forward, when talking about using an IVP solver, we explicitly refer to using an RK4 solver (Runge Kutta - fourth order) [3]. For ease of exposition, we assume $g \in \mathcal{C}(\Omega) \implies f \in \mathcal{C}^2(\Omega)$. For information about numerical solution of ODEs, see the reference in the above line.

2.1 How does the shooting method work?

The shooting method takes the initial value approach, and modifies it in order to solve the BVP. In the case of pure Dirichlet boundary conditions, as the name implies, we shoot from the first boundary condition using an IVP approach, and depending on our initial guess for the derivative at the first boundary value point, see if we hit the second boundary value for the other given boundary value from the domain we are shooting from. From here, we take two approaches depending on whether or not the problem is second order linear or second order non linear. In the second order case, the two point BVP's take the form

$$\frac{d^2f}{d^2x} = g(x, f, f'), \quad x \in [a, b], \quad a, b \in \mathbb{R}, \tag{2.1.1}$$

$$f(a) = \phi \quad | \quad f(b) = \zeta, \quad \phi, \zeta \in \mathbb{R}. \tag{2.1.2}$$

The shooting method problem requires us to find ψ, given ϕ and ζ such that

$$\frac{d^2f}{d^2x} = g(x, f, f'), \quad x \in [a, b], \tag{2.1.3}$$

$$f(a) = \phi \quad | \quad \frac{df}{dx}(a) = \psi, \quad \text{solves 2.1.2}, \quad \psi \in \mathbb{R}. \tag{2.1.4}$$

For the interested reader, a full explanation of the second order linear approach can be found from [3](page 653), but simply involves taking two guesses for ψ and linearly interpolating, in order to find ψ such that $f(b) = \zeta$. Note the linearly interpolating is only possible in the linear case. We now give a very brief example.

Example 2.1 (Linear shooting method with non-homogeneous Dirichlet boundary conditions)**.** *We look to apply the linear shooting method to the following BVP;*

$$-\frac{d^2u}{dx^2} = e^x, \tag{2.1.5}$$

$$u(0) = -1 \quad | \quad u(1) = -e. \tag{2.1.6}$$

Then the shooting method proposes we find ψ where

$$-\frac{d^2u}{dx^2} = e^x, \tag{2.1.7}$$

$$u(0) = -1 \quad | \quad \frac{du}{dx}(0) = \psi, \tag{2.1.8}$$

such that $u(1) = -e$. One can check that indeed this is an easy example to solve, i.e. $u(x) = -e^x$, since this satisfies the ODE in 2.1.5 and the boundary conditions 2.1.6. Then by using the Runge-Kutta 4'th order IVP solver, and two guesses for ψ, we can solve this problem numerically. Let us propose two guesses;

$$\psi_1 = 0 \quad | \quad \psi_2 = 5.$$

Then after interpolating the correct gradient, we have the following graph of results.

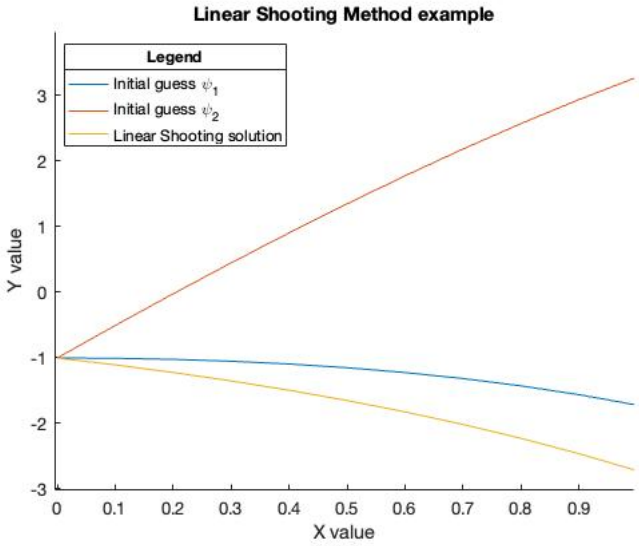


Figure 1: Linear interpolation of the shooting Method, Solution is given by $u(x) = -e^x$, initial guesses $\psi_1 = 0, \psi_2 = 5$, $\Delta x = 0.1$.

In order to interpolate the correct gradient, we have used the following calculation. Let ψ_3 be the correct gradient that ensures we solve 2.1.6, then

$$\begin{aligned} \frac{\psi_3 - \psi_1}{u(1)_{\psi_3} - u(1)_{\psi_1}} &= \frac{\psi_2 - \psi_1}{u(1)_{\psi_2} - u(1)_{\psi_1}}, \\ \implies \psi_3 &= \frac{(\psi_2 - \psi_1)(u(1)_{\psi_3} - u(1)_{\psi_1})}{(u(1)_{\psi_2} - u(1)_{\psi_1})} + \psi_1, \end{aligned} \tag{2.1.9}$$

where $u(1)_{\psi_i}$ denotes the value at $u(1)$ starting from the gradient ψ_i. Intuitively, one can think about this calculation as finding the value of the derivative we are looking for in which $u(1)_{\psi_3}$ intersects the line which connects $u(1)_{\psi_1}$ and $u(1)_{\psi_2}$.

By taking $\psi_2 = 5, \psi_1 = 0$, we can calculate $u(1)_{\psi_1} = -1.7183$ and $u(1)_{\psi_2} = 3.2817$ by using the Runge-Kutta solver. Since we are given $u(1)_{\psi_3} = -e$ from the boundary value of the problem, then

$$\psi_3 = \frac{(\psi_2 - \psi_1)(u(1)_{\psi_3} - u(1)_{\psi_1})}{(u(1)_{\psi_2} - u(1)_{\psi_1})} + \psi_1 = \frac{(5-0)(-e+1.7183)}{3.2817 + 1.7183} + 0 = -1.$$

From the solution, $u(x) = -e^x \implies \dfrac{du}{dx}(0) = -1$, and hence the result found. More information about this is found in the above reference.

The cost of solving a linear second order ODE is very low, since it only takes two guesses and some linear interpolating to find ψ to get the desired solution. Depending on the IVP approach one takes (i.e which numerical solver one uses) can dictate whether or not you look for accuracy, speed or both. This becomes important when we move to the Observational model, as here we require both speed, accuracy since multiple iterations of a root finding algorithm will be needed, of which we will see shortly.

2.2 Adapting the shooting method to the Observational model

As discussed in the above section, depending on whether or not the ODE is linear or non linear decides the approach one takes. Moreover, we also no longer have any type of standard boundary conditions, but instead nonlocal integral boundary conditions (1.5.2 or 1.5.4). In this approach, we seek to find the initial value conditions such that 1.5.2 or 1.5.4 are satisfied. We pose this problem as the following; given β, r and γ, find ϕ and ψ such that

$$z(0) = \phi \quad | \quad \frac{dz}{dt}(0) = \psi, \tag{2.2.1}$$

solves 1.5.1 with constraints 1.5.2.

In our previous example 2.1.1, we only needed to find the value of the derivative at the first boundary value point in order to solve the problem. Here, we need to find both the derivative and its value on the first boundary point such that when we integrate the values obtained, we satisfy the integral boundary conditions. Thanks to [1], we know that there exists a unique solution exists for the Observational model, given sensible parameters. Before moving any further, we must make some comments on quadrature (numerical integration) and multi-dimensional root finding. An excellent place for information regarding root finding is found in [3]. The need for quadrature becomes apparent when we need to calculate integrals, given we only have point values. Moreover, due to its easy implementation, we will be using the trapezium method going forward. If unfamiliar with quadrature, please visit the supplemental section at the end of the dissertation before moving on, or also visit [3]. In order to find ϕ and ψ, it is equivalent to finding the roots of the following functions

$$f^0 = \int_0^1 e^{z(s)}\, ds - X_0 = 0 \quad | \quad f^1 = \int_1^2 e^{z(s)}\, ds - X_1 = 0. \tag{2.2.2}$$

We note going forward, the subscript denotes a vector index, and further down, the superscript denotes iterates. When we look to find roots of equations, we normally look to a newtons method (since for close enough guesses, we have quadratic convergence). Newtons iteration scheme for the following problem is of the following form; given ϕ_i and ψ_i and after calculating f^0, f^1,

$$\begin{bmatrix} \phi_{i+1} \\ \psi_{i+1} \end{bmatrix} = \begin{bmatrix} \phi_i \\ \psi_i \end{bmatrix} - \begin{bmatrix} \dfrac{\partial f_i^0}{\partial \phi} & \dfrac{\partial f_i^0}{\partial \psi} \\ \dfrac{\partial f_i^1}{\partial \phi} & \dfrac{\partial f_i^1}{\partial \psi} \end{bmatrix}^{-1} \begin{bmatrix} f_i^0 \\ f_i^1 \end{bmatrix}, \tag{2.2.3}$$

with f^0 and f^1 calculated with the RK4 (IVP) solver and then applying the trapezium rule. Moreover, one notices that the jacobian matrix wants to take partial derivatives with respect to the initial condition and initial derivative which, is not *easily* analytically obtainable. Therefore we need to compute the jacobian numerically.

A sensible estimate of the jacobian was invented by Broyden [2], in which two initial guesses are given for $\{\phi_0,\ \psi_0\}$ and $\{\phi_1,\ \psi_1\}$, and an initial jacobian matrix is estimated and updated after consecutive iterations. The initial jacobian matrix is calculated using the finite difference approximation given by the two (sensible) initial guesses, i.e.,

$$\frac{\partial f_i^0}{\partial \phi} \approx \frac{f_{i+1}^0 - f_i^0}{\phi_{i+1} - \phi_i}. \tag{2.2.4}$$

The rest are analogous, for $i \geqslant 1$, let

$$J_i := \begin{bmatrix} \dfrac{\partial f_i^0}{\partial \phi} & \dfrac{\partial f_i^0}{\partial \psi} \\ \dfrac{\partial f_i^1}{\partial \phi} & \dfrac{\partial f_i^1}{\partial \psi} \end{bmatrix} \quad | \quad \Delta F_i := \begin{bmatrix} f_{i+1}^0 - f_i^0 \\ f_{i+1}^1 - f_i^1 \end{bmatrix} \quad | \quad \Delta X_i := \begin{bmatrix} \phi_{i+1} - \phi_i \\ \psi_{i+1} - \psi_i \end{bmatrix}. \tag{2.2.5}$$

Broyden showed that for jacobian's with non-trivial analytical derivatives that a more cost effective guess, based off the one dimensional secant root finding method, is given by

$$J_{i+1} = J_i + \frac{(\Delta F_i - J_i \Delta X_i)\,(\Delta X_i)^T}{\|\Delta X_i\|^2}, \tag{2.2.6}$$

where $()^T$ denotes the transpose of a vector and $\|\cdot\|$ denotes the norm. Then inserting the new approximation for the jacobian into newtons scheme (2.2.3) gives the following modified scheme;

$$\begin{bmatrix} \phi_{i+1} \\ \psi_{i+1} \end{bmatrix} = \begin{bmatrix} \phi_i \\ \psi_i \end{bmatrix} - \left[J_i + \frac{(\Delta F_i - J_i \Delta X_i)\,(\Delta X_i)^T}{\|\Delta X_i\|^2} \right]^{-1} \begin{bmatrix} f_i^0 \\ f_i^1 \end{bmatrix}. \tag{2.2.7}$$

Implementing this into an algorithm would mean we would need to set a stopping criteria, either for convergence (i.e we've found the roots), or we diverge perhaps due to not giving good initial guesses. As a consequence, if the guesses are too far away, this may also make the jacobian approximation ill-conditioned and hence the algorithm will fail, this will be touched on more later. The algorithm of the Newton-Broyden method for solving the Observational method can be found in subsection 5.1.

2.3 Shooting method results

We now present some results about the effectiveness of the shooting method. In order to find out if the shooting method converges to the solution, we need to know what the actual solution is. For the SIR equations, there is no closed sensible form worth noting here, only that we can calculate the numerical solution when we provide initial conditions. Simply using the RK4 solver is enough to find the numerical solution to the SIR equations, we omit the numerical calculating of the SIR equations here. In order to find out the effectiveness, we proceed in the following steps

1. Specify initial conditions S_0, I_0, R_0 and paramaters β, γ, r, N.
2. Solve the SIR equations, specifically we are interested in the infected solutions on the interval $[0, 2]$.
3. Artificially calculate the data points X_0 and X_1 by using the trapezium method (1.5.4).
4. Insert these data points into the the Newtons-Broyden's algorithm, along with given parameters.
5. Insert two initial guesses for ϕ and ψ.
6. Run the algorithm, find the errors and then calculate $\mathcal{L}^2$ error norm and $\mathcal{L}^\infty$ error norm.

We now take a look at two cases. An example of raising cases, and an example of decreasing cases (just for some variety). In all examples going forward, we will be using the shooting method for the Observational model (1.5.1), and by using the substitution $z(t) = \ln(r\gamma I(t))$, getting the results desired.

Example 2.2 (Shooting method example 1 - increasing cases)**.** *Let us specify the following initial parameters of this example. Let* $N = 1000$ *(i.e, like a small village),* $\beta = 1.5, \gamma = 1, r = 0.75, \Delta t = 0.0025$*. The initial conditions are given as* $S(0) = 980, I(0) = 20, R(0) = 0$*. Since* $\beta > \gamma$*, i.e. since on average more people are being infected then being removed from the infectious compartment, these parameters indicate we will have rising cases. By calculating the SIR solution, specifically the infectious cases, we can calculate* X_0 *and* X_1*. These are calculated as* $X_0 = 18.9739, X_1 = 28.6179$*. We note that in general, we would expect our data points to be exact, and hence some ambiguity to our pre-fabricated data points is needed. Furthermore, we will set the tolerance as highlighted in the algorithm as* 10^{-6}*, and max-iterations as* 100*.*

We now need to go about deducing what two sensible initial guesses are, given we only have access to the parameters and the data points X_0, X_1*. In this case, the difference in size between the data points is not too dissimilar, so here we shouldn't have too much of a problem giving some good educated guesses about where we are starting, but for the most part this is just guesswork, especially as the size difference between the data points increases. Let us take two initial guesses,*

$$I^0(0) = 22, \left(\frac{dI}{dt}\right)^0(0) = 9, \qquad I^1(0) = 18, \left(\frac{dI}{dt}\right)^1(0) = 8, \tag{2.3.1}$$

$$\implies z^0(0) = \ln(0.75 \times 22), \left(\frac{dz}{dt}\right)^0(0) = \frac{9}{22}, \qquad z^1(0) = \ln(0.75 \times 18), \left(\frac{dz}{dt}\right)^1(0) = \frac{8}{18}.$$

Then by inserting these parameters into the Newton-Broyden algorithm gives the following results. Below we show the infectious cases solution, our Observational model solution along with the results of the two initial guesses. We also note the following results related to this specific example. (See figure 2).

$$\mathcal{L}^\infty \text{ error} = 1.0806 \times 10^{-6} \quad | \quad \mathcal{L}^2 \text{ error} = 1.3993 \times 10^{-6}.$$
$$\text{Time till completion: } 0.144899 \text{ seconds.}$$
$$\text{Iterations taken: } 14.$$

We can see here that we get some great convergence, given the error norms. From the SIR (infectious) equation, we can calculate $\frac{dI}{dt} = 9.4$ *by inserting* $I(0) = 20$ *and* $S(0) = 980$ *into 1.1.2.*

So if we take a slightly closer guess to what the true solution is, we look to see how the error norms change, and whether or not the computational time and iterations needed decreases. Let us now take;

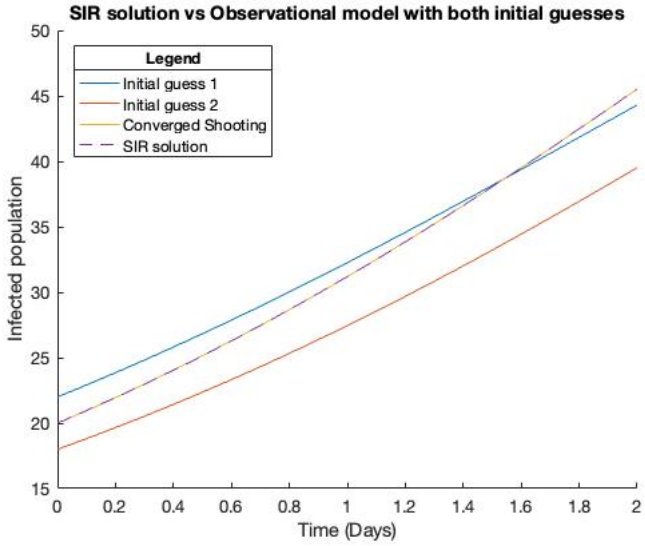


Figure 2: I(0)= 20, $\Delta t = 0.0025$, Initial guesses: 2.3.1.

$$I^0(0) = 21, \left(\frac{dI}{dt}\right)^0(0) = 9, \qquad I^1(0) = 19, \left(\frac{dI}{dt}\right)^1(0) = 8.5, \tag{2.3.2}$$
$$\implies z^0(0) = \ln(0.75 \times 21), \left(\frac{dz}{dt}\right)^0(0) = \frac{9}{21}, \qquad z^1(0) = \ln(0.75 \times 19), \left(\frac{dz}{dt}\right)^1(0) = \frac{8.5}{19}.$$

We can then plot the results and also note the following results related to these specific initial conditions. See figure 3.

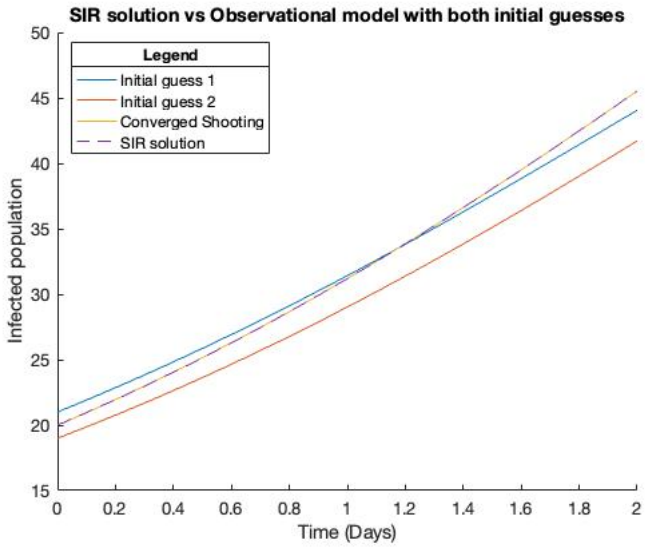


Figure 3: I(0)= 20, $\Delta t = 0.0025$, Initial guesses: 2.3.2.

$$\mathcal{L}^\infty \text{ error} = 2.8041 \times 10^{-7} \quad | \quad \mathcal{L}^2 \text{ error} = 3.7123 \times 10^{-7}.$$
Time till completion: 0.178309 *seconds.*
Iterations taken: 13.

From the results, we can deduce that as we approach the solution, the error norms get smaller (whilst keeping the same tolerance). We needed one less iteration and roughly the same computational time for MATLAB to run the code.

We now look at a case where $\beta < \gamma$ in which prompts us to see we will have decreasing cases.

Example 2.3 (Shooting method example 2 - decreasing cases)**.** *Let us specify the following initial parameters of this example. Let* $N = 1000$, $\beta = 0.6, \gamma = 1, r = 0.75, \Delta t = 0.0025$. *The initial conditions are given as* $S(0) = 816, I(0) = 184, R(0) = 0$. *By calculating the SIR solution, specifically the infectious cases, we can calculate* X_0 *and* X_1. *These are calculated as* $X_0 = 107.3485, X_1 = 62.0702$. *Furthermore, we will set the tolerance as highlighted in the algorithm as* 10^{-6}, *and max-iterations as* 100.

We can see here that there is a much larger difference in data points X_0 *and* X_1. *Deducing that the initial condition is within the vicinity of* 184 *is not exactly trivial, which here is the major let down of this method, though the sensitivity of the initial guesses will be briefly discussed later. Let us take two initial guesses*

$$I^0(0) = 190, \left(\frac{dI}{dt}\right)^0(0) = -80, \qquad I^1(0) = 170, \left(\frac{dI}{dt}\right)^1(0) = -90, \tag{2.3.3}$$
$$\implies z^0(0) = \ln(0.75 \times 190), \left(\frac{dz}{dt}\right)^0(0) = -\frac{80}{190}, \qquad z^1(0) = \ln(0.75 \times 170), \left(\frac{dz}{dt}\right)^1(0) = -\frac{90}{170}.$$

These yield the following errors and other relevant information. See figure 4

$$\mathcal{L}^\infty \text{ error} = 7.0021 \times 10^{-9} \quad | \quad \mathcal{L}^2 \text{ error} = 4.6221 \times 10^{-9}.$$
Time till completion: 0.169251 *seconds.*
Iterations taken: 13.

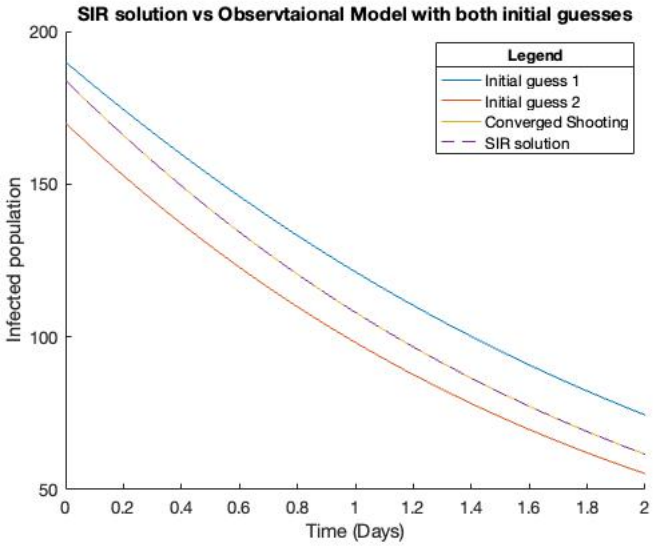


Figure 4: I(0)= 184, $\Delta t = 0.0025$, Initial guesses: 2.3.3.

We now look to see how far we can push away from the solution (within reason) as to when we converge or diverge, which will be the next topic we will briefly talk about. Let us take new initial guesses, given by

$$I^0(0) = 240, \left(\frac{dI}{dt}\right)^0(0) = -150, \qquad I^1(0) = 130, \left(\frac{dI}{dt}\right)^1(0) = -20, \tag{2.3.4}$$

$$\implies z^0(0) = \ln(0.75 \times 240), \left(\frac{dz}{dt}\right)^0(0) = -\frac{150}{240}, \qquad z^1(0) = \ln(0.75 \times 130), \left(\frac{dz}{dt}\right)^1(0) = -\frac{20}{130}.$$

With results

$$\mathcal{L}^\infty \ error = 6.6254 \times 10^{-8} \quad | \quad \mathcal{L}^2 \ error = 5.7602 \times 10^{-8}.$$

Time till completion: 0.169752 *seconds.*

Iterations taken: 10.

With the plot of results given by figure 5

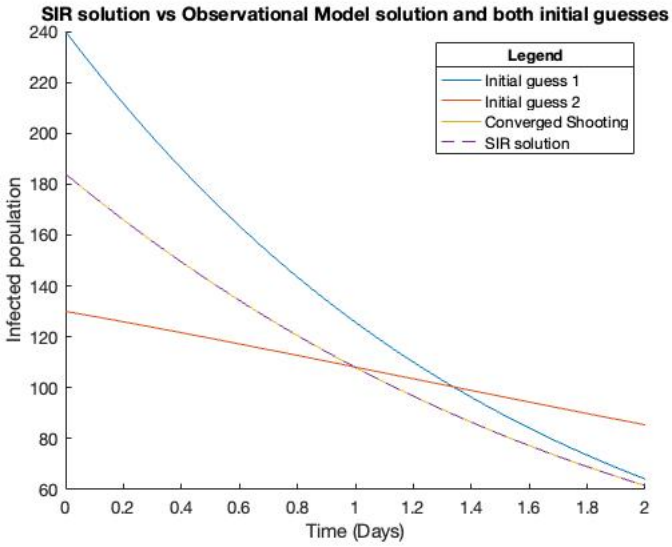


Figure 5: I(0)= 184, $\Delta t = 0.0025$, Initial guesses: 2.3.4.

2.4 Sensitivity of the Newton-Broyden algorithm

One issue not yet touched on, is how sensitive the Algorithm is to initial guesses. On the last example (2.3.4), we can see that our initial guess 2 is not a good representation of the solution, but yet, we still have convergence in this example. A sensible question to ask is, do we always converge? Much like Newtons method outside of the Observational model, the answer is not always. If, in the 1D case, we are to far away from the root such that the convergence criteria are not met, or the root is a stationary point, then we will more then likely diverge (or not converge) [3]. In the multidimensional case, if we are again too far away from the root, or our jacobian becomes singular (or is ill-conditioned) [2], we will again not converge. We now demonstrate an example.

Example 2.4 (Divergence, increasing cases)**.** *We again look at the examples given in 2.3.1 and 2.3.2, using the same parameters but slightly different initial guesses. Let us now take;*

$$I^0(0) = 25, \left(\frac{dI}{dt}\right)^0(0) = 10, \qquad I^1(0) = 16, \left(\frac{dI}{dt}\right)^1(0) = 8, \tag{2.4.1}$$
$$z^0(0) = \ln(0.75 \times 25), \left(\frac{dz}{dt}\right)^0(0) = \frac{10}{25}, \qquad z^1(0) = \ln(0.75 \times 16), \left(\frac{dz}{dt}\right)^1(0) = \frac{8}{16}.$$

Of which given the parameters from example 2.2, this initially does not seem like such a bad set of initial guesses. By following the steps of the Newton-Broyden algorithm, we can see where things start to go wrong. We again note that the below will be in terms of $z(t)$, and will distinguish between $z(t)$ and $I(t)$. On calculating the initial jacobian, our first Newtons iteration becomes

$$\begin{bmatrix}\phi_3\\ \psi_3\end{bmatrix} = \begin{bmatrix}2.4849\\ 0.5000\end{bmatrix} - \begin{bmatrix}-13.6333837618108 & -66.8706025009116\\ -13.6062357417434 & -69.2837350091016\end{bmatrix}^{-1} \begin{bmatrix}3.23669618409084\\ -4.37891690813301\end{bmatrix} \tag{2.4.2}$$
$$\implies \begin{bmatrix}\phi_3\\ \psi_3\end{bmatrix} = \begin{bmatrix}17.3798418913776\\ -2.48833365631926\end{bmatrix} \tag{2.4.3}$$

Now here we are given our new guess for the iterations as $z(0) \approx 17.38$ and $\frac{dz}{dt}(0) \approx -2.49$. By using

$$z(t) = \ln(r\gamma I(t)) \implies I(t) = \frac{1}{r\gamma}\exp(z(t)) \implies I(0) = \frac{1}{0.75}\exp(17.38) >> N \text{ (population).}$$

And hence, our solution has blown up, in-fact way bigger than even sensible for the paramaters of the model. This is more then likely due to the poor guesses illustrated with Broyden's jacobian calculation. Since Broyden's method relies on a good initial jacobian and updates it on each iteration, with Newtons method relying heavily on a good initial guess with an accurate jacobian, then we shouldn't be shocked we do not see convergence here. Further work on criteria for intervals of convergence given the data points X_0, X_1 is needed.

2.5 Estimating the order of convergence for the Newton-Broyden algorithm

We first recall the definition for the order of convergence for a fixed point scheme (in 1D) [5].

Definition 2.1 (Order of Convergence - Fixed Point Iteration)**.** Let $\{y_1, y_2, \ldots, y_n\}$ be a sequence that converges to some value $\hat{y}$. Then, if there exists some values $\alpha \geqslant 1$ and $\beta > 0$ such that

$$\lim_{n\to\infty} \frac{|y_{n+1} - \hat{y}|}{|y_n - \hat{y}|^\alpha} = \beta, \tag{2.5.1}$$

then we call α the order of convergence of the sequence $\{y_1, \ldots, y_n\}$. If we define $e_{n+1} = y_{n+1} - \hat{y}$, then the above is equivalent to

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \beta, \tag{2.5.2}$$

then we can notice by taking the limit as $n \to \infty$ that

$$|e_{n+1}| \approx \beta|e_n|^\alpha \quad | \quad |e_n| \approx \beta|e_{n-1}|^\alpha, \tag{2.5.3}$$
$$\implies \frac{|e_{n+1}|}{|e_n|} \approx \frac{\beta|e_n|^\alpha}{\beta|e_{n-1}|^\alpha} = \left(\frac{|e_n|}{|e_{n-1}|}\right)^\alpha \implies \alpha \approx \frac{\ln(e_{n+1}/e_n)}{\ln(e_n/e_{n-1})}. \tag{2.5.4}$$

In the standard literature, if $\alpha = 1$, this is called linear convergence, $\alpha = 2$ is called quadratic convergence and for $1 < \alpha < 2$, this is called super linear convergence.

[3] It is well documented in the standard literature that for the Newtons root finding algorithm, if we are close enough to the solution, we have quadratic convergence ($\alpha = 2$). [6] For the Secant method, it has been shown that we converge super-linearly, with the exact convergence being the golden ratio ($\alpha = \frac{1+\sqrt{5}}{2} \approx 1.618$). Considering that our Newton-Broyden's method is a combination of the multi-dimensional Secant method and Newton's method, it is sensible to ask what order of convergence do we expect? Since in any dimension, for a close enough guess, we would expect quadratic convergence for Newtons method, but it is unclear how this is affected by Broyden's estimation of the jacobian in higher dimensions. Moreover with the adaption of solving the Observational model, how does one measure the 'error'?

As previously mentioned, the error we measure in the Newtons-Broyden's algorithm is given by

$$norm(f_0^i, f_1^i) = \sqrt{\left(f_0^i\right)^2 + \left(f_1^i\right)^2},$$

and as $norm(f_0^i, f_1^i) \to 0$, we converge to the solution (provided we are converging). Whats important here is that in the 1D case, we don't always know what the limit is we are approaching. In the Observational model, we know that to converge to the unique solution, we must satisfy the following by definition;

$$\lim_{i\to\infty} f_0^i = 0 \quad | \quad \lim_{i\to\infty} f_1^i = 0.$$

Therefore, it is sensible to take the error norm as

$$e_i = \sqrt{\left(f_0^i\right)^2 + \left(f_1^i\right)^2} - \sqrt{(0)^2 + (0)^2} = \sqrt{\left(f_0^i\right)^2 + \left(f_1^i\right)^2}.$$

Then, by 2.5.4, we can take α as

$$\alpha_i \approx \frac{\ln(e_{i+1}/e_i)}{\ln(e_i/e_{i-1})}. \tag{2.5.5}$$

If we now show the errors for each of the initial conditions shown above (2.3.1,2.3.2,2.3.3 and 2.3.4), by taking the **last 5 errors** from each, we can calculate the following estimated orders of convergence (EOC).

2.3.1 errors	2.3.2 errors	2.3.3 errors	2.3.4 errors
$e_{10} = 0.0374$	$e_9 = 0.0354$	$e_9 = 0.1038$	$e_6 = 0.0486$
$e_{11} = 0.0079$	$e_{10} = 0.0019$	$e_{10} = 0.0359$	$e_7 = 3.4480 \times 10^{-4}$
$e_{12} = 0.0034$	$e_{11} = 3.8262 \times 10^{-4}$	$e_{11} = 8.3852 \times 10^{-4}$	$e_8 = 7.3798 \times 10^{-5}$
$e_{13} = 6.2960 \times 10^{-5}$	$e_{12} = 2.7905 \times 10^{-5}$	$e_{12} = 2.5793 \times 10^{-6}$	$e_9 = 1.9723 \times 10^{-6}$
$e_{14} = 8.0001 \times 10^{-7}$	$e_{13} = 1.8443 \times 10^{-7}$	$e_{13} = 3.3670 \times 10^{-9}$	$e_{10} = 5.4555 \times 10^{-9}$

Table 1: Errors for each of the four examples visited, each with the last 5 iterations up-to, and including, convergence.

2.3.1 EOC's	2.3.2 EOC's	2.3.3 EOC's	2.3.4 EOC's
$\alpha_{11} = 0.542$	$\alpha_{10} = 0.548$	$\alpha_{10} = 3.538$	$\alpha_7 = 0.312$
$\alpha_{12} = 4.731$	$\alpha_{11} = 1.634$	$\alpha_{11} = 1.540$	$\alpha_8 = 2.350$
$\alpha_{13} = 1.094$	$\alpha_{12} = 1.917$	$\alpha_{12} = 1.148$	$\alpha_9 = 1.626$

Table 2: Estimated orders of convergence using 2.5.5

As we can see from table 2, it seems we have at-least linear convergence in all of the cases (if we look at the last α before convergence). Of-course, it would be interesting to see what would happened if we let the tolerance be smaller than 10^{-6}, would that massively impact our EOC, or are there other factors including better/worse initial guesses? More investigation is needed.

As mentioned earlier, the Newton-Broyden method for solving the Observational model by using the shooting method does not come without its problems. In the next section, we look to see if we can improve on reliability issues of guessing we currently face in the shooting method.

3 The Finite Element Method

So far, we have already seen one way of solving the Observational model using a combination of a root finding and the shooting method, and now we look to see if we can improve on our results using a finite elements method. For those familiar with FEM, the problems we face in trying to solve 1.5.1 or 1.5.3 are apparent, but for those not familiar, several good places to read up can be found here [4, 7]. We will also provide an example of applying finite elements to a linear problem with an integral constraint shortly.

The first problem we need to address is how do we convert 1.5.1 or 1.5.3 into something we can apply a finite element scheme to, whilst enforcing the nonlocal boundary conditions? There is more then one way to tackle this problem. In the route that we take, we adopt the use of Lagrange multipliers via calculus of variations. For more information about Lagrange multipliers in the context of optimization, visit [4].

In any second order linear finite element problem, our approach is always the same. We multiply by a test function ($v(x) \in \mathcal{C}(\Omega)$), perform IBP on the term with two derivatives, and move the problem from an infinite space to a finite space by moving the problem to a finite element space. This means we need to discretize our interval from infinitely many points to a carefully selected finite amount. Depending on the problem, either a uniform or non uniform mesh is chosen. For the examples we discuss, a uniform mesh will be used, which means we can make an arbitrary partition

$$0 = x_1 < x_2 < x_3 < \cdots < x_{n-1} < x_n < x_{n+1} = 1, \tag{3.0.1}$$

such that by choosing a (sensible) $n \in \mathbb{N}$,

$$\Delta x = \frac{1-0}{n}, \quad n \in \mathbb{N},$$
$$x_{i+1} = x_i + \Delta x, \quad \forall i \in [1, n].$$

Since we have now discretized our problem, we now set our finite element space where we want to solve this problem as

$$V_\Omega^h = \{v^h(x) \in \mathcal{C}(\Omega) : v^h(x) \in \mathcal{C}^1(x_i, x_{i+1}) \ \forall i = [1:n]\} \subset \mathcal{H}^1(0,1). \tag{3.0.2}$$

Since V^h is a linear space, an extremely useful property about working in this space is that it has a basis. We can carefully choose this basis as the function $\phi_i(x)$ as being equal to

$$\phi_i(x) = \begin{cases} 1 - \dfrac{x_i - x}{\Delta x} & x \in [x_{i-1}, x_i], \\ 1 - \dfrac{x - x_i}{\Delta x} & x \in [x_i, x_{i+1}], \\ 0 & \text{Otherwise}, \end{cases} \quad \Bigg| \quad \frac{d}{dx}(\phi_i(x)) = \begin{cases} \dfrac{1}{\Delta x} & \phi_i(x) \in [x_{i-1}, x_i), \\ -\dfrac{1}{\Delta x} & \phi_i(x) \in (x_i, x_{i+1}], \\ 0 & \text{Otherwise}. \end{cases} \tag{3.0.3}$$

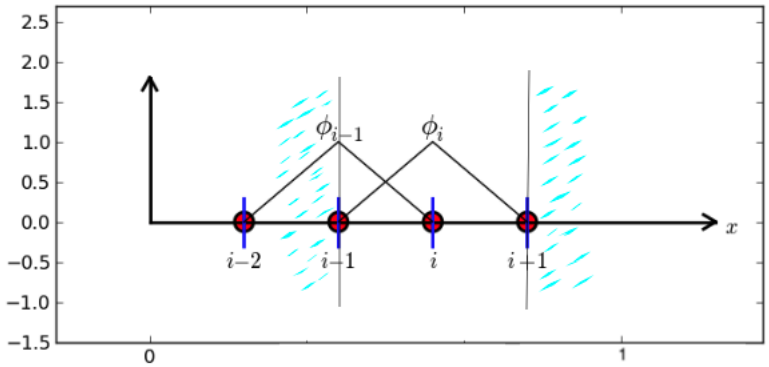


Figure 6: Linear Basis functions (also called "hat functions") as defined in 3.0.3.

Of which admits the following property (by definition of the basis being linearly independent)

$$\phi_i(x_j) = \delta_{ij} = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases} \tag{3.0.4}$$

where δ denotes the Kronecker delta function. For the example in the next section, we will refer back here. We also note that $\mathcal{H}^1(0,1)$ is a Hilbert space, defined as

$$\mathcal{H}^1(0,1) := \left\{ f : (0,1) \to \mathbb{R} \;\middle|\; \|f\|^2_{\mathcal{L}^2(0,1)} + \left\| \frac{df}{dx} \right\|^2_{\mathcal{L}^2(0,1)} < \infty \right\}. \tag{3.0.5}$$

For more information about function spaces and norms, please visit the function spaces and norms section at the beginning, or [7].

3.1 The Euler Lagrange equations

Some motivation for the formulation we will soon derive comes from the calculus of variations. Given a functional (a function that inputs a function and outputs another function), one can use the Euler-Lagrange equations to find the extremum (minimum or maximum) of said functional.

Theorem 3.1 (1D Euler - Lagrange equations)**.** *Let Y be defined as an integral of the form*

$$Y(y) = \int_\Omega m(t, y, y')\, dt, \tag{3.1.1}$$

with $y' = \dfrac{dy}{dt}$, then the functional Y has an extremal function y if

$$\frac{\partial m}{\partial y} - \frac{d}{dt}\left(\frac{\partial m}{\partial y'}\right) = 0. \tag{3.1.2}$$

Proof. The proof for the E-L equations is well known and involves taking small perturbations of m, of which can be found in various places, e.g. [8] □

An example of a well known functional is Poisson's principle (in which a special case admits Dirichlet's principle), of which we show below.

Remark 1 (Poisson's Principle)**.** *Let $\Omega \subset \mathbb{R}$, $f, u : \mathbb{R} \to \mathbb{R}$, we define the following functional $\mathcal{G}$ as the following*

$$\mathcal{G}(u) := \int_\Omega \left(\frac{1}{2}\left(\frac{du}{dx}\right)^2 - uf \right) dx, \tag{3.1.3}$$

with $u \in \mathcal{C}^2(\Omega) \bigcap \mathcal{C}\left(\bar{\Omega}\right)$ taking some specified boundary values. Then the function that minimizes $\mathcal{G}$ over Ω is also the solution to Poisson's equation.

Proof. By applying the Euler-Lagrange equation to 3.1.3, we have that

$$\begin{aligned} m = \left(\frac{1}{2}\left(\frac{du}{dx}\right)^2 - uf \right) &\implies \frac{\partial m}{\partial u} = -f, \\ \frac{\partial m}{\partial u'} = \frac{du}{dx} &\implies \frac{d}{dx}\left(\frac{\partial m}{\partial u'}\right) = \frac{d}{dx}\left(\frac{du}{dx}\right) = \frac{d^2u}{dx^2}, \\ \therefore \frac{\partial m}{\partial u} - \frac{d}{dx}\left(\frac{\partial m}{\partial u'}\right) &= -f - \frac{d^2u}{dx^2} = 0, \\ &\implies -\frac{d^2u}{dx^2} = f, \end{aligned} \tag{3.1.4}$$

with u taking the specified boundary values as mentioned the above remark, and $u' := \dfrac{du}{dx}$. □

Remark 2. *By taking $f = 0$ in Poisson's principle, we have Laplace's equation (Dirichlet's Principle).*

3.2 The Isoperimetric problem

Isoperimetric problems are optimization problems that involve some type of boundary condition and integral condition. A homogeneous Dirichlet boundary value example is given by;

$$\text{find the extremum of:} \qquad Y(y) = \int_\Omega m(t, y, y')\, dt, \tag{3.2.1}$$

$$\text{subject to:} \quad W(y) = \int_\Omega w(t, y, y')\, dt = A, \tag{3.2.2}$$

$$\textit{and} \qquad y = 0, \text{ on } \partial\Omega. \tag{3.2.3}$$

Then in order to solve this problem, it is equivalent to solve the following minimization problem.

Proposition 3.1. *Given 3.2.1 , with its respective boundary and integral conditions, then it is equivalent to solve;*

$$\textit{find the extremum of:} \quad \hat{Y}(y) = \int_\Omega m(t, y, y') - \lambda \left(w(t, y, y') - \frac{A}{|\Omega|} \right) dt, \tag{3.2.4}$$

where $|\Omega|$ is the length of the interval $(\Omega \subset \mathbb{R})$ and λ is an unknown Lagrange multiplier (to be determined). Then, by applying the Euler - Lagrange equation, we know the extremal function of $\hat{Y}$ is satisfied by

$$\frac{\partial L}{\partial y} - \frac{d}{dt}\left(\frac{\partial L}{\partial y'} \right) = 0, \tag{3.2.5}$$

$$\textit{with} \quad L := m(t, y, y') - \lambda \left(w(t, y, y') - \frac{A}{|\Omega|} \right).$$

We now show how we solve an Isoperimetric problem using the Lagrange multiplier method using linear finite elements. Let us take an example of Poisson's principle (3.1.3), with two Neumann boundary conditions and an integral constraint (note that without the integral constraint, the problem is ill-posed - see section 6.2). We want to emphasise (and show) how we are using Lagrange multipliers in this example, since the way one solves this problem is by enforcing integral constraints. This is important as we will use an analogous approach in the Observational model later on.

3.2.1 Derivation and calculations

Example 3.1.

$$\textit{Minimize:} \quad \mathcal{R}(u) := \int_0^1 \left(\frac{1}{2} \left(\frac{du}{dx} \right)^2 + u(x) \right) dx, \tag{3.2.6}$$

$$\textit{subject to:} \int_0^1 u(x)\, dx = 0 \mid \frac{du}{dw}(0) = 1 \mid \frac{du}{dw}(1) = 2.$$

Notice that this is indeed Poisson's principle by taking $f = -1$ with natural boundary conditions. By taking approach 3.1, the solution to 3.2.6 is given by

$$-\frac{d^2u}{dx^2} + \lambda = -1, \tag{3.2.7}$$

with λ to be determined from the integral boundary condition. The finite element approach is to multiply the equation we are looking to solve by a test function $v(x) \in \mathcal{C}(0,1)$ and integrate over $\Omega = (0,1)$. This allows us to perform integration by parts to remove a derivative from the u term with two derivatives with respect to t. Multiplying 3.2.7 by a test function and integrating over $(0,1)$ yields

$$\int_0^1 -\frac{d^2u}{dx^2} v + \lambda v\, dx = \int_0^1 -v\, dx, \quad \forall v(x) \in \mathcal{C}(0,1). \tag{3.2.8}$$

We now recall the integration by parts formula, i.e. by integrating the product rule and using the fundamental theorem of calculus (FTC), we have for $a, b : \mathbb{R} \to \mathbb{R},\ a, b \in \mathcal{C}^1(\Omega),\ \Omega \subset \mathbb{R}$

$$\frac{d}{dx}\big(a(x)b(x)\big) = \frac{da}{dx}b(x) + \frac{db}{dx}a(x),$$

$$\implies \int_\Omega \left[\frac{d}{dx}\big(a(x)b(x)\big)\right] dx = \int_\Omega \left[\frac{da}{dx}b(x) + \frac{db}{dx}a(x)\right] dx,$$

$$\overset{FTC}{\implies} \int_\Omega \frac{da}{dx}b(x)\,dx = a(x)b(x)\Big|_{\partial\Omega} - \int_\Omega \frac{db}{dx}a(x)\,dx, \tag{3.2.9}$$

which implies that

$$\int_0^1 -\frac{d^2u}{dx^2}v\,dx = -\left[\left(\frac{du}{dx}v\right)\Big|_0^1 - \int_0^1 \frac{du}{dx}\frac{dv}{dx}\,dx\right] = \int_0^1 \frac{du}{dx}\frac{dv}{dx}\,dx - \left(\frac{du}{dx}v\right)\Big|_0^1.$$

This implies that 3.2.8 is equivalent to

$$\int_0^1 \frac{du}{dx}\frac{dv}{dx} + \lambda v\,dx = \int_0^1 -v\,dx + \left(\frac{du}{dx}v\right)\Big|_0^1 \quad \forall v(x) \in \mathcal{C}(0,1). \tag{3.2.10}$$

Equation 3.2.10 is what we refer to as a weak formulation. An important step now is to specify what function space we are working in and consequently what space our functions live in. Since the first derivative of u and v are within integrals, for the question to be well defined, we must have that

$$u, v \in \mathcal{H}^1(0,1), \tag{3.2.11}$$

where $\mathcal{H}^1(\Omega)$ defines a Hilbert space (3.0.5) where the function and its (weak) derivative are square integrable within the domain specified and hence, the integral equation given is well defined. Then by discretising the problem as in 3.0.1 by choosing a (sensible) $n \in \mathbb{N}$, we can create our finite element space as

$$V^h = \{v^h(x) \in \mathcal{C}([0,1]) : v^h(x) \in \mathcal{C}^1(x_i, x_{i+1})\ \forall i = [1:n]\} \subset \mathcal{H}^1(0,1). \tag{3.2.12}$$

Since V^h is a linear space, then it admits a basis of functions (3.0.3). Then, in the finite element space, our weak formulation becomes find $u^h \in V^h$ such that

$$\int_0^1 \frac{du^h}{dx}\frac{dv^h}{dx} + \lambda v^h(x)\,dx = \int_0^1 -v^h(x)\,dx + \left(\frac{du^h}{dx}v^h(x)\right)\Big|_0^1 \quad \forall v^h(x) \in V^h. \tag{3.2.13}$$

Since we now want to find $u^h \in V^h$, we can express u^h in terms of the V^h basis functions from 3.0.3. Let

$$u^h(x) = \sum_{i=1}^{n+1} u_i\phi_i(x), \tag{3.2.14}$$

$$v^h(x) = \phi_j(x). \tag{3.2.15}$$

Then by inserting 3.2.14 and 3.2.15 into 3.2.13, our finite element problem becomes

$$\int_0^1 \left(\sum_{i=1}^{n+1} u_i\left(\frac{d}{dx}\phi_i(x)\right)\right)\left(\frac{d}{dx}\phi_j(x)\right) + \lambda\phi_j(x)\,dx = \int_0^1 -\phi_j(x)\,dx + \left(\frac{du}{dx}\phi_j(x)\right)\Big|_0^1. \tag{3.2.16}$$

To solve this problem, we know by the properties of 3.0.4, for any arbitrary $j \neq \{1, n+1\}$, $\phi_j(x)$ is only non zero in the interval (x_{j-1}, x_{j+1}). Therefore, we can simplify the above into $(n+2)$ linear simultaneous equations, where $(n+1)$ of these equations come from the the ordinary differential equation, and the last comes form enforcing the boundary condition onto the Lagrange multiplier. Since the sum is finite, by Fubini's theorem, we may interchange the summation and integral, so that 3.2.16 becomes

$$\sum_{i=1}^{n+1} u_i\left(\int_{x_{j-1}}^{x_{j+1}} \left(\frac{d}{dx}\phi_i(x)\right)\left(\frac{d}{dx}\phi_j(x)\right) dx\right) + \lambda\int_{x_{j-1}}^{x_{j+1}} \phi_j(x)\,dx = \int_{x_{j-1}}^{x_{j+1}} -\phi_j(x)\,dx + \left(\frac{du}{dx}\phi_j(x)\right)\Big|_0^1. \tag{3.2.17}$$

If we now assign

$$A_{ij} = \int_{x_{j-1}}^{x_{j+1}} \left(\frac{d}{dx}\phi_i(x)\right)\left(\frac{d}{dx}\phi_j(x)\right) dx, \tag{3.2.18}$$
$$C_j = \int_{x_{j-1}}^{x_{j+1}} \phi_j(x)\, dx \quad | \quad F_j = \int_{x_{j-1}}^{x_{j+1}} -\phi_j(x)\, dx + \left(\frac{du}{dx}\phi_j(x)\right)\Big|_0^1.$$

Then in ***block*** *matrix notation, it is equivalent to solve the following system (since λ and U are the unknown quantities we are searching for where we define U as the vector of the u_i $(= u(x_i))$ components)*

$$\begin{bmatrix} A & C^T \\ C & 0 \end{bmatrix}\begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} F \\ M \end{bmatrix} \quad | \quad A \in \mathbb{R}^{(n+1)\times(n+1)}, C^T, F, U \in \mathbb{R}^{(n+1)}, \quad \lambda, M \in \mathbb{R}. \tag{3.2.19}$$

In which after some brief calculations,

$$A_{i,j} = \begin{cases} -\dfrac{1}{\Delta x} & \text{if} \quad \{j = i-1\} \bigcap \{i \neq 1\}, \\ \dfrac{2}{\Delta x} & \text{if} \quad \{j = i\} \bigcap \{i \neq 1, n+1\}, \\ -\dfrac{1}{\Delta x} & \text{if} \quad \{j = i+1\} \bigcap \{i \neq n+1\}, \\ \dfrac{1}{\Delta x} & \text{if} \quad \{j = i\} \bigcap \{i = 1, n+1\}, \end{cases} \tag{3.2.20}$$

$$F_j = \begin{cases} -\Delta x & \text{if} \quad \{j \neq 1, n+1\}, \\ -\dfrac{\Delta x}{2} - 1 & \text{if} \quad \{j = 1\}, \\ -\dfrac{\Delta x}{2} + 2 & \text{if} \quad \{j = n+1\}, \end{cases} \quad | \quad C_j = \begin{cases} \Delta x & \text{if} \quad \{j \neq 1, n+1\}, \\ \dfrac{\Delta x}{2} & \text{if} \quad \{j = 1\}, \\ \dfrac{\Delta x}{2} & \text{if} \quad \{j = n+1\}. \end{cases} \tag{3.2.21}$$

Proof. Left to the interested reader. We hint that the vector C uses the trapezium rule to enforce the boundary conditions. To calculate F, simply look at the definition of the basis function over any interval. To calculate A, it suffices to show that on for any ϕ_j is only non zero on the interval (x_{j-1}, x_{j+1}). Further information on these calculations can be found at [7]. □

Note that M denotes the integral constraint (given value is 0). The reason we have the vector C, is due to using the ***trapezium*** *rule on the u_i components to enforce the boundary condition, in order to find the value of the unknown Lagrange multiplier λ. All we need to do now is implement this into a numerical program, of which we will use MATLAB.*

3.2.2 Results

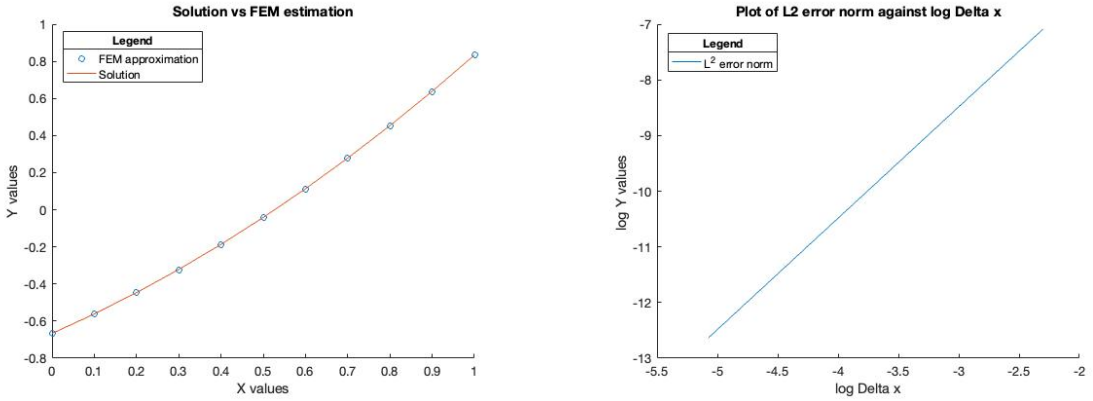
We claim that the solution to the minimization problem given as the Poisson ODE from

$$-\frac{d^2u}{dx^2} = -1 \quad | \quad u \in (0,1),$$
$$\frac{du}{dx}(0) = 1 \quad | \quad \frac{du}{dx}(1) = 2,$$
$$\int_0^1 u(x)\, dx = 0,$$

is given by

$$u(x) = \frac{1}{2}u^2 + u - \frac{2}{3}.$$

By usual methods, the interested reader can indeed check this satisfies all the conditions. Knowing the analytical solution is useful as it allows us to measure the error between our finite element approximation and the true solution. Furthermore, we can show even with the use of Lagrange multipliers that we still have quadratic order of convergence with respect to the $\mathcal{L}^2$ and $\mathcal{L}^\infty$ norm.



(a) FEM approximation against the solution, $\Delta x = 0.1$. (b) $\ln \mathcal{L}^2$ errors vs $\ln \Delta x \implies \text{EOC} = 2$ (from table 3).

With the Lagrange multiplier value calculated as -6.66×10^{-16}. We now present a table with varying (uniform) mesh sizes, where each Δx_i denotes the distance between each nodal point, in which we calculate the $\mathcal{L}^2$ and $\mathcal{L}^\infty$ errors, and present plots of the estimated order of convergence for both the $\mathcal{L}^2$ error and $\mathcal{L}^\infty$ error.

Δx	$\mathcal{L}^2$ error	$\mathcal{L}^2$ EOC	$\mathcal{L}^\infty$ error	$\mathcal{L}^\infty$ EOC
$\Delta x_1 = 1/10$	8.33×10^{-4}	-	8.33×10^{-4}	-
$\Delta x_2 = 1/20$	2.08×10^{-4}	2	2.08×10^{-4}	2
$\Delta x_3 = 1/40$	5.21×10^{-5}	2	5.21×10^{-5}	2
$\Delta x_4 = 1/80$	1.30×10^{-5}	2	1.30×10^{-5}	2
$\Delta x_5 = 1/160$	3.26×10^{-6}	2	3.26×10^{-6}	2

Table 3: A table of the estimated order of convergence (EOC) calculations taken from our isoperimetric example for the $\mathcal{L}^2$ errors and $\mathcal{L}^\infty$ errors.

3.3 Adapting the Isoperimetric problem to solve the Observational model

In the example just conducted, we had a linear problem (w.r.t u within the finite element formulation) with Neumann boundary conditions. As seen previously in the shooting method section, the Observational model does not come with any standard prescribed boundary conditions, only the nonlocal integral boundary conditions expressed by the data, along with a high level of non-linearity. Let us recall the Observational model once more (w.r.t $I(t)$)

$$\frac{d^2I}{dt^2} = \frac{dI}{dt}\left(\frac{1}{I}\frac{dI}{dt} - \frac{\beta I}{N}\right) - \frac{\beta\gamma I^2}{N}, \tag{3.3.1}$$

$$X_0 = r\gamma \int_0^1 I(s)\,ds \quad | \quad X_1 = r\gamma \int_1^2 I(s)\,ds. \tag{3.3.2}$$

An immediate problem we see before we can even think about making a weak formulation is how we deal with the $I(t)^{-1}$ term inside the bracket. The simplest and most obvious thing we can do is multiply all terms by $I(t)$ so that everything is well versed for creating a weak formulation. However, to ensure that this is valid for all time, we must ensure that

$$\lim_{I\to 0}\left(\frac{1}{I}\frac{dI}{dt}\right) \text{ is bounded.} \tag{3.3.3}$$

So that multiplying by $I(t)$ is well-posed. In fact, by substituting 1.1.1 into 3.3.3, we see that

$$\lim_{I\to 0}\left(\frac{1}{I}\frac{dI}{dt}\right)=\lim_{I\to 0}\frac{1}{I}\left(\beta\frac{I}{N}S-\gamma I\right)=\lim_{I\to 0}\frac{\beta}{N}S(t)-\gamma\in(-\gamma,\beta-\gamma), \tag{3.3.4}$$

by steady state analysis [1], since at the end of an epidemic - there are no more infectious people. Since the limit does not blow up, we can continue getting the Observational model in the right shape for a finite element scheme. By multiplying 3.3.1 by $I(t)$, we formally define the function P as

$$P(I)=-\frac{d^2I}{dt^2}I+\frac{dI}{dt}\left(\frac{dI}{dt}-\frac{\beta I^2}{N}\right)-\frac{\beta\gamma I^3}{N}=0. \tag{3.3.5}$$

Proposition 3.2. *We define $P_{E-L}(I,I',t)$ to be the functional that, when we look to find the extremum of this functional over its domain subject to 3.3.1, we produce the function P, i.e.,*

$$\textit{find the extremum of:}\quad \int_0^1 P_{E-L}(I,I',t)\,dt,\quad \textit{subject to } r\gamma\int_0^1 I(s)\,ds=X_0, \tag{3.3.6}$$

$$\textit{find the extremum of:}\quad \int_1^2 P_{E-L}(I,I',t)\,dt,\quad \textit{subject to } r\gamma\int_1^2 I(s)\,ds=X_1. \tag{3.3.7}$$

Then the weak formulation for each integral constraint is given by

$$F_0(I,I',\lambda_0,v_0)=\int_0^1 2\left(\frac{dI}{dt}\right)^2 v_0+\left(\frac{dI}{dt}\right)\left(\frac{dv_0}{dt}\right)I-\frac{\beta I^2}{N}\left(\frac{dI}{dt}\right)v_0-\frac{\beta\gamma I^3}{N}v_0 \tag{3.3.8}$$
$$+\lambda_0(r\gamma)v_0\,dt-\frac{dI}{dt}Iv_0\Big|_0^1=0$$

$$F_1(I,I',\lambda_1,v_1)=\int_1^2 2\left(\frac{dI}{dt}\right)^2 v_1+\left(\frac{dI}{dt}\right)\left(\frac{dv_1}{dt}\right)I-\frac{\beta I^2}{N}\left(\frac{dI}{dt}\right)v_1-\frac{\beta\gamma I^3}{N}v_1 \tag{3.3.9}$$
$$+\lambda_1(r\gamma)v_1\,dt-\frac{dI}{dt}Iv_1\Big|_1^2=0$$

$$\forall v_0\in\mathcal{H}^1(0,1)\quad \textbf{\textit{and}}\quad \forall v_1\in\mathcal{H}^1(1,2).$$

Proof. By using Proposition 3.1 and Lagrange multipliers, it is equivalent to finding;

$$\text{find the extremum of:}\quad \int_0^1 P_{E-L}(I,I',t)+\lambda_0\left(r\gamma I(t)-X_0\right)\,dt, \tag{3.3.10}$$

$$\text{find the extremum of:}\quad \int_1^2 P_{E-L}(I,I',t)+\lambda_1\left(r\gamma I(t)-X_1\right)\,dt. \tag{3.3.11}$$

For ease of exposition, we will only work with 3.3.10, since 3.3.11 is analogous. We define

$$\Gamma_0:=P_{E-L}(I,I',t)+\lambda_0\left(r\gamma I(t)-X_0\right).$$

Then, by applying the Euler- Lagrange equation to 3.3.10, and by using the properties that differentiation is a linear operator,

$$\begin{aligned}\frac{\partial\Gamma_0}{\partial I}-\frac{d}{dt}\left(\frac{\partial\Gamma_0}{\partial I'}\right)=0 &\implies \left(\frac{\partial P_{E-L}}{\partial I}+\lambda_0 r\gamma\right)-\frac{d}{dt}\left(\frac{\partial P_{E-L}}{\partial I'}\right)=0,\\ &\implies \left[\frac{\partial P_{E-L}}{\partial I}-\frac{d}{dt}\left(\frac{\partial P_{E-L}}{\partial I'}\right)\right]+\lambda_0 r\gamma=0,\\ &\overset{!}{\implies} -\frac{d^2I}{dt^2}I+\frac{dI}{dt}\left(\frac{dI}{dt}-\frac{\beta I^2}{N}\right)-\frac{\beta\gamma I^3}{N}+\lambda_0 r\gamma=0,\end{aligned}$$

noting that (!) uses the assumption we have made at the start of the proposition. By further multiplying by a test function $v_0(t)\in\mathcal{H}^1(0,1)$ and integrating over $(0,1)$, then by following the steps already outlined in 3.2.9 for integrating by parts the term with two derivatives with respect to t, we have the desired result. □

Then our weak formulation is to find $I\in\mathcal{H}^1(0,1)$ **and** $I\in\mathcal{H}^1(1,2)$, such that

$$F(I,I',\lambda_0,\lambda_1,v_0,v_1)=[F_0(I,I',\lambda_0,v_0),F_1(I,I',\lambda_1,v_1)]=0, \tag{3.3.12}$$

$$\forall v_0\in\mathcal{H}^1(0,1)\quad \textbf{and}\quad \forall v_1\in\mathcal{H}^1(1,2).$$

3.3.1 Choosing a finite element scheme - Newtons method

Since we have defined our weak formulation and function spaces, we are now at the stage that we need to decide on a sensible scheme to deploy into our weak formulation and then move into the FEM space. We admit sensible here is ambiguous, since the problem we are trying to tackle is novel and not well studied. The first thoughts for a scheme that come to mind are fixed point methods (Picard iterations), linearization techniques through Taylor's expansion and various versions of Newton methods (i.e. damped and non-damped). Further discussions of which methods are appropriate can be found in[4], depending on the problem at hand. We will focus on using Newtons method. To use Newtons method, we will first recall Taylor's expansion for multi-variate functions.

Theorem 3.2 (Taylor's theorem for multi-variate functions (1st order approximation))**.** *Let $\eta : \mathbb{R}^n \to \mathbb{R}^n$ and be a k $(\geqslant 2)$ times differentiable function at a point $\mu \in \mathbb{R}^n$ $(x \in \mathbb{R}^n)$, then we can express η as*

$$\eta(x) = \eta(\mu) + \left(\sum_{i,j=1}^{n} \frac{\partial \eta_i}{\partial x_j}(u) \right) \times [x - \mu]_{j,1} + O(x - \mu)^2, \tag{3.3.13}$$

or in more compact notation,

$$\eta(x) = \eta(\mu) + J(u)(x - \mu) + O(x - \mu)^2. \tag{3.3.14}$$

Where J is the $(n \times n)$ jacobian matrix of the function η, and $O(x-\mu)^2$ denotes terms of $(x-\mu)^2$ and above.

As usual in Newtons method, we take a function and want to find its roots, so by setting a function equal to 0, or in this case, our Taylor's expansion - we have that a first order approximation is given by

$$\eta(x) \approx \eta(\mu) + J(u)(x - \mu) = 0 \implies (x - \mu) \approx -(J(u))^{-1}\eta(\mu), \tag{3.3.15}$$

where $(J(u))^{-1}$ denotes the inverse of the jacobian mapping. Currently it is not clear how we can use the above in a finite element scheme, but this will be revealed further down. We must now discretize the domain into a (uniform) arbitrary partition with

$$0 = t_1 < t_2 < t_3 < \cdots < t_{n-1} < t_n < t_{n+1} = 2, \tag{3.3.16}$$

such that by choosing an (**even**) $n \in (2 \times \mathbb{N})$,

$$\Delta t = \frac{2 - 0}{n} \quad | \quad t_{i+1} = t_i + \Delta t \quad \forall i \in [1, n].$$

The reason for choosing an **even** n is due to how we enforce the Lagrange multipliers, and that $\exists\ t_i$ such that $t_i = 1$ $(t_{\frac{n}{2}+1} = 1)$. If $n \notin (2 \times \mathbb{N})$, we are not able to enforce the nonlocal integral boundary conditions correctly. We are now ready to move our weak formulation in 3.3.12 into a finite element space before we implement the above Newtons method. We define

$$V_0^h := \{v_0^h(t) \in \mathcal{C}([0,1]) : v_0^h(t) \in \mathcal{C}^1(t_i, t_{i+1})\ \forall i = [1 : (n/2)]\} \subset \mathcal{H}^1(0,1), \tag{3.3.17}$$

$$V_1^h := \{v_1^h(t) \in \mathcal{C}([1,2]) : v_1^h(t) \in \mathcal{C}^1(t_i, t_{i+1})\ \forall i = [(n/2) + 1 : n]\} \subset \mathcal{H}^1(1,2). \tag{3.3.18}$$

Then our weak formulation in the finite element space becomes find $I^h \in V_0^h$ **and** $I^h \in V_1^h$ such that

$$F(I^h, (I^h)', \lambda_0, \lambda_1, v_0^h, v_1^h) := [F_0(I^h, (I^h)', \lambda_0, v_0^h), F_1(I^h, (I^h)', \lambda_1, v_1^h)] = 0, \tag{3.3.19}$$

$$\forall v_0^h \in V_0^h \textbf{ and } \forall v_1^h \in V_1^h.$$

Remark 3 (Continuity of the solution)**.** *Since $I \in \mathcal{C}^2(0,2)$ [1], we need to enforce $I \in C(0,2)$ into our finite element formulation. Currently we have two vectors being solved simultaneously, but this does not imply that the resulting solution provided will be continuous. Since we require $I \in \mathcal{C}(0,2)$, we can combine (add) the respective boundary vector values from each vector at $(t = 1)$ to enforce continuity between the two weak finite element formulations.*

Since V_0^h and V_1^h admits a basis of functions, we can express $I^h(t)$, $v_0^h(t)$ and $v_1^h(t)$ as

$$I^h(t) = \sum_{i=1}^{n+1} I_i \phi_i(t), \tag{3.3.20}$$

$$v_0^h(t) = v_1^h(t) = \phi_j(t), \tag{3.3.21}$$

where $\phi_j(t)$ is the linear basis function we previously defined in 3.0.3.

3.3.2 Calculating the finite element formulation

We now approach the task of calculating a rather daunting finite element formulation. We first approach calculating $F(I^h, (I^h)', \lambda_0, \lambda_1, v_0^h, v_1^h) = 0$ by inserting 3.3.20 and 3.3.21 into 3.3.19. Then using the Newtons approach, by incorporating an initial guess I^0 we will talk about shortly, we then look to calculate the jacobian and talk about how we enforce the Lagrange multipliers from the integral boundary constraints to solve for $I(t)$.

Proposition 3.3 (Finite element formulation)**.** *We first start by noting that $F(I^h, (I^h)', \lambda_0, \lambda_1, v_0^h, v_1^h) \in \mathbb{R}^{(n+3)}$, where the first $(n+1)$ rows are from the finite element formulation calculations. The remaining two entries we will come back to later. For $j \neq \{1, n+1\}$, we state the following calculations, given all the necessary parameter's. We note that going down, we are moving into the finite element formulation of equation 3.3.8 from left to right.*

$$\begin{aligned}
&\int_{t_{j-1}}^{t_{j+1}} 2\left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)^2 \phi_j(t)\,dt, \\
&= \sum_{\zeta=j-1}^{j} \int_{t_\zeta}^{t_{\zeta+1}} 2\left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)^2 \phi_j(t)\,dt, \\
&= \frac{1}{\Delta t}\left(I_j^2 + I_{j-1}^2 - 2I_jI_{j-1}\right) + \frac{1}{\Delta t}\left(I_j^2 + I_{j+1}^2 - 2I_jI_{j+1}\right). \qquad (3.3.22)
\end{aligned}$$

$$\begin{aligned}
&\int_{t_{j-1}}^{t_{j+1}} \left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)\left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\left(\frac{d\phi_j(t)}{dt}\right) dt, \\
&= \sum_{\zeta=j-1}^{j} \int_{t_\zeta}^{t_{\zeta+1}} \left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)\left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\left(\frac{d\phi_j(t)}{dt}\right) dt, \\
&= \frac{1}{2\Delta t}\left(I_j^2 - I_{j-1}^2\right) + \frac{1}{2\Delta t}\left(I_j^2 - I_{j+1}^2\right). \qquad (3.3.23)
\end{aligned}$$

$$\begin{aligned}
&\int_{t_{j-1}}^{t_{j+1}} -\frac{\beta}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^2\left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\phi_j(t)\,dt, \\
&= \sum_{\zeta=j-1}^{j} \int_{t_\zeta}^{t_{\zeta+1}} -\frac{\beta}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^2\left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\phi_j(t)\,dt, \\
&= -\frac{\beta}{N}\left((I_j)^3\left(\frac{1}{4}\right) + (I_{j-1})^3\left(-\frac{1}{12}\right) + (I_{j-1})^2I_j\left(-\frac{1}{12}\right) + (I_j)^2I_{j-1}\left(-\frac{1}{12}\right)\right) \\
&\quad -\frac{\beta}{N}\left((I_j)^3\left(-\frac{1}{4}\right) + (I_{j+1})^3\left(\frac{1}{12}\right) + (I_{j+1})^2I_j\left(\frac{1}{12}\right) + (I_j)^2I_{j+1}\left(\frac{1}{12}\right)\right). \qquad (3.3.24)
\end{aligned}$$

$$\begin{aligned}
&\int_{t_{j-1}}^{t_{j+1}} -\frac{\beta\gamma}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^3\phi_j(t)\,dt, \\
&= \sum_{\zeta=j-1}^{j} \int_{t_\zeta}^{t_{\zeta+1}} -\frac{\beta\gamma}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^3\phi_j(t)\,dt, \\
&= -\frac{\beta\gamma}{N}\left((I_{j-1})^3\left(\frac{\Delta t}{20}\right) + (I_{j-1})^2I_j\left(\frac{\Delta t}{10}\right)(I_j)^2I_{j-1}\left(\frac{3\Delta t}{20}\right) + (I_j)^3\left(\frac{\Delta t}{5}\right)\right) \\
&\quad -\frac{\beta\gamma}{N}\left((I_j)^3\left(\frac{\Delta t}{5}\right) + (I_j)^2I_{j+1}\left(\frac{3\Delta t}{20}\right) + I_j(I_{j+1})^2\left(\frac{\Delta t}{10}\right) + (I_{j+1})^3\left(\frac{\Delta t}{20}\right)\right). \qquad (3.3.25)
\end{aligned}$$

$$\lambda_0 r\gamma \int_{t_{j-1}}^{t_{j+1}} \phi_j(t)\,dt, = \lambda_0 r\gamma(\Delta t) \quad | \quad (t_{j-1}, t_{j+1}) \in (0,1), \tag{3.3.26}$$

$$\lambda_1 r\gamma \int_{t_{j-1}}^{t_{j+1}} \phi_j(t)\,dt = \lambda_1 r\gamma(\Delta t) \quad | \quad (t_{j-1}, t_{j+1}) \in (1,2). \tag{3.3.27}$$

Proof. We will prove 3.3.24, the rest are analogous after we show the integral calculations.

$$\int_{t_{j-1}}^{t_{j+1}} -\frac{\beta}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^2 \left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\phi_j(t)\,dt,$$

$$= \sum_{\zeta=j-1}^{j} \int_{t_\zeta}^{t_{\zeta+1}} -\frac{\beta}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^2 \left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\phi_j(t)\,dt.$$

For $\zeta = j-1$

$$\int_{t_{j-1}}^{t_j} -\frac{\beta}{N}\left(\sum_{i=1}^{n+1} I_i\phi_i(t)\right)^2 \left(\sum_{i=1}^{n+1} I_i\left(\frac{d}{dt}\phi_i(t)\right)\right)\phi_j(t)\,dt, \tag{3.3.28}$$

$$= -\frac{\beta}{N}\int_{t_{j-1}}^{t_j} \left(I_{j-1}\phi_{j-1}(t) + I_j\phi_j(t)\right)^2 \left(I_{j-1}\frac{d}{dt}(\phi_{j-1}(t)) + I_j\frac{d}{dt}(\phi_j(t))\right)\phi_j(t)\,dt. \tag{3.3.29}$$

From 3.0.3, we know that

$$\left(I_{j-1}\frac{d}{dt}(\phi_{j-1}(t)) + I_j\frac{d}{dt}(\phi_j(t))\right) = I_{j-1}\left(\frac{-1}{\Delta t}\right) + I_j\left(\frac{1}{\Delta t}\right), \quad (\text{on } (t_{j-1}, t_j)).$$

Therefore 3.3.29 is equal to

$$= -\frac{\beta}{N}\frac{1}{\Delta t}\left(I_j - I_{j-1}\right)\int_{t_{j-1}}^{t_j} \left(I_{j-1}\phi_{j-1}(t) + I_j\phi_j(t)\right)^2\phi_j(t)\,dt,$$

$$= -\frac{\beta}{N}\frac{1}{\Delta t}\left(I_j - I_{j-1}\right)\int_{t_{j-1}}^{t_j} I_{j-1}^2\phi_{j-1}^2(t)\phi_j(t) + I_j^2\phi_j^3(t) + 2I_{j-1}I_j\phi_{j-1}(t)\phi_j^2(t)\,dt,$$

$$(\dagger) = -\frac{\beta}{N}\frac{1}{\Delta t}\left(I_j - I_{j-1}\right)\left(I_{j-1}^2\int_{t_{j-1}}^{t_j}\phi_{j-1}^2(t)\phi_j(t)\,dt + I_j^2\int_{t_{j-1}}^{t_j}\phi_j^3(t)\,dt + 2I_{j-1}I_j\int_{t_{j-1}}^{t_j}\phi_{j-1}(t)\phi_j^2(t)\,dt\right).$$

We now show how one goes about calculating the product of the basis functions. By definition,

$$\int_{t_{j-1}}^{t_j}\phi_{j-1}^2(t)\phi_j(t)\,dt = \int_{t_{j-1}}^{t_j}\left(1 - \frac{t-t_{j-1}}{\Delta t}\right)^2\left(1 - \frac{t_j - t}{\Delta t}\right)\,dt. \tag{3.3.30}$$

If we introduce a substitution $y = \dfrac{t-t_{j-1}}{\Delta t} \implies dy = \dfrac{1}{\Delta t}\,dt$, then 3.3.30 is equivalent too

$$\Delta t\int_0^1 (1-y)^2(1+(y-1))\,dy = \Delta t\int_0^1 y - 2y^2 + y^3\,dy = \Delta t\left(\frac{1}{12}\right). \tag{3.3.31}$$

The rest are analogous. Therefore † is equivalent to

$$-\frac{\beta}{N}\frac{1}{\Delta t}\left(I_j - I_{j-1}\right)\left(I_{j-1}^2\left(\frac{\Delta t}{12}\right) + I_j^2\left(\frac{\Delta t}{4}\right) + I_{j-1}I_j\left(\frac{\Delta t}{6}\right)\right). \tag{3.3.32}$$

Of which after expanding and simplifying gives the first half of the result. We will only make minor comments for $\zeta = j$ since most of the calculations are the same.

For $\zeta = j$, after expansion, the first line looks like;

$$-\frac{\beta}{N}\int_{t_j}^{t_{j+1}} \left(I_{j+1}\phi_{j+1}(t) + I_j\phi_j(t)\right)^2\left(I_{j+1}\frac{d}{dt}(\phi_{j+1}(t)) + I_j\frac{d}{dt}(\phi_j(t))\right)\phi_j(t)\,dt. \tag{3.3.33}$$

Here we note that since we are now looking at integrating over (t_j, t_{j+1}), the weak derivative of $\phi_j(t)$ changes from the previous computation, since we are now on a different interval. We leave the rest to the interested reader, but note all tools needed have been shown in the above calculations.

□

We also need to talk about some specific points, primarily $j = 1$, $j = n + 1$ and finally the midpoint $j = \frac{n}{2} + 1$. For $j = 1$, $\phi_1(t)$ is non zero only on the interval (t_1, t_2), and hence, combining all of the terms together, we formally give $F_1(I^h, (I^h)', \lambda_0, \lambda_1, v_0^h, v_1^h) = F_1$ as

$$\begin{aligned} F_1 = \frac{1}{\Delta t}\left(I_1^2 + I_2^2 - 2I_1I_2\right) + \frac{1}{2\Delta t}\left(I_1^2 - I_2^2\right) \\ -\frac{\beta}{N}\left((I_1)^3\left(-\frac{1}{4}\right) + (I_2)^3\left(\frac{1}{12}\right) + (I_2)^2I_1\left(\frac{1}{12}\right) + (I_1)^2I_2\left(\frac{1}{12}\right)\right) \\ -\frac{\beta\gamma}{N}\left((I_1)^3\left(\frac{\Delta t}{5}\right) + (I_1)^2I_2\left(\frac{3\Delta t}{20}\right) + I_1(I_2)^2\left(\frac{\Delta t}{10}\right) + (I_2)^3\left(\frac{\Delta t}{20}\right)\right) \\ +\frac{\Delta t}{2}r\gamma\lambda_0 + I_1\left(\frac{dI_1}{dt}\right). \end{aligned} \tag{3.3.34}$$

For $j = n + 1$,

$$\begin{aligned} F_{n+1} = \frac{1}{\Delta t}\left(I_{n+1}^2 + I_n^2 - 2I_{n+1}I_n\right) + \frac{1}{2\Delta t}\left(I_{n+1}^2 - I_n^2\right) \\ -\frac{\beta}{N}\left((I_{n+1})^3\left(\frac{1}{4}\right) + (I_n)^3\left(-\frac{1}{12}\right) + (I_n)^2I_{n+1}\left(-\frac{1}{12}\right) + (I_{n+1})^2I_n\left(-\frac{1}{12}\right)\right) \\ -\frac{\beta\gamma}{N}\left((I_n)^3\left(\frac{\Delta t}{20}\right) + (I_n)^2I_{n+1}\left(\frac{\Delta t}{10}\right)(I_{n+1})^2I_n\left(\frac{3\Delta t}{20}\right) + (I_{n+1})^3\left(\frac{\Delta t}{5}\right)\right) \\ +\frac{\Delta t}{2}r\gamma\lambda_1 - I_{n+1}\left(\frac{dI_{n+1}}{dt}\right). \end{aligned} \tag{3.3.35}$$

Noting here that at the end of $j = 1$ and $j = n+1$, we have included the appropriate boundary condition. When implementing this into a numerical solver, the finite element approximation is given at these points.

We now make special note for $j = \frac{n}{2} + 1$ of which is the nodal point for the time value 1. We now look to implement remark 3 into our finite element scheme, so that we can ensure continuity of the solution. This means $\phi_{\frac{n}{2}+1}$ will be in both intervals $(0, 1)$ and $(1, 2)$.

For $j = \dfrac{n}{2} + 1$,

$$\begin{aligned} F_{(n/2)+1} = \frac{1}{\Delta t}\left(I_j^2 + I_{j-1}^2 - 2I_jI_{j-1}\right) \\ +\frac{1}{\Delta t}\left(I_j^2 + I_{j+1}^2 - 2I_jI_{j+1}\right) + \frac{1}{2\Delta t}\left(I_j^2 - I_{j-1}^2\right) + \frac{1}{2\Delta t}\left(I_j^2 - I_{j+1}^2\right) \\ -\frac{\beta}{N}\left((I_j)^3\left(\frac{1}{4}\right) + (I_{j-1})^3\left(-\frac{1}{12}\right) + (I_{j-1})^2I_j\left(-\frac{1}{12}\right) + (I_j)^2I_{j-1}\left(-\frac{1}{12}\right)\right) \\ -\frac{\beta}{N}\left((I_j)^3\left(-\frac{1}{4}\right) + (I_{j+1})^3\left(\frac{1}{12}\right) + (I_{j+1})^2I_j\left(\frac{1}{12}\right) + (I_j)^2I_{j+1}\left(\frac{1}{12}\right)\right) \\ -\frac{\beta\gamma}{N}\left((I_{j-1})^3\left(\frac{\Delta t}{20}\right) + (I_{j-1})^2I_j\left(\frac{\Delta t}{10}\right)(I_j)^2I_{j-1}\left(\frac{3\Delta t}{20}\right) + (I_j)^3\left(\frac{\Delta t}{5}\right)\right) \\ -\frac{\beta\gamma}{N}\left((I_j)^3\left(\frac{\Delta t}{5}\right) + (I_j)^2I_{j+1}\left(\frac{3\Delta t}{20}\right) + I_j(I_{j+1})^2\left(\frac{\Delta t}{10}\right) + (I_{j+1})^3\left(\frac{\Delta t}{20}\right)\right) \\ +\frac{\Delta t}{2}r\gamma(\lambda_0 + \lambda_1). \end{aligned} \tag{3.3.36}$$

When evaluated at $j = \frac{n}{2} + 1$, for ease of exposition I will leave this un-evaluated. Note in passing that this is different to every other value (not including $j = 1$ and $j = n + 1$) since instead of having one λ_i value, we have both since we cover both intervals here. Something that may be missed by some is that when adding the boundary conditions together, they consequently cancel out. So we now know the first $(n + 1)$ rows of the vector $F = 0$. As highlighted in 3.3.15, when looking to use Newtons method, we look to solve for a difference around the point we are seeking to find by using an initial guess. For now, let I^0 be our initial guess for the function I, then we are looking to solve

$$\begin{aligned} \delta I = I - I^0 = -J(I^0)^{-1}F(I^0), \\ \text{Set: } I = I^0 + \delta I. \end{aligned} \tag{3.3.37}$$

Before looking at the assembly of the jacobian, we will briefly talk about what the last two rows are for in the vector $F \in \mathbb{R}^{n+3}$. Much like in the example of the linear finite elements problem we solved, we need to make sure that the integral boundary conditions were satisfied. If we start with an initial guess (curve), then the difference that δI must satisfy on each interval **must** be the difference between

$$X_i - r\gamma \int_i^{i+1} I^0(s)\, ds, \quad i = 0, 1.$$

If we under-guess the solution, it must come up to satisfy the boundary conditions and vice versa on each interval. Therefore

$$F_{n+2} = X_0 - r\gamma \int_0^1 I^0(s)\, ds \quad | \quad F_{n+3} = X_1 - r\gamma \int_1^2 I^0(s)\, ds.$$

Where on each iteration, we will use a quadrature rule (trapezium rule) to estimate the given integrals, and hence estimate the difference values F_{n+2} and F_{n+3}. We now look to assemble our jacobian matrix. We formally define our jacobian as

$$J(I^0) := \begin{bmatrix} \frac{dF_1}{d_{I_1}} & \cdots & \cdots & \frac{dF_1}{d_{I_{(n/2)+1}}} & \cdots & \cdots & \frac{dF_1}{d_{I_{n+1}}} & \frac{dF_1}{d_{\lambda_0}} & \frac{dF_1}{d_{\lambda_1}} \\ \vdots & & & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & & & \ddots & \ddots & \vdots \\ \frac{dF_{n+1}}{d_{I_1}} & \cdots & \cdots & \frac{dF_{n+1}}{d_{I_{(n/2)+1}}} & \cdots & \cdots & \frac{dF_{n+1}}{d_{I_{n+1}}} & \frac{dF_{n+1}}{d_{\lambda_0}} & \frac{dF_{n+1}}{d_{\lambda_1}} \\ \frac{\Delta t}{2} & \Delta t & \cdots & \frac{\Delta t}{2} & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cdots & 0 & \frac{\Delta t}{2} & \Delta t & \cdots & \frac{\Delta t}{2} & 0 & 0 \end{bmatrix}^{I^0}, \tag{3.3.38}$$

where we have added an extra two rows at the bottom of the jacobian in which describe the integral constraints of how much the initial guess can move (by δI) compared to the data points X_0 and X_1 we are given. To be clear,

$$\begin{bmatrix} \frac{\Delta t}{2} & \Delta t & \cdots & \frac{\Delta t}{2} \end{bmatrix} := \begin{bmatrix} \frac{\Delta t}{2} & \Delta t & \Delta t & \cdots & \Delta t & \Delta t & \frac{\Delta t}{2} \end{bmatrix}. \tag{3.3.39}$$

Of which is the direct use of the trapezium method (6.1.1). The component calculations of the jacobian are fairly straight-forward since we already have the values of the vector F. We will show particular calculations of interest. By usual rules of differentiation, we have

$$\begin{aligned} \frac{dF_1}{dI_1} = \frac{2}{\Delta t}(I_1 - I_2) + \frac{1}{\Delta t}(I_1) \\ -\frac{\beta}{N}\left((I_1)^2\left(-\frac{3}{4}\right) + (I_2)^2\left(\frac{1}{12}\right) + (I_1)I_2\left(\frac{1}{6}\right)\right) \\ -\frac{\beta\gamma}{N}\left((I_1)^2\left(\frac{3\Delta t}{5}\right) + (I_1)I_2\left(\frac{3\Delta t}{10}\right) + (I_2)^2\left(\frac{\Delta t}{10}\right)\right) + \left(\frac{dI_1}{dt}\right). \end{aligned} \tag{3.3.40}$$

$$\begin{aligned} \frac{dF_{n+1}}{dI_{n+1}} = \frac{1}{\Delta t}(2I_{n+1} + -2I_n) + \frac{1}{2\Delta t}(2I_{n+1}) \\ -\frac{\beta}{N}\left((I_{n+1})^2\left(\frac{3}{4}\right) + (I_n)^2\left(-\frac{1}{12}\right) + (I_{n+1})I_n\left(-\frac{1}{6}\right)\right) \\ -\frac{\beta\gamma}{N}\left((I_n)^2\left(\frac{\Delta t}{10}\right)(I_{n+1})I_n\left(\frac{3\Delta t}{10}\right) + (I_{n+1})^2\left(\frac{3\Delta t}{5}\right)\right) - \left(\frac{dI_{n+1}}{dt}\right). \end{aligned} \tag{3.3.41}$$

In which we take the finite element approximation of

$$\frac{dI_1}{dt} \approx \frac{1}{\Delta t}(I_2 - I_1) \quad | \quad \frac{dI_{n+1}}{dt} \approx \frac{1}{\Delta t}(I_{n+1} - I_n).$$

The rest are easily found by simply differentiating the results in remark 3.3 with respect to the correct variable. Furthermore, looking at our unknown Lagrange multiplier terms, some more results at interesting points are as follows

$$\begin{aligned} \frac{dF_1}{\lambda_0} = \frac{dF_1}{\lambda_1} &= \frac{\Delta t}{2} r\gamma, \\ \frac{dF_{(n/2)+1}}{\lambda_0} = \frac{dF_{(n/2)+1}}{\lambda_1} &= \frac{\Delta t}{2} r\gamma, \\ \frac{dF_{n+1}}{\lambda_0} = \frac{dF_{n+1}}{\lambda_1} &= \frac{\Delta t}{2} r\gamma. \end{aligned}$$

We then have that for other values of $j \neq \{1, \frac{n}{2}+1, n+1\}$,

$$\frac{dF_j}{d\lambda_0} = \frac{dF_j}{d\lambda_1} = (\Delta t)\, r\gamma. \tag{3.3.42}$$

We now have everything we need except a way of choosing an initial guess.

3.3.3 Choosing the initial finite element approximation

Choosing an excellent initial guess given the data points is not easy. If we consider that the data given (X_0, X_1) could vary in size (or not at all), choosing one method for an initial guess is definitely not simple or perhaps even sensible, since we know a Newtons method needs an initial guess close enough to the solution to converge. We now show the method derived in our attempt at giving a sensible initial guess, given any data points X_0, X_1.

The motivation for our method comes from many observations that the SIR equations exhibit exponential curves at the very beginning of a pandemic. Therefore, it makes sense that we let our initial curve be some form of an exponential curve, given in the form

$$I^0(t) = a \times e^{bt}, \quad a, b \in \mathbb{R}.$$

Since we are given two data points X_0, X_1, we can only have two parameters in our initial curve. Then we can solve for a and b, i.e.,

$$r\gamma \int_0^1 a \times e^{bt}\, dt = X_0 \quad | \quad r\gamma \int_1^2 a \times e^{bt}\, dt = X_1. \tag{3.3.43}$$

We note that at this point, the paramaters r and γ are known. Then by solving and re-arranging, we deduce that the parameters in terms of the data are given by

$$a = \frac{1}{r\gamma} \frac{(X_0)^2 \ln\left(\frac{X_1}{X_0}\right)}{X_1 - X_0} \quad | \quad b = \ln\left(\frac{X_1}{X_0}\right).$$

Then plugging these values into 3.3.43 gives our initial guess, which also satisfies the integral constraints.

$$\therefore I^0(t) = \frac{1}{r\gamma} \frac{(X_0)^2 \ln\left(\frac{X_1}{X_0}\right)}{X_1 - X_0} \times \exp\left(\ln\left(\frac{X_1}{X_0}\right) t\right). \tag{3.3.44}$$

3.3.4 Assembling the scheme

There is still a lot which has not been mentioned yet in terms of how the scheme runs and updates. So far, we have chosen an initial guess for the infectious cases (I^0), but we need to also chose an initial guess for our Lagrange multipliers λ_0^0 and λ_0^1. It is sensible to set these to 0 initially, since the algorithm will make them what it needs to be for the integral boundary conditions to be satisfied on further iterations. Convergence criteria here is specified in terms of the difference of the residuals (R_i) between iterations. Let

$$R_i := I^{i+1} - I^i \ \in \mathbb{R}^{n+1}, \quad i \geqslant 1, \tag{3.3.45}$$

where I^i is the *vector* value solution to the finite element approximation at the *i'th* iteration. Then we can use the norm of the residuals as a stopping criteria, say once $\|R_i\| < 10^{-1.5}$. As discussed in later results, more work needs to be done on choosing a sensible stopping criteria, dependant on the size of the data points given. We now mention the algorithm in which uses everything we have talked about and finalises all the details needed to use the scheme, can be found in the algorithms section found in subsection 5.2.

3.4 Finite element results

Example 3.2 (Finite element scheme - decreasing cases)**.** *Take the second example we looked at in the shooting method, example 2.3. To save us scrolling back, we reproduce the the paramaters and information given. Let* $N = 1000$, $\beta = 0.6, \gamma = 1, r = 0.75, \Delta t = 0.0025$. *The initial conditions are given as* $S(0) = 816, I(0) = 184, R(0) = 0$. *By calculating the SIR solution, specifically the infectious cases, we can calculate* X_0 *and* X_1. *These are calculated as* $X_0 = 107.3485, X_1 = 62.0702$. *Furthermore, we will set the tolerance as highlighted in the algorithm as* $10^{-1.5}$, *and max-iterations as* 100. *Then we have the following results.*

$\mathcal{L}^\infty$ *error* 0.5248 | $\mathcal{L}^2$ *error* = 0.2144.
Time till completion: 2.747331 *seconds.*
Iterations taken: 100.

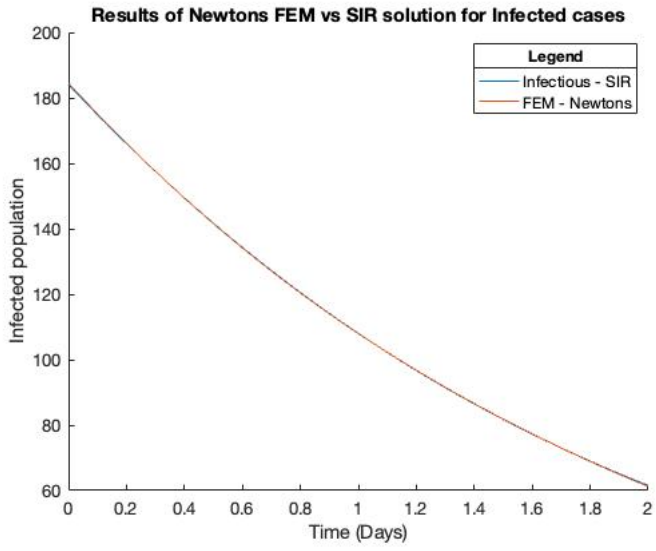


Figure 8: Finite element approximation, I(0)= 184, $\Delta t = 0.0025$ with parameters in 2.3.

We notice with our initial guess here, that with the tolerance given, we did not reach the norm of the residuals to be less than $10^{-1.5}$, *and so we reached our maximum iteration count. We notice also that since we appear to converge, but not hitting the convergence criteria, our initial guess from the data must have been close enough to the true solution. We now look at first example shown in the shooting method with increasing cases.*

Example 3.3 (Finite element scheme - increasing cases)**.** *Given the following paramaters; let* $N = 1000$, $\beta = 1.5, \gamma = 1, r = 0.75, \Delta t = 0.0025$. *The initial conditions are given as* $S(0) = 980, I(0) = 20, R(0) = 0$. *By calculating the SIR solution, specifically the infectious cases, we can calculate* X_0 *and* X_1. *These are calculated as* $X_0 = 18.9739, X_1 = 28.6179$. *Furthermore, we will set the tolerance as highlighted in the algorithm as* $10^{-1.5}$, *and max-iterations as* 100. *Then in the first plot, we will show what using our initial guess highlighted in 3.3.43.*

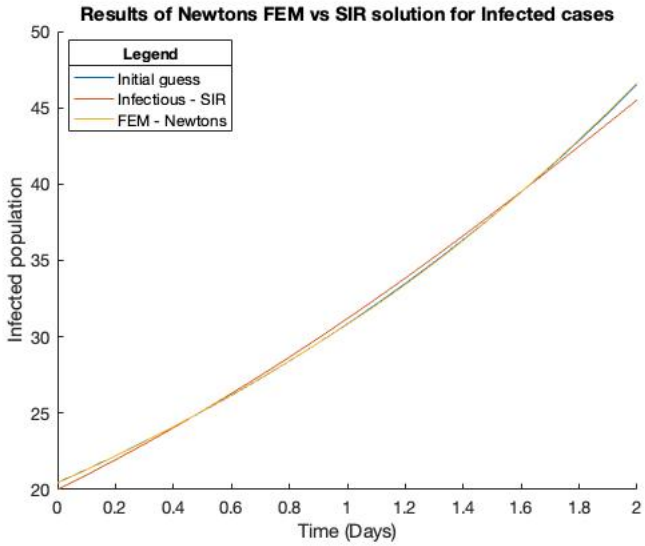


Figure 9: Finite element approximation using our derived initial guess, I(0)=20, $\Delta t = 0.0025$.

$$\mathcal{L}^\infty \textit{ error } 1.1072 \quad | \quad \mathcal{L}^2 \textit{ error} = 0.5093.$$
$$\textit{Time till completion: } 0.684001 \textit{ seconds.}$$
$$\textit{Iterations taken: } 4.$$

As we can see here, we have convergence in 4 iterations, but there are errors associated clearly. We note here, that the shape of the initial curve is definitely not good enough to converge to the solution, in comparison to the example of decreasing cases where we are very close. We now show the same example, but using a small translation (-10) *of the Infectious cases solution as our initial FEM approximation.*

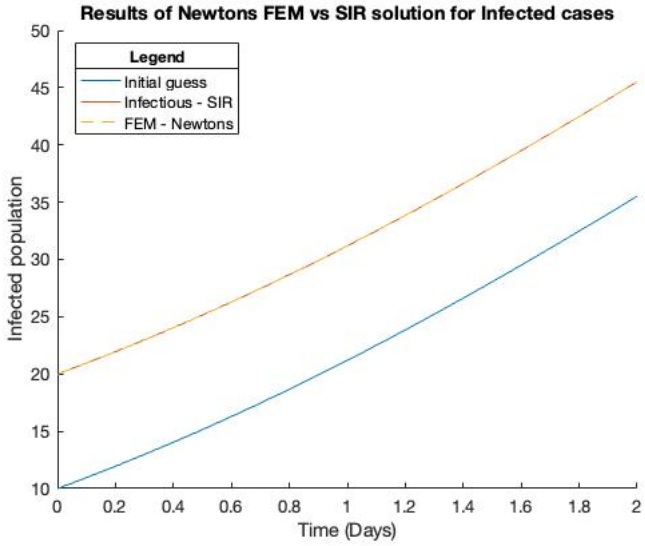


Figure 10: Finite element approximation using a translation of the solution, I(0)=20, $\Delta t = 0.0025$.

$\mathcal{L}^\infty$ *error* 0.1330 | $\mathcal{L}^2$ *error* = 0.0637.
Time till completion: 0.711791 *seconds.*
Iterations taken: 5.

We notice here even though the integral constraints from the initial guess is not satisfied initially, since the shape of the initial guess is correct, we appear to converge in this instance to the solution. This leads us to think that the shape of the initial guess is much more important then satisfying the initial integral boundary constraints, since the next iteration of the FEM approximation will have more room to move to satisfy the ODE.

4 Discussion of results

Now we have seen both the finite element method and shooting method at work in order to tackle the Observational model. We set out to see if we could utilise the finite element method to give better results without the worry of having to give initial guesses every time we looked to solve the model. We now briefly give some comments on the results.

Accuracy of the solution: From the examples conducted, we can see comparing the relevant results from the FEM and shooting method example sections that when the solution converged (in both cases), the shooting method had a significantly smaller $\mathcal{L}^2$ and $\mathcal{L}^\infty$ error norms than compared to the finite element scheme.

Computational time taken to converge: Of-course, this section will depend on the software ran and computer used in order to do the calculations. However, re-checking the sections of results, we can see there is a clear disparity in terms of time taken to converge (providing we converge). On average, the shooting method took around 0.2 seconds on average to converge using around 10-13 iterations from the examples seen. On the other hand, the FEM scheme took approximately 0.7-0.8 seconds provided it converged. We also saw the example where we did not hit the tolerance for the residuals and we were taken to 100 iterations which took the time to approximately 3 seconds.

We also note that in order to get the results we did with the FEM scheme, we had to take Δt extremely small ($\Delta t = 0.0025$), which means on every operation we are inverting a $(803)^2$ matrix every iteration, which of course is much more cost heavy than inverting the $(2)^2$ matrix from the Newton-Broyden's algorithm. By taking Δt larger in the FEM calculations, we found that the errors were larger than by taking Δt smaller.

Initial Guess/Guesses: From the results of the shooting method, we showed various different guesses for the two examples highlighted. We showed that for sensible guesses, we mostly converged, and also when the guesses were not as well chosen, there was still sometimes convergence. Using our derivation of the exponential curve from the FEM section, we saw that when this guess was close to the solution, we appear to converge to the solution. When this initial guess diverged away from the solution, primarily in shape, the Newtons scheme did not favour well to converging to the true solution.

Reliability of the results: Most importantly, we want to know given data points X_0 and X_1, without knowing the initial condition which method would be 'all-round' more suitable. Since more work needs to be done on choosing initial guesses for the finite element scheme, it is in my opinion that the shooting method is currently the preferred method for now, since the FEM scheme is extremely sensitive to the initial guess. On every choice of sensible guesses for the shooting method, we converged to the true solution - and so no guess work had to be done in terms of checking our solution. Of-course, this is limited by how good our initial guesses are, but when we converge, we know we've got the true solution.

Further work: Since the work on the FEM Newtons scheme with the two integral boundary conditions is novel, we feel more work in this area would be of great benefit. We note that in our work, we have only used linear finite element basis functions, there are of-course many other excellent choices for basis functions which will give much more accurate results, some examples are quadratic and cubic basis functions. More work is being conducted into the derivation of the Observational model in terms of expanding compartments to make the model more applicable.

5 Algorithms

5.1 Shooting method algorithm

Algorithm 1 Newton-Broyden's algorithm for solving the Observational model.

Algorithms: Runge-Kutta 4th order (RK4), Trapezium method (TM)
Input : Data: X_0, X_1 — Parameters: γ, β, r, N — Initial Guesses
Set : Tolerance, Max-iterations , $\Delta t << 1$
Output : Solution for Infectious Cases (IC)

$IC_0 \longleftarrow RK4(\phi_0, \psi_0)$
$IC_1 \longleftarrow RK4(\phi_1, \psi_1)$

for $i = 0, 1$ **do**
  $f_0^i \longleftarrow (r\gamma) \times TM(IC_0^i) - X_0$
  $f_1^i \longleftarrow (r\gamma) \times TM(IC_1^i) - X_1$
end

for $i = 0$ **do**
  for $j = 0, 1$ **do**
    $\dfrac{df_i^j}{d\phi} \longleftarrow \dfrac{f_{i+1}^j - f_i^j}{\phi_{i+1} - \phi_i}$
    $\dfrac{df_i^j}{d\psi} \longleftarrow \dfrac{f_{i+1}^j - f_i^j}{\psi_{i+1} - \psi_i}$
  end
  $\Delta F_i \longleftarrow \begin{bmatrix} f_{i+1}^0 - f_i^0 \\ f_{i+1}^1 - f_i^1 \end{bmatrix}$
  $\Delta X_i \longleftarrow \begin{bmatrix} \phi_{i+1} - \phi_i \\ \psi_{i+1} - \psi_i \end{bmatrix}$
  $J_i \longleftarrow \begin{bmatrix} \dfrac{\partial f_i^0}{\partial \phi} & \dfrac{\partial f_i^0}{\partial \psi} \\ \dfrac{\partial f_i^1}{\partial \phi} & \dfrac{\partial f_i^1}{\partial \psi} \end{bmatrix}$
end

$i \longleftarrow 0$
$\text{norm}(R^i) \longleftarrow$ Tolerance

while $((i <$ *Max-iterations*$)$ ***and*** $norm(R^i) <$ *Tolerance*$)$ **do**
  $J_{i+1} \longleftarrow J_i + \dfrac{(\Delta F_i - J_i \Delta X_i)(\Delta X_i)^T}{\|\Delta X_i\|^2}$
  $\begin{bmatrix} \phi_{i+2} \\ \psi_{i+2} \end{bmatrix} \longleftarrow \begin{bmatrix} \phi_{i+1} \\ \psi_{i+1} \end{bmatrix} - [J_{i+1}]^{-1} \begin{bmatrix} f_i^0 \\ f_i^1 \end{bmatrix}$
  $IC_{i+2} \longleftarrow RK4(\phi_{i+2}, \psi_{i+2})$
  $f_0^{i+2} \longleftarrow (r\gamma) \times TM(IC_0^{i+2}) - X_0$
  $f_1^{i+2} \longleftarrow (r\gamma) \times TM(IC_1^{i+2}) - X_1$
  $\text{norm}\left(R^{i+1}\right) \longleftarrow \sqrt{\left(f_0^{i+2}\right)^2 + \left(f_1^{i+2}\right)^2}$
  $i + 1 \longleftarrow i$
end

5.2 Finite element algorithm

Algorithm 2 Finite element algorithm for the Observational model.

Algorithms: Trapezium method (TM)
Input : Data: X_0, X_1 — Parameters: γ, β, r, N — $I^0(t) \longleftarrow$ 3.3.44
Set : Tolerance, Max-iterations , $\Delta t << 1$
Output : Solution for Infectious Cases (IC)

$\lambda_0^0 \longleftarrow 0$
$\lambda_1^0 \longleftarrow 0$

for $i = 1 \longrightarrow n+1$ **do**
  $I_i^0 \longleftarrow I^0(t_i)$
end

$k \longleftarrow 0$
$\text{norm}(R^k) \longleftarrow$ Tolerance

while $(0 < k <$ *Max-iterations* ***and*** *norm*$(R^k) <$ *Tolerance*$)$ **do**
  for $i = 1 \longrightarrow n+1$ **do**
    $F_i^k \longleftarrow$ Proposition 3.3$|^{I^k}$
    for $j = 1 \longrightarrow n+1$ **do**
      $J_{i,j}^k \longleftarrow \dfrac{dF_i^k}{dI_j}$
    end
  end
  $F_{n+2}^k \longleftarrow X_0 - (r\gamma) \times TM(I^k(t))\restriction_{(0,1)}$
  $F_{n+3}^k \longleftarrow X_1 - (r\gamma) \times TM(I^k(t))\restriction_{(1,2)}$
  for $i = 2 \longrightarrow \frac{n}{2}$ **do**
    $J_{n+2,i}^k \longleftarrow \Delta t$
    $J_{i,n+2}^k \longleftarrow (\Delta t) r\gamma$
  end
  for $i = \frac{n}{2} + 2 \longrightarrow n$ **do**
    $J_{n+3,i}^k \longleftarrow \Delta t$
    $J_{i,n+3}^k \longleftarrow (\Delta t) r\gamma$
  end
  $J_{1,n+2}^k \longleftarrow J_{(n/2)+1,n+2}^k \longleftarrow J_{(n/2)+1,n+3}^k \longleftarrow J_{n+1,n+3}^k \longleftarrow \frac{\Delta t}{2}$
  $J_{n+2,1}^k \longleftarrow J_{n+2,(n/2)+1}^k \longleftarrow J_{n+3,(n/2)+1}^k \longleftarrow J_{n+3,n+1}^k \longleftarrow \frac{\Delta t}{2} r\gamma$
  $\delta I^k = \left(J^k\right)^{-1} F^k$
  for $i = 1 \longrightarrow n+1$ **do**
    $I_i^{k+1} = I_i^k + \delta I_i^k$
    $R_i^{k+1} = |I_i^{k+1} - I_i^k|$
  end
  $\lambda_0^{k+1} \longleftarrow \lambda_0^k + I_{n+2}^{k+1}$
  $\lambda_1^{k+1} \longleftarrow \lambda_1^k + I_{n+3}^{k+1}$
  $k+1 \longleftarrow k$
end

6 Supplemental section

6.1 Quadrature - trapezium method

In order to see whether or not the values calculated by the shooting and finite element methods satisfy the integral boundary conditions, we must use some form of quadrature to estimate the integral values. The most famous and well known quadrature rules are formed as part of the Newton-Cotes [3] group, which were developed by Sir Isaac Newton and Roger Cotes. Whilst extremely interesting, we will only talk about the trapezium rule (two point closed Newton Cotes method). The group of Newton-Cotes *closed* formulas rely on taking a uniformly spaced mesh on the real line, between (and including) the bounds of integration. Let us define an arbitrary uniform partition on $[a, b]$ by taking $n \in \mathbb{N}$ (the number of nodal points we would like to have) as

$$a = x_1 < x_2 < \cdots < x_n < x_{n+1} = b$$

$$\Delta x = \frac{b-a}{n}, \quad x_i = a + (\Delta x)(i-1) \ \forall i = \{1, 2, \ldots, n, n+1\}$$

Then we can formally define the trapezium method as the following calculation

$$\int_a^b f(x)\,dx = \frac{\Delta x}{2}(f(a) + f(b)) + \Delta x \sum_{i=2}^{n} f(x_i) + O((\Delta x)^2) \tag{6.1.1}$$

$$\implies \int_a^b f(x)\,dx \approx \frac{\Delta x}{2}(f(a) + f(b)) + \Delta x \sum_{i=2}^{n} f(x_i) \tag{6.1.2}$$

Where Δx describes the distance between any two arbitrary uniformly spaced nodal points, and $O(\Delta x^2)$ denotes terms of Δx^2 and above. As suggested by the formula, we notice since 6.1.1 is order $(\Delta x)^2$, then as $\Delta x \to 0$, the trapezium approximation approaches the true solution. More closely, if we look at any Δx interval, the method approximates the area by taking the Lagrange polynomial between the values of the nodal points at either side of the interval, and hence why it is called the trapezium method (the area approximated is the shape of a trapezium).

6.2 Well-posedness example of Poisson's equation with Neumann boundary values and an integral constraint

To show Poisson's equation with Neumann boundary values is ill posed without an integral constraint, take the example

$$\begin{aligned} -\frac{d^2\omega}{dx^2} &= x, \quad | \quad x \in (0,1), \\ \frac{d\omega}{dx}(0) &= -1 \quad | \quad \frac{d\omega}{dx}(1) = -\frac{3}{2}. \end{aligned} \tag{6.2.1}$$

Then $\omega(x) = -\frac{x^3}{6} - x + \alpha$ is a solution, with $\alpha \in \mathbb{R}$, since by taking derivatives of the solution, we note that we satisfy the ODE and the boundary conditions, but α is a free parameter and hence the problem ill posed. By introducing the integral constraint

$$\int_0^1 \omega(x)\,dx = 0$$

Then this implies that $\alpha = \frac{13}{24}$.

Proof.

$$\int_0^1 -\frac{x^3}{6} - x + \alpha\,dx = \left[-\frac{x^4}{24} - \frac{x^2}{2} + \alpha x\right]\Big|_0^1 = -\frac{1}{24} - \frac{1}{2} + \alpha = 0$$

$$\implies \alpha = \frac{13}{24}$$

□

Therefore the solution is now unique after introducing an integral constraint.

7 Source Code for MATLAB Simulations

We now attach the URL where the reader can try for them selves to have a go at using the discussed numerical methods to solve for the Observational model: Link to GitHub repository.

References

[1] Campillo-Funollet E, Wragg H, Van Yperen J, Duong DL, Madzvamuse A. Reformulating the SIR model in terms of the number of COVID-19 detected cases: well-posedness of the observational model. Philosophical Transactions of the Royal Society A (to appear).

[2] Teukolsky, S.A., Flannery, B.P., Press, W.H. and Vetterling, W.T., 1992. Numerical recipes in C. SMR.

[3] Burden, R.L., Faires, J.D. and Burden, A.M., 2015. Numerical analysis. Cengage learning.

[4] Langtangen, H.P. and Mardal, K.A., 2016. Introduction to numerical methods for variational problems. University of Oslo.

[5] Senning, J.R., 2007. Computing and estimating the rate of convergence.

[6] Vianello, M. and Zanovello, R., 1992. On the superlinear convergence of the secant method. The American mathematical monthly, 99(8), pp.758-761.

[7] Venkataraman, C. Numerical solution of PDE's, lecture notes from the University of Sussex.

[8] Jost, J., Jost, J. and Li-Jost, X., 1998. Calculus of variations (Vol. 64). Cambridge University Press.