

# Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Difference between implicit and explicit methods .....</b>	<b>3</b>
<b>Method of finite difference .....</b>	<b>4</b>
Methodology .....	4
Results .....	7
General validation of numerical simulation results and exact solutions .....	7
Effect of relaxation on the final profile for different reaction orders .....	8
Effect of assumed initial profiles on final profile .....	10
<b>Shooting method .....</b>	<b>11</b>
Methodology .....	11
Results .....	15
General validation of numerical simulation results and exact solutions .....	15
Effect of different integration methods on the final profile .....	16
Effect of Thiele modulus ( $\varphi$ ) on the final profile .....	17
<b>Adam-Bashforth (AB) method .....</b>	<b>18</b>
Methodology .....	18
Results .....	19
General validation of numerical simulation results and exact solutions .....	19
Effect of different order methods on the final profile .....	20
Effect of Thiele modulus ( $\varphi$ ) on the final profile .....	21
<b>Effect of Thiele modulus on concentration profile (different reaction orders) .....</b>	<b>22</b>
<b>Conclusion .....</b>	<b>22</b>
<b>References .....</b>	<b>23</b>
<b>Appendix .....</b>	<b>24</b>
Finite difference method results .....	24
Shooting method results .....	25
Adam-Bashforth method results .....	26
Derivation of reference Thiele modulus .....	27
Variation of concentration profile with changing Thiele modulus .....	27

## 1. Introduction

Reaction-diffusion in a spherical porous catalyst can be modelled by the following ordinary differential equation (ODE).

$$\phi'' + \frac{2}{Y} \phi' - \phi^2 \phi^n = 0$$

where  $\phi$  is the non-dimensional concentration

$Y$  is the non-dimensional radius

$\Phi$  is Thiele modulus ( $\Phi = R \sqrt{\frac{kC_{AS}^{n-1}}{D_e}}$ )

$n$  is the reaction order of the chemical reaction taking place in the porous catalyst

At present, this ODE can be solved exactly only for reaction orders 1 and 5. Their exact solutions are as follows.

$$\phi = \frac{\sinh(\phi_1 Y)}{Y \cdot \sinh(\phi_1)} \quad (n = 1)$$

$$\phi = \sqrt{2} \left[ 1 + Y^2 + \left( 1 + \frac{4}{3} \phi_5^2 \right)^{1/2} \cdot (1 - Y^2) \right]^{-1/2} \quad (n = 5)$$

For other reaction orders, this ODE has to be still solved numerically in order to obtain the concentration profile. Hence, the objective of this project is to explore some promising numerical methods and determine the numerical method which is able to provide the most accurate results for a wide range of reaction and diffusion conditions. Particularly, since exact solutions are already available for reaction orders 1 and 5, the numerical method which is able to reproduce the exact solutions most accurately will be concluded as the best method for solving the ODE. The 3 numerical methods that will be explored in this project are namely method of finite difference, shooting method and Adam-Bashforth method.

## 2. Difference between implicit and explicit methods

When solving a problem numerically, one is often faced with the choice between explicit and implicit methods. In general, explicit methods involve obtaining a new solution based on existing information. For example, in time-dependent problems, the future state of a system can be predicted explicitly based on its current and/or past states. This makes explicit methods simple to implement but they can be unstable if too large a step is taken since there is no guarantee that the predicted future state will be a meaningful one. In contrast, implicit methods take into account the unknown future state in the solution process and, therefore, provide some guarantee that the future state is realizable. In other words, there is some in-built stability to implicit methods. This allows the use of much larger steps, but they are generally harder to implement and more expensive compared to explicit methods. Depending on the nature of the problem, one approach could be more suitable and efficient than the other.

### 3. Method of finite difference

#### 3.1 Methodology

$$\phi'' + \frac{2}{Y} \phi' - \phi^2 \phi^n = 0 \quad (1)$$

Given that this is a non-linear ordinary differential equation (ODE) and method of finite difference is an implicit method which provides exact results only for linear ODE, this non-linear equation has to be transformed to the following linear form.

$$\phi'' + \frac{2}{Y} \phi' - \phi^2 \phi_0^{n-1} \phi = 0 \quad (2)$$

If  $n = 1$ , Eq. 2 becomes a linear ODE which can be solved exactly for  $\phi$  by method of finite difference. However, in order to solve Eq. 1 for any arbitrary order  $n$ , an iterative version of finite difference method has to be adopted. The procedure for this iterative finite difference method is described below.

1. Discretize  $Y$  (non-dimensional radius) with a step size  $\Delta\tau$  over the range 0 to 1 (inclusive). In this analysis,  $\Delta\tau$  will be taken to be 0.01. The boundary conditions at  $Y = 0$  and  $Y = 1$  will be enforced at  $Y = 0.01$  and  $Y = 0.99$  respectively.
2. Assume an initial profile for  $\phi$  (e.g.  $\phi = Y^2$ ).
3. Approximate the derivatives in Eq. 2 with finite difference formulas for each of the discretized  $Y$  values.

#### Case 1: $Y = 0.01$

$$\phi'_{Y=0} = 0 \quad (\text{Boundary condition 1})$$

The boundary condition at  $Y = 0$  is approximated using 3 points forward approximation.

$$\phi'_{Y=0} = \frac{-3\phi_{Y=0} + 4\phi_{Y=0.01} - \phi_{Y=0.02}}{2\Delta\tau} = 0 \quad (3)$$

Rearranging Eq. 3,

$$\phi_{Y=0} = \frac{\phi_{Y=0.02} - 4\phi_{Y=0.01}}{-3} \quad (4)$$

Approximating Eq. 2 with finite difference formulas,

$$\frac{\phi_{Y=0.02} - 2\phi_{Y=0.01} + \phi_{Y=0}}{(\Delta\tau)^2} + \frac{2}{Y} \frac{\phi_{Y=0.02} - \phi_{Y=0}}{2\Delta\tau} - \varphi^2 \phi_0^{n-1} \phi_{Y=0.01} = 0 \quad (5)$$

where  $\phi_0^{n-1}$  is a constant since  $\phi_0$  represents the solution at  $Y = 0.01$  from the previous iteration

Substituting Eq. 4 into Eq. 5,

$$\frac{\phi_{Y=0.02} - 2\phi_{Y=0.01} + \frac{\phi_{Y=0.02} - 4\phi_{Y=0.01}}{-3}}{(\Delta\tau)^2} + \frac{2}{Y} \frac{\phi_{Y=0.02} - \frac{\phi_{Y=0.02} - 4\phi_{Y=0.01}}{-3}}{2\Delta\tau} - \varphi^2 \phi_0^{n-1} \phi_{Y=0.01} = 0 \quad (6)$$

Rearranging Eq.6,

$$\left( \frac{-2}{3(\Delta\tau)^2} - \frac{4}{3Y\Delta\tau} - \varphi^2 \phi_0^{n-1} \right) \phi_{Y=0.01} + \left( \frac{2}{3(\Delta\tau)^2} + \frac{4}{3Y\Delta\tau} \right) \phi_{Y=0.02} = 0 \quad (7)$$

### Case 2: $0.01 < Y < 0.99$

For each of the discretized  $Y$  value in the range 0.01 to 0.99 (exclusive),

$$\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{(\Delta\tau)^2} + \frac{2}{Y} \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta\tau} - \varphi^2 \phi_{i0}^{n-1} \phi_i = 0 \quad (8)$$

where  $\phi_{i0}^{n-1}$  is a constant since  $\phi_{i0}$  represents  $\phi_i$  solved from the previous iteration

Rearranging Eq. 8,

$$\left( \frac{1}{(\Delta\tau)^2} - \frac{1}{Y\Delta\tau} \right) \phi_{i-1} + \left( \frac{-2}{(\Delta\tau)^2} - \varphi^2 \phi_{i0}^{n-1} \right) \phi_i + \left( \frac{1}{(\Delta\tau)^2} + \frac{1}{Y\Delta\tau} \right) \phi_{i+1} = 0 \quad (9)$$

### Case 3: $Y = 0.99$

$$\phi_{Y=1} = 1 \quad (\text{Boundary condition 2})$$

Approximating Eq. 2 with finite difference formulas,

$$\frac{\phi_{Y=0.98} - 2\phi_{Y=0.99} + \phi_{Y=1}}{(\Delta\tau)^2} + \frac{2}{Y} \frac{\phi_{Y=1} - \phi_{Y=0.98}}{2\Delta\tau} - \varphi^2 \phi_0^{n-1} \phi_{Y=0.99} = 0 \quad (10)$$

where  $\phi_0^{n-1}$  is a constant since  $\phi_0$  represents the solution at  $Y = 0.99$  from previous iteration

Substituting boundary condition 2 into Eq. 10,

$$\left(\frac{1}{(\Delta\tau)^2} - \frac{1}{Y\Delta\tau}\right)\phi_{Y=0.98} + \left(\frac{-2}{(\Delta\tau)^2} - \phi^2\phi_0^{n-1}\right)\phi_{Y=0.99} = \frac{-1}{(\Delta\tau)^2} - \frac{1}{Y\Delta\tau} \quad (11)$$

4. Using equations 7, 9 & 11, create a system of linear equations and solve for X vector iteratively until convergence.

$$\begin{array}{l}
 \text{for eqn 1} \rightarrow \\
 \text{for eqn i} \rightarrow \\
 \text{for eqn k1} \rightarrow
 \end{array}
 \begin{array}{c}
 \begin{array}{cccccc}
 \text{for } \phi_{Y=0.01} & & \text{for } \phi(i-1) & \text{for } \phi(i) & \text{for } \phi(i+1) & \text{for } \phi_{Y=0.99} \\
 \downarrow & & \downarrow & \downarrow & \downarrow & \downarrow
 \end{array} \\
 \left[ \begin{array}{cccccc}
 \dots & & & & & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \dots & & & & & \dots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 \dots & & & & & \dots
 \end{array} \right]
 \begin{array}{c}
 \left[ \begin{array}{c}
 \phi_{Y=0.01} \\
 \vdots \\
 \phi(i-1) \\
 \phi(i) \\
 \phi(i+1) \\
 \vdots \\
 \phi_{Y=0.99}
 \end{array} \right]
 \end{array}
 \end{array}
 =
 \begin{array}{c}
 \left[ \begin{array}{c}
 0 \\
 \vdots \\
 0 \\
 \vdots \\
 \frac{-1}{(\Delta\tau)^2} - \frac{1}{Y\Delta\tau}
 \end{array} \right]
 \end{array}$$

**A**
**X**
**b**

where X is a vector containing variables  $\phi$

A is a matrix system containing the coefficients of  $\phi$  in each equation

b is a vector containing constants

For  $n > 1$ , the solution has to be obtained through an iterative procedure defined above. Given the vector  $\phi^{q-1}$  at the  $(q-1)^{\text{th}}$  iteration, the vector  $\phi^q$  at the next iteration is obtained by solving the following system of linear equation.

$$A(\phi^{q-1}) \cdot \phi^q = b(\phi^{q-1})$$

\* Note that this expression is a different form of the same matrix system shown earlier.

However, since the solution at the  $(q-1)^{\text{th}}$  iteration is not exact, solving the above system of linear system results in an error at the qth iteration. Therefore, the solution  $\phi^q$  can be written as  $\phi^q = \phi^{\text{exact}} + \epsilon^n$ . The iterative procedure converges only if the error term  $\epsilon^n$  decreases as q increases. It was observed that performing the iterations without any relaxation led to divergence.

In this numerical simulation (shown below), the solution could not converge even after 1500 iterations. Intuitively, this can be seen as the effect of making large changes to the solution which causes the error to grow unbounded. Hence, to stabilize the procedure, the solution was under-relaxed at each iteration to limit the change in the solution from one iteration to the next. This proved to be an essential step in achieving convergence. The following figure shows the importance of relaxation for the ODE to converge.

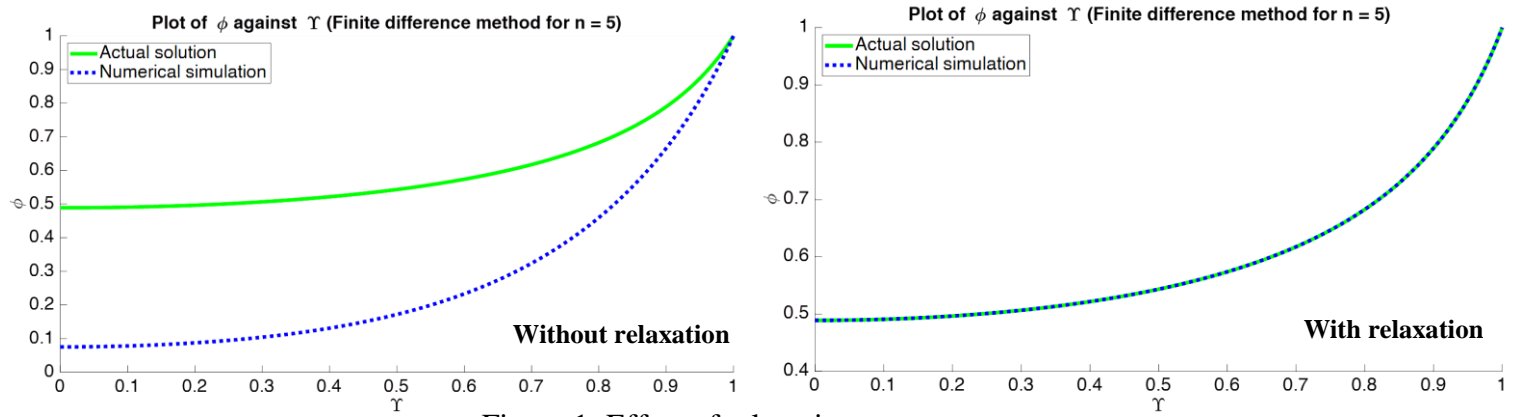


Figure 1: Effect of relaxation on convergence

Therefore, an additional step for relaxation was included in the iterative procedure as follows after step 4.

$$\phi_{\text{new}}^q = (1 - r)\phi^{q-1} + r\phi_{\text{old}}^q \quad (13)$$

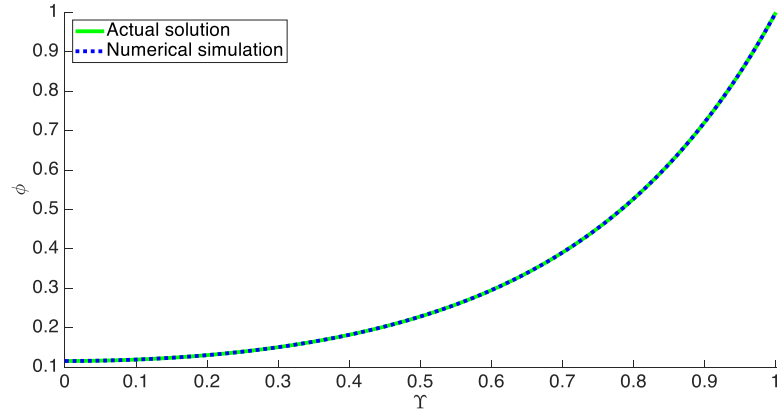
where  $r$  is the relaxation parameter such that  $0 \leq r \leq 1$ .

## 3.2 Results

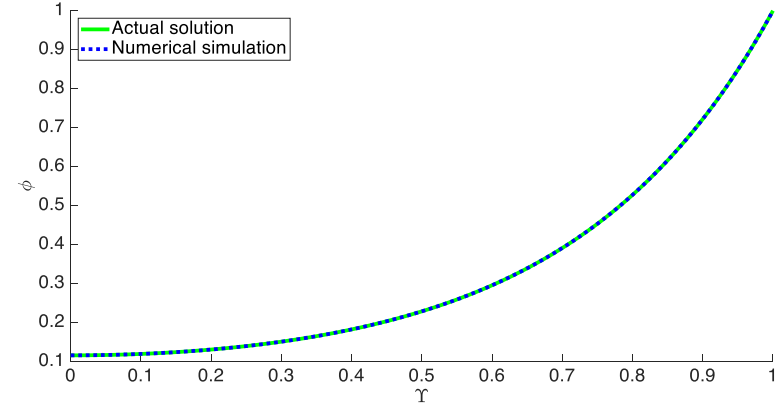
### 3.2.1 General validation of numerical simulation results and exact solutions

From Appendix section 9.1, it can be seen that the simulation results obtained from finite difference method are generally in agreement with the exact solutions for reaction orders 1 and 5. Further analysis will be done to validate if this numerical method is stable under different conditions (i.e. different relaxation values, different initial profiles & different  $q$  values).

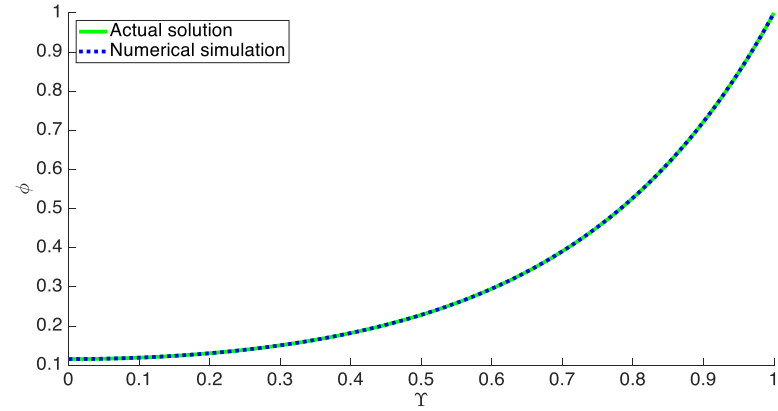
### 3.2.2 Effect of relaxation on the final profile for different reaction orders



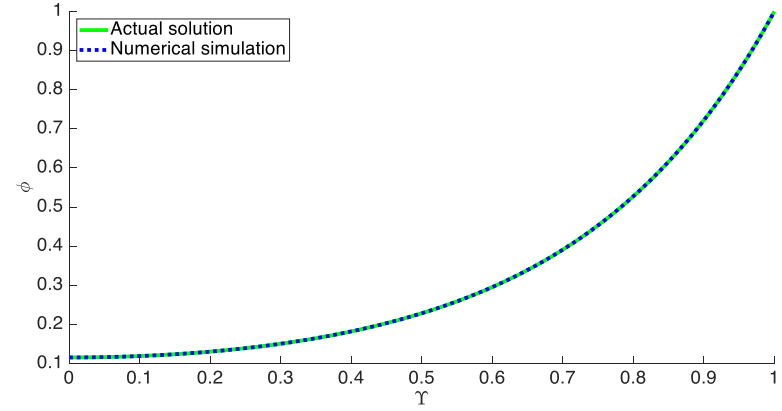
**Relaxation = 0** (Converges in 1 iteration & elapsed time = 0.19 s)



**Relaxation = 0.3** (Converges in 14 iterations & elapsed time = 0.36 s)



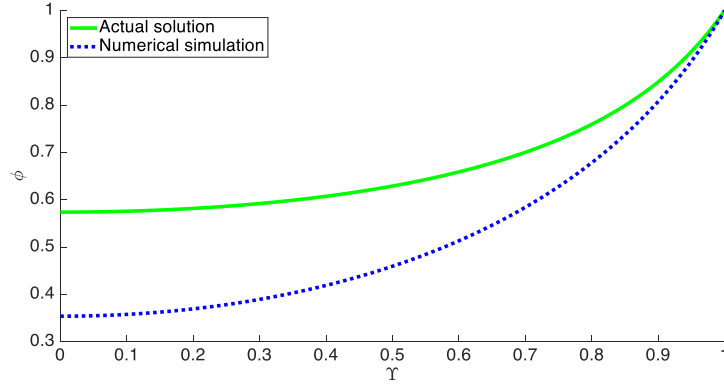
**Relaxation = 0.6** (Converges in 32 iterations & elapsed time = 0.55 s)



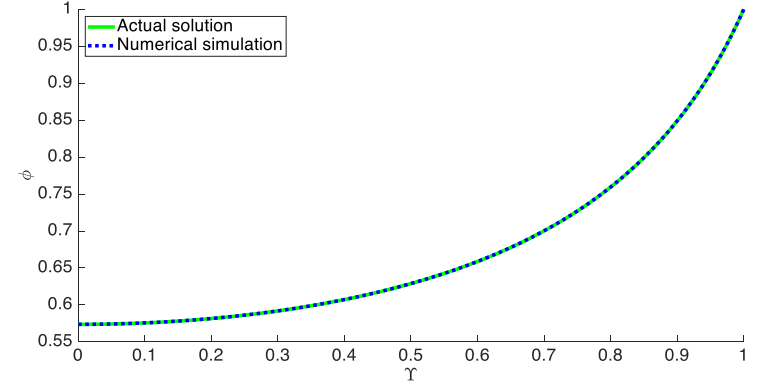
**Relaxation = 0.9** (Converges in 138 iterations & elapsed time = 1.89 s)

Figure 2: Different relaxation values (reaction order  $n = 1$ , step size = 0.001 &  $\varphi = 4.31$ )

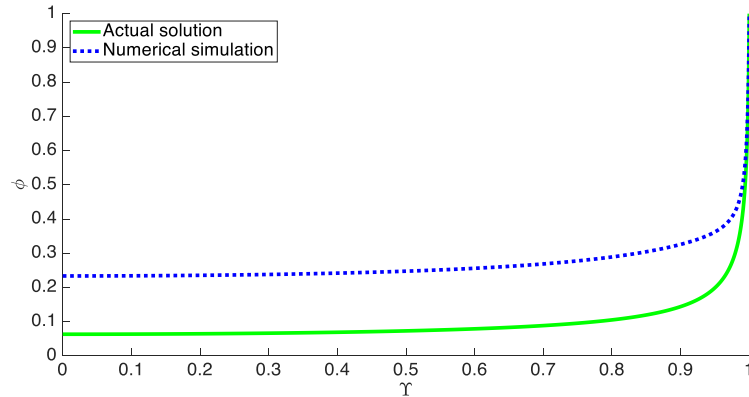
Relaxation values does not affect the convergence of the numerical simulation for reactions with order 1 as the matrix system does not contain any non-linear terms since  $n = 1$ . Hence, 1 iteration is sufficient for the system of equations to converge.



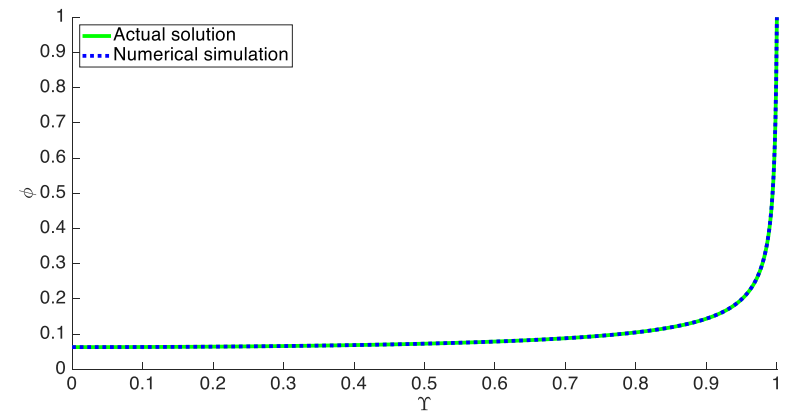
**Relaxation = 0 &  $\varphi = 4.31$**  (Did not converge even after 1500 iterations)



**Relaxation = 0.1 &  $\varphi = 4.31$**  (Converges in 21 iterations & elapsed time = 0.4 s)



**Relaxation = 0.3 &  $\varphi = 430.95$**  (Did not converge even after 1500 iterations)



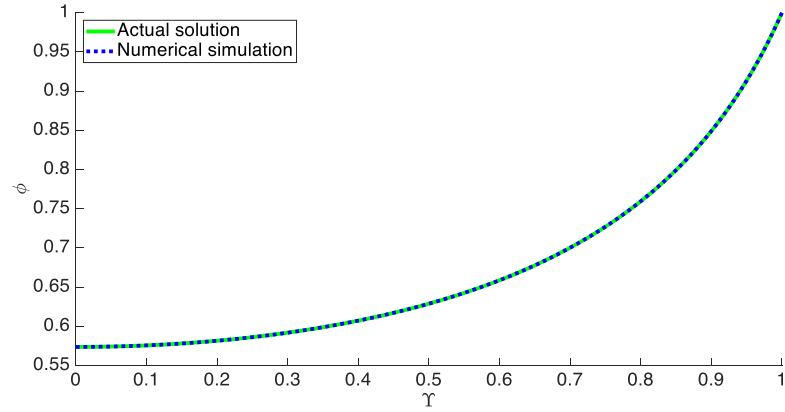
**Relaxation = 0.9 &  $\varphi = 430.95$**  (Converges in 111 iterations & elapsed time = 1.55 s)

Figure 3: Different relaxation values (reaction order  $n = 5$  & step size = 0.001)

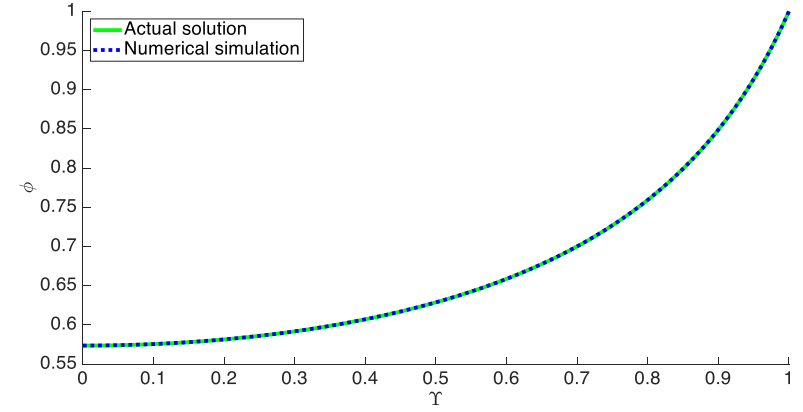
Relaxation values do affect the convergence of the numerical simulation for reactions with order 5 as the matrix system contains a non-linear term. Hence, iteration becomes necessary. Particularly, as Thiele modulus ( $\varphi$ ) becomes larger, a larger relaxation factor is required for the system to converge. Hence, the conclusion is to set the relaxation factor to a high value (i.e. 0.9) so that the method converges for all cases.



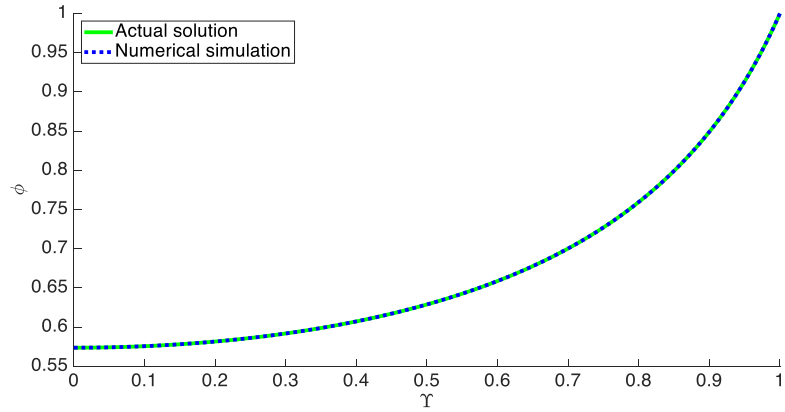
### 3.2.3 Effect of assumed initial profiles on final profile



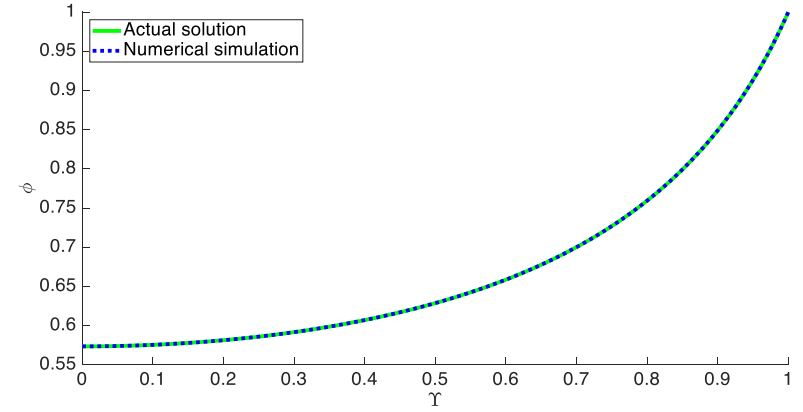
$\phi = Y^2$  (Converges in 98 iterations & elapsed time = 1.43 s)



$\phi = 1$  (Converges in 99 iterations & elapsed time = 1.45 s)



$\phi = Y$  (Converges in 123 iterations & elapsed time = 1.72 s)



$\phi = 0$  (Converges in 137 iterations & elapsed time = 1.93 s)

Figure 4: Different initial profiles (reaction order  $n = 5$ ,  $\varphi = 4.31$ , step size = 0.001 & relaxation = 0.9)

Choice of different initial profiles does not determine if the system of equations will converge or not. However, it can be inferred that the closer the initial profile resembles the final profile, the faster the rate of convergence (i.e. lesser iterations).

## 4. Shooting method

### 4.1 Methodology

Shooting method is an explicit method used for solving a boundary value problem by reducing it to an initial value problem. In essence, we 'shoot' out trajectories in different directions until a trajectory with the desired boundary value is found.

The original differential equation is:

$$\phi'' + \frac{2}{Y} \phi' = \varphi_n^2 \phi^n \quad (1)$$

With boundary conditions:

1. At  $Y = 1, \phi = 1$
2. At  $Y = 0, \phi' = \frac{d\phi}{dY} = 0$  ( $\phi \geq 0$ )

Let  $Z = \phi'$ , Eq. 1 can be transformed to:

$$Z' + \frac{2}{Y} Z = \varphi_n^2 \phi^n \quad (2)$$

Hence, the original 2<sup>nd</sup> order ordinary differential equation can be transformed into a system of two 1<sup>st</sup> order differential equations as follows.

$$\begin{cases} Z' = \varphi_n^2 \phi^n - \frac{2}{Y} Z \\ \phi' = Z \end{cases}$$

Since there is no complete information for value of  $Z$  and  $\phi$  at either ends ( $Y = 1$  and  $Y = 0$ ), this problem is a boundary value problem. Hence, shooting method is employed to solve this system of ordinary differential equations.

The initial condition used is:

$$\begin{aligned} \text{At } Y = 0: \quad & Z = \phi' = 0 \\ & \phi \geq 0 \end{aligned}$$

Exact value of  $\phi$  at  $Y = 0$  is not given. However, given that  $0 \leq \phi \leq 1$ , different initial values of  $\phi$  in the range of  $[0,1]$  will be tried to calculate the value of  $\phi$  at  $Y = 1$  until  $\phi = 1$  at  $Y = 1$  is satisfied (the other boundary is satisfied).

Before using MATLAB, the above system of two 1<sup>st</sup> order differential equations is transformed as shown below.

$$y = \begin{bmatrix} y(1) \\ y(2) \end{bmatrix} = \begin{bmatrix} \phi \\ Z \end{bmatrix}$$

$$dydt = \begin{bmatrix} \phi' \\ z' \end{bmatrix} = \begin{bmatrix} y(2) \\ \phi_n^2 y(1)^n - \frac{2}{Y} y(2) \end{bmatrix}$$

Euler's method, 2<sup>nd</sup> order Runge-Kutta Method (RK2) and 4<sup>th</sup> order Runge-Kutta Method are used to solve the above differential equation.

### Euler's method

Euler's method is a numerical method to solve a 1<sup>st</sup> order differential equation with a given initial condition. It is the most basic explicit method for numerical integration of differential equations.

For a differential equation  $\frac{dy}{dx} = f(x, y)$  with an initial condition  $y(a)$ , Euler's method assumes the solution is expressed in the form of a Taylor's Series.

$$y(x + h) \approx y(x) + hy'(x) + \frac{h^2 y''(x)}{2!} + \frac{h^3 y'''(x)}{3!} + \dots$$

A reasonably accurate approximation can be obtained if a small step size of  $x$ , namely  $h$  is chosen and sufficiently large number of terms are taken.

For Euler's method, only first 2 terms are used, and the rest of the terms are truncated.

$$y(x + h) \approx y(x) + hy'(x)$$

Since  $y'(x) = f(x, y)$ ,

$$y(x + h) \approx y(x) + hf(x, y)$$

This method can be graphically expressed as follows. Suppose  $\frac{dy}{dt} = yt^2$  with initial condition  $y = 1$  when  $t = 1$  is given, let the graph of  $\frac{dy}{dt}$  vs  $t$  be as follows:

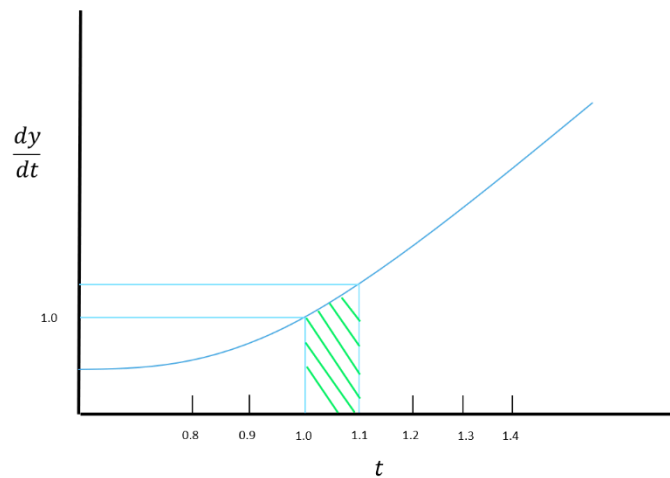


Figure 5: Illustration of integration

Since initial condition of  $y(1) = 1$  is given,  $\frac{dy}{dt}$  at  $t = 1$  can then be calculated as  $\frac{dy}{dt} = yt^2 = 1$ . It can be found that, the area under the graph of  $\frac{dy}{dt}$  vs  $t$ , within the interval between  $t = 1$  and  $t = 1.1$  can be calculated as:

$$\text{Area} = I = \int_{1.0}^{1.1} \frac{dy}{dt} dt = y(1.1) - y(1)$$

Hence,  $y$  at  $t = 1.1$  can be calculated as follows.

$$y(1.1) = y(1) + I$$

Since the area under the curve  $I$  cannot be determined exactly, Euler's method uses the rectangular area  $A1$  to approximate the area  $I$  as shown below.

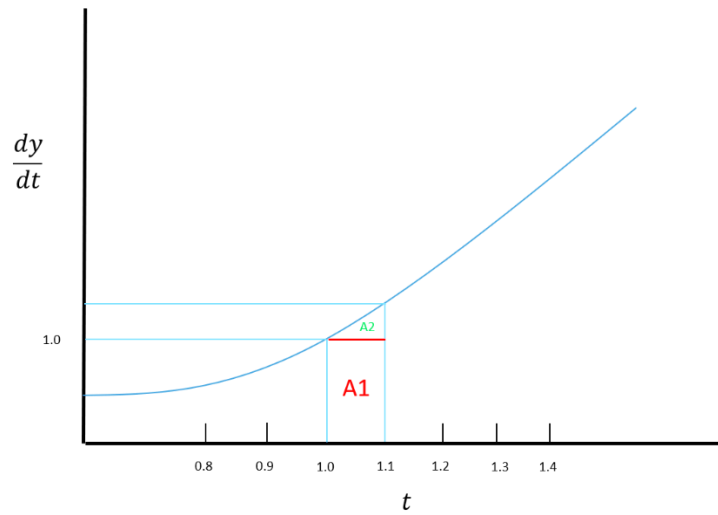


Figure 6: Illustration of Euler's method

In this case, area  $A2$  is the error incurred as a result of the approximation. With the approximated area, value of  $y$  at  $t = 1.1$  is then calculated. Subsequently, value of  $\frac{dy}{dx}$  at  $t = 1.1$  can be obtained using the original differential equation. This entire process is then repeated for the entire range of  $t$ .

## 2<sup>nd</sup> order Runge-Kutta Method (RK2 Method)

RK2 method is an improved version of Euler's method. The basic idea of RK2 method is to use Euler's method as an approximation first and then refine the area under the curve using trapezoidal rule.

The same example is used as an illustration as shown below.

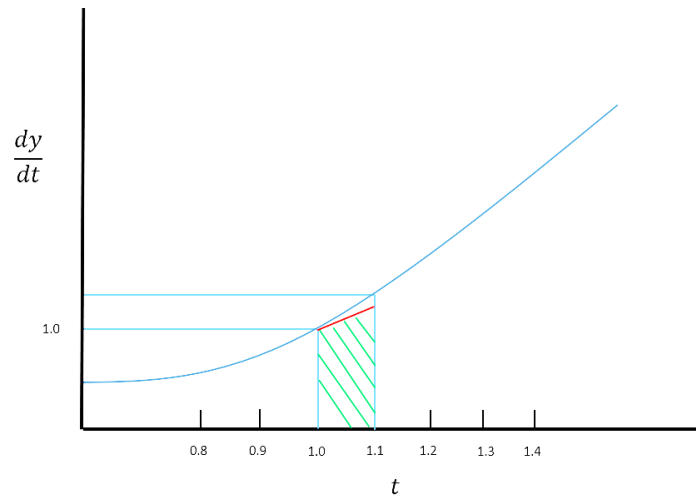


Figure 7: Illustration of RK2 method

$y$  at  $t = 1.1$  is calculated using Euler's method and value of  $\frac{dy}{dx}$  at  $t = 1.1$  is obtained using the original equation  $\frac{dy}{dt} = yt^2$ . Hence, the area under the curve of  $\frac{dy}{dt}$  vs  $t$ , within the interval between  $t = 1$  and  $t = 1.1$  is further refined using trapezoidal rule which is a more accurate approximation than a rectangle.

With the refined area known,  $y$  and  $\frac{dy}{dt}$  values at  $t = 1.1$  can then be refined using the same procedures illustrated previously.

#### 4<sup>th</sup> order Runge-Kutta Method (RK4 Method)

RK4 method is another improved version of Euler's method. The basic idea of RK4 method is using Euler's method as an approximation first and then refine the area under the curve using 1/3 Simpson's rule. The same example is used as an illustration as shown below.

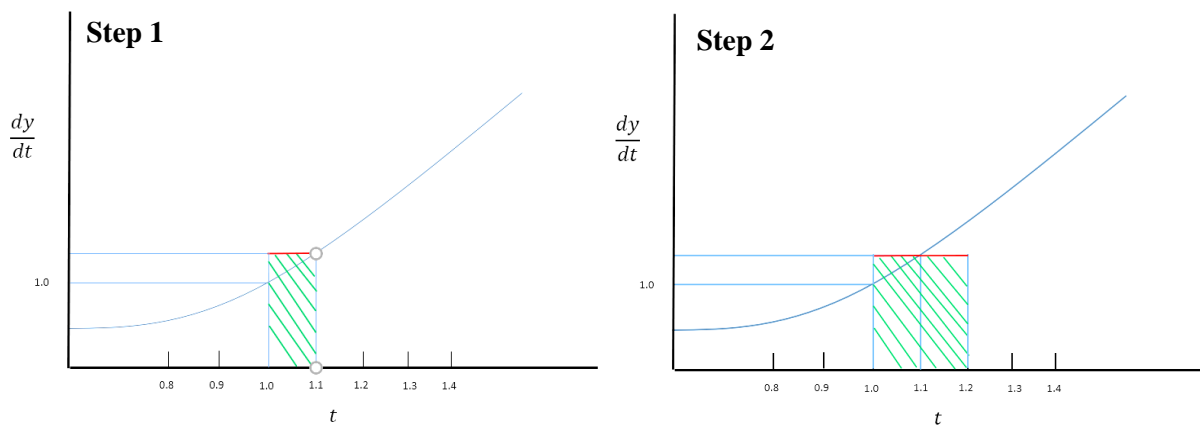


Figure 8: Illustration of RK4 method

RK4 method is explained in the following steps.

1.  $\frac{dy}{dt}$  at  $t = 1$  can be calculated using differential equation since  $y$  at  $t = 1$  is given. Denote this  $\frac{dy}{dt}$  value as dydt1 (step 1).
2.  $y$  at  $t = 1.1$  is calculated using Euler's method first, hence  $\frac{dy}{dt}$  at  $t = 1.1$  is known. Denote this  $\frac{dy}{dt}$  value as dydt2.1.
3. Next, dydt2.1 is used as the new height of the rectangle to refine the area of the rectangle as shown above. New value of  $y$  at  $t = 1.1$  is then calculated using this new area via Euler's method and new values of  $\frac{dy}{dt}$  at  $t = 1.1$  can then be obtained. Denote this  $\frac{dy}{dt}$  value as dydt2.2
4. dydt2.2 is used as the height of the rectangle and distance between  $t = 1.1$  and  $t = 1.2$  is used as the base to calculate the area of a new rectangle (step 2).
5. With the area of this new rectangle known,  $y$  at  $t = 1.2$  is then calculated using Euler's method and values of  $\frac{dy}{dt}$  at  $t = 1.2$  can be obtained. Denote this  $\frac{dy}{dt}$  value as dydt3
6. Average value of dydt2.1 and dydt2.2 is taken, denoted as dydt2
7. Area under the curve of  $\frac{dy}{dt}$  vs  $t$ , within the interval between  $t = 1$  and  $t = 1.2$  is approximated using 1/3 Simpson's rule:

$$\text{Area} = I = \frac{h}{6} \times (\text{dydt1} + 4 \times \text{dydt2} + \text{dydt3})$$

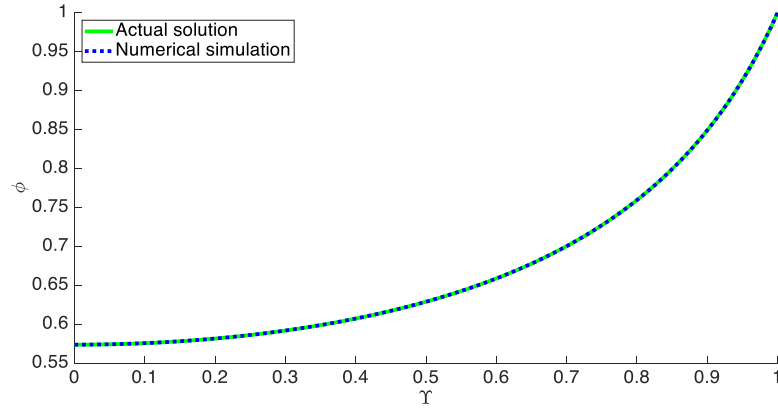
where  $h$  is twice the step size of  $t$  as 3 points are used in RK4 method.

## 4.2 Results

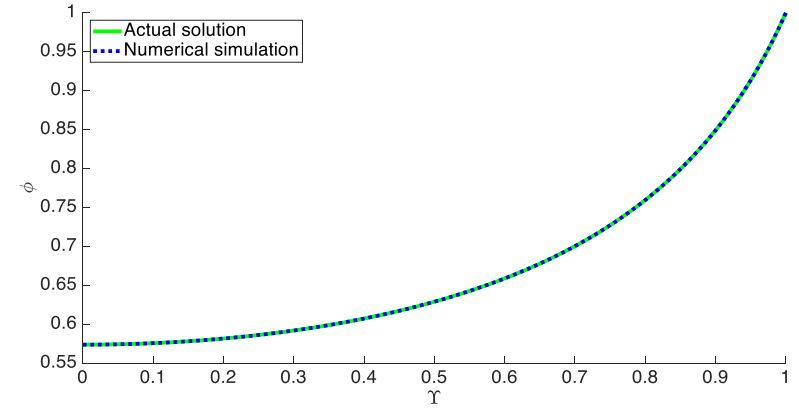
### 4.2.1 General validation of numerical simulation results and exact solutions

From Appendix section 9.2, it can be seen that the simulation results obtained from shooting method are generally in agreement with the exact solutions for reaction orders 1 and 5. Further analysis will be done to validate if this numerical method is stable under different conditions (i.e. different integration methods and  $\varphi$  values).

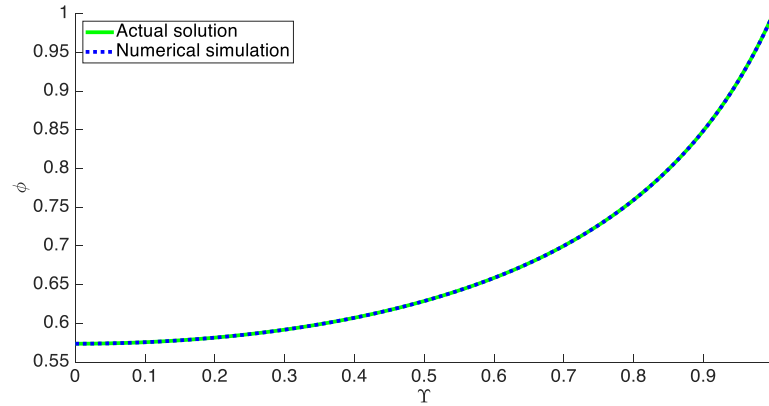
#### 4.2.2 Effect of different integration methods on the final profile



**Euler's method** (Error = 0.0083 & elapsed time = 18.6 s)



**RK2 method** (Error = 0.0017 & elapsed time = 31.6 s)

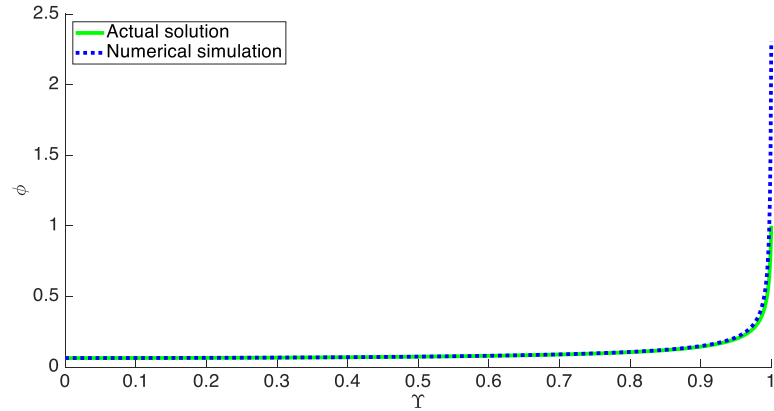


**RK4 method** (Error = 0.0017 & elapsed time = 44.8 s)

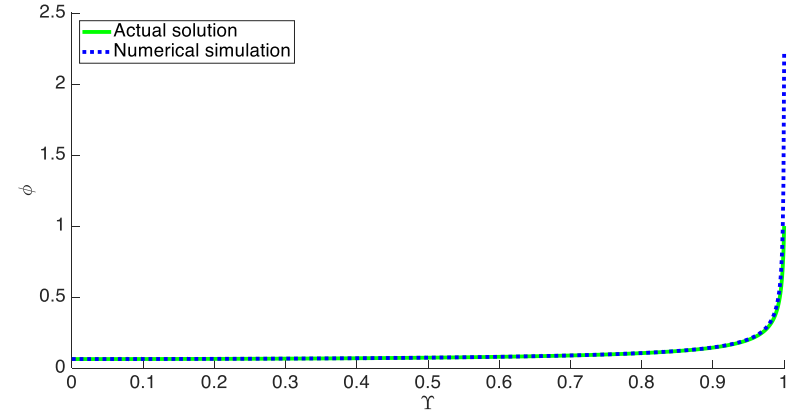
Figure 9: Different integration methods (reaction order  $n = 5$ , step size = 0.001 &  $\varphi = 4.31$ )

The error was computed as the norm of the difference between the calculated and actual profile. All 3 methods produced highly accurate results as inferred from the very low error. However, Euler's method is significantly faster than RK2 and RK4 in converging. This is probably because the arithmetic operations involved in Euler's method is less complex than RK2 & RK4 methods as seen in the methodology section.

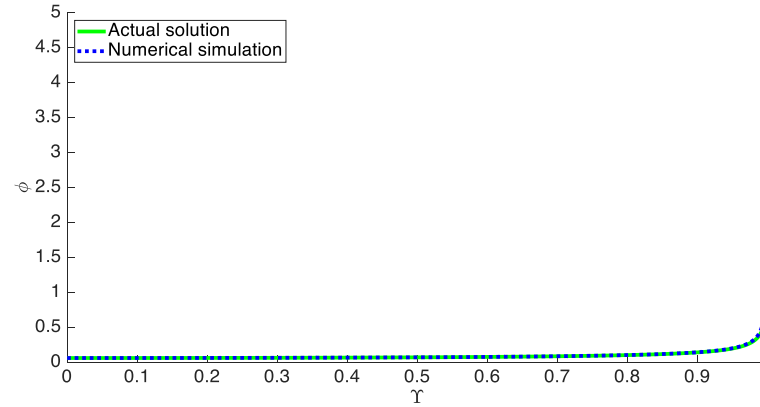
### 4.2.3 Effect of Thiele modulus ( $\varphi$ ) on the final profile



**Euler's method** (Error = 1.67 & elapsed time = 1.60 s)



**RK2 method** (Error = 1.3809 & elapsed time = 2.78 s)



**RK4 method** (Error = 4.031 & elapsed time = 5.06 s)

Figure 10: Different integration methods (reaction order  $n = 5$ , step size = 0.001 &  $\varphi = 430.95$ )

For very large Thiele modulus ( $\varphi = 430.95$ ), all 3 methods are not able to converge accurately. The step size used was the same as finite difference method. However, finite difference method converged faster and produced accurate results for the same parameters. Hence, finite difference method is superior to shooting method.



## 5. Adam-Bashforth (AB) method

### 5.1 Methodology

Adam-Bashforth is an linear multi-step explicit numerical method which utilizes the information from previous steps to approximate the function for the next step.

Let us consider the derivation of the second order Adam-Bashforth method for a first-order ordinary differential equation (ODE) of the following form.

$$\frac{dy}{dt} = f(y, t) \quad (1)$$

Integrating the ODE with respect to time from the  $n$ th time step to the  $(n + 1)$ th time step results in the following.

$$y^{n+1} = y^n + \int_{t^n}^{t^{n+1}} f(y, t) dt \quad (2)$$

If a suitable approximation to the integral on the right-hand side could be derived, it can be used to obtain  $y^{n+1}$  from  $y^n$ . To this end, consider a linear approximation of the function  $f(y, t)$ ,

$$f(y, t) \approx f(y^n, t^n) + \alpha \cdot (t - t^n) \quad (3)$$

where  $\alpha$  is the slope of the line. This slope could be approximated using the backward finite difference formula given below.

$$\alpha \approx \frac{f(y^n, t^n) - f(y^{n-1}, t^{n-1})}{h} \quad (4)$$

This is schematically shown in Figure 11. Next, the linear approximation of  $f(y, t)$  from Eq. (3) is substituted into Eq. (1) and integrated as follows. Note that the variable substitution  $\tilde{t} = t - t^n$  is used.

$$\begin{aligned} y^{n+1} &= y^n + \int_{t^n}^{t^{n+1}} f(y^n, t^n) + \frac{f(y^n, t^n) - f(y^{n-1}, t^{n-1})}{h} \cdot (t - t^n) dt \\ &= y^n + \int_0^h f(y^n, t^n) + \frac{f(y^n, t^n) - f(y^{n-1}, t^{n-1})}{h} \cdot \tilde{t} d\tilde{t} \\ &= y^n + \left[ f(y^n, t^n) \tilde{t} + \frac{f(y^n, t^n) - f(y^{n-1}, t^{n-1})}{h} \frac{\tilde{t}^2}{2} \right]_0^h \\ &= y^n + h \left[ \frac{3}{2} f(y^n, t^n) - \frac{1}{2} f(y^{n-1}, t^{n-1}) \right] \end{aligned} \quad (5)$$

The final expression of Eq. (5) is the second-order Adams-Bashforth formula.

In order to obtain higher order formulas, the function  $f(y, t)$  has to be approximated using higher order polynomials (quadratic, cubic, etc.). Since higher order polynomials contain

more coefficients, additional values of  $f(y, t)$  from past iterations are required to determine the unknown coefficient using finite difference. For instance, for a quadratic polynomial,  $f(y^{n-2}, t^{n-2})$  will be required in addition to  $f(y^n, t^n)$  and  $(y^{n-1}, t^{n-1})$ .

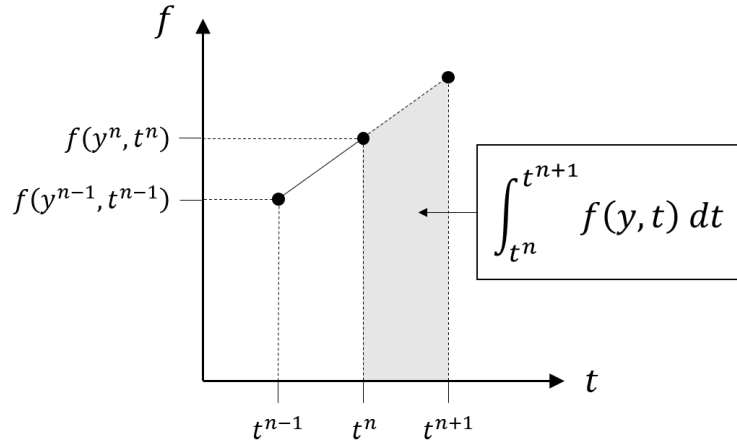


Figure 11: Adam-Bashforth illustration

The Adam-Bashforth methods for orders 1, 2, 3 and 4 are as follows.

$$y^{n+1} = y^n + hf(y^n, t^n) \text{ (Same as Euler's method)}$$

$$y^{n+1} = y^n + h \left[ \frac{3}{2}f(y^n, t^n) - \frac{1}{2}f(y^{n-1}, t^{n-1}) \right]$$

$$y^{n+1} = y^n + h \left[ \frac{23}{12}f(y^n, t^n) - \frac{16}{12}f(y^{n-1}, t^{n-1}) + \frac{5}{12}f(y^{n-2}, t^{n-2}) \right]$$

$$y^{n+1} = y^n + h \left[ \frac{55}{24}f(y^n, t^n) - \frac{59}{24}f(y^{n-1}, t^{n-1}) + \frac{37}{24}f(y^{n-2}, t^{n-2}) - \frac{9}{24}f(y^{n-3}, t^{n-3}) \right]$$

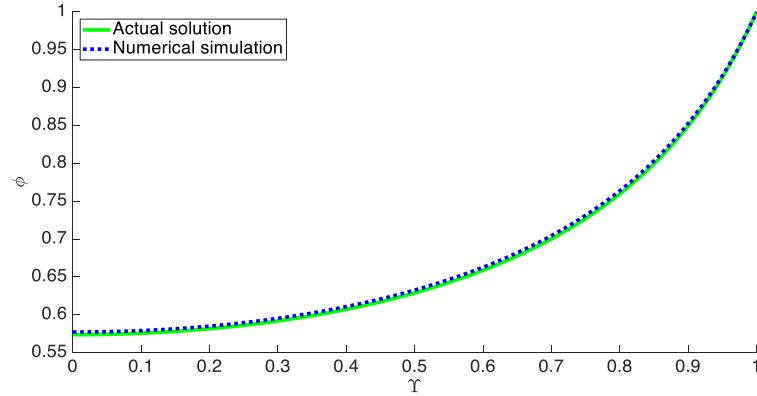
Adam-Bashforth method was carried out in a similar fashion to shooting method except that Euler, RK2 and RK4 methods were replaced with the appropriate  $n^{\text{th}}$  order Adam-Bashforth method to compute the  $\phi$  values. When carrying out  $n^{\text{th}}$  order Adam-Bashforth method (where  $n > 1$ ), the first  $(n-1)$   $\phi$  values were computed using the lower order Adam-Bashforth methods. For example, when 3<sup>rd</sup> order method was implemented, the second and third  $\phi$  values were computed using 1<sup>st</sup> order and 2<sup>nd</sup> order methods respectively.

## 5.2 Results

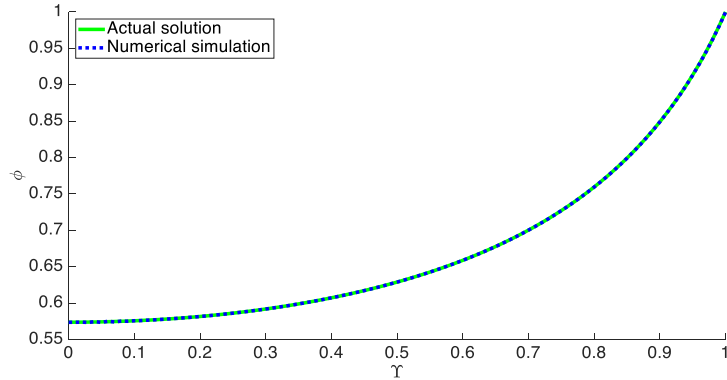
### 5.2.1 General validation of numerical simulation results and exact solutions

From Appendix section 9.3, it can be seen that the simulation results obtained from Adam-Bashforth method are generally in agreement with the exact solutions for reaction orders 1 and 5. Further analysis will be carried out on the different order methods to solve this ODE system.

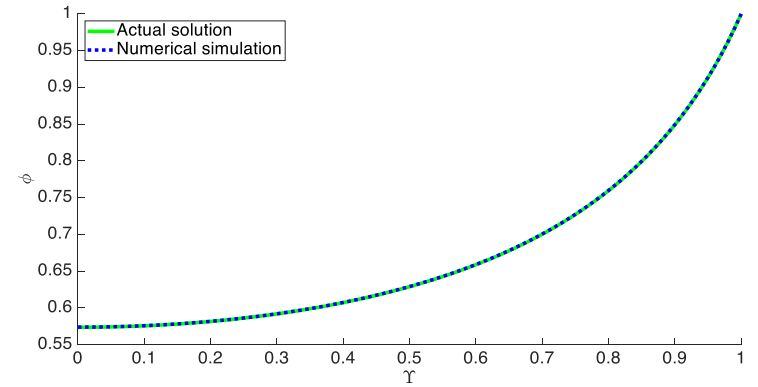
### 5.2.2 Effect of different order methods on the final profile



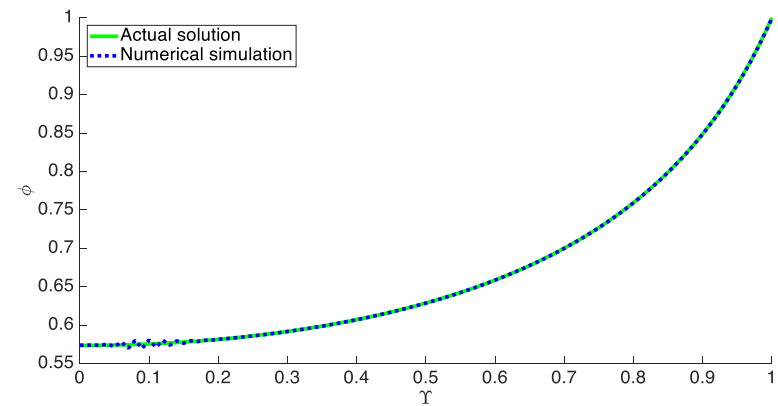
**1<sup>st</sup> order method** (Error = 0.0366 & elapsed time = 1.74 s)



**3<sup>rd</sup> order method** (Error = 0.0018 & elapsed time = 3.91 s)



**2<sup>nd</sup> order method** (Error = 0.0020 & elapsed time = 2.78 s)

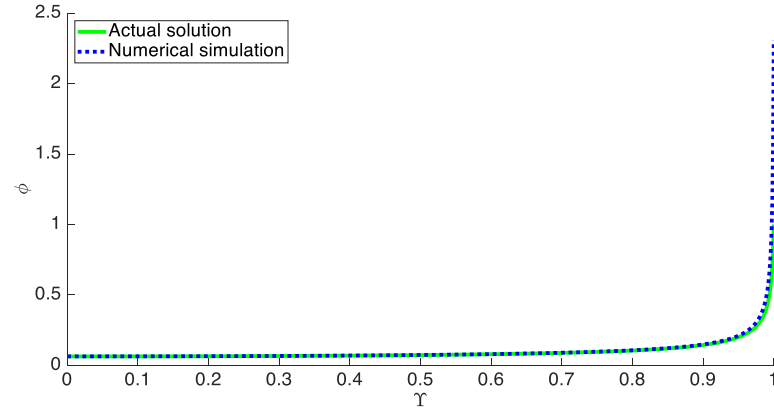


**4<sup>th</sup> order method** (Error = 0.0134 & elapsed time = 5.06 s)

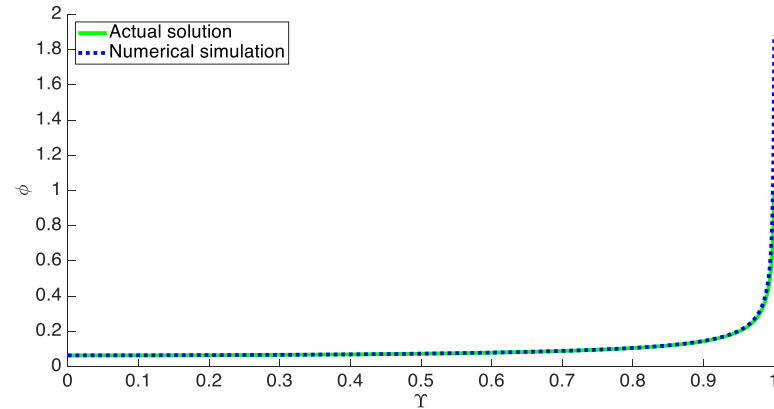
Figure 12: Different order methods (reaction order  $n = 5$ , step size = 0.01 &  $\varphi = 4.31$ )

The error was computed as the norm of the difference between the calculated and actual profile. From the results, it can be seen that higher order methods do not necessarily guarantee higher accuracy as they might be less stable since more previous points are taken into account in higher order Adam-Bashforth methods. Therefore, there is a higher chance for errors from previous points to propagate forward in the profile. For this ODE system, 2<sup>nd</sup> order Adam-Bashforth method is able to produce accurate results despite the fact that step size is not too small (0.01).

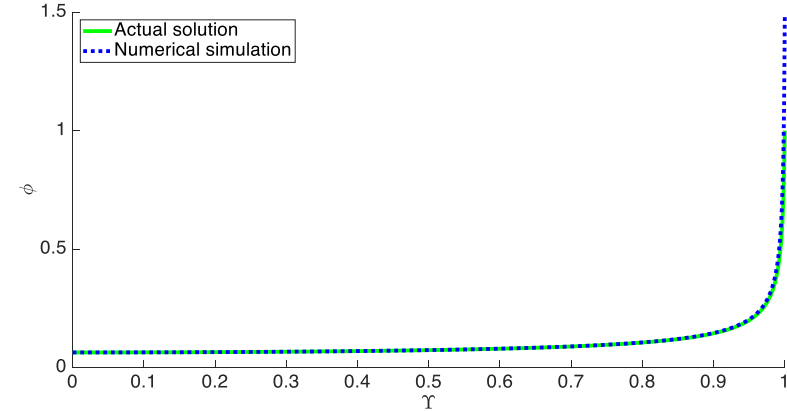
### 5.2.3 Effect of Thiele modulus ( $\varphi$ ) on the final profile



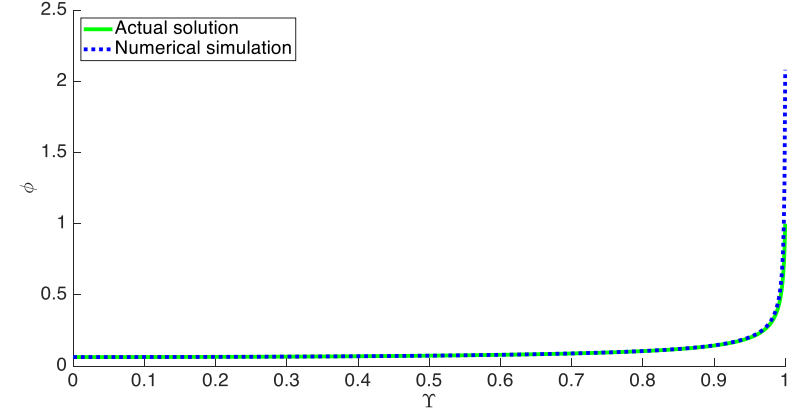
**1<sup>st</sup> order method** (Error = 1.67 & elapsed time = 1.65 s)



**3<sup>rd</sup> order method** (Error = 1.06 & elapsed time = 3.98 s)



**2<sup>nd</sup> order method** (Error = 0.6448 & elapsed time = 2.90 s)



**4<sup>th</sup> order method** (Error = 1.26 & elapsed time = 5.07 s)

Figure 13: Different order methods (reaction order  $n = 5$ , step size = 0.001 &  $\varphi = 430.95$ )

Despite making the step size 10 times smaller, for very large Thiele modulus ( $\varphi = 430.95$ ), all 4 methods are not able to converge accurately. The result produced by 2<sup>nd</sup> order method was closest to the exact solution. Even under these conditions, 4<sup>th</sup> order method could not produce as accurate result as lower order methods. Hence, method of finite difference is definitely superior to Adam-Bashforth method.

## 6. Effect of Thiele modulus on concentration profile (different reaction orders)

In this project, 43.09 was used as the reference Thiele modulus value (**refer to Appendix section 9.4 for derivation**). Based on this reference value, 4 other Thiele modulus values (430.95, 136.28, 13.63 and 4.31) were arbitrarily chosen to study the effect of Thiele modulus magnitude on the concentration profile. Since method of finite difference provides most accurate numerical simulation results, it was utilized to obtain the plots for reaction orders 1, 3 and 5 (**refer to Appendix section 9.5**).

For each reaction order, the concentration at a given radius always decreases as Thiele modulus increases. This is because a larger Thiele modulus either corresponds to a faster reaction rate or smaller diffusion coefficient. Hence, lesser reactants will reach the center of the catalyst particle with increasing Thiele modulus, thus explaining the decreasing concentration at the center.

An interesting observation is that the concentration at center of particle (at  $Y = 0$ ) is higher when  $n = 5$  as compared to the concentration when  $n = 1$ . This appears to be inconsistent with our understanding. As reactants diffuse into the porous catalyst particle, a reaction with higher order of reaction should consume the reactants faster than the same reaction with a smaller order. Hence, a smaller concentration at center of catalyst particle is expected for  $n = 5$  rather than  $n = 1$ . Although no literature can be found to explain this phenomenon, one plausible hypothesis is proposed: as reaction with a higher order consumes the reactants faster, there will be a higher concentration difference between site of reaction and region outside the catalyst particle. Hence, there is a steeper concentration gradient for diffusion. As a result, the rate of diffusion is increased. Such increase may outweigh the consumption of reactants and result in a higher concentration at center.

## 7. Conclusion

All in all, it can be conclusively seen that method of finite difference, which is an implicit method, is indeed the most accurate numerical method for simulating the concentration profile of a reaction-diffusion process in spherical porous catalyst for a wide range of conditions.

## 8. References

1. Magyari, E. (2010). Exact analytical solutions of diffusion reaction in spherical porous catalyst. *Chemical Engineering Journal*, 158(2), 266-270.
2. Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.
3. Chapra, S. C., & Canale, R. P. (2010). *Numerical methods for engineers*. Boston: McGraw-Hill Higher Education,.
4. *Diffusion coefficients of Gases in Water*. Retrieved from Engineering Toolbox:  
[https://www.engineeringtoolbox.com/diffusion-coefficients-d\\_1404.html](https://www.engineeringtoolbox.com/diffusion-coefficients-d_1404.html)
5. *Carbonic acid*. Retrieved from Science direct:  
<https://www.sciencedirect.com/topics/chemistry/carbonic-acid>

## 9. Appendix

### 9.1 Finite difference method results

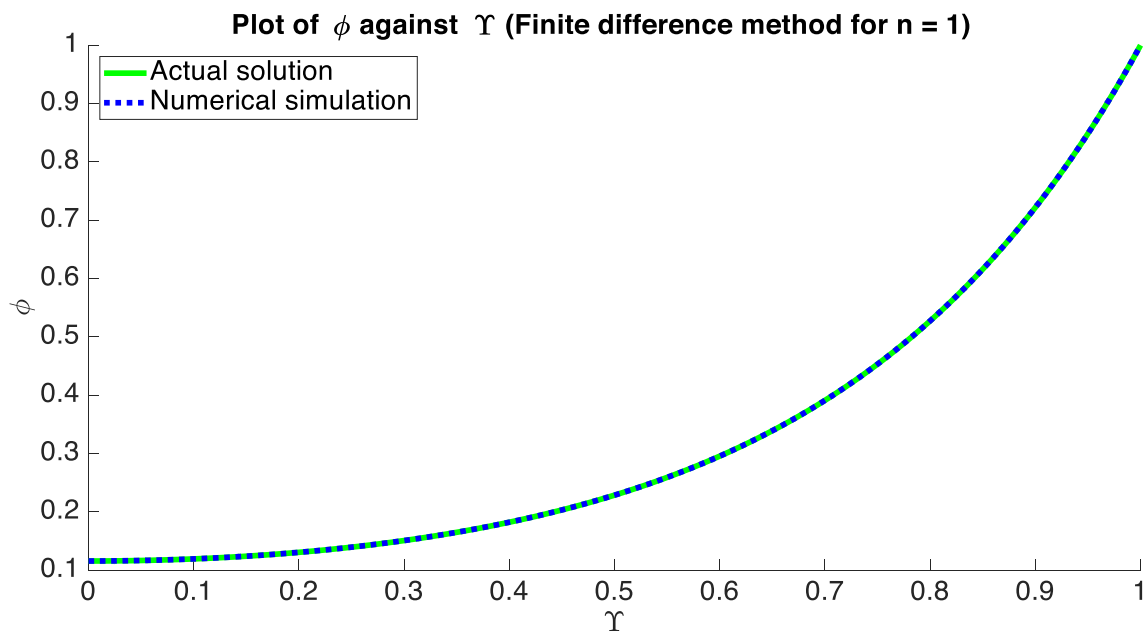


Figure 14: Comparison of actual solution and numerical simulation results (reaction order = 1 &  $\varphi = 4.31$ )

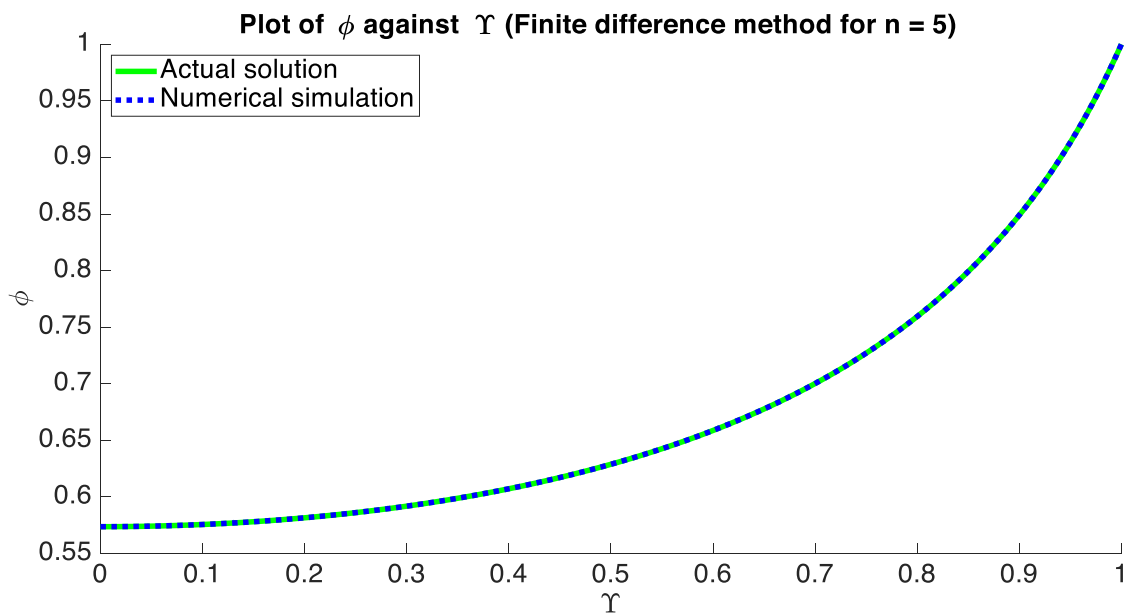


Figure 15: Comparison of actual solution and numerical simulation results (reaction order = 5 &  $\varphi = 4.31$ )

## 9.2 Shooting method results

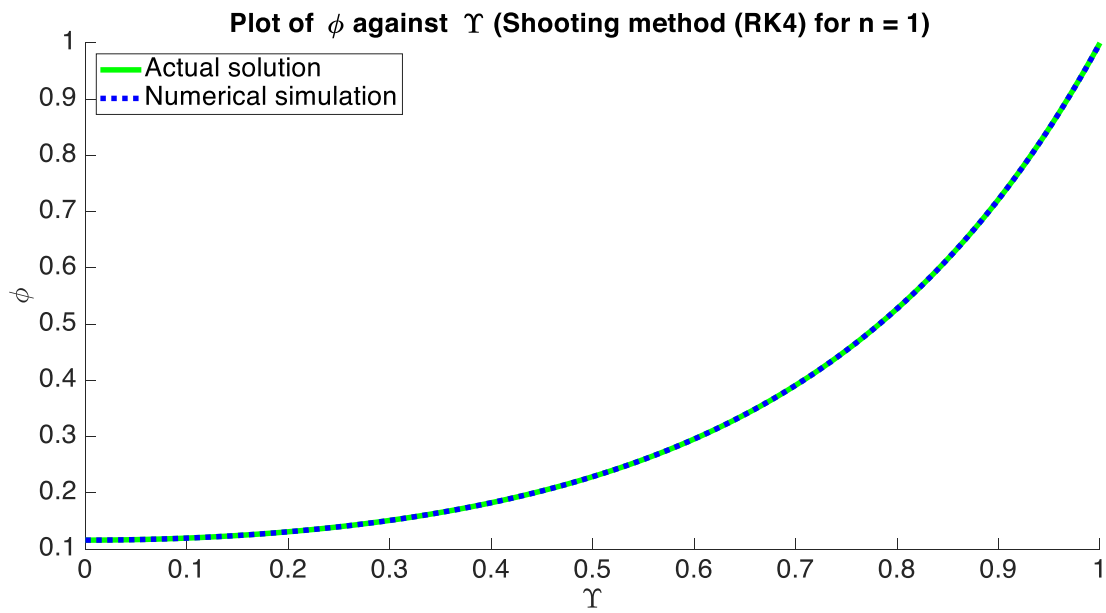


Figure 16: Comparison of actual solution and numerical simulation results  
(reaction order = 1 &  $\varphi = 4.31$ )

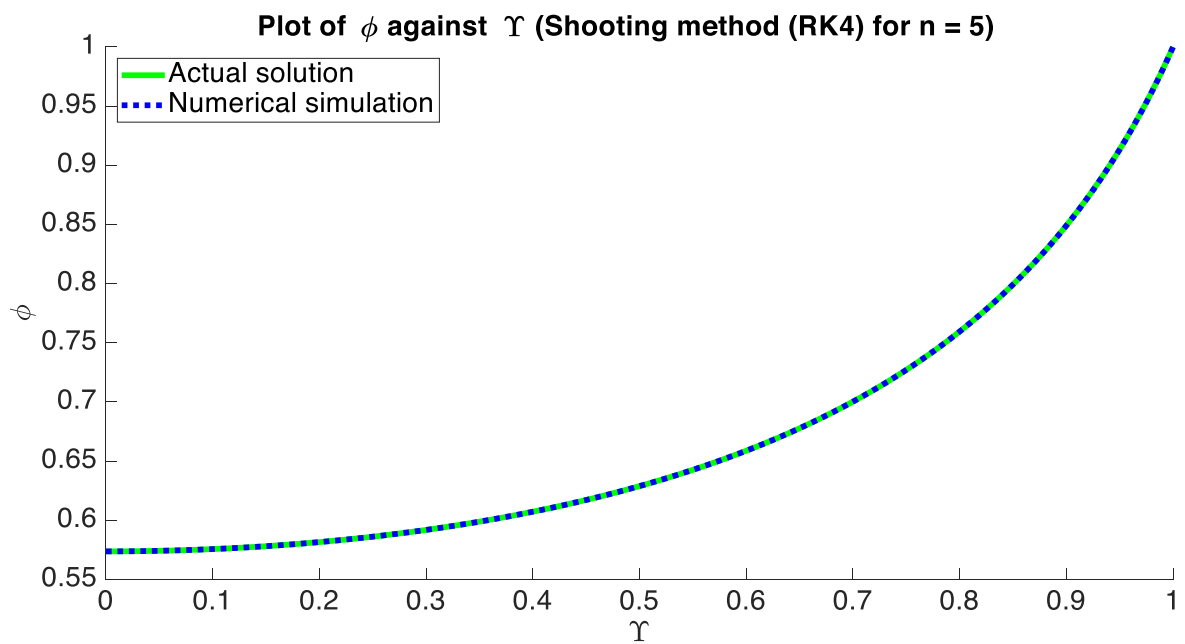


Figure 17: Comparison of actual solution and numerical simulation results  
(reaction order = 5 &  $\varphi = 4.31$ )



### 9.3 Adam-Bashforth method results

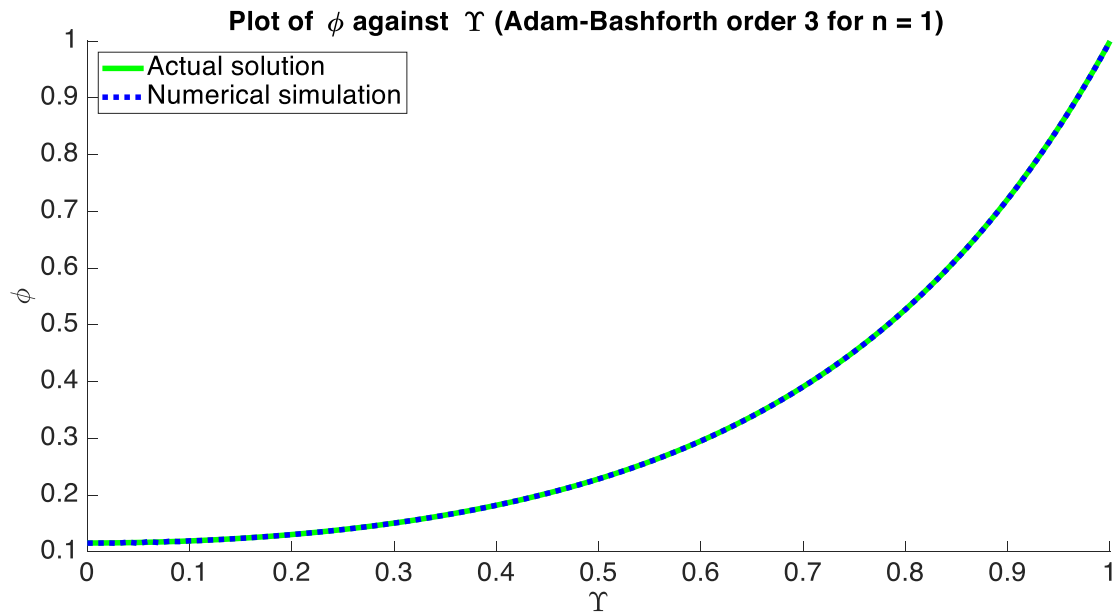


Figure 18: Comparison of actual solution and numerical simulation results  
(reaction order = 1 &  $\varphi = 4.31$ )

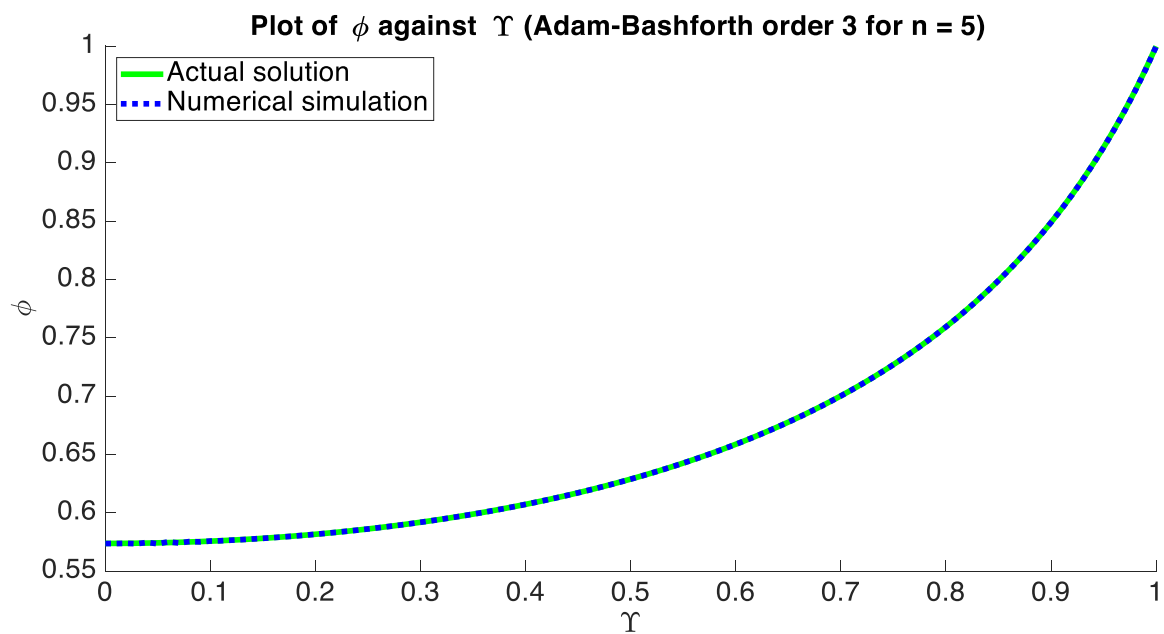


Figure 19: Comparison of actual solution and numerical simulation results  
(reaction order = 5 &  $\varphi = 4.31$ )

## 9.4 Derivation of reference Thiele modulus

Diffusivity of carbon dioxide in water ( $D_e$ ) =  $2.1 \times 10^{-5} \text{ cm}^2/\text{s} = 2.1 \times 10^{-9} \text{ m}^2/\text{s}$

Reaction rate constant for carbon dioxide in water to form carbonic acid ( $k$ ) =  $0.039 \text{ s}^{-1}$

Radius of the spherical catalyst particle ( $R$ ) =  $0.01 \text{ m}$  (assumed)

Surface concentration of carbon dioxide ( $C_{AS}$ ) =  $1 \text{ kmol/m}^3$  (assumed)

By choosing the surface concentration of carbon dioxide to be  $1 \text{ kmol/m}^3$ , the reference Thiele modulus will become the same for all reaction orders.

$$\text{Thiele modulus, } \phi = R \sqrt{\frac{k C_{AS}^{n-1}}{D_e}} = 0.01 \sqrt{\frac{0.039 \times 1}{2.1 \times 10^{-9}}} = 43.09 \quad (2 \text{ d p})$$

## 9.5 Variation of concentration profile with changing Thiele modulus

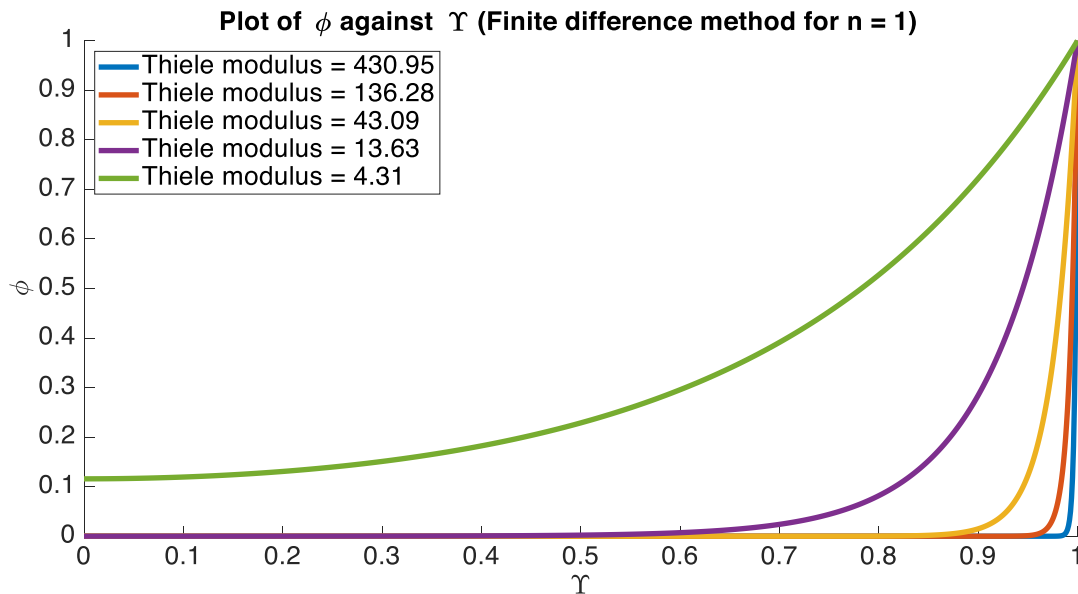


Figure 20: Variation of concentration profile with Thiele modulus (reaction order 1)

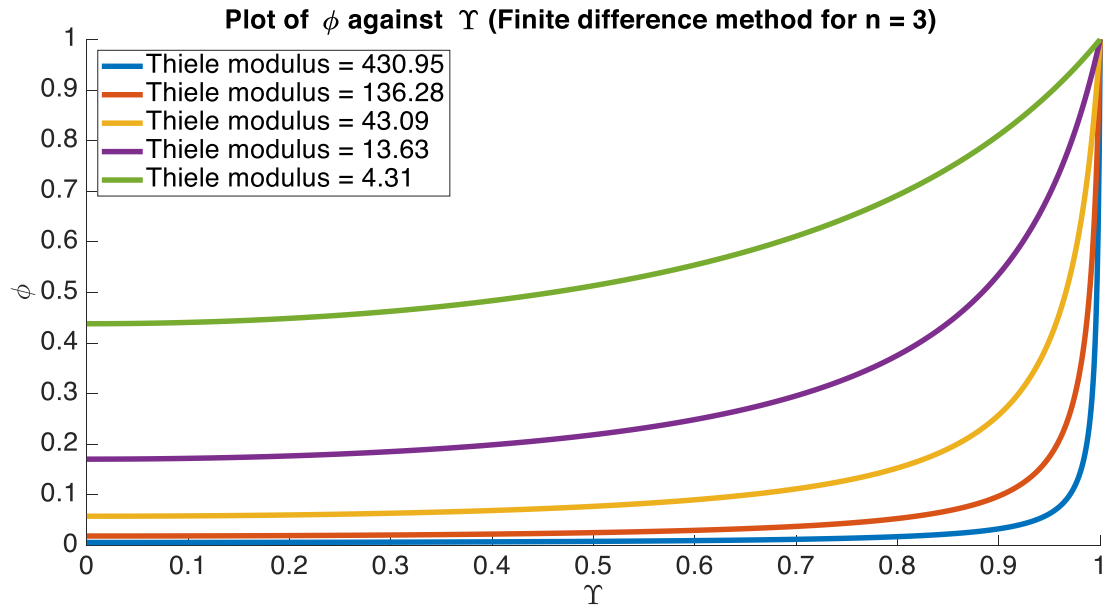


Figure 21: Variation of concentration profile with Thiele modulus (reaction order 3)

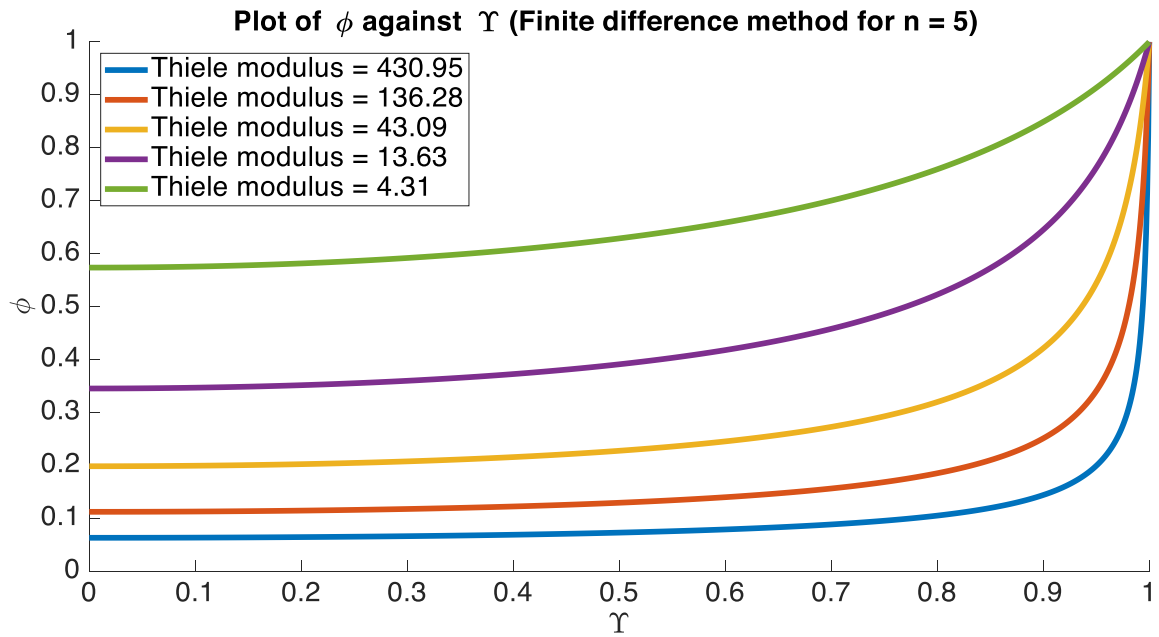


Figure 22: Variation of concentration profile with Thiele modulus (reaction order 5)