Imperial College London
Department of Mathematics
Dr Igor Shevchenko

Numerical Solution of Ordinary Differential Equations

I pledge that the work submitted for this coursework, both the report and the MATLAB code, is my own unassisted work unless stated otherwise.

First Name .... Marwan

Family Name......... Riach

CID 01349928

Coursework 5

Before you start working on this coursework, please read coursework wines. Fill in this page and include it as a cover sheet to your report, otherwise the coursework will not be marked. Any marks received for this coursework are only indicative and may be subject to moderation and scaling. *The mastery component is marked with a star.* **Make sure both the report and the code are located in the directory CW5_FirstName_FamilyName_CID.**

Exercise 1 (Stability analysis of linear BVP) % of course mark: /5.0

Design your own numerical method for the boundary value problem

$$u_t = \varepsilon u_{xx} + \gamma(1-u)\,, \quad u(t,0) = u(t,L)\,, \quad x \in [0,L], \quad t=(0,10], \tag{1}$$

and show it is convergent.

Exercise 2 (Numerical solution of linear BVP) % of course mark: /5.0

Solve the boundary value problem (1) with the method you have developed in Exercise 1. The initial conditions and parameters for (1) are

$$u(0,x) = \mathrm{e}^{-L^2(x-L/2)^2}, \quad \gamma = 1/\varepsilon, \quad L=10, \quad \varepsilon = 0.1. \tag{2}$$

Present the results as a waterfall graph (*waterfall* in MATLAB) with a reasonable step between time slices (solution in the domain at time t_i).

Exercise 3 (Numerical solution of linear BVP) % of course mark: /3.0

Show that the global error for (1) decreases as the time step, Δt, and the space step, Δx, decreases:

$$\Delta t = 2^{-i}\Delta t_0, \qquad \Delta x = 2^{-i}\Delta x_0, \quad i = 0,1,2,3,4\,.$$

Take Δt_0 and Δx_0 to ensure no blow-up solution. *In order to calculate the analytical solution for the global error, take a function $\phi(t,x)$ (at least once differentiable in time and twice in space) that satisfies the boundary condition and compute the right hand side of* (1) *by plugging ϕ into* (1). *Thus, equation* (1) *with the right hand side has the solution ϕ.*

Exercise 4 (Numerical solution of nonlinear BVP) % of course mark: /10.0

Solve the nonlinear boundary value problem

$$u_t + vu_x = \varepsilon u_{xx} + \gamma u^2(1-u)\,, \quad u_x(t,0) = u_x(t,L) = 0\,, \quad x\in[0,L], \quad t=(0,10], \tag{3}$$

 In hand date: 16:00 GMT, 12 Dec 2020 **Due date: 23:59 GMT, 06 Jan 2021**

$$u(0,x) = \cos(2\pi x/L)\sin(10\pi x/L), \quad \gamma = 1, \quad v = \{-1.0, 0.0, 1.0\}, \quad L = 10, \quad \varepsilon = 0.1\,,$$

with the implicit Runge–Kutta method developed in Coursework 4. Show that the global error for (3) decreases as the time and space steps decrease (as in Exercise 3). Present the results as a waterfall graph with a reasonable step between time slices.

Exercise 5 (Numerical solution of nonlinear BVP)	**% of course mark:**	**/10.0★**

Solve the Camassa–Holm equation

$$m_t + m u_x + (mu)_x = 0, \quad m = u - \alpha^2 u_{xx}, \quad \alpha = 1.0, \quad x \in [0, 100], \quad t = (0, 400], \tag{4}$$

with the implicit RK method developed in Coursework 4. The initial condition and boundary conditions are given by

$$u(0,x) = \frac{1}{10}\mathrm{e}^{-\frac{1}{10}(x - L\frac{1}{6})^2} - \frac{1}{10}\mathrm{e}^{-\frac{1}{10}(x - L\frac{5}{6})^2}, \quad u(t,0) = u(t,L), \quad L = 100.$$

For the space approximation use the following finite difference scheme:

$$m_t^n + m^n D u^n + D^-(m^n u^n) = 0, \quad m^n = u^n - D^- D^+ u^n,$$

where D^+ and D^- denote the forward and backward difference operators, and $D = (D^+ + D^-)/2$. Show that the global error for (3) decreases as the time and space steps decrease (as in Exercise 3). Present the results as a waterfall graph with a reasonable step between time slices.

Exercise 6 (Numerical solution of ODEs)	**% of course mark:**	**/☺**

Have a Merry Christmas and a Happy New year!

Coursework mark: **% of course mark**

In order to produce the plots and figures in this document, run the *exercise.m* files in the CW5 repository in MATLAB.

1 Exercise One (Stability Analysis of Linear BVP)

The method we will use for the boundary value problem will be the Backward Euler method. We will therefore be required to show that the method is both consistent and stable. From lectures, we know that both the Backward Euler method and 2nd order central difference schemes are consistent. Thus, all that needs to be shown is that the Backward Euler method, when applied to this BVP, is stable according to the Von Neumann stability analysis, as this is a linear problem.

In order to perform Von Neumann stability analysis, we will translate the problem by one unit ($v = u - 1$). This is to ensure that when the equation (obtained via substitution) is divided through by $e^{ikj\Delta x}$, we are left with a linear equation in $\zeta(k)$. Furthermore, by the property of linearity of convergence, we know that if the Backward Euler method converges for v, then it converges for u. The problem becomes:

$$v_t = \varepsilon v_{xx} - \gamma v$$
$$(v = u - 1 \Rightarrow \; v_{xx} = u_{xx}, \; v_t = u_t)$$

The Backward Euler method when applied to the above BVP is as follows:

$$v_j^{n+1} = v_j^n + \Delta t(\varepsilon \frac{v_{j+1}^{n+1} - 2v_j^{n+1} + v_{j-1}^{n+1}}{(\Delta x)^2} - \gamma v_j^{n+1}) \tag{1}$$

As per the Von Neumann stability analysis, let us assume that the numerical scheme admits a solution of the form:

$$v_j^n = \zeta^n(k) e^{ikj\Delta x}$$

Substitution of this gives:

$$\zeta^{n+1} e^{ikj\Delta x} = \zeta^n e^{ikj\Delta x} + \Delta t(\varepsilon \frac{\zeta^{n+1} e^{ik(j+1)\Delta x} - 2\zeta^{n+1} e^{ikj\Delta x} + \zeta^{n+1} e^{ik(j-1)\Delta x}}{(\Delta x)^2} - \gamma \zeta^{n+1} e^{ikj\Delta x})$$
$$\Rightarrow \zeta = 1 + \Delta t(2\varepsilon \frac{cos(k\Delta x) - 1}{(\Delta x)^2} - \gamma)\zeta$$
$$\Rightarrow \zeta = \frac{1}{1 - \Delta t(2\varepsilon \frac{cos(k\Delta x)-1}{(\Delta x)^2} - \gamma)}$$

The Von Neumann stability condition is given by:

$$|\zeta(k)| \leq 1, \; \forall k$$

This is satisfied when the following holds:

$$\Delta t(2\varepsilon \frac{cos(k\Delta x) - 1}{(\Delta x)^2} - \gamma) \leq 0$$
$$\Rightarrow \frac{cos(k\Delta x) - 1}{(\Delta x)^2} \leq \frac{\gamma}{2\varepsilon}$$

This is satisfied $\forall \Delta x > 0$ as the RHS is positive and the LHS is at most 0 (if $k = 0$). Thus, there are no constraints on Δx or Δt for the stability of this scheme. Ergo, this method is stable (and hence convergent) for the boundary problem with respect to v and by the preservation of convergence under a shift, the boundary problem is convergent with respect to u. Indeed, from lectures we know that this method is absolutely stable given its independence of the size of the time step and space step.

2 Exercise Two (Numerical Solution of Linear BVP)

Solving the BVP with the Backward Euler gives a linear system of equations that must be solved at each time step, as shown by (1) on the previous page with the substitution $u = v + 1$. The value at the 0th time step is given by the following initial condition with the values of the constants also given:

$$u(0, x) = e^{-L^2(x-L/2)^2}, \ \gamma = 10, \ \varepsilon = 0.1, \ L = 10$$

The set of equations in (1) (after the substitution) can be represented as an M by M matrix which is tridiagonal except for the first and and last row. The matrix is M by M rather than $M+1$ by $M+1$ since the boundary conditions are such that the solution at position L is the same as at position 0 ($u_0^n = u_M^n$):

$$\begin{pmatrix} (1+\Delta t\gamma + \frac{2\varepsilon\Delta t}{(\Delta x)^2}) & -\frac{\varepsilon\Delta t}{(\Delta x)^2} & 0 & \cdots & -\frac{\varepsilon\Delta t}{(\Delta x)^2} \\ -\frac{\varepsilon\Delta t}{(\Delta x)^2} & (1+\Delta t\gamma + \frac{2\varepsilon\Delta t}{(\Delta x)^2}) & -\frac{\varepsilon\Delta t}{(\Delta x)^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\varepsilon\Delta t}{(\Delta x)^2} & 0 & \cdots & -\frac{\varepsilon\Delta t}{(\Delta x)^2} & (1+\Delta t\gamma + \frac{2\varepsilon\Delta t}{(\Delta x)^2}) \end{pmatrix} \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_{M-1}^{n+1} \end{pmatrix} = \begin{pmatrix} u_0^n + \Delta t\gamma \\ u_1^n + \Delta t\gamma \\ \vdots \\ u_{M-1}^n + \Delta t\gamma \end{pmatrix}$$

The system is solved in MATLAB using the *inv* function at each time step. Below is the numerical solution to the BVP using the $waterfall$ feature in MATLAB:

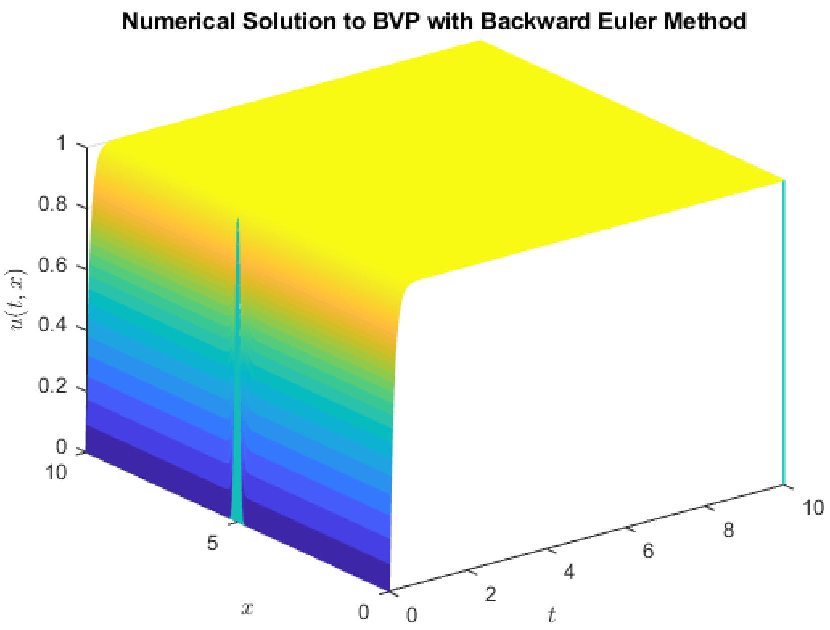


Figure 1: Solution to linear BVP with $\Delta t = \Delta x = 0.01$

3 Exercise Three (Numerical Solution of Linear BVP)

In order to investigate the behaviour of the global error of the Backward Euler method with regards to the BVP, we will consider a function, $\phi(t, x)$, that satisfies the boundary conditions and survives differentiation with respect to t once and x twice. We will then add a function of x and t to our original BVP such that $\phi(t, x)$ is an analytical solution and this will give us our modified BVP. We will then find the global error for the modified equation and study how it behaves as Δt and Δx are changed. If the solution to the modified BVP converges as Δt and Δx approach zero, then the solution to the original BVP will converge as Δt and Δx approach zero.

Our original BVP is as follows:

$$u_t = \varepsilon u_{xx} + \gamma(1-u),\ u(t,0) = u(t,L),\ x \in [0,L],\ t = [0,10]$$

If we take $\phi(t,x) = t \cdot cos(\frac{2\pi x}{L})$, then this satisfies the boundary conditions and is differentiable to a sufficient degree that it survives the BVP. The initial condition is simply $\phi(0,x) = 0$.

Modifying the BVP as follows ensures that $\phi(t,x)$ is an analytical solution to the modified BVP:

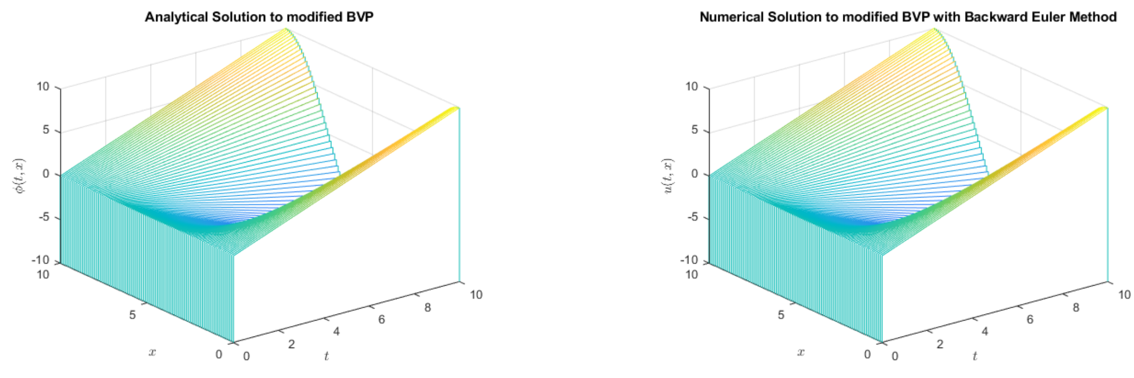
$$u_t = \varepsilon u_{xx} - \gamma u + cos(\frac{2\pi x}{L})(1 + \varepsilon t(\frac{2\pi}{L})^2 + \gamma t)$$

Applying this modified BVP to the numerical solution used in exercise 2, we find that the new set of equations can be written as follows:

$$\begin{pmatrix} (1+\Delta t\gamma + \frac{2\varepsilon\Delta t}{(\Delta x)^2}) & -\frac{\varepsilon\Delta t}{(\Delta x)^2} & 0 & \cdots & -\frac{\varepsilon\Delta t}{(\Delta x)^2} \\ -\frac{\varepsilon\Delta t}{(\Delta x)^2} & (1+\Delta t\gamma + \frac{2\varepsilon\Delta t}{(\Delta x)^2}) & -\frac{\varepsilon\Delta t}{(\Delta x)^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{\varepsilon\Delta t}{(\Delta x)^2} & 0 & \cdots & -\frac{\varepsilon\Delta t}{(\Delta x)^2} & (1+\Delta t\gamma + \frac{2\varepsilon\Delta t}{(\Delta x)^2}) \end{pmatrix} \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_{M-1}^{n+1} \end{pmatrix}$$
$$= \begin{pmatrix} u_0^n + \Delta t \cdot cos(\frac{2\pi x_0}{L})(1 + \varepsilon t_{n+1}(\frac{2\pi}{L})^2 + \gamma t_{n+1}) \\ u_1^n + \Delta t \cdot cos(\frac{2\pi x_1}{L})(1 + \varepsilon t_{n+1}(\frac{2\pi}{L})^2 + \gamma t_{n+1}) \\ \vdots \\ u_{M-1}^n + \Delta t \cdot cos(\frac{2\pi x_{M-1}}{L})(1 + \varepsilon t_{n+1}(\frac{2\pi}{L})^2 + \gamma t_{n+1}) \end{pmatrix}$$

where t_{n+1} denotes the time at step $n+1$ and x_i denotes the position at step i.

Below shows the solution according to the analytical solution and the numerical solution. As can be seen, there is strong similarity suggesting convergence with this numerical method.



(a) $\phi(t,x)$

(b) $u(t,x)$

Figure 2: Analytical solution (left) and Numerical solution (right) to modified BVP

The global error in the numerical solution is computed by taking the modulus of the difference between the solutions at all points at time, $t = 10$, by making use of the *norm* function in MATLAB. We start with $\Delta t_0 = \Delta x_0 = 0.4$ and halve both the time step and space step 4 times. The figure is given below, showing that as Δt and Δx approach zero, the global error also approaches zero, thus showing that the numerical method converges.

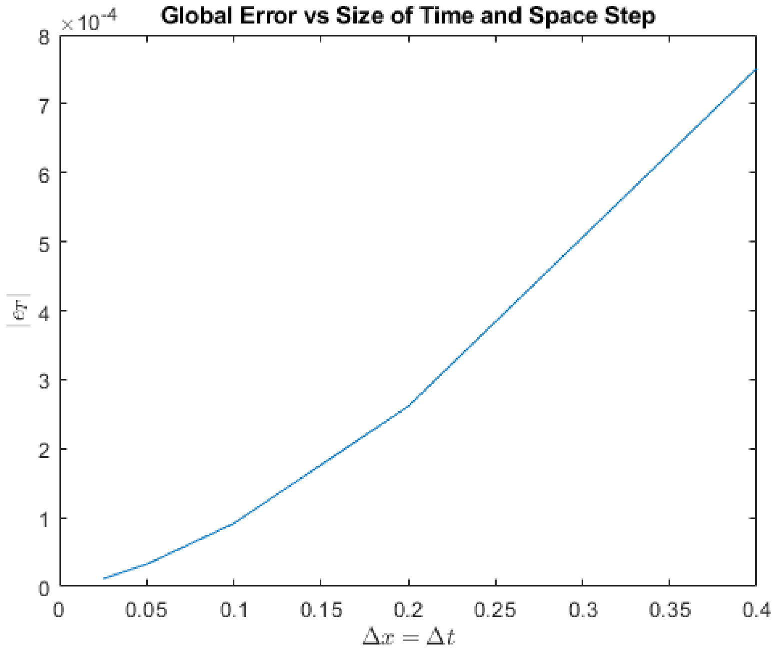


Figure 3: Global error of modified BVP with $\Delta t = \Delta x$ at 5 different values

4 Exercise Four (Numerical Solution of Nonlinear BVP)

The BVP we are now considering is as follows:

$$u_t = \varepsilon u_{xx} - v u_x + \gamma u^2(1-u),\ u_x(t,0) = u_x(t,L),\ x \in [0,L],\ t = [0,10]$$
$$u_t^n = f(t_n, u_n)$$

The IRK method being used is as follows:

$$k_1 = f(t_n, u_n),$$
$$k_2 = f(t_n + \frac{h}{3}, u_n + \frac{h}{3}k_1),$$
$$k_3 = f(t_n + \frac{2h}{3}, u_n - \frac{h}{3}k_1 + hk_2),$$
$$k_4 = f(t_n + h, u_n + hk_1 - hk_3 + hk_4),$$
$$u_{n+1} = u_n + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4)$$

where $h = \Delta t$ and k_i has $M+1$ components with the jth component denoted as $k_i(j)$.

To compute the derivatives with respect to x we will use first and second order central difference schemes where required. This makes the derivative at position j as follows:

$$(u_j^n)_t = \varepsilon \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} - \gamma v_j^n - v\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \gamma (u_j^n)^2(1 - u_j^n) \quad (2)$$

At the boundaries where $j = 0, M$, the derivative with respect to x is zero and hence we can say that $u_1^n = u_{-1}^n$ and $u_{M+1}^n = u_{M-1}^n$ and plug this into the above equation at the cases where j occurs peripherally.

The initial conditions and constants are given as follows:

$$u(0,x) = cos(2\pi x/L)sin(10\pi x/L),\ \gamma = 1,\ \varepsilon = 0.1,\ L = 10$$

As the method is implicit, we will require the Newton method to obtain the value of k_4 which in this instance has $M+1$ dimensions. We will denote the ith element of k_4 as $k_4(i)$. Up to k_4, all k values are explicit and so in the context of finding k_4 they, along with u_n, are constants. We will denote all terms in f except for k_4 as a constant, a, with components $a(i)$ ($a = u_n + hk_1 - hk_3$). Furthermore, since at no point is f dependent on t, we will consider f as only taking in one argument.

The $\mathbf{F}$ in our Newton method is given as:

$$\mathbf{F}(k_4) = k_4 - f(a + hk_4)$$

with f being given as (2) on the previous page.

The Jacobian matrix, $\mathbf{F}'(k_4)$, is computed by taking the partial derivatives of $\mathbf{F}$ with respect to $k_4(i)$, giving us a tridiagonal $M+1$ by $M+1$ matrix as follows:

$$\mathbf{F}'(k_4) = \begin{pmatrix} c_1 & -\frac{2\varepsilon\Delta t}{(\Delta x)^2} & 0 & \dots & 0 \\ -\frac{\varepsilon\Delta t}{(\Delta x)^2} - \frac{v\Delta t}{2\Delta x} & c_2 & -\frac{\varepsilon\Delta t}{(\Delta x)^2} + \frac{v\Delta t}{2\Delta x} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -\frac{2\varepsilon\Delta t}{(\Delta x)^2} & c_{M+1} \end{pmatrix}$$

with $c_i = 1 + \frac{2\varepsilon\Delta t}{(\Delta x)^2} + \gamma(2(\Delta t)^2 k_4(i) + 2a(i)\Delta t)(1 - a(i) - \Delta t k_4(i)) - \Delta t(a(i)^2 + 2a(i)\Delta t k_4(i) + (\Delta t)^2 k_4(i)^2)$

This thus makes the Newton method:

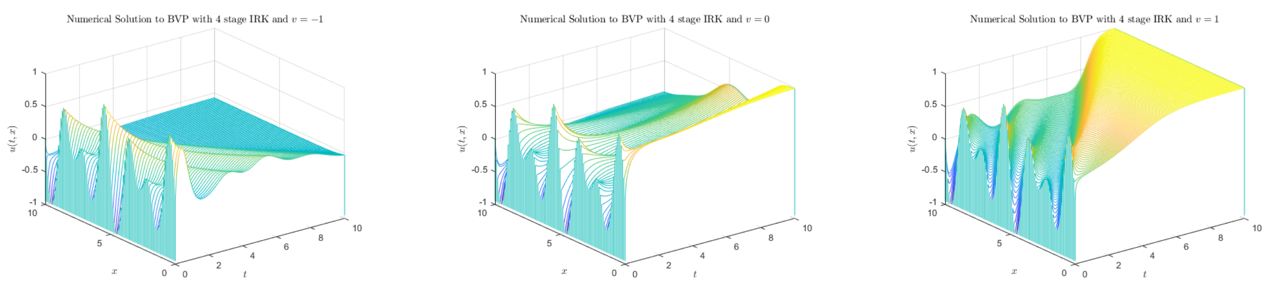
$$k_4^{i+1} = k_4^i - \mathbf{F}'(k_4^i)^{-1}\mathbf{F}(k_4^i), \ i = 0, 1, 2, ..., m.$$

To execute the implicit method with the Newton method we require an initial guess, k_4^0. Consider the following:

$$k_4 = f(t_n + h, u_n + hk_1 - hk_3 + hk_4) \approx f(t_n + h, u_n + hk_1 - hk_3 + hf(t_n, x_n)) = f(t_n + h, u_n + 2hk_1 - hk_3)$$

We will therefore estimate our value of k_4 for k_4^0 as $f(t_n + h, x_n + 2hk_1 - hk_3)$ which we know the value of. To speed up the implementation of the algorithm we will add a stopping criterion with a tolerance of $\epsilon = 10^{-5}$ to stop us always having to iterate through m times for each time step.

Below is the numerical solution to the problem at the values of $v = -1, 0, 1$. The time step was chosen to ensure convergence as we know this IRK method is not stable for all values of Δt:



(a) $v = -1$ (b) $v = 0$ (c) $v = 1$

Figure 4: Numerical solution to BVP with $\Delta t = 0.01$ and $\Delta x = 0.1$, $m = 5$

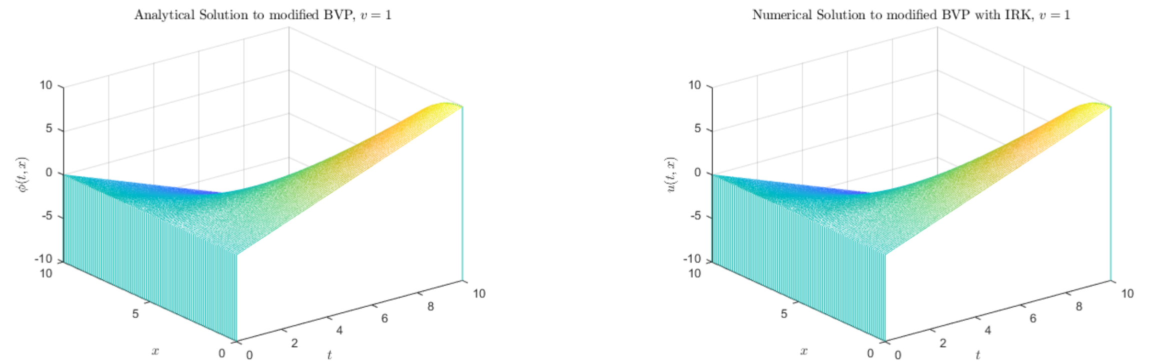
To investigate the global error, we will proceed as we did in the previous exercise. This time, we will take take $\phi(t, x) = t \cdot \sin(\frac{\pi x}{L} + \frac{\pi}{2})$, which satisfies the boundary conditions and is differentiable to a sufficient degree that it survives the BVP. The initial condition is simply $\phi(0, x) = 0$.

Modifying the BVP as follows ensures that $\phi(t, x)$ is an analytical solution to the modified BVP:

$$u_t = \varepsilon u_{xx} - v u_x + \gamma u^2(1-u) + \sin(\frac{\pi x}{L} + \frac{\pi}{2}) + \varepsilon \frac{\pi^2}{L^2} t \sin(\frac{\pi x}{L} + \frac{\pi}{2}) + v \frac{\pi}{L} t \cos(\frac{\pi x}{L} + \frac{\pi}{2})$$
$$-\gamma t^2 \sin^2(\frac{\pi x}{L} + \frac{\pi}{2}) \cdot (1 - t \sin(\frac{\pi x}{L} + \frac{\pi}{2}))$$

The method is applied as before, with the sin and cosine terms added on at the derivative with respect to t. Regarding the Newton method, there is little change, since the Jacobian is identical.

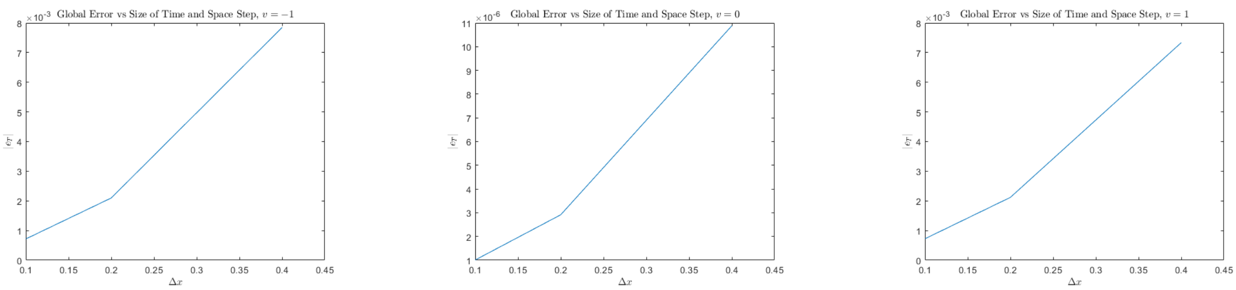
The figures below show the analytical solution alongside the numerical solution. There is significant resemblance, validating the IRK method as a way of solving the nonlinear BVP.



(a) $\phi(t, x)$ (b) $u(t, x)$

Figure 5: Analytical solution (left) and Numerical solution (right) to modified BVP at $v = 1$

The global error is computed in the same way as in exercise 3. The main difference here is that we start with $\Delta x_0 = 0.4 = 200\Delta t_0$ due to the fact that this method is not convergent for all space and time steps. With this borne in mind, we compute the global error for each of the 3 v values at 3 different values of Δx_0 instead of 5, given that as you halve the time step, the Jacobian increases by a factor of 4 and so inversion takes a very long time for small Δt; we halve Δx_0 and Δt_0 twice.



(a) $v = -1$ (b) $v = 0$ (c) $v = 1$

Figure 6: Global error of modified BVP with $\Delta x = 200\Delta t$ at 3 different values for 3 v values

As can be seen from all three plots, as Δx and Δt diminish, the numerical solution approaches the analytical solution. The solution to the modified BVP converges and ergo the solution to the original BVP converges too.

5 Exercise Five (Numerical Solution of nonlinear BVP)

The BVP we are now considering is as follows:

$$m_t = -mu_x - (mu)_x,\ m = u - u_{xx},\ u_x(t,0) = u_x(t,L),\ x \in [0,L],\ t = [0,400]$$
$$u_t^n = f(t_n, u_n)$$

The IRK method being used is as follows:

$$\begin{aligned}
k_1 &= f(t_n, u_n),\\
k_2 &= f(t_n + \frac{h}{3}, u_n + \frac{h}{3}k_1),\\
k_3 &= f(t_n + \frac{2h}{3}, u_n - \frac{h}{3}k_1 + hk_2),\\
k_4 &= f(t_n + h, u_n + hk_1 - hk_3 + hk_4),\\
u_{n+1} &= u_n + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4)
\end{aligned}$$

where $h = \Delta t$ and k_i has $M+1$ components with the jth component denoted as $k_i(j)$.

To compute the derivatives with respect to x we will use forward and backward difference operators as given in the assignment. This makes the derivative of m at position j as follows:

$$(m_j^n)_t = -(u_j^n - \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2})(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}) - (\frac{(u_j^n)^2 - (u_{j-1}^n)^2}{\Delta x})$$
$$+\frac{u_j^n(u_{j+1}^n - 2u_j^n + u_{j-1}^n) - u_{j-1}^n(u_j^n - 2u_{j-1}^n + u_{j-2}^n)}{(\Delta x)^3}$$

The boundaries imply that the solution is periodic spatially and so we use the fact that $u_j^n = u_{M+j}^n$ and plug this into the above equation at the cases where j occurs peripherally.

The initial conditions and constants are given as follows:

$$u(0,x) = \frac{1}{10}e^{-\frac{1}{10}(x-L/6)^2} - \frac{1}{10}e^{-\frac{1}{10}(x-5L/6)^2},\ L = 100$$

In order to carry out the IRK method, we require that u_t is isolated and that k_4 is approximated using a numerical scheme that converges.

To isolate u_t on the left hand-side, we will write the above system of equations in m_t as a matrix Au_t. This can be done since m involves u and u_{xx} and the derivative with respect to space represents a linear transformation which can consequently be written as a matrix. u_{xx} is computed using the second order central difference scheme, and so the matrix A can be written as follows:

$$m_t = (I - D^-D^+)u_t = \begin{pmatrix} (1+\frac{2}{(\Delta x)^2}) & -\frac{1}{(\Delta x)^2} & \cdots & -\frac{1}{(\Delta x)^2} & 0 \\ -\frac{1}{(\Delta x)^2} & (1+\frac{2}{(\Delta x)^2}) & -\frac{1}{(\Delta x)^2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -\frac{1}{(\Delta x)^2} & \cdots & -\frac{1}{(\Delta x)^2} & (1+\frac{2}{(\Delta x)^2}) \end{pmatrix} u_t = Au_t$$

Isolating u_t requires that we take the inverse of the $M+1$ by $M+1$ matrix, A, where M is the number of space steps in our method (excluding the zeroth step). This is constant for the whole procedure and so only needs to be computed once.

In order to compute k_4, we will make use of fixed point iteration. While FPI is not as fast to converge as the Newton method used in the previous exercise, it is simpler to code and is therefore more appropriate given the convoluted nature of this BVP.

As in exercise 4, the initial guess of k_4, k_4^0, will be as follows:

$$k_4^0 = f(t_n + h, u_n + 2hk_1 - hk_3)$$

The FPI scheme is as follows:

$$k_4^{i+1} = f(t_n + h, u_n + hk_1 - hk_3 + hk_4^i)$$

This scheme will converge to the true solution at a linear rate (rather than at a quadratic rate with the Newton method), and so the number of iterations will be increased ($m = 15$) and the tolerance for the difference in k_4 at consecutive steps will be $\epsilon = 10^{-16}$.

Applying the method as above gives us the following numerical solution to the BVP. It shows us the evolution of the solution of a peakon and an antipeakon:

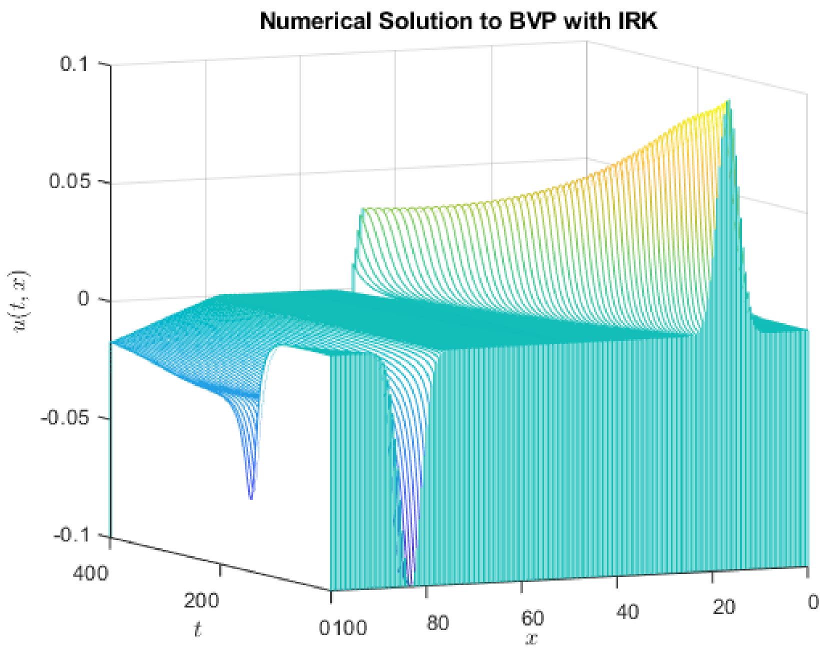


Figure 7: Solution to nonlinear BVP with $\Delta t = \Delta x = 0.5$

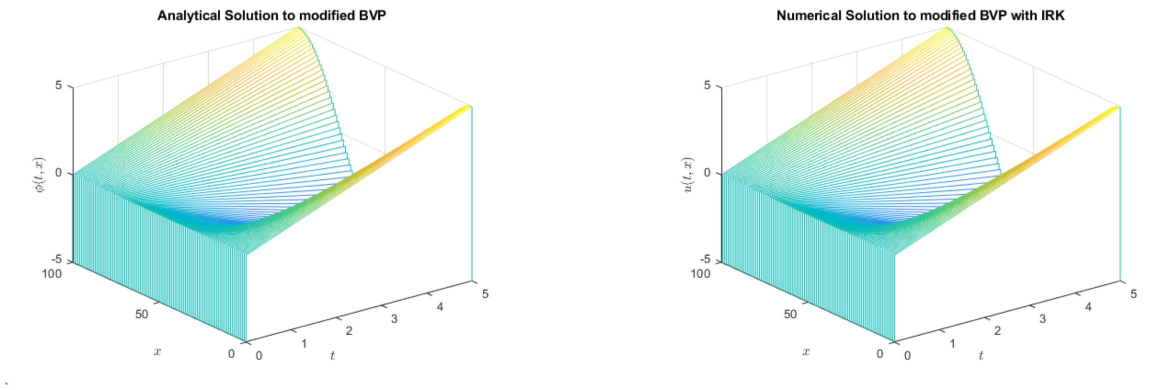
In order to explore the global error of this method, we will modify our BVP so that is satisfies the solution $\phi(t, x) = t \cdot cos(\frac{2\pi x}{L})$, the same solution as used in exercise 3.

Modifying the BVP as follows satisfies the analytical solution:

$$m_t = -mu_x - (mu)_x + (1 + (\frac{2\pi}{L})^2)\cos(\frac{2\pi x}{L}) - \frac{3\pi}{L}(1 + (\frac{2\pi}{L})^2)t^2\sin(\frac{4\pi x}{L}),$$
$$m = u - u_{xx}$$

The method carried out is exactly the same as with the unmodified BVP, except that the RHS now has functions that depend on space, time and trigonometric functions. To explore the global error, we will truncate the time period in order that we can investigate 5 different space and time steps and that the whole process won't take hours to run.

The figures below show the analytical solution alongside the numerical solution. There is significant resemblance, validating the IRK method as a way of solving the nonlinear BVP.



(a) $\phi(t, x)$ (b) $u(t, x)$

Figure 8: Analytical solution (left) and Numerical solution (right) to modified BVP

The global error is computed in the same way as in exercise 3. The main difference here is that we start with $\Delta x_0 = 1 = 10\Delta t_0$ due to the fact that this method is not convergent for all space and time steps. Here, no Jacobian needs to be computed at each time step (due to using the FPI method instead of the Newton method) so it will not take significantly longer to execute the algorithm for smaller space and time steps. We measure the global error at 5 different steps, with Δx_0 and Δt_0 being halved four times. As can be seen from the below plot, the global error approaches zero as Δx_0 and Δt_0 approach zero, showing that the method is convergent for this BVP.

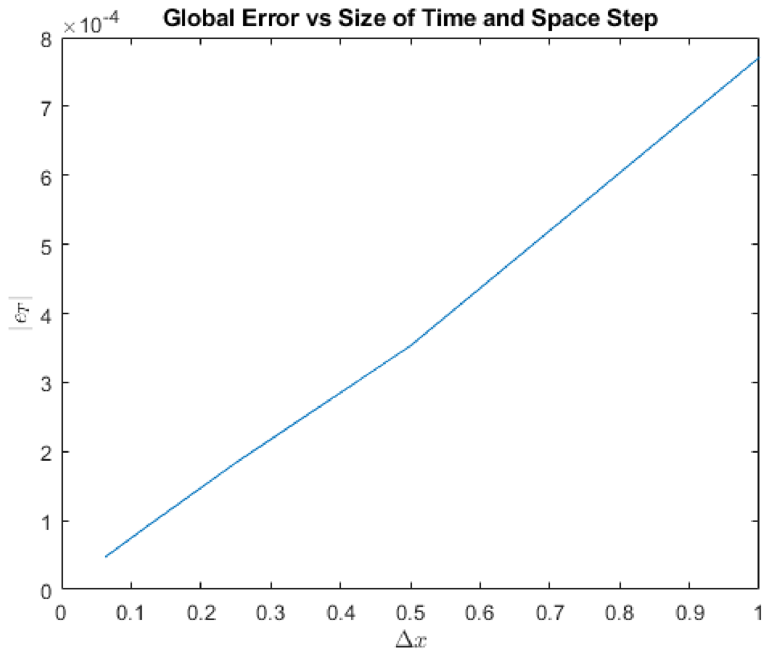


Figure 9: Global error of modified BVP with $10\Delta t = \Delta x$ at 5 different values