Numerical Methods for the Solution of Differential Equations (AMS 213B)
Homework 3 - Due Monday May 20

Instructions

Please submit to CANVAS one PDF file (your solution to the assignment), and one .zip file that includes any computer code you develop for the assignment. The PDF file must be a document compiled from Latex source code (mandatory for PhD students), or a PDF created using any other other word processor (MS and SciCAM students). No handwritten work should be submitted.

Question	points
1	40
2	60
Extra credit	30

Question 1 (40 points). Consider the following boundary value problem (BVP) for the Poisson's equation in two dimensions

$$\begin{cases} \dfrac{\partial^2 U(x,y)}{\partial x^2} + \dfrac{\partial^2 U(x,y)}{\partial y^2} = f(x,y) & (x,y) \in \Omega \\ U(x,y) = g(x,y) \qquad (x,y) \in \partial\Omega, \end{cases} \tag{1}$$

where $\Omega = [0,2] \times [0,1]$, $\partial\Omega$ is the boundary of Ω,

$$g(x,y) = 2 - x^2 - 2\sin(\pi y), \qquad \text{and} \qquad f(x,y) = -20 + 3x^2 + 4y^2. \tag{2}$$

(a) (30 points) Write a computer code to compute the numerical solution of the BVP (1)-(2) using second-order centered finite differences. To this end, discretize the domain using a tensor product grid with N points in x (including endpoints) and M points in y (including endpoints).

(b) (5 points) Plot the numerical solution you obtain for $(N, M) = (81, 51)$ as a surface plot.

(c) (5 points) Plot the numerical solution at $y = 0.2$ and $y = 0.5$ versus x (two graphs in the same figure).

Solution:

(a) We solve the boundary value problem by inverting of the matrix system at page 14 of the course note 7, i.e., equation (77). In order to apply this method we first transform (1)-(2) to a BVP with zero Dirichlet conditions. To this end, we define

$$\eta(x,y) = U(x,y) - g(x,y) \qquad \Leftrightarrow \qquad U(x,y) = \eta(x,y) + g(x,y). \tag{3}$$

A substitution of (3) into (1)-(2) yields

$$\begin{cases} \dfrac{\partial^2 \eta(x,y)}{\partial x^2} + \dfrac{\partial^2 \eta(x,y)}{\partial y^2} = f(x,y) - \underbrace{\left(\dfrac{\partial^2 g(x,y)}{\partial x^2} + \dfrac{\partial^2 g(x,y)}{\partial y^2}\right)}_{-2-4\pi\cos(\pi y^2)+8\pi^2 y^2 \sin(\pi y^2)} & (x,y) \in \Omega \\ \eta(x,y) = 0 \qquad (x,y) \in \partial\Omega, \end{cases}$$

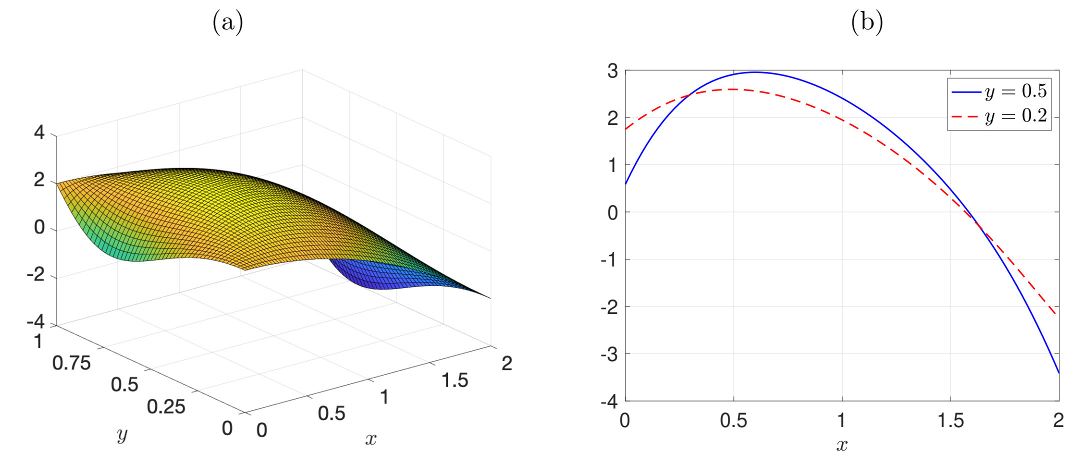


Figure 1: (a) Numerical solution of (1)-(2) obtained by inverting the matrix system (4) on a grid with $(N, M) = (81, 51)$. (b) Solution at $y = 0.2$ and $y = 0.5$ versus x.

The fully discrete finite-difference form of this problem can be written as in equation (77) of the course note 7. The matrix system $\boldsymbol{L}$ that corresponds to the finite-dimensional version of the Laplace operator (with the ordering discussed at page 14 of course note 7) is coded in the attached function `compute_matrix_system.m`. The solution is obtained by solving

$$\boldsymbol{L}\boldsymbol{\eta} = \boldsymbol{f} - \boldsymbol{L}\boldsymbol{g}, \tag{4}$$

where $\boldsymbol{\eta}$, $\boldsymbol{f}$, and $\boldsymbol{f}$ are column vectors representing the relabeling of $\eta(x_i, y_j)$, $f(x_i, y_j)$ and $g(x_i, y_j)$ at the inner nodes discussed at page 14 of course note 2. Once the solution of (4) is obtained, we simply translate it back to $U(x, y)$ using (3). This method is coded in the attached Matlab function `Poisson_solver.m`[1].

(b) In Figure 1(a) we plot the numerical solution of (1)-(2) obtained by solving the linear system (4).

(c) In Figure 1(b) we plot two cross-sections of the solution surface at $y = 0.2$ and $y = 0.5$ versus x.

Question 2 (60 points). Consider the following initial-boundary value problem for the heat

[1] We also solved the Poisson's equation using unconstrained optimization of the residual

$$R(\hat{\boldsymbol{U}}) = \sum_{i=2}^{N-1} \sum_{j=2}^{M-1} \left[\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} - f(x_i, y_j) \right]^2, \tag{5}$$

where $\hat{\boldsymbol{U}}$ is the matrix representing the solution at the interior grid points, while $\boldsymbol{u}$ is the matrix representing the full numerical solution (boundary + interior) – see the Matlab function `Poisson_solver.m`.

equation

$$\begin{cases} \dfrac{\partial U}{\partial t} = \dfrac{\partial^2 U}{\partial x^2} \qquad t \geq 0 \qquad x \in [-1, 1] \\ U(x, 0) = (3 + x) + 5(1 - x^2)^2 \qquad \text{(initial condition)} \\ U(-1, t) = 2, \qquad U(1, t) = 4 \qquad \text{(boundary conditions)} \end{cases} \tag{6}$$

(a) (15 points) Determine the analytical solution of (6). (Hint: the solution of (6) can be decomposed as

$$U(x, t) = (3 + x) + \eta(x, t) \tag{7}$$

where $\eta(x, t)$ satisfies the PDE

$$\frac{\partial \eta}{\partial t} = \frac{\partial^2 \eta}{\partial x^2} \tag{8}$$

with initial condition $\eta(x, 0) = 5(1 - x^2)^2$ and zero Dirichlet boundary conditions at $x = -1$ and $x = 1$).

(b) (5 points) Plot the analytical solution as a surface plot on a grid with 100×100 evenly-spaced points in $(x, t) \in [-1, 1] \times [0, 2]$.

(c) (15 points) Write a code to compute the numerical solution of (6) using second-order finite differences in space on an evenly-spaced grid with N points (N to be chosen later) in $[-1, 1]$, including the endpoints. Integrate the semi-discrete form of the PDE in time using the Crank-Nicolson method. Set $\Delta t = 10^{-4}$.

(d) (15 points) Write a code to compute the numerical solution of (6) using the Gauss-Chebyshev-Lobatto collocation method on a Gauss-Chebyshev-Lobatto grid with N points (N to be chosen later) in $[-1, 1]$, including the endpoints. As before, integrate the semi-discrete form in time using the Crank-Nicolson method. Set $\Delta t = 10^{-4}$.

(e) (10 points) Plot maximum pointwise error

$$e(T) = \max_{i=1,\dots,N} |U(x_i, T) - u_i(T)| \tag{9}$$

in a log scale between the analytical solution $U(x, t)$ you obtained in (a) and the numerical solutions $\{u_i(t)\}$ you obtained in (b) and (c) at time $T = 2$ versus the number of grid points N for $N = \{5, 10, 15, 20, 25, 30, 50, 100, 150, 200\}$ (two "semilogy" error plots in the same Figure, each with 10 points). Which method converges faster? Based on the analysis of the error plots, what can you say about the convergence order of each method?

Solution:

(a) To solve the heat equation (8) in the variable η, we define the transformations

$$\eta(x, t) = \xi(y, t) \qquad y = \frac{1}{2}x + \frac{1}{2} \tag{10}$$

It can be shown that ξ satisfies the PDE

$$\frac{\partial \xi}{\partial y} = \frac{1}{4}\frac{\partial^2 \xi}{\partial y^2} \tag{11}$$

with boundary conditions

$$\xi(0,t) = \eta(-1,t) = 0 \qquad \xi(1,t) = \eta(1,t) = 0 \tag{12}$$

and initial condition

$$\xi(y,0) = \eta(x,0) = 5\left[1-(2y-1)^2\right]^2, \tag{13}$$

We find the analytical solution using the method of separation of variables. To this end, we assume that ξ has the form

$$\xi(y,t) = Y(y)T(t).$$

Substituting this into the heat equation (11) yields

$$T'(t)Y(y) = Y''(y)T(t) \Rightarrow \frac{T'(t)}{T(t)} = \frac{1}{4}\frac{Y''(y)}{Y(x)} = -\lambda^2$$

where the choice of constant $-\lambda^2$ will be made clear later. The equation for T is a first-order separable ODE and has the general solution

$$T(t) = C\exp\left(-\frac{\lambda^2}{4}t\right).$$

The equation for X is a second-order homogeneous ODE and can be solved by considering its characteristic equation given by the second degree polynomial

$$r^2 + \lambda^2 = 0$$

which has roots $\pm i\lambda$. Since we have purely imaginary roots, the general solution is

$$Y(y) = A\cos(\lambda y) + B\sin(\lambda y).$$

Using the boundary condition yields

$$\begin{aligned} 0 =& X(0) = A \\ 0 =& X(1) = B\sin(\lambda) \Rightarrow \lambda = n\pi \end{aligned}$$

which gives the eigenfunctions

$$Y_n(y) = \sin(n\pi y).$$

By the principle of superposition, we have

$$\xi(y,t) = \sum_{n=1}^{\infty} C_n \exp\left(-\frac{(n\pi)^2}{4}t\right)\sin(n\pi y)$$

In the original coordinates, we have

$$\eta(x,t) = \sum_{n=1}^{\infty} C_n \exp\left(-\frac{(n\pi)^2}{4}t\right)\sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right) \tag{14}$$

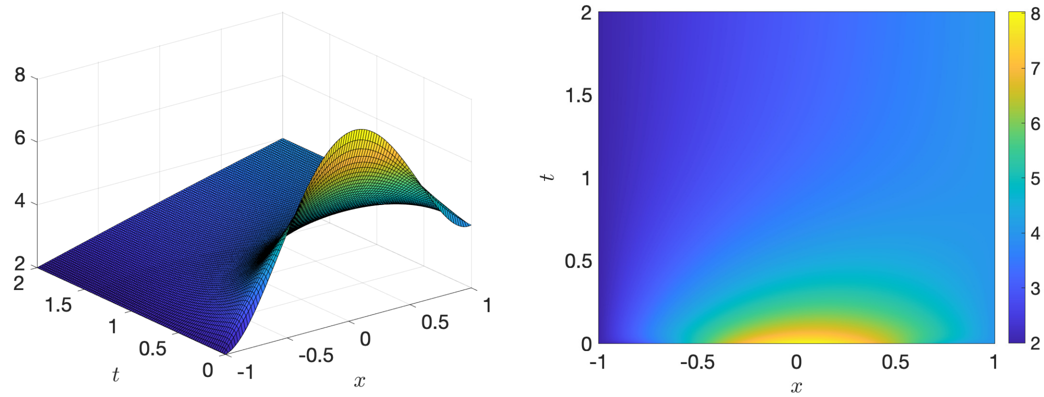


Figure 2: Analytical solution $U(x,t)$ of the initial value problem (6) given by the formula (15).

The constants C_n are determined by projecting the initial condition

$$\eta(x,0)=\sum_{n=1}^{\infty} C_n \sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)=5(1-x^2)^2.$$

onto the orthogonal basis $\sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)$. Performing this projection gives

$$\begin{aligned} C_n\int_{-1}^{1}\sin^2\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)dx &= \int_{-1}^{1}\sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)5(1-x^2)^2 dy \\ \Rightarrow C_n=\frac{\int_{-1}^{1}\sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)5(1-x^2)^2dx}{\int_{-1}^{1}\sin^2\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)dx} &= \frac{320}{\pi^5 n^5}\left((\pi^2n^2-12)((-1)^n-1)\right) \end{aligned}$$

So we have

$$\eta(x,t)=\sum_{n=1}^{\infty}\frac{320}{\pi^5n^5}\left((\pi^2n^2-12)((-1)^n-1)\right)\exp\left(-\frac{(n\pi)^2}{4}t\right)\sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right)$$

and consequently, the final solution has the form

$$U(x,t)=(3+x)+\sum_{n=1}^{\infty}\frac{320}{\pi^5n^5}\left((\pi^2n^2-12)((-1)^n-1)\right)\exp\left(-\frac{(n\pi)^2}{4}t\right)\sin\left(n\pi\left(\frac{x}{2}+\frac{1}{2}\right)\right). \tag{15}$$

(b) In Figure 2 we plot the analitical solution (15) for n running from 1 to 2000.

(c) To find the numerical solution, let us consider the evenly-spaced grid on $[-1,1]$ given by

$$x_j=-1+(j-1)\Delta x,\quad \Delta x=\frac{2}{N-1},\quad j=1,2,\dots N.$$

On this grid, we can write the system in semi-discrete form

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{D}^2_{\mathrm{FD}}\boldsymbol{u} + \boldsymbol{h}$$

where

$$\boldsymbol{D}^2_{\mathrm{FD}} = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -2 & 0 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 0 & \dots & \dots & \dots & 0 & 1 & -2 \end{bmatrix} \quad \boldsymbol{u} = \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ \vdots \\ \vdots \\ u_{N-3} \\ u_{N-2} \end{bmatrix} \quad \boldsymbol{h} = \begin{bmatrix} 2/\Delta x^2 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 3/\Delta x^2 \end{bmatrix}.$$

Using the Crank-Nicolson method to solve this linear system of ODEs gives

$$(\boldsymbol{I} - \frac{\Delta t}{2}\boldsymbol{D}^2_{\mathrm{FD}})\boldsymbol{u}^{k+1} = (\boldsymbol{I} + \frac{\Delta t}{2}\boldsymbol{D}^2_{\mathrm{FD}})\boldsymbol{u}^k + \Delta t\boldsymbol{h}. \tag{16}$$

Hence, at every iteration we need to solve a linear system of equations. The attached Matlab code `solve_heat_FD.m` implements the recursion (16).

(d) For the Gauss-Chebyshev-Lobatto (GCL) collocation method, we seek solutions of the form

$$u_N(x,t) = \sum_{j=1}^{N} u(x_j,t)\ell_j(x)$$

where ℓ_j are the Lagrange characteristic polynomials corresponding to the Gauss-Chebyshev-Lobatto quadrature points. A substitution of this expression into the heat equation and assuming that the residual vanishes at the interior points yields the system of $N-2$ equations

$$\begin{aligned}\frac{du_N(x_j,t)}{dt} &= \sum_{k=0}^{N} D^2_{jk}u_N(x_k,t), \\ &= D_{j1}U_N(x_1,t) + D_{jN}U_N(x_N,t) + \sum_{k=2}^{N-1} D^2_{jk}u_N(x_k,t) \quad j = 2,\dots,N-1\end{aligned}$$

where D^2_{jk} is the second-order differentiation matrix corresponding to the GCL quadrature points (see the Appendix of course note 1 or the Appendix of course note 7). We can write the system above in matrix vector form

$$\frac{d\boldsymbol{u}}{dt} = 2\boldsymbol{h}_1 + 4\boldsymbol{h}_N + \hat{\boldsymbol{D}}^2\boldsymbol{u},$$

where $\boldsymbol{h}_1, \boldsymbol{h}_N \in \mathbb{R}^{N-2}$ are the first and last columns of the matrix $\boldsymbol{D}^2$ with the first and last entries deleted. Note that $\hat{\boldsymbol{D}}^2$ is a matrix of dimension $(N-2)\times(N-2)$ formed by

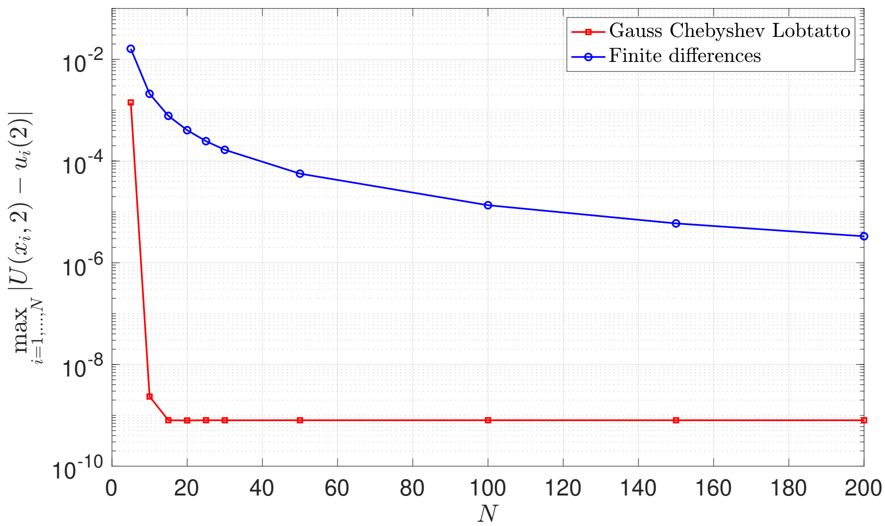


Figure 3: Errors obtained at $t = 2$ by solving the IVP (6) with Finite Differences and Gauss-Chebyshev-Lobatto space discretization and Crank-Nicolson time discretization.

deleting the first and last rows and the first and last columns of the matrix $\boldsymbol{D}$. Using the Crank-Nicolson Method to solve this linear system of ODE's results in the recursion

$$(\boldsymbol{I} - \frac{\Delta t}{2}\hat{\boldsymbol{D}}^2)\boldsymbol{u}^{k+1} = (\boldsymbol{I} + \frac{\Delta t}{2}\hat{\boldsymbol{D}}^2)\boldsymbol{u}^k + \Delta t(\boldsymbol{h}_1 u_1 + \boldsymbol{h}_N u_n). \tag{17}$$

Again, at every iteration k we need to solve a linear equation. The attached Matlab code `solve_heat_GCL.m` implements the recursion (17).

(d) In Figure 3, we plot the error (9) in linear-log coordinates for both of the methods described above. We can deduce from Figure 3, that the GCL collocation method converges much faster than the finite difference method. Specifically, the collocation method converges exponentially fast while the finite difference method converges with order 2.

Extra Credit (30 points).

(a) (20 points) Write a code to compute the numerical solution of the Kuramoto-Sivashinsky initial-boundary value problem

$$\begin{cases} \dfrac{\partial U}{\partial t} + u\dfrac{\partial U}{\partial x} + \dfrac{\partial^2 U}{\partial x^2} + \dfrac{\partial^4 U}{\partial x^4} = 0 \qquad t \geq 0 \qquad x \in [-25, 25] \\ U(x, 0) = \sin(x)e^{-(x-10)^2/2} \\ \text{Periodic boundary conditions} \end{cases} \tag{18}$$

using second-order centered finite-differences in space and the two-step Adams-Bashforth method in time. Run your simulations with $N = 200$ evenly-spaced spatial grid points in $[-25, 25]$, including endpoints, up $T = 100$ time units using time step $\Delta t = 10^{-4}$.

(b) (5 points) Plot the numerical solution as a surface on a 200×1001 space-time grid[2] (200 spatial points in $[-25, 25]$ and 1001 time instants in $[0, 100]$.

(c) (5 points) Plot the numerical solution at time $t = 62$ as a function of x.

Solution:

(a) We approximate the derivatives with centered finite differences on the grid given by

$$x_j = -25 + (j-1)\Delta x, \quad \Delta x = \frac{50}{N-1}, \quad j = 1, \dots N.$$

To this end, we obtain the approximations

$$\begin{aligned}
\frac{\partial U(x_j,t)}{\partial x} &\simeq \frac{U_{j+1}(t) - U_{j-1}(t)}{2\Delta x} \\
\frac{\partial^2 U(x_j,t)}{\partial x^2} &\simeq \frac{U_{j-1}(t) - 2U_j(t) + U_{j+1}(t)}{\Delta x^2} \\
\frac{\partial^4 U(x_j,t)}{\partial x^4} &\simeq \frac{U_{j-2}(t) - 4U_{j-1}(t) + 6U_j(t) - 4U_{j+1}(t) + U_{j-2}(t)}{\Delta x^4}.
\end{aligned}$$

Inserting these expressions into the PDE gives the semi-discrete form

$$\frac{du_j}{dt} = -u_j \frac{u_{j+1} - u_{j-1}}{2\Delta x} - \frac{u_{j-1} - 2u_j + u_{j+1}}{\Delta x^2} - \frac{u_{j-2} - 4u_{j-1} - 6u_j + 4u_{j+1} + u_{j-2}}{\Delta x^4} \tag{19}$$

for $j = 1, 2 \dots, N-1$ with periodic boundary conditions

$$u_{j+N}(t) = u_j(t) \quad \text{for} \quad \text{all } j$$

and initial condition

$$u_j(0) = U_0(x_j) \quad \text{for} \quad \text{all } j.$$

Here, $u_j(t)$ denotes the finite-difference approximation of the solution to (18). There are a couple ways to implement the right hand side vector field in (19). One can create a function that takes input $\boldsymbol{u}, t, \Delta x$ and fills the elements of the output vector field. This procedure is implemented in the MATLAB script RHS.m. Alternatively, we can cast the nonlinear system above into semi-discrete form

$$\frac{d\boldsymbol{u}}{dt} = -\boldsymbol{u} \circ \boldsymbol{D}^1_{\mathrm{FD}}\boldsymbol{u} - \boldsymbol{D}^2_{\mathrm{FD}}\boldsymbol{u} - \boldsymbol{D}^4_{\mathrm{FD}}\boldsymbol{u} \tag{20}$$

[2] To this end, you can use the Matlab commands:

```
surf(X,T,U);    shading interp;    view(0,90).
```

$$
\boldsymbol{D}_{\text{FD}}^{1} = \frac{1}{2\Delta x}\begin{bmatrix} 0 & 1 & 0 & 0 & \dots & \dots & -1 \\ -1 & 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & -1 & 0 & 1 \\ 1 & \dots & \dots & \dots & 0 & -1 & 0 \end{bmatrix}, \boldsymbol{D}_{\text{FD}}^{2} = \frac{1}{\Delta x^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & 1 & -2 & 1 \\ 1 & \dots & \dots & \dots & 0 & 1 & -2 \end{bmatrix}
$$

$$
\boldsymbol{D}_{\text{FD}}^{4} = \frac{1}{\Delta x^4}\begin{bmatrix} 6 & -4 & 1 & 0 & \dots & 1 & -4 \\ -4 & 6 & -4 & 1 & \dots & \dots & 1 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 1 & & & 1 & -4 & 6 & -4 \\ -4 & 1 & \dots & \dots & 1 & -4 & 6 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}.
$$

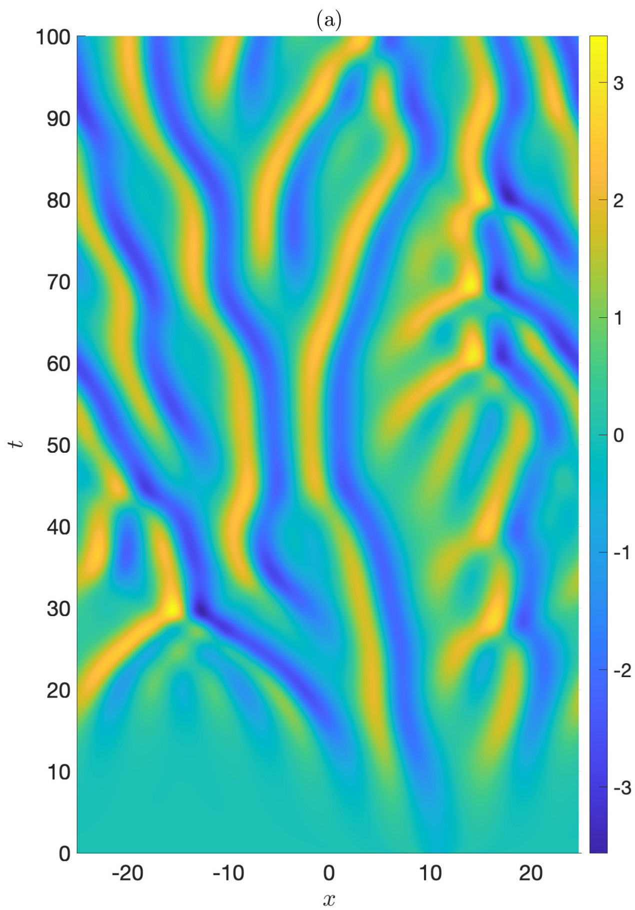
where $\circ$ denotes element-wise multiplication (Hadamard product). For computational efficiency, we implemented the first approach because it requires less number of floating point operations as it avoids the multiplication by zeros in the matrix vector multiplication in (20). Finally, we apply the Adams-Bashforth 2 step method given by

$$
\boldsymbol{u}_{k+2} = \boldsymbol{u}_{k+1} + \frac{\Delta t}{2}\left(3\boldsymbol{f}(\boldsymbol{u}_{k+1}) - \boldsymbol{f}(\boldsymbol{u}_k)\right)
$$

where $\boldsymbol{f}$ is the right hand side given in (19). We note that this is a two-step method and therefore requires a start up. In the script, we compute $\boldsymbol{u}_1$ using the Heun method.

(b) In figure 4(a), we have the plot the numerical solution with 200 space grid points and time step $\Delta t = \times 10^{-4}$. This is implemented in the Matlab code scripts `solve_KS.m`.

(b) In figure 4(b) we plot the numerical solution at time $t = 62$ as a function of x.



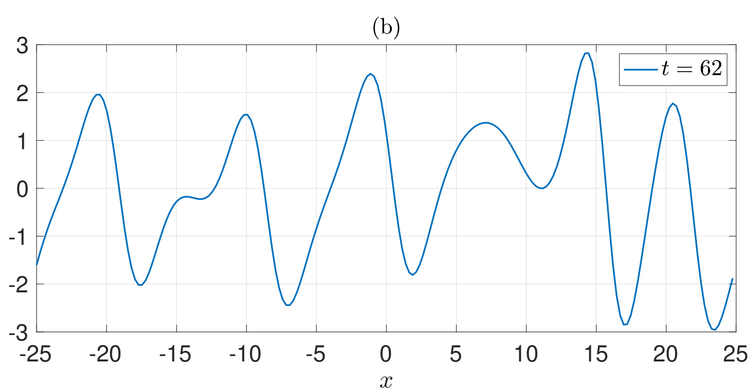


Figure 4: (a) Plot of the numerical solution of the IVP (18) on the requested space-time grid. (b) Solution at time $t = 62$ as a function of x.