

Homework 1. Due Feb. 8

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, github, google drive, etc. and place links to them in your pdf file with your solutions.

1. **(3 pts)** Solve Exercise 1 from my Lecture notes `ODEsolvers.pdf` (see Section 1.4). Note, $y(t)$ is a scalar function.
2. **(3 pts)** Solve Exercise 2 from my Lecture notes `ODEsolvers.pdf` (see Section 1.4). Note, $y(t)$ is a scalar function.
3. **(3 pts)** Solve Exercise 3 from my Lecture notes `ODEsolvers.pdf` (see Section 1.5).
4. **(3 pts)** Solve Exercise 5 from my Lecture notes `ODEsolvers.pdf` (see Section 3.4).
5. **(3 pts)** Show that the implicit midpoint rule (a symplectic method)

$$k = f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k\right), \quad u_{n+1} = u_n + hk. \quad (1)$$

is consistent (at least) of order 2.

6. **(6pts)**

- (a) Use the method of undetermined coefficients to determine a_0 , a_1 , b_0 , and b_1 that make the linear two-step explicit method consistent of as high order as possible:

$$u_{n+1} - a_0u_n - a_1u_{n-1} = h(b_0f_n + b_1f_{n-1}). \quad (2)$$

- (b) Show that this method is unstable by applying it to the ODE $y' = 0$ and mimicking the technique shown in Section 3.5.
- (c) Apply this method to the 2D gravity problem with a unit-circle solution:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix} = \begin{bmatrix} u \\ v \\ -\frac{x}{x^2+y^2} \\ -\frac{y}{x^2+y^2} \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ u \\ v \end{bmatrix} (0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (3)$$

Integrate the numerical solution for two periods. Show that the solution blows up by reporting the norm of the solution at time 4π for $h = 2\pi/N$ with $N = 20, 40, 80$. Also plot the x and y components of the solution in the xy -plane.

Homework 2. Due Feb. 15

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. The goal of this exercise to become familiar with various ODE solvers available in Matlab and/or Python and apply them to a few test problems.
 - (a) Read the description of available built-in ODE solvers in Matlab <https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html> and in Python https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html#scipy.integrate.solve_ivp. (For Python, see also [more syntax options here](#) and [examples here](#))
 - (b) **(5 pts)** Test the ODE solvers available in Matlab **or** Python on [the Van Der Pol oscillator](#)
 - Matlab: `ode45` (explicit Runge-Kutta method of order 5 DOPRI5(4)) and `ode15s` (linear multistep method of variable order from 1 to 5 for stiff problems);
 - Python: `RK45` (explicit Runge-Kutta method of order 5 DOPRI5(4)) and `LSODA` (linear multistep method based on Adams and BDF with automatic stiffness detection):

$$\begin{aligned}y_1' &= y_2, \\ y_2' &= \mu((1 - y_1^2)y_2) - y_1.\end{aligned}$$

with $\mu = 10, 10^2, \text{ and } 10^3$. The greater is μ , the stiffer is the problem. Set $t_{\max} = 1000.0$. Set error tolerances $\epsilon \equiv atol \equiv rtol$, $\epsilon = 10^{-6}, 10^{-9}, \text{ and } 10^{-12}$. You can pick the initial condition $y_1(0) = 2, y_2(0) = 0$. Plot the solution on the phase plane (y_1, y_2) . For each value of μ , make a single plot of the solution. Observe how the limit cycle changes at μ increases. Measure the CPU time T_{CPU} required to compute one cycle for each of the solvers for each of the values of μ and plot $\log \epsilon$ versus $\log(\text{CPUtime})$. Comment on how the CPU time depends on the error tolerance for each value of μ for each method. Try to explain your observations.

- (c) **(5 pts)** Now test appropriate ODE solvers on [the Arenstorf problem](#). [More details are found in this note on pages 3-4](#). Use exactly the same values for the parameters and the initial conditions as in the note. First set the T_{\max} equal to the period, compute the periodic orbit using `DOPRI5(4)`, and plot it (x vs y). Then set $T_{\max} = 100$. Set error tolerance $\epsilon \equiv atol \equiv rtol = 10^{-12}$.

- If you program in Matlab, compute the numerical solution using `ode45`, `ode78`, and `ode89`.
- In Python, use `RK45`, `DOP853`, and `Radau`.

Measure CPU times and plot the orbit of the satellite for each solver. Comment on the CPU times and plots that you are observing and compare the solvers.

2. **(5 pts)** On the same coordinate plane, plot the regions of the absolute stability for the following methods ERK methods:
 - Forward Euler;
 - Midpoint rule with Euler Predictor (this is the example in my code `RK_RAS.ipynb` available on ELMS in Files/Codes);
 - Kutta's method (three-stage, third order) (see the last page of this PDF file);
 - the standard 4-stage, fourth order Runge-Kutta method (see the first method in [this Wiki article](#));
 - DOPRI5(4) – see the last page of this PDF file). Use \hat{y} for the method of order 5.
3. **(5 pts)** Prove convergence for the backward Euler method from scratch. You should mimic the general proof of convergence for one-step methods, but simplify and adapt it for backward Euler.

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Table 1: Kutta's method

0		
$\frac{1}{3}$	$\frac{1}{3}$	
$\frac{2}{3}$	0	$\frac{2}{3}$
$\frac{3}{3}$	$\frac{1}{4}$	$\frac{3}{4}$

Table 2: Heun's method

0						
$\frac{1}{5}$	$\frac{1}{5}$					
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$			
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$		
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$	
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$
y_1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$
\hat{y}_1	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$
						$\frac{1}{40}$

Table 3: Embedded Runge-Kutta Method: Dormand and Prince 5(4).

Homework 3. Due Feb. 22

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. **(5 pts)** Solve exercise 13 in my lecture notes `ODEsolvers.pdf`.
2. **(5 pts)** Solve exercise 14 in my lecture notes `ODEsolvers.pdf`.
3. **(6 pts)** There is a curious phenomenon called *order reduction* observed when DIRK methods are applied to stiff problems with time steps h such that $h\lambda$ where $-\lambda$ is the largest negative eigenvalue of the linear part of the right-hand side of an ODE is not small – see papers by B. Seibold’s group <https://arxiv.org/pdf/1712.00897.pdf> and <https://arxiv.org/pdf/1811.01285.pdf>. The goal of this exercise is to examine the performance of DIRK methods of orders 2 and 3 from the previous exercises on the Prothero-Robinson problem, plot the maximum absolute error as a function of the time step, and observe the two orders of convergence for each method, one for large hL , and one for small hL . You need to plot graphs similar to those in Fig. 2 in <https://arxiv.org/pdf/1811.01285.pdf>.

Consider the Prothero-Robinson problem

$$y' = -L(y - \phi(t)) + \phi'(t), \quad y(0) = y_0 \quad (1)$$

with $L = 10^4$ and $\phi(t) = \sin(t + \pi/4)$. Set the time interval $0 \leq t \leq T_{\max} = 10$. The exact solution to this problem is

$$y = e^{-Lt}(y_0 - \phi(0)) + \phi(t). \quad (2)$$

- (a) Pick the initial condition $y(0) = \sin(\pi/4)$. Compute the numerical solution using DIRK2 on the interval $[0, T_{\max}]$ with time step h for each h from the following set:

$$h = 10^{-p}, \quad \text{where } p \in \{1, 1 + d, 1 + 2d, \dots, 6\}, \quad d = 5/24. \quad (3)$$

Plot the numerical error $e(h) = \max_{0 \leq t_n \leq T_{\max}} |u_n - y(t_n)|$ vs h . Use the log-log scale. Observe error decay $e = C_1 h$ and $e = C_2 h^2$ for large and small values of h , respectively. For reference, plot lines with slopes 1 and 2, i.e., $e = C_1 h$ and $e = C_2 h^2$ where you need to choose C_1 and C_2 so that the plot looks nice. Do the same for the DIRK of order 3. What orders of error decay do you observe? Also, plot reference lines.

- (b) Repeat the task with $y(0) = \sin(\pi/4) + 10$. You will obtain a bit puzzling set of graphs. To understand what is going on, plot $|e(t)|$ for each method where $e(t)$ is the difference between the numerical and the exact solutions for three values of h : $h = 10^{-1}$, $h = 10^{-2}$, and $h = 10^{-3}$. Set the log scale in the y -axis. Do so for $T_{\max} = 10$ and $T_{\max} = 1$.
- (c) Summarize what you have learned about the behavior of the error for DIRK2 and DIRK of order 3 from the numerical experiments in this problem.
4. **(5 pts)** Derive a three-step Adams-Moulton method with a variable timestep. Then compute its coefficients for the case where the timestep is constant.

Homework 4. Due March 1

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. (4 pts) Solve exercise 16 in my lecture notes ODEsolvers.pdf.
2. (4 pts) Solve exercise 17 in my lecture notes ODEsolvers.pdf.
3. (4 pts) Solve exercise 18 in my lecture notes ODEsolvers.pdf.
4. (4 pts) Solve exercise 19 in my lecture notes ODEsolvers.pdf.
5. (5 pts)

(a) Derive the following formula for the 2-step BDF method with a variable timestep:

$$u_{n+1} - \frac{(1+\omega)^2}{1+2\omega}u_n + \frac{\omega^2}{1+2\omega}u_{n-1} = h_n \frac{1+\omega}{1+2\omega}f(t_{n+1}, u_{n+1}), \quad (1)$$

where $h_n := t_{n+1} - t_n$, $\omega = h_n/h_{n-1}$.

- (b) Prove that this method is stable provided that $\omega_n < 1 + \sqrt{2}$. *Hint: use the main theorem for methods with constant stepsize and **Vieta's formulas for quadratic equations**.*
6. (5 pts) Solve the **stiff Robertson's problem** on the time interval $[0, 40]$ using the **2-step BDF method** programmed from scratch. Use fixed timestep. I am providing my code `RobertsonDIRK2.ipynb` where this task is accomplished using DIRK2 from the previous homework with a fixed timestep $h = 10^{-3}$. You can merely add to my code. If you prefer Matlab, please feel free to rewrite it in Matlab. To initialize BDF2, you will need to make the first step by DIRK2. Plot the three components of the solution in different figures as they have very different scales. Measure the CPU time for Robertson's problem for DIRK2 and BDF2 with the same timestep.

Homework 5. Due March 8

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. (10 pts) Solve exercise 2 in my lecture notes `SymplecticMethods.pdf`.
2. (10 pts) Consider the motion in the gravitational field with the Hamiltonian

$$H(u, v, x, y) = \frac{1}{2}u^2 + \frac{1}{2}v^2 - \frac{1}{\sqrt{x^2 + y^2}}, \quad (1)$$

where x, y are the coordinates and u, v are the momenta in the reduced units.

- (a) Write the Hamiltonian equations of motion. Set the initial conditions to $u(0) = 0$, $v(0) = \frac{1}{2}$, $x(0) = 2$, $y(0) = 0$. Check that the total energy is negative, hence the motion will follow an elliptic trajectory.

The exact motion according to these Hamiltonian equations with the initial conditions $u(0) = 0$, $v(0) = \frac{1}{2}$, $x(0) = 2$, $y(0) = 0$ occurs by the elliptical orbit with one focus at the origin, the major semiaxis $a = 4/3$, eccentricity $e = 1/2$, and the exact period of revolution $T = 2\pi a^{3/2} = 9.673596609249161$. Hence, for the exact orbit, $x_{\max} = a(1 + e) = 2$, and $x_{\min} = -a(1 - e) = -2/3$.

- (b) Integrate the system for 10 revolutions using the implicit midpoint rule. Proceed as follows. Define

$$z := \begin{bmatrix} u \\ v \\ x \\ y \end{bmatrix}, \quad f(z) := \frac{dz}{dt};$$

then

$$k = f(z_n + \frac{h}{2}k), \quad z_{n+1} = z_n + hk.$$

At each time step, you will need to solve the 4D nonlinear system

$$F(k) := k - f(z_n + \frac{h}{2}k) = 0. \quad (2)$$

Find the initial approximation for k by linearizing f and solving the linear system:

$$k = f(z_n) + \frac{h}{2}Df(z_n)k,$$

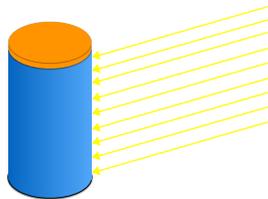
where $Df(z_n)$ is the Jacobian matrix of f evaluated at z_n . Then find the solution of Eq. (2) using Newton's iteration. Plot x and y components of your numerical solutions on the same xy -plane. Plot the Hamiltonian versus time for your numerical solution. Do this task with time steps such that there are 100, 1000, and 10000 steps per period. You should generate a total of 6 figures.

- (c) Integrate the same system using the Stoermer-Verlet method described in the previous problem using the same time steps and generate the same plots.
 - (d) Conclude which method is more accurate, while both of them are 2nd order.
3. **(3 pts)** Find the region of absolute stability of the implicit midpoint rule described in the previous problem.

Homework 6. Due March 15

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, github, google drive, etc. and place links to them in your pdf file with your solutions.

1. **(10 pts.) To be done on paper.** Consider Laplace's equation $u_{xx} + u_{yy} = 0$ in the given domain Ω with the given boundary conditions (BCs). Discretize the problem on the specified mesh using the 5-point stencil and write out an appropriate system of linear algebraic equations $Au = f$ for the numerical solution u as it is done in `elliptic.pdf`. Highlight the block structure of the matrix A .
 - (a) $\Omega = [0, 1] \times [0, 1]$, the mesh and the BCs are shown in Fig. (a).
 - (b) $\Omega = [0, 1] \times [0, 1]$, the mesh and the BCs are shown in Fig. (b).
 - (c) $\Omega = [0, 1] \times [0, 1]$, the mesh and the BCs are shown in Fig. (c). (BCs are periodic in y and Dirichlet on the left and right boundaries.)
 - (d) $\Omega = [0, 1] \times [0, 1]$, the mesh and the BC are shown in Fig. (d). (BCs are periodic in x and Dirichlet on the top and bottom boundaries.)
 - (e) Ω is the L-shaped domain shown in Fig. (e). Assume that the left and the bottom boundaries have length 1. The mesh and the BCs are shown in Fig. (e) as well.
2. **(10 pts.) A programming task.** Suppose an empty cylindrical tin covered with an insulating top is standing on ice. The sun located low above the horizon is shining on one side of the tin as shown in the figure.

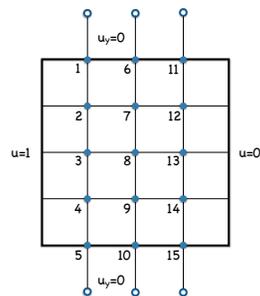


Assume that the height of the cylinder is 2 and the radius is 1. Take all physical constants such as the heat conductance and the intensity of the sun shine equal to 1. Then the stationary heat distribution satisfies the equation

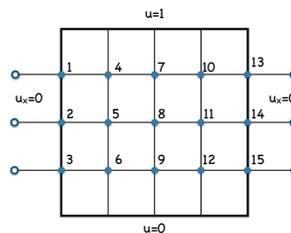
$$u_{xx} + u_{yy} = \begin{cases} -\cos(x), & -\pi/2 \leq x \leq \pi/2, \\ 0, & \text{otherwise,} \end{cases}$$

$-\pi \leq x \leq \pi, 0 \leq y \leq 2$. Pick adequate boundary conditions and write down the boundary value problem (BVP). Solve the BVP numerically and find the stationary temperature distribution. Plot a figure showing the distribution.

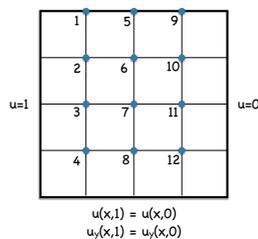
Submit the formulation of the BVP, the figure with the stationary heat distribution, and a printout of your code.



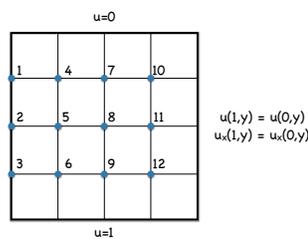
(a)



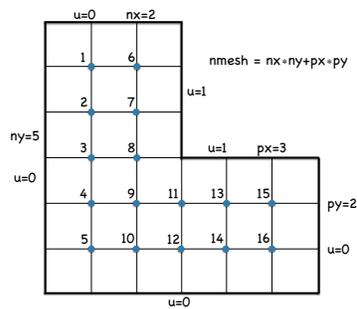
(b)



(c)



(d)



(e)

Homework 7. Due March 29

1. **(5 pts)** Consider a 1D boundary-value problem

$$-\frac{d}{dx} \left(k(x) \frac{d}{dx} u \right) = 0, \quad a < x < b, \quad u(a) = u_a, \quad u(b) = u_b, \quad (1)$$

where the heat conductance coefficient $k(x)$ is the following piecewise constant function

$$k(x) = \begin{cases} k_1, & a \leq x < c \\ k_2, & c < x \leq b \end{cases}. \quad (2)$$

It follows from the integral form of Fourier's law that the temperature u and the heat flux $k(x)u_x$ must be continuous at $x = c$.

- (a) Find the exact solution to this problem analytically.
 (b) Set $u_a = 0$, $u_b = 10$, $k_1 = 10$, $k_2 = 1$, $a = 0$, $b = 10$, $c = 4$. Choose $h = 1.0$. Use the finite difference scheme

$$L_h U_P = -\frac{1}{h^2} (U_W k_w + U_E k_e - U_P (k_e + k_w)) \quad (3)$$

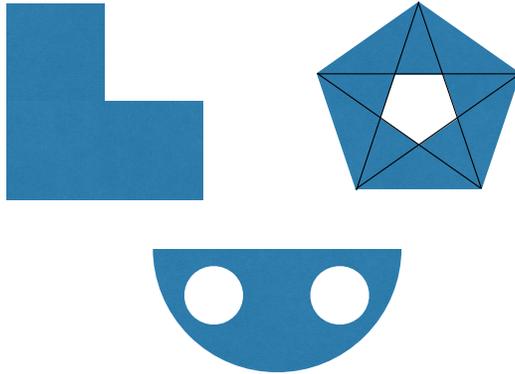
where W is to the left of P , E is to the right of P , and e and w are the midpoints between E and P and W and P respectively. You need to evaluate $k(x)$ at these midpoints. In this case, it is not a problem that $k(x)$ is not defined at c . Compute the numerical solution U using this scheme. Plot it and also plot the exact solution in the same figure. You should see that these solutions exactly coincide.

2. **(5 pts)** *Triangular mesh generation using `distmesh2d.m` by P.-O. Persson.* Read [1], at least its first 12 pages (at least up to Section 6 “Mesh Generation in Higher Dimensions”).

If you prefer Matlab, download the `distmesh` package `distmesh.zip` available at <http://persson.berkeley.edu/distmesh/>.

If you prefer Python, you can download my Python version of P.-O. Persson's code available at [GitHub](#), user `mar1akc`, package `transition_path_theory_FEM_distmesh`, file `distmesh.py`.

Mesh the shapes in the Figure below using `distmesh2d.m` following examples in [1].



You can pick arbitrary sizes as soon as topologies are preserved, and you can do uniform meshing.

3. **5 pts** Consider the following BVP in 1D:

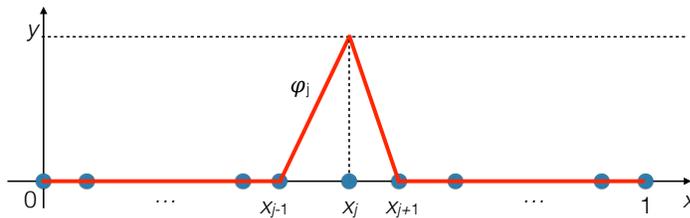
$$-u_{xx} = f(x), \quad 0 < x < 1, \quad u(0) = u(1) = 1.$$

Work out all steps of the FEM on it.

- (a) Let $w(x)$ be a twice continuously differentiable function on $(0, 1)$ such that $w(0) = w(1) = 0$. Use integration by parts to reduce the BVP to an integral equation.
- (b) Partition the interval $[0, 1]$:

$$0 = x_0 < x_1 < \dots < x_N < x_{N+1} = 1.$$

Define the basis functions $\phi_i(x)$, $1 \leq i \leq N$ as shown in the figure below ($\phi_i(x_i) = 0$, $\phi_i(x_j) = 0$, $j \neq i$, ϕ_i is piecewise linear).



$$\phi_i(x) = \begin{cases} 0, & x \leq x_{i-1}, \quad x \geq x_{i+1}, \\ \frac{x-x_{i-1}}{x_i-x_{i-1}}, & x_{i-1} < x \leq x_i, \\ \frac{x_{i+1}-x}{x_{i+1}-x_i}, & x_i < x < x_{i+1}. \end{cases}$$

Calculate the stiffness matrix and the load vector.

- (c) In what case the FEM solution would coincide with the finite difference solution using the central difference scheme?

4. (5 pts)

- (a) Let η_1 , η_2 , and η_3 be linear functions in a triangle T with vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) such that $\eta_j = 1$ at (x_j, y_j) and $\eta_j = 0$ at the other two vertices of T , $j = 1, 2, 3$. Prove that the matrix G introduced in Section 5 in [2] is equal to

$$G = \begin{bmatrix} \frac{\partial \eta_1}{\partial x} & \frac{\partial \eta_1}{\partial y} \\ \frac{\partial \eta_2}{\partial x} & \frac{\partial \eta_2}{\partial y} \\ \frac{\partial \eta_3}{\partial x} & \frac{\partial \eta_3}{\partial y} \end{bmatrix}. \quad (4)$$

- (b) Let u be a finite element solution to some problem. This means that u is continuous, piecewise linear, and linear within each mesh triangle. Let T be a mesh triangle with vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) , and $u(x_j, y_j) = u_j$, $j = 1, 2, 3$. Suppose that we need to compute the gradient of u . Find an exact expression for ∇u within the mesh triangle in terms of (x_j, y_j) and u_j , $j = 1, 2, 3$.

References

- [1] [P.-O. Persson, G. Strang, A Simple Mesh Generator in MATLAB. SIAM Review, Volume 46 \(2\), pp. 329-345, June 2004 \(PDF\)](#)
- [2] [Jochen Albrety, Carsten Carstensen and Stefan A. Funken, Remarks around 50 lines of Matlab: short finite element implementation](#)

Homework 8. Due Wednesday, April 5

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. **(10 pts)** You can use Python or Matlab. If you choose Matlab, you can modify the code *MyFEMcat.m*. If you choose Python, you can modify *FEM_TPT.py* and any *ipynb* notebook available on my GitHub: [Github](#).

Consider the problem of finding the density of the electric current in a thin square plate Ω made out of two metals with different conductivities $a = a_1$ and $a = a_2$ as shown in Fig. 1. The voltage $u = 0$ at the left side of the square, and $u = 1$ on the right side. Use the finite element method (FEM) to find the voltage and the density of the electric current inside the plate. Plot the voltage and the absolute value of the current density. Consider two cases: (a): $a_1 = 1.2$ and $a_2 = 1$; (b): $a_1 = 0.8$ and $a_2 = 1$. Comment on the distribution of the current in the plate. Link your codes to your pdf file.

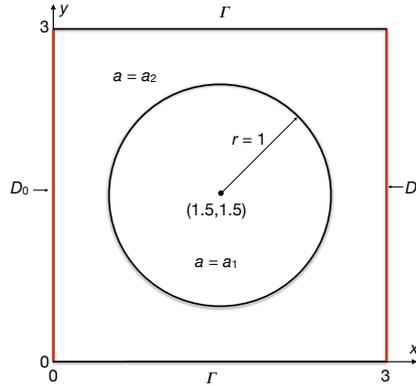


Figure 1: The conducting plate. An illustration to Problem 1.

The voltage is the solution to the following boundary value problem (BVP):

$$\begin{cases} -\nabla \cdot (a(x, y) \nabla u) = 0, & (x, y) \in \Omega := [0, 3]^2, \\ u = 0, & (x, y) \in D_0, \\ u = 1, & (x, y) \in D_1, \\ \frac{\partial u}{\partial n} = 0, & (x, y) \in \Gamma. \end{cases} \quad (1)$$

Note that the solution must satisfy the flux continuity condition: at the boundary separating the regions where $a(x, y) = a_1$ and $a(x, y) = a_2$, the voltage u and the current density $a(x, y)\nabla u$ must be continuous. This condition is automatically satisfied if you generate the mesh so that $a(x, y)$ is continuous in each mesh triangle and then solve the BVP using FEM. This can be easily achieved using e.g. `distmesh2d.m` requiring the points on the boundary separating these two regions to be fixed.

Once you have computed the voltage u , you can find the current density:

$$j = -a(x, y)\nabla u.$$

Use the formula for the gradient found in HW7, Problem 4(b). As a result, you have the gradient evaluated at the centers of the mesh triangles. Then compute the absolute value of the density of the current. In order to visualize it using `trisurf` you need to recompute it at the vertices of the mesh triangles. This can be done by averaging over all triangles adjacent to a given vertex:

```
abs_current_verts = zeros(Npts,1);
count_tri = zeros(Npts,1);
for j = 1:Ntri
    abs_current_verts(tri(j,:)) = abs_current_verts(tri(j,:)) ...
        + abs_current_centers(j);
    count_tri(tri(j,:)) = count_tri(tri(j,:)) + 1;
end
abs_current_verts = abs_current_verts./count_tri;
```

2. **(8 pts)** The goal of this problem is to show that the FEM solution of the Poisson equation is the linear interpolant of the exact solution. We will examine only the 1D case. This problem is a composite of problems 2.2a, 2.4, and 5.4 from [2].

Consider the BVP

$$-u'' = f(x), \quad 0 < x < 1, \quad u(0) = u(1) = 0, \quad (2)$$

where $f(x)$ is a given function integrable on $[0, 1]$.

- (a) Verify that the solution of Eq. (2) is given by

$$u(x) = \int_0^1 G(x, y)f(y)dy, \quad (3)$$

where $G(x, y)$ is Green's function defined in $[0, 1]^2$ by

$$G(x, y) = \begin{cases} (1-x)y, & 0 \leq y \leq x \leq 1, \\ x(1-y), & 0 \leq x \leq y \leq 1. \end{cases} \quad (4)$$

- (b) Verify that for any function $v(x) \in H_0^1([0, 1])$, the following identity holds:

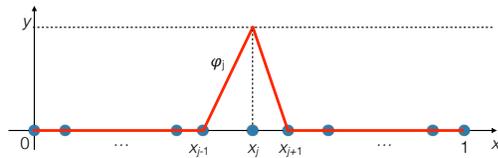
$$\int_0^1 v'(y) \frac{d}{dy} G(x, y) dy = v(x). \quad (5)$$

Conclude that since the exact solution u of Eq. (2) belongs to $H_0^1([0, 1])$, Eq. (5) holds for u for all $x \in [0, 1]$.

- (c) The linear interpolant of the exact solution $u(x)$, denoted by $I_h u$, is a function that is equal to u at the nodes

$$0 \equiv x_0 < x_1 < \dots < x_n < x_{n+1} \equiv 1,$$

and is linear in each interval $[x_j, x_{j+1}]$, $j = 0, 1, \dots, n$. (We do not assume that the nodes are equispaced.) Write $I_h u$ as a linear combination of the basis functions shown in fig. below.



- (d) Show that the set of functions $\psi_j(y) := G(x_j, y)$ is a basis in $S_h(x) = \text{span}\{\phi_1, \dots, \phi_n\}$.

- (e) Show that for the linear interpolant $I_h u$,

$$\int_0^1 [I_h u]'(y) \frac{d}{dy} G(x_j, y) dy = u(x_j). \quad (6)$$

- (f) Write out the definition of the FEM solution. Prove, using all the facts above that the FEM solution

$$U^{FEM}(x) = \sum_{j=1}^n U_j^{FEM} \phi_j(x)$$

coincides with $I_h u$.

3. (5 pts) This problem suggests a way to modify a non-self-adjoint elliptic equation to make it amenable for solving using FEM.

- (a) Let Ω be a compact domain in 2D. The boundary of Ω can be decomposed as $\partial\Omega = \Gamma_D \cup \Gamma_N$. Consider the following boundary-value problem:

$$\begin{cases} \beta^{-1} \Delta u + b(x) \cdot \nabla u = 0, & x \in \Omega \\ u = u_D & x \in \Gamma_D \\ \frac{\partial u}{\partial n} = g, & x \in \Gamma_N. \end{cases} \quad (7)$$

Let b be a smooth vector field and β be a constant. Suppose it can be decomposed into curl-free and divergence-free components:

$$b = -\nabla V(x) + f(x). \quad (8)$$

The curl-free component is ∇V and the divergence-free component is f : $\nabla \cdot f = 0$. Show that BVP (7) is equivalent to

$$\begin{cases} \nabla \cdot (e^{-\beta V(x)} \nabla u) + \beta e^{-\beta V(x)} f(x) \cdot \nabla u = 0, & x \in \Omega \\ u = u_D & x \in \Gamma_D \\ \frac{\partial u}{\partial n} = g, & x \in \Gamma_N. \end{cases} \quad (9)$$

(b) Consider the Maier-Stein drift field

$$b(x, y) = \begin{bmatrix} x - x^3 - 10xy^2 \\ -(1 + x^2)y \end{bmatrix}. \quad (10)$$

Decompose $b(x, y)$ into divergence-free and curl-free components. Find the potential function $V(x, y)$.

References

- [1] [Jochen Albrety, Carsten Carstensen and Stefan A. Funken, Remarks around 50 lines of Matlab: short finite element implementation](#)
- [2] S. Larsson and V. Thomee, *Partial Differential Equations with Numerical Methods*, Springer-Verlag Berlin Heidelberg, 2003, 2009 (soft cover)

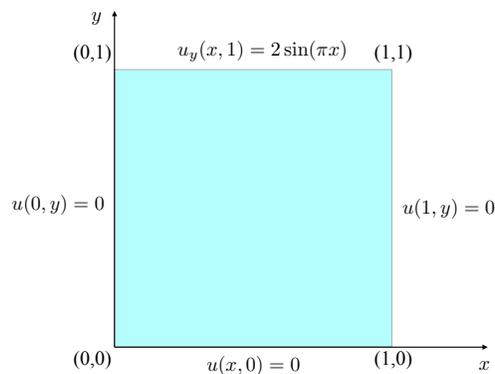
Homework 9. Due Wednesday, April 12

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. (10 pts) Set up and solve Problem 6 from Lagaris, Likas, and Fotiadis (1998), a boundary-value problem for the Poisson equation

$$u_{xx} + u_{yy} = f(x, y) \quad \text{where} \quad f(x, y) = (2 - \pi^2 y^2) \sin(\pi x), \quad (x, y) \in [0, 1]^2, \quad (1)$$

with mixed boundary conditions:



The exact solution to this problem is given by

$$u_{ex}(x, y) = y^2 \sin(\pi x). \quad (2)$$

To solve this problem, you can mimic the provided codes for Problem 5 from Lagaris et al.:

- Lagaris5.zip, a Matlab package written by me from scratch, or
- Lagaris5.ipynb, a Python code written by me from scratch, or
- Lagaris_Margot.ipynb, a Python written by Margot Yuan (AMSC) with the use of automatic differentiation and built-in Adam optimizer.

Use a neural network with one hidden layer and $n = 10$ neurons of the form:

$$\mathcal{N}(x, y; w) = \sum_{j=0}^{n-1} w_{3j} \sigma(w_{0j}x + w_{1j}y + w_{2j}), \quad (3)$$

where $\sigma(z) = \tanh(z)$ acts entrywise. The total number of parameters to optimize is $4n$.

The proposed form of the solution $U(x, y; w)$ is given in Lagaris et al. in Eqs. (24)–(25). Take $N_{tr} = 49$ training points forming a uniform 7×7 `meshgrid` in $[0, 1]^2$ (set a 9×9 meshgrid and strip its boundaries). Set up the least squares loss function

$$\mathcal{L}(w) = \frac{1}{2} \sum_{i=0}^{N_{tr}-1} |U_{xx}(x_i, y_i; w) + U_{yy}(x_i, y_i; w) - f(x_i, y_i)|^2. \quad (4)$$

Plot the computed solution and the numerical error using level sets or a heatmap. Plot the loss function versus the iteration number using a log scale for the y-axis.

2. **(5 pts)** Consider the following IBVP for the heat equation:

$$u_t = u_{xx}, \quad 0 \leq x \leq \pi, \quad t \geq 0, \quad (5)$$

$$u(x, 0) = x, \quad (6)$$

$$u(0, t) = 0, \quad \text{the left end is kept at temperature 0,} \quad (7)$$

$$u_x(\pi, t) = 0, \quad \text{the right end is insulated.} \quad (8)$$

Solve Eq. (5)–(8) analytically using the method of separation of variables (if you are not familiar with it look it up e.g. in [Wikipedia](#)). You will obtain the solution of the form of infinite series involving sines $\sin[(n + 1/2)x]$, $n = 0, 1, 2, \dots$. Calculate the coefficients for each harmonic of this series. Plot graphs of the sums of the first m terms of the series at time 0 and time 2 for $m = 1, 2, 5, 10, 100$. Find the maximum norm of the difference of the sum of the first 100 terms of the series at time 0 and the function $x = u(x, 0)$.

Homework 10. Due April 19

1. (5 pts) Consider the heat equation in the unit square $[0, 1]^2$

$$u_t = \Delta u, \quad (x, y) \in [0, 1]^2. \quad (1)$$

Impose the homogeneous Dirichlet boundary conditions: $u = 0$ on $\partial\Omega$. Consider the uniform square mesh $(J+1) \times (J+1)$ in the unit square $[0, 1]^2$ and discretize Laplace's operator to this mesh using central differences and taking the boundary conditions into account. The resulting "method of lines" equation will be of the form

$$\frac{d}{dt}U = AU, \quad (2)$$

where A is the discretized operator Δ , an $(J-1)^2 \times (J-1)^2$ matrix.

- (a) Verify that the eigenvectors of A are

$$v_{k_x, k_y}(x_r, y_s) = \sin(k_x x_r) \sin(k_y y_s), \quad k_x, k_y = \pi, 2\pi, \dots, (J-1)\pi, \quad (3)$$

where (x_r, y_s) are the mesh points. Find the corresponding eigenvalues.

- (b) Suppose we are applying the forward Euler time discretization to (2). Find the relationship between the time step Δt and the mesh step h such that this method is stable for all J .

2. (10 pts) Consider the following Initial and Boundary Value Problem (IBVP) in 2D:

$$u_t = \Delta u + 1, \quad (x, y) \in \Omega = \{(x, y) \in \mathbb{R}^2 \mid 1 < r < 2\}, \quad (4)$$

$$u|_{t=0} = r + \cos(\phi), \quad (5)$$

$$u|_{r=1} = u|_{r=2} = 0, \quad (6)$$

where r and ϕ are the polar coordinates. Solve this problem using the finite element method and a scheme based on the trapezoidal rule:

$$u_{n+1} = u_n + \frac{1}{2}\Delta t (\Delta u_{n+1} + \Delta u_n) + \Delta t.$$

- (a) Derive equations for the weak and the FEM solutions of the IBVP (4)-(6) analogous to Eq. (13) and the two unnumbered equations right below it in Section 9 on page 127 in [Remarks around 50 lines of Matlab: short finite element implementation](#). Use time step `dt = 0.01`.
- (b) Make your program plot the following figures:
- with the computed solution at $t = 0.1$ (use `trisurf`);

- with the computed solution at $t = 1$ (use `trisurf`);
- with the computed solution at time $t = 1$ as a function of r . You can do it e.g., as follows:

```

u = U(:,N+1); % N+1 corresponds to t=1.
r = sqrt(coordinates(:,1).^2 + coordinates(:,2).^2);
[rsort, isort] = sort(r, 'ascend');
usort = u(isort);
plot(rsort, usort, 'Linewidth', 2);

```

At $t = 1$, the function u will virtually reach the stationary solution $\Delta u + 1 = 0$ satisfying the BC (6). This stationary solution can be found exactly:

$$u(r) = \frac{1 - r^2}{4} + \frac{3 \log(r)}{4 \log 2}. \quad (7)$$

Plot the graph of the exact stationary solution (7) in the same figure.

Hint: You might find helpful my code `MyFEMheat.m` implementing the Backward Euler time integrator described in “Remarks around 50 lines of Matlab: ...”. This code is found on ELMS in Files/Codes/Elliptic/MyFEMcat.

Homework 11. Due April 26

Reading:

- [1], Chapter 10, in particular, Section 10.9 (Modified equations) and Appendix E3 (Fourier analysis).
- My lecture notes `Hyperbolic.pdf`.

1. (15 pts) Consider the wave equation of an infinite line:

$$\begin{aligned} u_{tt} &= a^2 u_{xx}, \\ u(x, 0) &= \phi(x), \quad u_t(x, 0) = \psi(x). \end{aligned} \tag{1}$$

The parameter $a > 0$ is the speed of wave propagation.

(a) Verify that the solution is given by d'Alembert's formula

$$u(x, t) = \frac{1}{2}(\phi(x + at) + \phi(x - at)) + \frac{1}{2a} \int_{x-at}^{x+at} \psi(s) ds. \tag{2}$$

(b) The domain of influence of a point x_0 is the set of points on the (x, t) -plane such that $u(x, t)$ depends on the initial conditions at the point x_0 . More precisely, a point (x, t) belongs to the domain of influence of x_0 , if for any neighborhood of x_0 there exist perturbations of the initial conditions supported within this neighborhood such that the solution changes at the point (x, t) due to these perturbations.

Plot the domain of influence of a point x_0 on the (x, t) -plane.

The domain of dependence of a point (x, t) is the minimal set of points on the line $(x, 0)$ such that the solution at the point (x, t) is completely determined by the initial conditions restricted to this set. (The word minimal in the clause "minimal set satisfying some conditions" means that no its proper subset satisfies these conditions.)

Plot the domain of dependence of a point (x, t) .

(c) Rewrite Eq. (1) as a hyperbolic system. Proceed as follows. Introducing the new vector-function

$$w = \begin{bmatrix} u_t \\ u_x \end{bmatrix}.$$

Derive the equation for w : $w_t = Aw_x$, where

$$A = \begin{bmatrix} 0 & a^2 \\ 1 & 0 \end{bmatrix}.$$

Express the initial conditions for w in terms of ϕ and ψ .

- (d) Diagonalize the matrix A , i.e., find its eigenvalues and eigenvectors (λ_1, v_1) and (λ_2, v_2) and rewrite it of the form $A = C\Lambda C^{-1}$, where $C = [v_1, v_2]$, $\Lambda = \text{diag}\{\lambda_1, \lambda_2\}$. Introduce the new variable

$$y = \begin{bmatrix} \xi \\ \eta \end{bmatrix} \equiv C^{-1}w.$$

Observe that the system for y is decoupled into two independent advection equations:

$$\xi_t = \lambda_1 \xi_x, \quad \eta_t = \lambda_2 \eta_x. \quad (3)$$

Obtain the initial conditions for ξ and η in terms of ϕ and ψ .

- (e) Set $\phi = \max\{1 - |x|, 0\}$, $\psi = 0$, $a = \sqrt{2}$, $h = 0.05$, and a reasonable time step k . Pick the numerical domain $-6 \leq x \leq 6$ and periodic boundary conditions. Solve the equations for ξ and η numerically using Lax-Friedrichs, appropriate Upwind (left for one and right for the other depending on signs of λ), Lax-Wendroff, and appropriate Beam-Warming methods. Return to the variable w and then to the original variable u . Plot the numerical solutions u obtained using each of the methods at times $t = 1/(2a)$, $t = 1/a$, $t = 2/a$, and $t = 4/a$ as well as the exact solution at these times given by Eq. (2). Write a summary of your observations.

2. (15 pts)

Consider the advection equation

$$u_t + au_x = 0, \quad u(x, 0) = \phi(x), \quad -\infty < x < \infty, \quad t \geq 0. \quad (4)$$

Assume that $a > 0$.

- (a) Derive the modified equations for the Lax-Friedrichs method (LaxFr), the Leapfrog (LF) and the Left Beam Warming (BWL) method (Eq. (10.26) in [1]). For convenience, you can introduce the parameter $\nu := ak/h$ where $k \equiv \Delta t$ is the step in t , h is the step in x and make coefficients the modified equations more compact using it.
- (b) Consider the modified equations for the Upwind Left (UL), Lax-Wendroff (LW) (derived in class, see [1]) and the ones that you derived for LaxFr, LF, and BWL. UL and LaxFr are first order, while LW, LF, and BWL are second order. Use their modified equations for determining which of the first-order methods is most accurate and which of the second-order methods is the most accurate. Comment on the choice of the time step k given a and space step h that would lead to the smallest numerical error in each of these five methods.

- (c) Apply the Fourier transform (Eq. (E.18) in Appendix E in [1]) to the modified equations for UL, LaxFr, LW, LF, and BWL. Solve them in the Fourier space (i.e., solve the ODEs for \hat{u}). Comment on the behavior of the Fourier modes $\hat{u}(\xi, t)$ as $t \rightarrow \infty$. Obtain the dispersion relations (Eq. (E.48) in [1]) and find the phase and group velocities. What are the phase and the group velocity for the original advection equation? Run the code `advection.m` available on ELMS. The choice of method is controlled by the variable `method` in the code. Explain the appearance and the location of oscillations produced by these methods using their modified equations.

References

- [1] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM 2007 (Chapter 10, Appendix E)

Homework 12. Due May 3.

1. (10 pts) Consider the [Greenberg traffic model](#)

$$\rho_t + [-\rho \log(\rho)]_x = 0, \quad \rho(x, 0) = \rho_0(x). \quad (1)$$

Here, ρ is the density of cars, and the velocity v depends on the density according to $v(\rho) = v_{\max} \log\left(\frac{\rho_{\max}}{\rho}\right)$, where v_{\max} and ρ_{\max} are set to be 1 for convenience.

- (a) Find the formula for the characteristic $x(t)$ of Eq. (1) starting at the point $(x = x_0, t = 0)$ (the curve $x(t)$ passing through $(x = x_0, t = 0)$ along which ρ is constant, i.e., $\frac{d}{dt}\rho(x(t), t) = 0$).
- (b) Plot the characteristics and the shock line on the xt -plane for the Riemann problem

$$\rho_0(x) = \begin{cases} 0.1, & x < 0, \\ 0.9, & x > 0. \end{cases}$$

- (c) Suppose

$$\rho_0(x) = 0.5 + \frac{0.9}{\pi} \arctan(x). \quad (2)$$

Find the time when the shock appears. Then find the eventual shock speed.

References

- [1] [R. J. LeVeque, Numerical Methods for Conservation Laws, Second Edition, Birkhauser, Basel, Boston, Berlin, 1992](#)
- [2] M. Cameron's notes [burgers.pdf](#) available on ELMS.

Homework 13. Due Monday, May 10

1. (5 pt)

(a) Show that MacCormack's method

$$\begin{aligned} U_j^* &= U_j^n - \frac{k}{h} [f(U_{j+1}^n) - f(U_j^n)], \\ U_j^{n+1} &= \frac{1}{2} (U_j^n + U_j^*) - \frac{k}{2h} [f(U_j^*) - f(U_{j-1}^*)], \end{aligned} \quad (1)$$

reduces to the Lax-Wendroff method for $f(u) \equiv au$.

(b) Show that MacCormack's method is second-order consistent on smooth solutions.

(c) Determine a numerical flux function for MacCormack's method that allows us to rewrite it in the conservative form. Rewrite it in the conservative form. Show that the method is consistent.

2. (5 pts) Consider Godunov's method for solving $u_t + [f(u)]_x = 0$. In class we have established that if $f(u)$ is convex (if f is twice differentiable then $f''(u) > 0$), the following four cases exhaust all possibilities:(a) $f'(u_L) \geq 0$ and $f'(u_R) \geq 0$. Then $u^* = u_L$.(b) $f'(u_L) \leq 0$ and $f'(u_R) \leq 0$. Then $u^* = u_R$.(c) $f'(u_L) \geq 0 \geq f'(u_R)$. Then

$$u^* = \begin{cases} u_L, & \text{if } \frac{f(u_L) - f(u_R)}{u_L - u_R} > 0, \\ u_R, & \text{if } \frac{f(u_L) - f(u_R)}{u_L - u_R} < 0. \end{cases} \quad (2)$$

(d) $f'(u_L) < 0 < f'(u_R)$. Then $u^* = u_s$ (transonic rarefaction), where the value u_s is such that $f'(u_s) = 0$. It is called the *sonic point*. For example, for the Burgers equation $u_t + [u^2/2]_x = 0$, $u_s = 0$.

In the first three cases, the value u^* is either u_L and u_R , and it can be simply determined by Eq. (2). Note that in Cases 1 and 2, u^* is the same whether the physically correct weak solution to the Riemann problem is a shock wave or a rarefaction. Only in Case 4, the transonic rarefaction, the value of u^* differs from the one determined by Eq. (2). This is the value of u for which the characteristic speed is zero.

Verify that the numerical flux determined by Cases 1 - 4 can be rewritten more compactly as

$$F(u_L, u_R) = \begin{cases} \min_{u_L \leq u \leq u_R} f(u), & \text{if } u_L \leq u_R, \\ \max_{u_R \leq u \leq u_L} f(u), & \text{if } u_L > u_R. \end{cases} \quad (3)$$

Remark: It was proven that the numerical flux given by Eq. (3) gives the physically correct flux for scalar conservation laws even if $f(u)$ is non-convex.

3. **(5 pts)** Consider the Burgers equation $u_t + [\frac{1}{2}u^2]_x = 0$ with the initial condition $u_0(x) = 1$ on $[0, 1]$ and $u_0(x) = 0$ otherwise. Implement the following methods for conservation laws: Lax-Friedrichs, Richtmyer, MacCormack, and Godunov and apply them to the problem above. Compute the numerical solution by each of the methods with the same time step and plot it at times $t = 0, 1, 2, 3, 4, 5, 6$. Plot the exact solution as well. It is found in `Hyperbolic.pdf` in Section 8.3.
4. **5 pts** Read an article on the Kuramoto-Sivashinsky equation available at <http://people.maths.ox.ac.uk/trefethen/pdectb/kuramoto2.pdf>. A detailed description of the method can be found in `Kassam&Trefethen (2005)`.

Solve the equation

$$u_t + u_{xxxx} + u_{xx} + \frac{1}{2}(u^2)_x = 0, \quad u(x, 0) = \cos(x/16)(1 + \sin(x/16)) \quad (4)$$

on the interval $[0, 32\pi]$ with periodic boundary condition. Proceed as follows. Assume first that you need to solve

$$u_t = -u_{xxxx} - u_{xx} := Lu. \quad (5)$$

Write

$$u(x, t) = \sum_{k=-\infty}^{\infty} u_k(t)e^{ikx/16}.$$

Plug this into the equation and obtain an exact solution $u(x, t)$ of Eq. (5). Define the solution operator e^{tL} so that $u(x, t) = e^{tL}u(x, 0)$. Now return to Eq. (4). Note that $u_t = Lu + N(u)$ where $N(u) := -\frac{1}{2}(u^2)_x$. Define a new unknown function $v(x, t)$ by $u(x, t) = e^{tL}v(x, t)$. Plug this into $u_t = Lu + N(u)$ and obtain the following equation for $v(x, t)$:

$$v_t = e^{-tL}N(e^{tL}v). \quad (6)$$

Solve Eq. (6) using 4th order Runge-Kutta method on the time interval $[0, 200]$. Plot the surface $u(x, t)$ using the command `imagesc`. Compare it with the one in the article above.

Hint: modify the program `KdVrk.m` that solves the Korteweg-de Vries equation

$$u_t + u_{xxx} + \frac{1}{2}(u^2)_x = 0$$

using the proposed approach.