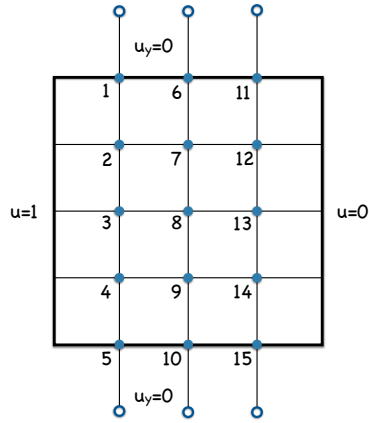


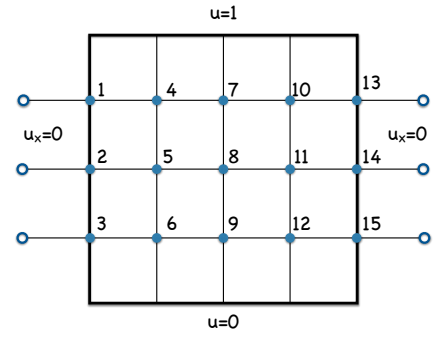
Homework 1. Due Feb. 10 (In class)

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.

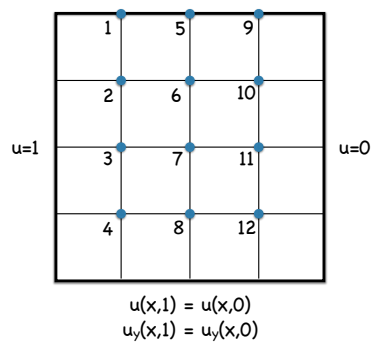
2. **(10 pts.) To be done in Matlab.** Write Matlab codes for solving Laplace's equation on the specified domain with the specified BCs. To set up the matrix A , use the commands `spdiags`, `speye`, `kron` (read Matlab's help about them if necessary). First set the mesh sizes as in the previous task and verify that the programmed matrix A coincides with the one you obtained. Then refine your mesh to make $nx = ny = 199$ (this will work if your matrix A is set up as sparse) and obtain the numerical solution. Set up `colormap gray` (then you do not need a colored printer) and use the command `contourf` to plot your numerical solution. Please submit printouts of your codes and plots of the solutions.
 - (a) Do it for the domain and the BCs shown in Fig. (d).
 - (b) Do it for the L-shaped domain with the BCs shown in Fig. (e) (in this case, the mesh should be refined so that $nx + px = ny = 199$).



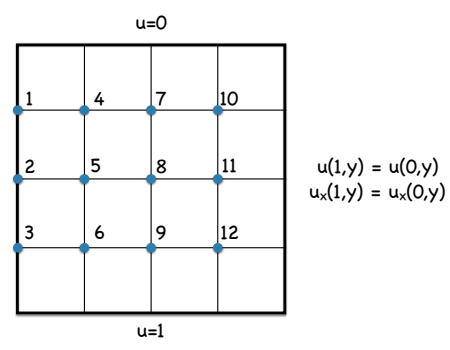
(a)



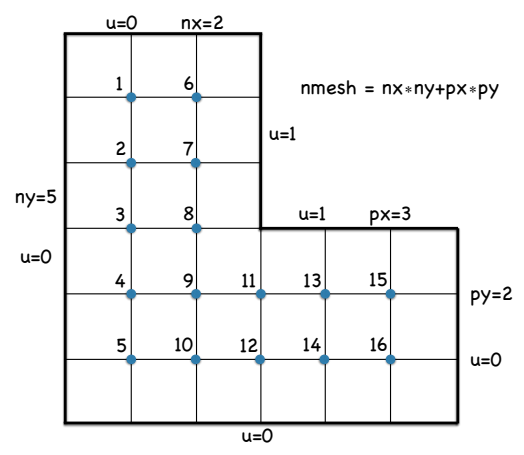
(b)



(c)



(d)



(e)

Homework 2. Due Feb. 17 (In class)

1. **(10 pts)** This problem is concerned with spectral error analysis of the finite difference method for the Poisson equation in 1D. This approach can be readily extended to 2D.

Consider the following 2-point BVP for the Poisson equation:

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

Suppose we have partitioned the interval $[0, 1]$ into $N + 2$ points $0 = x_0 < x_1 < \dots < x_N < x_{N+1} = 1$ where $x_j = jh$, $h = 1/(N + 1)$, and solved the BVP using the finite difference method. The numerical solution $U = [U_1, \dots, U_N]^T$ satisfies the linear equation

$$\frac{1}{h^2}AU = F + G \quad (2)$$

where

$$A := \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 \end{bmatrix}, \quad F = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{bmatrix}, \quad G = \begin{bmatrix} a/h^2 \\ 0 \\ \vdots \\ 0 \\ b/h^2 \end{bmatrix}.$$

- (a) Let $u(x)$ be the exact solution of BVP (1). Plug it into the difference equation (2) and calculate the truncation error $t = [t_1, \dots, t_N]^T$ using Taylor expansion.
 (b) Show that the error $e := U - u$ satisfies the equation

$$\frac{1}{h^2}Ae = -t. \quad (3)$$

- (c) Show that the eigenvalues of A are $\lambda_k = 4 \left(\sin \frac{\pi k}{2N+2} \right)^2$ and the corresponding eigenvectors are $v_k = [\sin \left(\frac{\pi k}{N+1} \right), \sin \left(\frac{2\pi k}{N+1} \right), \dots, \sin \left(\frac{N\pi k}{N+1} \right)]^T$, $k = 1, 2, \dots, N$.
 (d) Expand e and t in the basis $V = [v_1, \dots, v_N]$ as

$$e = V\gamma \equiv \sum_{k=1}^N v_k \gamma_k \quad \text{and} \quad t = V\tau \equiv \sum_{k=1}^N v_k \tau_k,$$

write $A = V\Lambda V^T$, and insert all of this into Eq. (3). Obtain the relationship

$$\gamma = -h^2 \Lambda^{-1} \tau. \quad (4)$$

- (e) Find the exact 2-norm of Λ^{-1} and an approximation to it for large N . (The 2-norm of $x \in \mathbb{R}^N$ is $\|x\|_2 \equiv (x^T x)^{1/2}$. The 2-norm of a *symmetric* matrix M is $\|M\|_2 = \max_{1 \leq k \leq N} |\mu_k|$ where μ_k s are eigenvalues of M .)
- (f) Observe that $\|e\|_2 = \|V\gamma\|_2 = (\gamma^T V^T V \gamma)^{1/2} = \|\gamma\|_2$ and, similarly, $\|t\|_2 = \|\tau\|_2$. Using this obtain an upper bound on $\|e\|_2$.

2. **(5 pts)** Consider the BVP

$$-\nabla \cdot (a(x, y) \nabla u) = f(x, y), \quad (x, y) \in \Omega := [0, 1]^2, \quad u = 0, \quad (x, y) \in \partial\Omega, \quad (5)$$

where $a(x, y)$ is a given smooth and positive function Ω , and $f(x, y)$ is a given continuous function in Ω . Consider the finite difference scheme given by

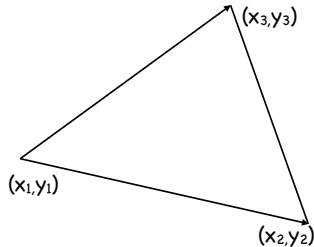
$$\frac{a_s \frac{u_S - u_P}{hy} - a_n \frac{u_P - u_N}{hy}}{hy} + \frac{a_e \frac{u_E - u_P}{hx} - a_w \frac{u_P - u_W}{hx}}{hx} = -f_P \quad (6)$$

(see Fig. 5 in `elliptic.pdf`), where a_n , a_s , a_e , and a_w are defined as

$$a_n = \frac{1}{2}(a_P + a_N), \quad a_s = \frac{1}{2}(a_P + a_S), \quad a_e = \frac{1}{2}(a_P + a_E), \quad a_w = \frac{1}{2}(a_P + a_W).$$

Calculate the truncation error for scheme (6).

3. **(5 pts)** Suppose the triangle Δ with vertices (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) is a triangular element of a mesh.



Consider basis functions $\phi_1(x, y)$ and $\phi_2(x, y)$ that are linear on this triangle and defined by

$$\begin{aligned} \phi_1(x_1, y_1) &= 1, & \phi_1(x_2, y_2) &= \phi_1(x_3, y_3) = 0, \\ \phi_2(x_2, y_2) &= 1, & \phi_2(x_1, y_1) &= \phi_2(x_3, y_3) = 0. \end{aligned}$$

- (a) Calculate $\nabla\phi_1$ and $\nabla\phi_2$. You should obtain rational functions of $x_1, x_2, x_3, y_1, y_2, y_3$. Observe that the denominators in your expressions are multiples of the area of the triangle

$$S = \frac{1}{2} \det \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}.$$

Simplify the expressions for $\nabla\phi_1$ and $\nabla\phi_2$ using this observation.

(b) Calculate

$$\iint_{\Delta} \nabla\phi_1(x, y) \cdot \nabla\phi_2(x, y) dx dy. \quad (7)$$

(c) Suppose all angles of the triangle Δ are less than $\pi/2$. What will be the sign of the integral (7)?

Homework 3. Due Feb. 24 (In class)

1. **(10 pts)** This problem is concerned with solving the Poisson equation using the finite Element method.

- Download all files from http://www.wire.tu-bs.de/docs/lehre/veranstaltungen/uebungen/pde2_s05/fem2d/ and read pages 117-126 in [1]. Also download the `mesh2d` package is available at mathwork file exchange. As you download the package, make sure to make the following corrections for the contemporary matlab versions:

```
mytsearch.m Line 68:  change to
i(j)=tsearchn([x y],t,[xi(j) yi(j)]);
meshfaces.m Line 199: change to
i = tsearchn(ph,th,p);
```

Put all files from these two downloads to the same directory.

- Create any two nontrivial smooth contours Γ_N and Γ_D so that Γ_D lies inside Γ_N . For example, you can take some image you like and extract two contours from there. Denote the region bounded by these two contours by Ω . Pose the following BVP:

$$-\nabla^2 u = 1, \quad (x, y) \in \Omega \quad (1)$$

$$\frac{\partial u}{\partial n} = 0, \quad (x, y) \in \Gamma_N, \quad (2)$$

$$u = 0, \quad (x, y) \in \Gamma_D. \quad (3)$$

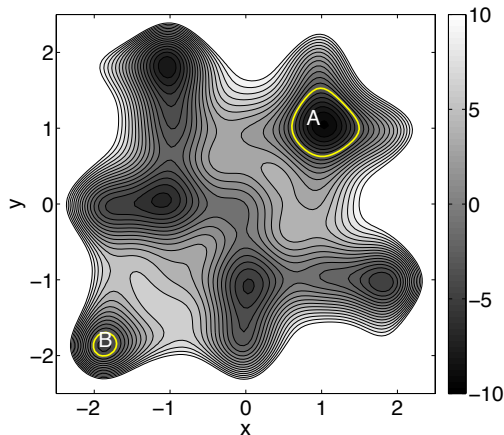
Remark. The solution of this problem has the following probabilistic interpretation. Suppose a particle is moving randomly in the domain Ω with reflecting boundary Γ_N according to the SDE $dz = dw$, $z = (x, y)$, where w is the standard Brownian motion (i.e., the velocity of the particle is the 2D white noise). Suppose the particle is at the point (x, y) at time $t = 0$. Then the solution $u(x, y)$ is the expected first passage time of this particle to the interior of the contour Γ_D .

- Solve BVP (1)-(3) using FEM on the mesh generated using the function `mesh2d`. Place all FEM-related codes (but not `mesh2d`) to the same m file. Submit a printout of your code and figure that its generate.

2. **(10 pts)** The 7-well potential

$$V(x, y) = x^4 + y^4 - 2(x - \cos(\pi y))^2 - 2(y - \cos(\pi x))^2 + x + 4$$

is shown in the figure below by its contour plot (the 7th minimum is the shallow minimum located near the origin).



Denote the outer contour corresponding to $V = 10$ by Γ_N . The contours ∂A and ∂B are two contours (out of 6) where $V = -3$. The region Ω is bounded by contours Γ_N , ∂A and ∂B . Consider the following BVP:

$$\nabla^2 u - \nabla V \cdot \nabla u = 0, \quad (x, y) \in \Omega \quad (4)$$

$$\frac{\partial u}{\partial n} = 0, \quad u \in \Gamma_N \quad (5)$$

$$u(\partial A) = 0, \quad u(\partial B) = 1. \quad (6)$$

Remark. The solution of this problem has the following probabilistic interpretation. Suppose a particle is moving inside the region Ω with reflecting boundary Γ_N according to the SDE $dz = -\nabla V(z)dt + \sqrt{2}dw$ (the overdamped Langevin dynamics). The potential force $-\nabla V(z)$ pushes it downhill while the random force dw pushes it randomly to different directions. Suppose at time $t = 0$ the particle is at the point (x, y) . Then $u(x, y)$ is the probability that the particle will reach the region B prior to region A .

(a) Show that Eq. (4) is equivalent to

$$\nabla \cdot \left(e^{-V(x,y)} \nabla u \right) = 0. \quad (7)$$

(b) Let u be the solution of Eq. (7) with BCs (5)-(6) and u_D be a smooth function such that $u_D = 1$ at ∂B and $u_D = 0$ outside a small neighborhood of ∂B . Define $v := u - u_D$. Note that v satisfies $v(\partial B) = v(\partial A) = 0$ and $\frac{\partial v}{\partial n} = 0$ on Γ_N . Show that then for any continuous and piecewise continuously differentiable function

w defined on Ω such that $w(\partial A) = w(\partial B) = 0$

$$\int_{\Omega} e^{-V(x,y)} \nabla v \cdot \nabla w dx dy = - \int_{\Omega} e^{-V(x,y)} \nabla u_D \cdot \nabla w dx dy. \quad (8)$$

- (c) Modify your FEM code developed for the previous problem and solve Eq. (7) with BCs (5)-(6) using it. Submit a printout of your code and the plot of $u(x, y)$ that it makes.

Hint: The code `contours7well.m` extracts the contours Γ_N and ∂A and ∂B and calls `mesh2d` so that the mesh has maximal diameter of triangles not exceeding 0.1.

References

- [1] Jochen Albrety, Carsten Carstensen and Stefan A. Funken, Remarks around 50 lines of Matlab: short finite element implementation

Homework 4. Due March 2 (In class)

1. **(10 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 [1].

Consider the BVP

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

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

- (a) Verify that the solution of Eq. (1) is given by

$$u(x) = \int_0^1 G(x, y) f(y) dy, \quad (2)$$

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

$$G(x, y) = \begin{cases} (1-x)y, & 0 \leq y \leq x \leq 1, \\ x(1-y), & 0 \leq x \leq y \leq 1. \end{cases} \quad (3)$$

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

$$\int_0^1 v'(y) \frac{d}{dy} G(x, y) dy = v(x). \quad (4)$$

Conclude that since the exact solution u of Eq. (1) belongs to $H_0^1([0, 1])$, Eq. (4) 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

$$\phi_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}, & x \in [x_{j-1}, x_j], \\ \frac{x-x_{j+1}}{x_j-x_{j+1}}, & x \in [x_j, x_{j+1}], \\ 0, & \text{otherwise,} \end{cases} \quad j = 1, \dots, n.$$

Show that for the linear interpolant $I_h u$,

$$\int_0^1 [I_h u]'(y) \frac{d}{dy} G(x_j, y) dy = u(x_j). \quad (5)$$

- (d) (There is nothing to write for this item, just think about it and make sure that all statements are clear.) Observe that for every fixed node x_j , the function $\psi_j(y) := G(x_j, y)$ is a linear combination of the basis functions $\phi_i(y)$. Observe that $\psi_j(y)$, $j = 1, \dots, n$ are linearly independent. Hence $\psi_j(y)$ constitute another basis in the space $S_h = \text{span}\{\phi_1, \dots, \phi_n\}$. Recall that the FEM solution U^{FEM} satisfies the identity

$$\int_0^1 [U^{FEM}(y)]' \chi'(u) dy = \int_0^1 f(y) \chi(y) dy \quad \text{for all } \chi \in S_h. \quad (6)$$

- (e) Now 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 $I_h u$.

2. (5 pts) The weighted Jacobi iteration for the equation $(D - U - L)x = b$ is defined by

$$x^{(n+1)} = (1 - \omega)x^{(n)} + \omega D^{-1}[(L + U)x^{(n)} + b]. \quad (7)$$

Consider the application of the weighted Jacobi iteration to solving the linear system arising from the 5-point stencil finite difference scheme for the Poisson equation $-\nabla^2 u = f(x, y)$ in a unit square $[0, 1]^2$ with homogeneous Dirichlet boundary conditions (i.e., $u = 0$ on the boundary of the unit square). Assume that the mesh has step $h = 1/N$ both in x and y directions. Hence, the total number of mesh points where u is to be determined is $(N - 1)^2$.

- (a) Verify that the mesh functions

$$v_{k_x, k_y}(x_i, y_j) = \sin(k_x x_i) \sin(k_y y_j), \quad x_i = hi, \quad y_j = hj,$$

are eigenvectors of the corresponding iteration matrix G for all

$$k_x, k_y = \pi, 2\pi, \dots, (N - 1)\pi.$$

Determine the corresponding eigenvalues.

- (b) Show that if $\omega \leq 0$, then the iteration diverges (use the theorem relating the eigenvalues of the iteration matrix G and the convergence.) Show that if $\omega > 1$, then for sufficiently small h the iteration diverges. Show that for all h the iteration converges for all $0 < \omega \leq 1$.
- (c) For $0 < \omega \leq 1$, determine the value of ω for which the asymptotic rate of convergence $r = -\log \rho(G)$ is maximal possible. Here $\rho(G) = \max |\lambda(G)|$ is the spectral radius.

- (d) The weighted Jacobi iteration is a reasonable method to use as a smoother for high harmonics in the multigrid method. In this case, ω is typically set to be $4/5$ in 2D. Explain this choice of ω . Reason as follows. Assume that the task is to make the components proportional to the eigenvectors with $\max\{k_x, k_y\} \geq \pi N/2$ decay as fast as possible. The other components can be ignored. Furthermore, the choice of ω should be good for all $h \rightarrow 0$.

Hint: You might find Section 13.2.2 in [2] helpful.

3. **(5 pt)** The goal of this problem is to find the optimal choice of ω for the SOR method. It is to be done in MATLAB.
- (a) Consider the linear system from the previous problem with $f(x, y) = 2(x^2 - x + y^2 - y)$. In this case the exact solution of both the BVP $-\nabla^2 u = f$ and its finite difference solution are $u(x, y) = xy(1 - x)(1 - y)$. Set $N = 2^4 + 1$. For each $\omega = 1 : 0.01 : 1.99$ solve the system using SOR and determine the convergence rate (if you do not know how to do it, you can mimic the procedure used in `basic_iterative_methods.m`). Find the value of ω for which the convergence rate is maximal.
- (b) Repeat the previous item for $N = 2^k + 1$ where $k = 5, 6, 7, 8$. To accelerate the process, use smaller ranges of values of ω since you can bracket the optimal value of ω . You might also want to use smaller step in ω .
- (c) Plot the found optimal values of ω versus h and superimpose your findings with the theoretical prediction

$$\omega_{opt} = \frac{2}{1 + \sin(\pi h)}.$$

References

- [1] S. Larsson and V. Thomee, Partial Differential Equations with Numerical Methods, Springer-Verlag Berlin Heidelberg, 2003, 2009 (soft cover)
- [2] Yousef Saad, Iterative Methods for Sparse Linear System, SIAM 2003 (Chapter 13)

Homework 5. Due March 9 (In class)

1. **(10 pts)** The goal of this project is to explore the multigrid method. It is to be done in MATLAB. Consider the BVP

$$\nabla \cdot (a(x, y)\nabla u) = f(x, y) \quad (x, y) \in \Omega := [0, 1]^2, \quad u = 0 \quad (x, y) \in \partial\Omega, \quad (1)$$

where $a(x, y) = 1 + x + 2y^2$ and $f(x, y)$ is set up so that $u(x, y) = xy(1 - x)(1 - y)$ is the exact solution. Set up the 1025×1025 mesh. $u(x, y)$ is to be found at 1023×1023 interior mesh points.

- (a) Compare the performance of multigrid with three different smoothers: weighted Jacobi with ω , Gauss-Seidel, and Red-and-Black Gauss-Seidel. For weighted Jacobi, experiment with different values of $\omega \in (0.8, 1)$ and pick ω for which you observe the fastest convergence. Set up the while-cycle that breaks either if the residual is less than 10^{-9} or if the number of V-cycles exceeds 40. For each method (if it converges), (a) find the number of V-cycles and (b) the CPU time necessary to make the residual to become less than 10^{-9} . For each method (whether it converges or not), plot the norm of the residual in the log scale versus the number of V-cycles. Report your findings and write a conclusion, which method and in which sense is the best.

Hint: You can use `multigrid.m` as a template. Red-and-Black Gauss-Seidel is vectorized for the case of the Poisson equation in `Smoothers4multigrid.m`. If you find a better approach to vectorize it, please show it to me (e.g. send me an email with your code:). You can look up a non-vectorized version of Red-and-Black Gauss-Seidel, e.g., in D. Bindel's lecture, in C. Rycroft's notes, or in Y. Saad's book [1].

- (b) Pick the best smoother for multigrid and experiment with W-cycles instead of V-cycles (see [1]). The code `multigrid.m` implements the V-cycle of depth 10 (the meshes used are $N \times N$ for $N = 2^p + 1$ where p first decreases from 10 to 1 and then increases back to 10. If necessary, replace the regular Gauss-Seidel with the smoother which works the best according to your results in the previous task. Measure the cpu time for this V-cycle. Then, replace this V-cycle, implement the $W(q)$ -cycle with meshes for $N = 2^p + 1$ where p first decreases from 10 to 1, then increases to q then decreases to 1, and then increases back to 10. Do so for $q = 2, 3, \dots, 9$ and for each q measure the cpu time. For which cycle, V or W, and if W, for which q the cpu time is the smallest one? Submit a report of your observations and a print-out of your code with the W-cycle.
2. **(5 pts)** The Jordan form of an $n \times n$ matrix A is given by $J = S^{-1}AS$. Suppose the two first Jordan blocks of J are 1×1 and 2×2 and given by $J_1 = 3$ and $J_2 = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$.

Consider the Krylov sequence for the matrix A and the starting vector $u = s_1 + s_2 + s_3$ where s_1 , s_2 , and s_3 are the first three columns of the matrix S .

- (a) The subspace spanned by columns of an $n \times k$ matrix B ($1 \leq k < n$) (denoted by $\text{span}\{B\}$) is an invariant subspace of A if for any vector $y \in \text{span}\{B\}$, $Ay \in \text{span}\{B\}$, i.e., $AB = BC$ for some $k \times k$ matrix C .

Show that $\text{span}\{s_1, s_2, s_3\}$ is an invariant subspace of A .

- (b) Show that the Krylov sequence terminates at 3, i.e.,

$$\text{span}\{u, Au, A^2u\} = \text{span}\{u, Au, A^2u, \dots, A^k u\} \quad \text{for all } k > 2.$$

3. **(5 pts)** Prove that the two versions of the conjugate gradient method (see Algorithms 5.1, p. 108 and Algorithm 5.2, p. 112 in [2]) are equivalent, i.e., prove that

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} = \frac{r_k^T r_k}{p_k^T A p_k} \quad \text{and} \quad \beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

References

- [1] Yousef Saad, Iterative Methods for Sparse Linear System, SIAM 2003 (Chapter 13)
 [2] J. Nocedal and S. Wright, Numerical Optimization, 2nd edition, Springer

Homework 6. Due March 23 (In class)

1. **(5 pts)** Consider a large linear system $Ax = b$ where A is a symmetric positive definite matrix. In many cases such a system can be efficiently solved using the preconditioned conjugate gradient (CG) method. In essence, preconditioning is the introduction of a new variable $y = Cx$ and the replacement of the original system with $C^{-T}AC^{-1}y = C^{-T}b$, where the condition number of the matrix $C^{-T}AC^{-1}$ is smaller than the one of A .

- (a) Derive the preconditioned conjugate gradient algorithm.

Input: $x_0, tol, M = C^T C$.

Initialization: $r_0 = Ax_0 - b$; solve $Mq_0 = r_0$ for q_0 ; $p_0 = -q_0$; $k = 0$;

while $\|r_k\| > tol$

$$\alpha_k = \frac{r_k^T q_k}{p_k^T A p_k};$$

$$x_{k+1} = x_k + \alpha_k p_k;$$

$$r_{k+1} = r_k + \alpha_k A p_k;$$

solve $Mq_{k+1} = r_{k+1}$ for q_{k+1} ;

$$\beta_{k+1} = \frac{r_{k+1}^T q_{k+1}}{r_k^T q_k};$$

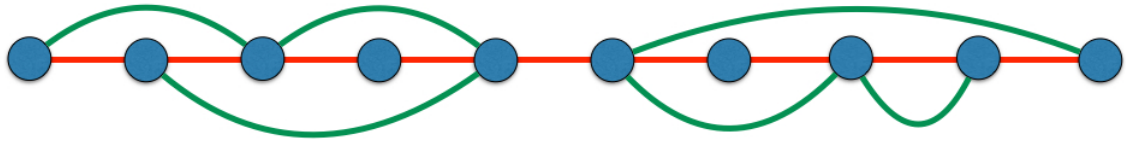
$$p_{k+1} = -q_{k+1} + \beta_{k+1} p_k;$$

$$k = k + 1;$$

end (while)

Hint: The CG algorithm with preconditioning is written above. Apply it to the equation $Qy = f$ where $Q \equiv C^{-T}AC^{-1}$ and $f \equiv C^{-T}b$. Denote residuals by ρ_k and directions by d_k . Show that $C^T \rho_k = r_k$ and $d_k = Cp_k$. Then return to the original variable x using these relationships.

- (b) For the CG algorithm above the orthogonality relationships for the directions and the residuals are $p_i^T A p_j = 0$, $r_i^T r_j = 0$, $i \neq j$. What are the orthogonality relationships for the directions and the residuals for the preconditioned CG algorithm?
2. **(15 pts)** Generate an electric circuit with n nodes connected into a chain and, in addition randomly selected $\leq k$ pairs of nodes are also connected. An example of such a circuit is shown in the figure below.



Conductances c_{ij} of the red conductors are random numbers between 0 and 1. The conductances c_{ij} of the green conductors are random numbers between 0 and k .

Set the voltage v at the first node to 0 and at the last one to 1. Then using the Kirchhoff circuit law $\sum_j I_{ij} = 0$ and Ohm's law $I_{ij} = c_{ij}(v_j - v_i)$ set up a linear system to find the voltage v_i at each node $i = 2, 3, \dots, n - 1$. All these is done in the script below.

```
%% generate the conductances

n = 1000;
k = 1000;
ad = 1 + round(rand(k,2)*(n-1));

ud = rand(n,1);
L = spdiags(ud,1,n,n);
for j = 1:k
    if( ad(j,1) ~= ad(j,2) )
        L(ad(j,1),ad(j,2)) = rand(1)*k;
    end
end

L = L + L';
p = sum(L,2);
L = L - spdiags(p,0,n,n);

%% form the linear system: the matrix A and the right-hand side b
A = -L(2:n-1,2:n-1);
b = L(2:n-1,:)*[zeros(n-1,1);1];

c = condest(A);
fprintf('cond(A) = %d\n',c);
```

- (a) Set $n = k = 1000$. Solve the resulting system $Ax = b$ using the CG method with and without the incomplete Cholesky preconditioning. Set the tolerance to $1e-12$ and stop when the norm of the residual is less than the tolerance. Compute the eigenvalues of A using `eig(full(A))`. Plot the residuals versus

step obtained by CG with Incomplete Choleski preconditioning and without it as well as the predicted residuals by Eqs. (5.34) and (5.35) in Nocedal and Wright [1]. Comment on your graphs. (You should converge in about 120 iterations with the preconditioning and in about 2500 without it (not 1000, because the arithmetics is not exact – note the difference!).

- (b) Experiment with different values of n and k as well as different rules for assigning conductances (for example, you can make them all the same rather than random). For each experiment, compare the performance of the CG with and without preconditioning. Comment on your observations.

3. (5 pts) Calculate the local truncation error of the Crank-Nicholson method for solving $u_t = u_{xx}$ (see Eq. 9.6 in [2]).

Hint: use Taylor expansion around $(n+1/2, j)$. Look up in the internet Taylor series in 2 variables.

4. (15 pts) Consider the following IBVP for the heat equation

$$u_t = u_{xx}, \quad 0 \leq x \leq \pi, \quad t \geq 0, \quad (1)$$

$$u(x, 0) = x, \quad (2)$$

$$u(0, t) = 0, \quad \text{the left end is kept at temperature 0} \quad (3)$$

$$u_x(\pi, t) = 0, \quad \text{the right end is insulated.} \quad (4)$$

- (a) Solve Eq. (1)-(4) 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 $x = u(x, 0)$.
- (b) Compute the solution using the Crank-Nicholson scheme (Eq. 9.6 in [2]) with $h = k = 2^{-12}\pi$ at time $t = 2$ and treat it as the exact solution. Experiment with solving Eq. (1)-(4) numerically on the interval $0 \leq t \leq 2$ using three different schemes (a) the explicit (forward Euler) scheme (Eq. 9.4 in [2]), (b) the implicit (backward Euler) scheme (change the superscripts to $n+1$ in the right-hand side of Eq. 9.4 in [2]), and (c) the Crank-Nicholson scheme with space discretization into $N = 2^p + 1$ points for $p = 4, 5, \dots, 10$. Find the CPU time for each calculation. Compare your numerical solution at time $t = 2$ for each h and each scheme with the “exact” solution. Plot the error versus h for each scheme in the log-log scale. Write a summary of your observations.

Hint: In the explicit scheme, observe that if you set the time step $k > h^2/2$ where h is the step in space, the solution blows up. Use $\lambda = k/h^2$ equal to 0.4

and 1/6. Check whether $\lambda = 1/6$ indeed leads to a higher order of convergence. In the implicit (backward Euler) and the Crank-Nicholson schemes, use $k = h$.

References

- [1] J. Nocedal and S. Wright, Numerical Optimization, 2nd edition, Springer
- [2] Randall J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM 2007 (Chapter 9)

Homework 7. Due April 1 (Friday)

(20 pts) Consider the Boussinesq equation describing ground water flow in a porous rock lying on an impermeable bed

$$\partial_t u = \frac{1}{2} \partial_{xx}^2 u^2 \quad \text{or, equivalently,} \quad \partial_t u = u \partial_{xx} u + (\partial_x u)^2, \quad (1)$$

where $u(x, t)$ is the height of the water dome. It is well-known that if the initial condition $u(x, 0) = u_0(x)$ has a compact support then the solution $u(x, t)$ also has a compact support. We will denote the left and right ends for the support of $u(x, t)$ by $x_L(t)$ and $x_R(t)$ respectively. We also define the midpoint of the support by $x_0(t)$ and the half-length of the support by $x_f(t)$, i.e.,

$$x_0(t) = \frac{1}{2}(x_L(t) + x_R(t)), \quad x_f(t) = \frac{1}{2}(x_R(t) - x_L(t)).$$

1. Check that Eq. (1) has a self-similar solution

$$u(x, t) = \begin{cases} \frac{1}{6} B^2 (t - t_0)^{-1/3} \left[1 - \left(\frac{x - x_0}{B(t - t_0)^{1/3}} \right)^2 \right], & |x - x_0| \leq B t^{1/3}, \\ 0, & |x - x_0| > B t^{1/3}, \end{cases} \quad (2)$$

where B , t_0 and x_0 are constants depending on the initial condition. Note that the quantity $u(x, t)/u_{\max}(t) = 6uB^{-2}(t - t_0)^{1/3}$ is a function of a single variable $\xi := (x - x_0)/x_f(t)$ where $x_f(t) = B(t - t_0)^{1/3}$. Therefore, at all times,

$$\frac{u}{u_{\max}} = 1 - \xi^2.$$

2. The self-similar solution given by Eq. (2) is special in the sense that it is an intermediate asymptotics for any solution with two free boundaries and compact support, i.e., any solution with two free boundaries and compact support approaches a function given by Eq. (2) for some B , t_0 and x_0 . Demonstrate this by solving Eq. (1) numerically starting with an arbitrary compactly supported initial condition. Do it as follows. Introduce a new variable

$$\xi := (x - x_0(t))/x_f(t) \in [-1, 1].$$

The flux continuity condition at the free boundaries leads to the following evolution of $x_L(t)$ and $x_R(t)$:

$$\frac{d}{dt} x_R(t) = -\frac{\partial_\xi u(1, t)}{x_f(t)}, \quad \frac{d}{dt} x_L(t) = -\frac{\partial_\xi u(-1, t)}{x_f(t)}. \quad (3)$$

Using Eq. (3) show that in the coordinates (ξ, t) Eq. (1) becomes

$$\partial_t u = \frac{1}{x_f(t)^2} \left[-\frac{1}{2} \partial_\xi u [(1 + \xi) \partial_\xi u(1, t) + (1 - \xi) \partial_\xi u(-1, t)] + u \partial_\xi^2 u + (\partial_\xi u)^2 \right]. \quad (4)$$

3. Code `alina.m` solves the nonlinear system (3)-(4) using the method of lines (MOL, see [2]) and the built-in MATLAB solver `ode15s` on the time interval such that $\{t : x_f(t) \leq 2\}$ and plots the graph of $u(\xi, t)/u_{\max}(t)$ (**figure(1)**) versus ξ and $u(x, t)$ versus x (**figure(2)**). Central differences are used in space at all inner mesh points, and one-sided second-order differences are used to approximate $\partial_\xi u$ at the endpoints $\xi = \pm 1$.

Experiment with the initial data `y0(1:n)`. Check that no matter what is the initial profile, the solution approaches the parabola at $t \rightarrow \infty$. If necessary, comment out the stopping criterion due to `events`.

4. In this task the goal is to check whether the technique for moving the support of the solution is worth the trouble. Solve the Boussinesq equation (1) on the interval $[-2, 2]$ without moving the support of the solution explicitly. Do not introduce variables x_L and x_R . Discretize Eq. (1) using central differences. Start from the initial condition giving rise to the self-similar solution:

$$u(x, 0) = \begin{cases} 1 - x^2, & |x| \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

- (a) First use the code `alina.m`. Make the initial condition in `alina.m` $1 - \xi^2$ as it was originally. Compute the solution at time $t_f = 1.166666666666345$ (this is the time at which the support becomes $(-2, 2)$). Find the maximal deviation of $u(\xi, t_f)/u_{\max}$ from the parabola $1 - \xi^2$.
- (b) Solve Eq. (1) using MOL with $h = 0.01$ and `ode15s` without moving the support explicitly on the time interval $[0, t_f]$. In other words, solve the system of ODEs

$$\frac{dU_j}{dt} = U_j \frac{U_{j+1} - 2U_j + U_{j-1}}{h^2} + \left(\frac{U_{j+1} - U_{j-1}}{2h} \right)^2, \quad j = 1, \dots, n-1,$$

where the functions $U_j(t)$ approximate $u(x_j, t)$, using `ode15s`. Measure the CPU time. Find the maximal deviation of $u(x, t_f)/u_{\max}$ from the parabola $1 - (x/2)^2$.

- (c) Program the forward Euler central difference explicit scheme, i.e.,

$$\frac{v_j^{n+1} - v_j^n}{k} = v_j^n \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2} + \left(\frac{v_{j+1}^n - v_{j-1}^n}{2h} \right)^2.$$

First find out for which ratios k/h^2 this method is stable. Set $k = 0.5h^2$ and find the maximal deviation of $u(x, t_f)/u_{\max}$ from the parabola $1 - (x/2)^2$. Measure the CPU time for $k = 0.5h^2$.

- (d) Program the semi-implicit scheme

$$\frac{v_j^{n+1} - v_j^n}{k} = v_j^n \frac{v_{j+1}^{n+1} - 2v_j^{n+1} + v_{j-1}^{n+1}}{h^2} + \frac{v_{j+1}^n - v_{j-1}^n}{2h} \frac{v_{j+1}^{n+1} - v_{j-1}^{n+1}}{2h}.$$

Note that you need to solve a linear system at each time step. Write out the linear system to be solved at each time step. Check that the method is stable even if you use $h = k$. For each $k = h$, $k = 0.1h$ and $k = h^2$ find the maximal deviation of $u(x, t_f)/u_{\max}$ from the parabola $1 - (x/2)^2$. Measure the CPU time.

- (e) Write the summary of your observations. Comment on advantages and disadvantages of each of these four methods.

If you would like to learn more details about the derivation of the self-similar solution for Eq. (1) and the numerical experiment, please see [1]. The code `alina.m` is named due to Prof. Alina Chertock (NCSU), one of the authors of [1], who taught me the technique for moving the support of the solution implemented in the code.

References

- [1] G. I. Barenblatt, M. Bertsch, A. E. Chertock, V. M. Prostokishin, *Self-similar intermediate asymptotics for a degenerate parabolic filtration-absorption equation*, PNAS, 97 (2000), pp. 9844-9848.
- [2] Randall J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM 2007 (Chapter 9)

Homework 8. Due April 13 (Wednesday)

1. (5 pt) Consider the advection equation

$$u_t + au_x = 0, \quad u(x, 0) = \phi(x). \quad (1)$$

- (a) The modified equation for the Left Upwind method is derived in [1] (see Eq. (10.44) on page 220). Derive the modified equation for the Lax-Friedrichs method. Compare it with the one for the Left Upwind. Comment on which method adds more smearing.
- (b) Assume that Eq. (1) is solved on the infinite line $-\infty < x < \infty$. Apply the Fourier transform (Eq. (E.18) on page 318 in [1]) to both modified equations and solve them in the Fourier space. Comment on the behavior of the Fourier modes $\hat{v}(\xi, t)$ as $t \rightarrow \infty$. Using the solutions in the Fourier domain, determine necessary stability conditions on h , k , and a for the Left Upwind and the Lax-Friedrichs methods. (We know that these conditions are also sufficient from the spectral analysis conducted in class.)
2. (5 pt) Consider the advection equation Eq. (1) on the infinite line $-\infty < x < \infty$. Assume that $a > 0$. The modified equation for the Lax-Wendroff method is derived in [1] (see Eq. (10.45) on page 221). Derive the modified equation for the Left Beam-Warming method (Eq. (10.26) on p. 212 in [1]). Apply the Fourier transform (Eq. (E.18) on page 318 in [1]) to both modified equations and solve them in the Fourier space. Comment on the behavior of the Fourier modes $\hat{v}(\xi, t)$ as $t \rightarrow \infty$. Obtain the dispersion relations and the phase and group velocities. Compare the phase and group velocities for the Lax-Friedrichs and the Left Beam-Warming methods. Explain the appearance and the location of oscillations produced by these methods. (You may want to refresh your memory by running the code `advection.m` available on ELMS. Make sure appropriate values to the variable `method` in the code.)
3. (5 pt) Consider the wave equation on the infinite line $-\infty < x < \infty$

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

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

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

$$u(x, t) = \frac{1}{2}(\phi(x + at) + \phi(x - at)) + \frac{1}{2a} \int_{x-at}^{x+at} \psi(s) ds. \quad (3)$$

- (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) Let $\phi(x) = \max\{1 - |x|, 0\}$ and $\psi(x) = 0$. These initial conditions correspond to an infinite string released from being plucked at time 0. Plot the solution (without using the computer) at times $t = 1/(2a)$, $t = 1/a$, $t = 2/a$, and $t = 4/a$.
- (d) Let $\phi(x) = 0$ and

$$\psi(x) = \begin{cases} 1, & -1 \leq x \leq 1, \\ 0, & |x| > 1. \end{cases}$$

These initial conditions correspond to an infinite string hit by a hammer at time 0. Plot the solution (without using the computer) at times $t = 1/(2a)$, $t = 1/a$, $t = 2/a$, and $t = 4/a$.

4. **(10 pt)** The goal of this problem is to adjust for the full wave equation the numerical methods developed for the advection equation.

- (a) Rewrite Eq. (2) 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 .

- (b) 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:

$$\xi_t = \lambda_1 \xi_x, \quad \eta_t = \lambda_2 \eta_x. \quad (4)$$

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

- (c) Set $\phi = \max\{1 - |x|, 0\}$, $\psi = 0$, $a = \sqrt{2}$, $h = 0.05$, and k satisfying $|ka/h| = 0.8$. 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. (3). Write a summary of your observations.

References

- [1] Randall J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM 2007 (Chapter 10 and Appendix E.3)

Homework 9. Due April 20 (Wednesday)

1. (5 pt) Consider the Burgers equation $u_t + [u^2/2]_x = 0$ on $-\infty < x < \infty$, $t \geq 0$, with the initial data

$$u_0(x) = \begin{cases} 0, & x < -1 \text{ and } x > 5, \\ 2, & -1 < x < 0, \\ 4, & 0 < x < 1, \\ 3, & 1 < x < 3, \\ 1, & 3 < x < 5. \end{cases} \quad (1)$$

- (a) Draw characteristics on the (x, t) plane. Calculate all necessary shock speeds and highlight the shock lines along which discontinuities propagate. Find the values (x_i^*, t_i^*) , $i = 1, 2$, at which the shock lines collide and accurately mark these points on the diagram. Indicate regions on the (x, t) -plane where the solution u is constant. Mark those constant values of u .
- (b) Plot the physically relevant weak solution $u(x, t)$ at times $t = 1/2$, $t = 1$ and $t = 3/2$.

2. (5 pt)

Show that MacCormack's method

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

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

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

3. (10 pt)

- (a) Solve numerically the Burgers equation $u_t + [u^2/2]_x = 0$ with the initial data (1) using the Lax-Friedrichs method in the conservative form, Richtmyer's method (Eq. (23) in `burgers.pdf`), MacCormack's method, Godunov's method, and Glimm's method. Do this using $h = 0.1$ and 0.01 . Plot the numerical solutions at times $t = 1/2$, $t = 1$, $t = 2$, and $t = 3$ and compare them with the exact solutions found in Problem 1(b). Write a summary of your observations.
- (b) Solve numerically the Burgers equation $u_t + [u^2/2]_x = 0$ with the initial data $u_0(x) = \exp(-x^2)$ using the Lax-Friedrichs method in the conservative form, Richtmyer's method, MacCormack's method, Godunov's method, and Glimm's method. Do this using $h = 0.1$ and 0.01 . Find analytically the time at which the shock arises. Is it consistent with your numerical results? Write a summary of your observations.

Homework 10. Due April 29 (Friday)

1. **(5 pt)** Let $\phi_n(x)$, $n = 1, 2, \dots$ be a family of functions with respect to the inner product (\cdot, \cdot) , i.e., $(\phi_m, \phi_n) = 0$ for all m and n such that $m \neq n$. Let $f(x)$ be a given function. Let the norm $\|\cdot\|$ be induced by the inner product, i.e., $\|g\| \equiv \sqrt{(g, g)}$ for any function $g(x)$. Suppose we want to approximate $f(x)$ by a finite linear combination of N functions $\phi_n(x)$ as accurately as possible with respect to the norm $\|\cdot\|$, i.e., we want to find coefficients c_1, \dots, c_N such that

$$\left\| f - \sum_{n=1}^N c_n \phi_n \right\|^2 = \min_{a_1, \dots, a_N} \left\| f - \sum_{n=1}^N a_n \phi_n \right\|^2.$$

Show that such coefficients are given by

$$c_n = \frac{(f, \phi_n)}{(\phi_n, \phi_n)}. \quad (1)$$

Remark The coefficients c_n given by Eq. (1) are called the Fourier coefficients for the function f and the family of n orthogonal functions $\{\phi_n\}_{n=1}^{\infty}$. Note that the formula for the Fourier coefficients is the same for any inner product.

2. **(5 pt)**

- (a) Find the family of eigenvalues and the corresponding eigenfunctions of the differential operator $L = \frac{d}{dx}$ acting on the space of 2π -periodic functions, i.e, find all pairs (λ_k, ϕ_k) such that

$$\frac{d\phi_k}{dx} = \lambda_k \phi_k, \quad \phi(0) = \phi(2\pi).$$

- (b) Find the family of eigenvalues and the corresponding eigenvectors of the N -point central difference operator with 2π -periodic boundary conditions

$$L_N = \frac{N}{4\pi} \begin{bmatrix} 0 & 1 & 0 & \dots & -1 \\ -1 & 0 & 1 & \dots & 0 \\ 0 & -1 & 0 & \dots & \dots \\ & \dots & \dots & \dots & 1 \\ 1 & 0 & \dots & -1 & 0 \end{bmatrix}.$$

Comment on how the eigenvalues and eigenvectors/eigenfunctions of L and L_N relate when N is finite and fixed and when $N \rightarrow \infty$.

3. **(5 pt)** The discrete Fourier transform **fft** and the inverse discrete Fourier transform **ifft** are defined, respectively, by

$$y(k) = \sum_{j=0}^{N-1} x(j) \omega_N^{jk}, \quad x(j) = \frac{1}{N} \sum_{k=0}^{N-1} y(k) \omega_N^{-jk}, \quad \text{where } \omega_N = e^{-\frac{2\pi i}{N}}. \quad (2)$$

Verify that the functions **fft** and **ifft** are mutually inverse.

4. **5 pts** Read an article on the Kuramoto-Sivashinsky equation available at <http://people.maths.ox.ac.uk/trefethen/pdectb/kuramoto2.pdf>

Solve the equation

$$u_t + u_{xxxx} + u_{xx} + \frac{1}{2}(u^2)_x = 0, \quad u(x, 0) = \cos(x/16)(1 + \sin(x/16)) \quad (3)$$

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

$$u_t = -u_{xxxx} - u_{xx} := Lu. \quad (4)$$

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. (4). Define the solution operator e^{tL} so that $u(x, t) = e^{tL}u(x, 0)$. Now return to Eq. (3). 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 $v(x, t)$:

$$v_t = e^{-tL}N(e^{tL}v). \quad (5)$$

Solve Eq. (5) 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 KdVrk1.m that solves the Korteweg-de Vries equation

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

using the proposed approach.

5. **5 pts**

- (a) Prove that the N -point Discrete Fourier Transform (2) aliases a high frequency m not in the range $\{0, 1, \dots, N-1\}$ to a uniquely defined frequency $m_A \in \{0, 1, \dots, N-1\}$. *Hint: write $f(x) = \sum_{m=-\infty}^{\infty} \alpha_m e^{imx}$. Its m -th Fourier component is $f_m := \alpha_m e^{imx}$. Apply the discrete Fourier transform to the function $f_m(x_j) = \alpha_m e^{imx_j}$, $x_j = \frac{2\pi j}{N}$. Then apply the inverse discrete Fourier transform and see what happens.*

- (b) Let $f(x) = \sin(50x)$ and $N = 32$. What is $\text{ifft}(\text{fft}(f))$?

6. **5 pts** Devise the Fast Fourier Transform for $N = 3^m$. Work out the formulas for $N = 3^2 = 9$ as it is done in the lecture notes. Calculate the number of flops and compare it with the number of flops for $N = 2^m$.

Homework 11. Due May 9 (Monday)

1. **(5pt)** Write a book report explaining the construction of Daubechies db2 (or D4) wavelet and the corresponding scaling function shown in the top left figure in Wiki.
2. **(5 pt)** Write a program (from scratch) generating Daubechies wavelet and the corresponding scaling function using the iterative technique. At iteration j , ϕ and ψ are evaluated at the points $(2k+1)2^{-j}$, $k = 0, 1, \dots, 3 * 2^{j-1} - 1$. Do 10 iterations. Submit a printout of your code and the graphs of ϕ and ψ .
3. **(5 pt)** Read matlab help on `wavelet families`, `wavefun`, and `Two-Dimensional Discrete Wavelet Analysis`. Take any photo of your choice containing different features: texture, flat areas, face. Do not use "Lena". Compress it using wavelet decomposition and zeroing out coefficients below some threshold. You might find useful the code `WLena.m`. Experiment with Haar and Daubechies `dbn` wavelets for different n and different thresholds for zeroing out coefficients. Write a summary of your observations. Then add some noise. Try to corrupt 10%, 30%, 50% and 70% of pixels with noise. Then try to denoise the image using wavelets.