Math 586 Homework 4
Due soon
By Marvyn Bailly (GitHub: MarvynB)

Problem 1 *Consider the implicit upwind method for the advection equation* $u_t + au_x = 0$

$$U_j^{n+1} = U_j^n - \frac{ak}{h}\left(U_j^{n+1} - U_{j-1}^{n+1}\right).$$

Derive the modified equation for this method to order $O(k+h)$*. Compare the effective diffusion coefficient with that of the modified equation for*

$$U_j^{n+1} = U_j^n - \frac{ak}{h}\left(U_j^n - U_{j-1}^n\right),$$

which was derived in lecture.

Solution.
Consider the implicit upwind method for the advection equation $u_t + au_x = 0$

$$U_j^{n+1} = U_j^n - \frac{ak}{h}\left(U_j^{n+1} - U_{j-1}^{n+1}\right).$$

Let's assume that $U_n^j = v(x_j, t_n)$ for a smooth function $v(x,t)$ and plugging this into the method yields

$$\frac{v(x,t+k) - v(x,t)}{k} = -a\frac{v(x,t+k) - v(x-h,t+k)}{h}.$$

Expanding the RHS yields

$$\frac{v(x,t+k) - v(x,t)}{k} = v_t + \frac{k}{2}v_{tt} + \mathcal{O}(k^2),$$

and expanding the LHS

$$\begin{aligned}\frac{v(x,t+k) - v(x-h,t+k)}{h} &= v_x(x,t+k) - \frac{h}{2}v_{xx}(x,t+k) + \mathcal{O}(h^2)\\ &= \left(v_x + kv_{xt} + \mathcal{O}(k^2)\right) - \frac{h}{2}(v_{xx} + \mathcal{O}(k)) + \mathcal{O}(h^2)\\ &= v_x + kv_{xt} - \frac{h}{2}v_{xx} + \mathcal{O}(k^2 + kh + h^2).\end{aligned}$$

This gives

$$v_t = -av_x - akx_{xt} + \frac{ah}{2}v_{xx} - \frac{k}{2}v_{tt} + \mathcal{O}(k^2 + kh + h^2) = -av_x + \mathcal{O}(k+h).$$

Noticing that $v_t = -av_x + \mathcal{O}(k+h)$, we substitute $v_{xt} = -av_{xx} + \mathcal{O}(k+h)$ and $v_{tt} = a^2 v_{xx} + \mathcal{O}(k+h)$ yielding

$$\begin{aligned} v_t &= av_x + \left(a^2 k + \frac{ah}{2} - \frac{ka^2}{2}\right) v_{tt} + \mathcal{O}(k^2 + kh + h^2) \\ &= av_x + \left(\frac{ah}{2} + \frac{ka^2}{2}\right) v_{tt} + \mathcal{O}(k^2 + kh + h^2). \end{aligned}$$

Thus the diffusion constant is

$$\frac{ah}{2} + \frac{ka^2}{2},$$

and the PDE is well-posed if it is non-negative giving the condition

$$\frac{ah}{2} + \frac{ka^2}{2} > 0 \implies \frac{ak}{h} \geq -1,$$

for $a > 0$.

On the other hand, recall that the modified equation for

$$U_j^{n+1} = U_j^n - \frac{ak}{h}\left(U_j^n - U_{j-1}^n\right),$$

is given by

$$v_t = -av_x + \left(\frac{ah}{2} - \frac{a^2 k}{2}\right) v_{xx} + \mathcal{O}(h^2 + hk + k^2),$$

where the diffusion constant is

$$\frac{ah}{2} - \frac{a^2 k}{2}.$$

This PDE is well-posed if the diffusion constant is non-negative

$$\frac{ah}{2} - \frac{a^2 k}{2} \geq 0 \implies \frac{ah}{k} \geq 1.$$

□

Problem 2 *Consider the method of lines discretization of the advection equation with a Dirichlet boundary conditions:*

$$\begin{cases} u_t + au_x = 0, \quad a > 0, \\ u(x,0) = \eta(x), \\ u(0,t) = g_0(t), \end{cases}$$

given by

$$U'(t) = AU(t) + g(t),$$
$$A = \frac{-a}{2h}\begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & & 1 \\ & & & -1 & 0 \\ & & 1 & -4 & 3 \end{bmatrix}, \quad g(t) = \begin{bmatrix} \frac{ak}{2h} g_0(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

We have seen that if this is discretized using the leapfrog method then an instability is excited. One may want to stabilize this in the way we stabilized forward Euler to obtain the Lax-Friedrichs method. So, consider

$$U_j^{n+1} = \frac{1}{2}(U_{j-1}^{n-1} + U_{j+1}^{n-1}) - \frac{ak}{h}(U_{j+1}^n - U_{j-1}^n).$$

Perform the von Neumann stability analysis to find $s(\xi)$ (LeVeque calls this $g(\xi)$) and give numerical evidence (or a proof, if you can) that $|s(\xi)| > 1$ for some ξ regardless of k, h. From this, one might conclude that $k \approx ch^2$ is required for stability. Now, derive the modified equation and assess whether or not this restriction on k is acceptable.

Solution.

Consider the method of lines discretization of the advection equation with a Dirichlet boundary conditions:

$$\begin{cases} u_t + au_x = 0, \quad a > 0, \\ u(x,0) = \eta(x), \\ u(0,t) = g_0(t), \end{cases}$$

given by

$$U'(t) = AU(t) + g(t),$$
$$A = \frac{-a}{2h}\begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & & 1 \\ & & & -1 & 0 \\ & & 1 & -4 & 3 \end{bmatrix}, \quad g(t) = \begin{bmatrix} \frac{ak}{2h} g_0(t) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

We have seen that if this is discretized using the leapfrog method then an instability is excited. One may want to stabilize this in the way we stabilized forward Euler to obtain the Lax-Friedrichs method. So, consider

$$U_j^{n+1} = \frac{1}{2}(U_{j-1}^{n-1} + U_{j+1}^{n-1}) - \frac{ak}{h}(U_{j+1}^n - U_{j-1}^n).$$

To preform von Neumann stability analysis, let $U_j^n = s(\xi)^h e^{ij\xi h}$ and substituting this into our equation we have

$$s(\xi)^{n+1}e^{ij\xi h} = \frac{1}{2}\bigl(s(\xi)^{n-1}e^{i(j-1)\xi h} + s(\xi)^{n-1}e^{i(j-1)\xi h}\bigr) - \frac{ak}{h}\bigl(s(\xi)^n e^{i(j+1)} + s(\xi)^n e^{i(j-1)}\bigr),$$

and dividing through by $s(\xi)^{n-1}e^{ij\xi h}$ yields

$$s(\xi)^2 = \frac{1}{2}\bigl(e^{-i\xi h} + e^{i\xi h}\bigr) - \frac{ak}{h}s(\xi)\bigl(e^{i\xi h} - e^{-i\xi h}\bigr) = \cos(\xi h) - \frac{ak}{h}s(\xi)2i\sin(\xi h).$$

Now solving for $s(\xi)$ we get

$$s(\xi)^2 + \nu 2i\sin(\xi h)s(\xi) - \cos(\xi h) = 0,$$

where $\nu = \frac{ak}{h}$ and thus we find that

$$s(\xi) = -i\nu\sin(\xi h) \pm \sqrt{\cos(\xi h) - \nu\sin^2(\xi h)}.$$

Now we wish to show that $|s(\xi)| > 1$ for some ξ regardless of k, h. Consider when $\xi_1 = \pi/h$, then

$$s(\xi_1) = 0 \pm \sqrt{\cos(\pi) - 0} = \pm i,$$

and thus

$$|s(\xi_1)| = 1.$$

Now if we compute the derivative we get

$$s'(\xi) = -ih\nu\cos(\xi h) \mp \frac{h\sin(\xi h)(2\nu^2\cos(\xi h) + 1)}{2\sqrt{-\nu^2\sin^2\xi h + \cos(\xi h)}}.$$

Notice that $s'(\xi_1) = ih\nu$ and since $s(\xi)$ is known to be continuous, there must exist some ξ such that the real part is zero and $|s(\xi)| > 1$. Therefore we might conclude that $k \approx ch^2$ may be required for stability. To access if this restriction on k is acceptable, we derive the modified equation. Consider $U_n^j = v(x,t)$ for a sufficiently smooth function $v(x,t)$. Plugging this in yields

$$v(x,t+k) = \frac{1}{2}(v(x-h,t-k) + v(x+h,t-k)) - \frac{ak}{h}(v(x+h,t) - v(x-h,t)).$$

Now Taylor expanding the left-hand side gives

$$v(x,t+k) = v + kv_t + \frac{k^2}{2}v_{tt} + \mathcal{O}(k^3),$$

and Taylor expanding the first term on the right-hand side gives

$$\begin{aligned}\frac{1}{2}(v(x-h,t-k)+v(x+h,t-k)) &= \frac{1}{2}\big(2v(x,t-k) + h^2 v_{xx} + \mathcal{O}(h^4)\big)\\ &= v(x,t-k) + \frac{h^2}{2}v_{xx}(x,t-k) + \mathcal{O}(h^4)\\ &= \left(v - kv_t + \frac{k^2}{2}v_{tt} + \mathcal{O}(k^3)\right) + \frac{h^2}{2}(v_{xx} + \mathcal{O}(k)) + \mathcal{O}(h^4)\\ &= v - kv_t + \frac{k^2}{2}v_{tt} + \frac{h^2}{2}v_{xx} + \mathcal{O}(k^4 + h^2k + k^3).\end{aligned}$$

Taylor expanding the second term on the right-hand side gives

$$\begin{aligned}\frac{ak}{h}(v(x+h,t) - v(x-h,t)) &= \frac{ak}{h}\big(2hv_x + \mathcal{O}(h^3)\big)\\ &= 2akv_x + \mathcal{O}(kh^2).\end{aligned}$$

Thus the equation becomes

$$v + kv_t + \frac{k^2}{2}v_{tt} + \mathcal{O}(k^3) = v - kv_t + \frac{k^2}{2}v_{tt} + \frac{h^2}{2}v_{xx} + \mathcal{O}(k^4 + h^2k + k^3) + 2akv_x + \mathcal{O}(kh^2),$$

which we can solve for v_t to get the modified equation to be

$$v_t = -av_x + \frac{h^2}{4k}v_{xx} + \mathcal{O}(\frac{h^4}{k} + h^2 + k^2).$$

Then if we let $k \approx ch^2$ the equation becomes

$$v_t \approx= -av_x + \frac{1}{4c}v_{xx} + \mathcal{O}(h^2).$$

Notice that while the diffusion coefficient is nonnegative, the method is not consistent as $\frac{1}{4c}v_{xx} \not\to 0$ as $k, h \to \infty$ and therefore will not converge. Thus we can conclude that this restriction on k is not acceptable.

□

Problem 3 *Consider the wave equation with periodic boundary conditions on* $[0,1]$

$$\begin{cases} u_{tt} = c^2 u_{xx}, \quad c > 0, \\ u(x,0) = f(x), \\ u_t(x,0) = g(x), \\ u(0,t) = u(1,t), \\ u_x(0,t) = u_x(1,t). \end{cases}$$

Using $f(x) = \exp(-20(x-1/2)^2)$, $g(x) = \sin(4\pi x)$, $c = 1$, *implement a method to compute and plot the solution at* $t = 0, 1, 2, 3$ *with* $h = 0.01$. *The* k *you need to choose will depend on the method you use. Describe your method in detail.*

Solution.
Consider the wave equation with periodic boundary conditions on $[0,1]$

$$\begin{cases} u_{tt} = c^2 u_{xx}, \quad c > 0, \\ u(x,0) = f(x), \\ u_t(x,0) = g(x), \\ u(0,t) = u(1,t), \\ u_x(0,t) = u_x(1,t), \end{cases}$$

where $f(x) = \exp(-20(x-1/2)^2)$, $g(x) = \sin(4\pi x)$, $c = 1$. Observe that

$$u_{tt} - c^2 u_{xx} = (\partial_t - c\partial_x)(\partial_t + c\partial_x)u = 0,$$

and thus if we let $v = (v_1(x,t), v_2(x,t))$ where $v_1(x,t) = u(x,t)$ and $v_2(x,t) = (\partial_t + c\partial_x)u$ then

$$v_t + c\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} v_x = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} v.$$

Then the boundary conditions become

$$\begin{aligned} v_1(0,t) &= u(0,t) = u(1,t) = v_1(1,t), \\ v_2(0,t) &= u_t(0,t) + cu_x(0,t) = u_t(1,t) + cu_x(1,t) = v_2(1,t), \end{aligned}$$

and the initial conditions are

$$\begin{aligned} v_1(x,0) &= u(x,0) = f(x), \\ v_2(x,0) &= u_t(x,0) + cu_x(x,0) = g(x) + cf'(x). \end{aligned}$$

Now we are trying to solve the two equations

$$\begin{cases} v_{2,t} - cv_{2,x} = 0 \\ v_{1,t} + cv_{1,x} = v_2. \end{cases}$$

Noting that the first equation is completely independent of v_1, we can solve it using the Lax-Wendroff method to get

$$[V_{2,j}^{n+1}] = \left(I + \frac{ck}{2h}S - \frac{c^2k^2}{2h^2}A\right)[V_{2,j}^n],$$

where

$$S = \begin{bmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 1 & & & -1 & 0 \end{bmatrix},$$

and

$$A = \begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix}.$$

Since this method will give us the $V_{2,j}^n$ for any j, we can use Lax Wendroff to get

$$[V_{1,j}^{n+1}] = \left(I - \frac{ck}{2h}S - \frac{c^2k^2}{2h^2}A\right)[V_{1,j}^n] + k[V_{2,j}^n].$$

For stability, we require that

$$\left|\frac{ck}{h}\right| = \frac{ck}{h} \leq 1$$

and thus we let $h = 0.01$ and $k = h$. This method is implemented in Julia using the following code:

```
f = x -> exp(-20*(x - 1/2)^2)
df = x -> -40*(x - 1/2)*exp(-20*(x - 1/2)^2)
g = x -> sin(4 * pi * x)

function solve_wave(m,c,stop)
    h = 1/(1+m)
    k = h
    xs = h:h:1-h

    S = Tridiagonal(fill(-1.0,m-1),fill(0.0,m),fill(1.0,m-1)) |> sparse
    S[1,end] = -1
    S[end,1] = 1

    A = SymTridiagonal(fill(2.0,m),fill(-1.0,m-1)) |> sparse
```

```
        A[1,end] = -1
        A[end,1] = -1

        v1 = map(f, xs)
        v2 = map(g, xs) + c*map(df,xs)

        for t=0:k:stop
            v2 = (I + (c*k)/(2*h) * S - (c^2*k^2)/(2*h^2) * A)*v2;
            v1 = (I - (c*k)/(2*h) * S - (c^2*k^2)/(2*h^2) * A)*v1 + k*v2
        end
        return xs, v1
    end
```

Using the described method, we plot the solution for $t = 0, 1, 2, 3$ which can be seen in Figure 1. While the solution between times looks similar, we see the begins of an unstablity growing with time.

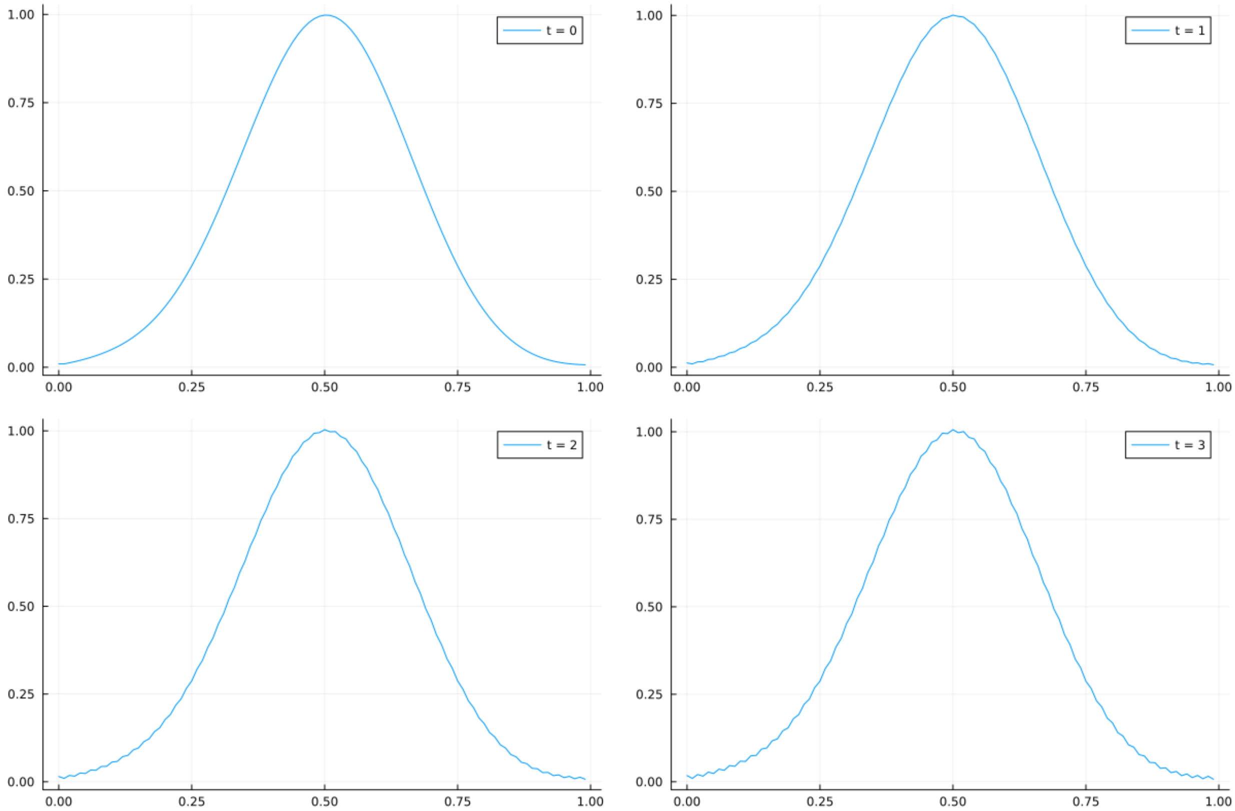

Figure 1: Solutions for $t \in [0, 1, 2, 3]$.

□

Problem 4 *Consider the non-constant coefficient reaction diffusion equation*

$$\begin{cases} u_t = \kappa(x)u_{xx} + u(1-u)(u-1/2), \quad \kappa(x) = \sin^4(2\pi x - \pi), \\ u(x,0) = \eta(x), \\ u(0,t) = u(1,t), \\ u_x(0,t) = u_x(1,t). \end{cases}$$

Use the initial condition

$$\eta(x) = \frac{(1+\tanh(20(x-0.25)))(1+\tanh(20(-x+0.75)))}{4}.$$

Solve this problem by forming the diffusion matrix

$$A_\kappa = \frac{1}{h^2}\operatorname{diag}(\kappa(x_1),\ldots,\kappa(x_m))A, \quad A = \begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ 1 & & & 1 & -2 \end{bmatrix}.$$

Write a function `CN(U,k)` *that performs a step of size k for trapezoid applied to the MOL discretization $U'(t) = A_\kappa U(t)$. Next use RK2 (5.30) to write a function* `RK2(U,k)` *performs one time step of $U'(t) = U(t)(1-U(t))(U(t)-1/2)$ with time step k. Note that while we are solving a PDE, the MOL discretization of a term without derivatives is a system of uncoupled ODEs. The Strang splitting scheme for this problem is then written as:*

```
U = RK2(U,k/2)
U = CN(U,k)
U = RK2(U,k/2)
```

Use this scheme to plot the solution at $t = 0.01, 0.10, 1.00$. Repeat with $\kappa(x) = \cos^4(2\pi x)$.

Solution.
Consider the non-constant coefficient reaction-diffusion equation

$$\begin{cases} u_t = \kappa(x)u_{xx} + u(1-u)(u-1/2), \\ u(x,0) = \eta(x), \\ u(0,t) = u(1,t), \\ u_x(0,t) = u_x(1,t), \end{cases}$$

where $\kappa(x) = \sin^4(2\pi x - \pi)$ and

$$\eta(x) = \frac{(1+\tanh(20(x-0.25)))(1+\tanh(20(-x+0.75)))}{4}.$$

To solve this equation, we will use the diffusion matrix

$$A_\kappa = \frac{1}{h^2}\text{diag}(\kappa(x_1), \ldots, \kappa(x_m))A, \quad A = \begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ 1 & & & 1 & -2 \end{bmatrix},$$

with the Crank-Nicolson method along with the RK2 method. We created the Crank-Nicolson method using the following Julia code:

```
function CN(U,k)
  h = k
  m = convert(Int64,1/h)-1
  xs = h:h:1-h
  A = SymTridiagonal(fill(-2.0,m),fill(1.0,m-1)) |> sparse
  A[1,end] = 1
  A[end,1] = 1
  diag = Diagonal(map(k,xs)) |> Matrix
  Ak = (1/h^2) * diag * A
  U = (I - (k/2) * Ak) \ (I + (k/2) * Ak)*U
end
```

We created the RK2 method that performs $U'(t) = U(t)(1 - U(t))(U(t) - 1/2)$ with time step k with the following Julia code:

```
function RK2(U,k)
  f = u -> u*(1 - u)*(u - 1/2)
  U1 = U + k/2*(map(f,U))
  U2 = U1 + k*map(f,U)
end
```

Then we used the Strange splitting scheme:

```
U = RK2(U,k/2)
U = CN(U,k)
U = RK2(U,k/2)
```

to solve the problem at $t \in [0.01, 0.10, 1.00]$ which can be seen in Figure 2. We repeated the process but with $\kappa(x) = \cos^4(2\pi x)$ to get the results seen in Figure 3.

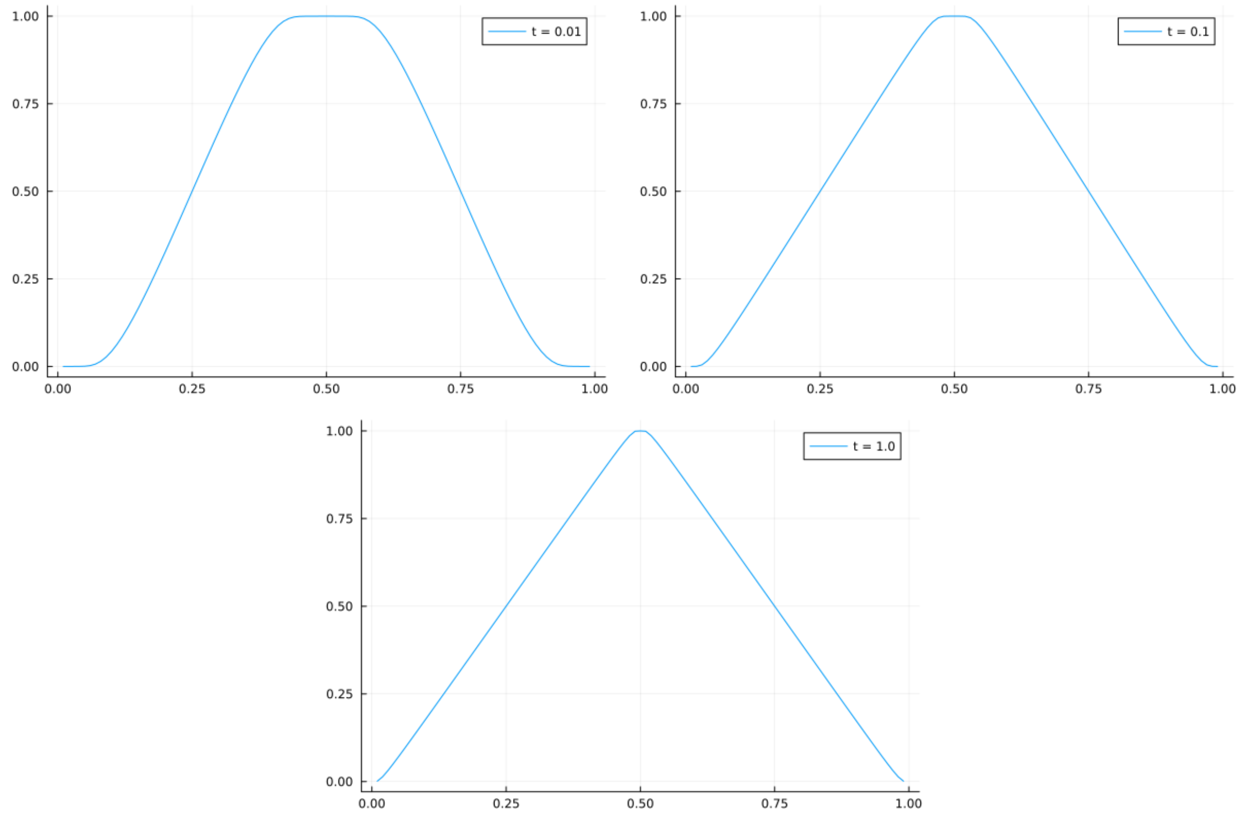

Figure 2: Solutions for $t \in [0.01, 0.10, 1.00]$ using the Strange splitting scheme with $\kappa(x) = \sin^4(2\pi x - \pi)$.

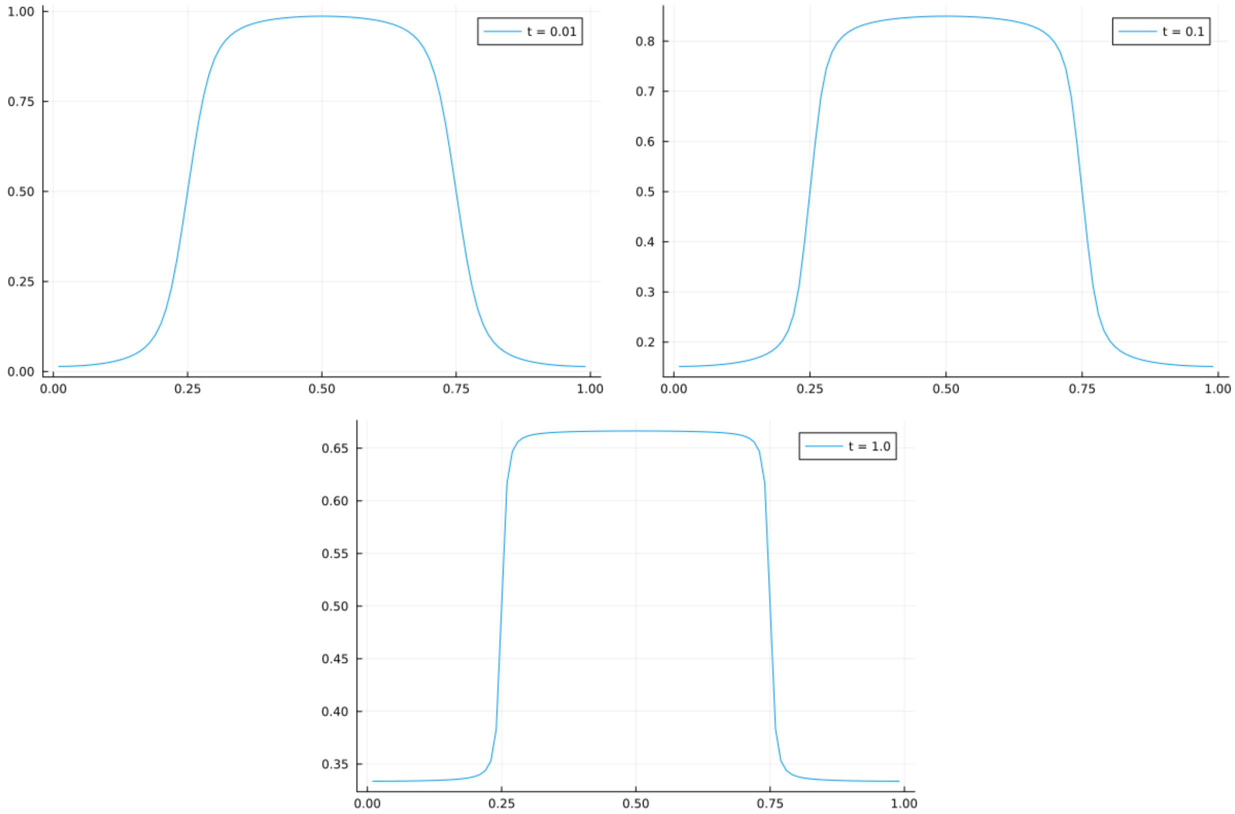

Figure 3: Solutions for $t \in [0.01, 0.10, 1.00]$ using the Strange splitting scheme with $\kappa(x) = \cos^4(2\pi x)$.

□

Problem 5 *Prove that for two $n \times n$ matrices A, B that*

$$e^{k(A+B)} - e^{k/2A}e^{kB}e^{k/2A} = O(k^3).$$

Solution.
Consider the two $n\times n$ matrices A and B. Observe that Taylor expanding matrix exponential yields

$$\begin{aligned}
e^{k(A+B)} &= I + k(A+B) + \frac{k^2}{2}(A+B)^2 + \mathcal{O}(k^3)\\
&= I + k(A+B) + \frac{k^2}{2}(A^2 + AB + BA + B^2) + \mathcal{O}(k^3)
\end{aligned}$$

and

$$\begin{aligned}
e^{k/2A}e^{kB}e^{k/2A} &= \left(I + \frac{kA}{2} + \frac{k^2A^2}{8} + \mathcal{O}(k^3)\right)\left(I + kB + \frac{k^2B^2}{2} + \mathcal{O}(k^3)\right)\left(I + \frac{kA}{2} + \frac{k^2A^2}{8} + \mathcal{O}(k^3)\right)\\
&= \left(I + k\left(\frac{A}{2} + B\right) + k^2\left(\frac{A^2}{8} + \frac{AB}{2} + \frac{B^2}{2}\right) + \mathcal{O}(k^3)\right)\left(I + \frac{kA}{2} + \frac{k^2A^2}{8} + \mathcal{O}(k^3)\right)\\
&= I + k(A+B) + k^2\left(\frac{A^2}{8} + \frac{AB}{2} + \frac{B^2}{2} + \frac{A^2}{8} + \frac{A^2}{4} + \frac{BA}{2}\right) + \mathcal{O}(k^3)\\
&= I + k(A+B) + k^2\left(\frac{A^2}{2} + \frac{BA}{2} + \frac{B^2}{2} + \frac{AB}{2}\right) + \mathcal{O}(k^3)\\
&= I + k(A+B) + \frac{k^2}{2}\big(A^2 + AB + BA + B^2\big) + \mathcal{O}(k^3).
\end{aligned}$$

Thus we have that

$$e^{k(A+B)} - e^{k/2A}e^{kB}e^{k/2A} = \mathcal{O}(k^3).$$

□

Problem 6 *Consider the Jacobi iteration (4.4) for the linear system $Au = f$ arising from a centered difference approximation of the boundary value problem $u_{xx}(x) = f(x)$. Show that this iteration can be interpreted as forward Euler time stepping applied to the MOL equations arising from a centered difference discretization of the heat equation $u_t(x,t) = u_{xx}(x,t) - f(x)$ with time step $k = \frac{1}{2}h^2$.*

Note that if the boundary conditions are held constant then the solution to this heat equation decays to the steady state solution that solves the boundary value problem. Marching to steady state with an explicit method is one way to solve the boundary value problem, though as we saw in Chapter 4 this is a very inefficient way to compute the steady state.

Solution.

Consider the boundary value problem $u_{xx} = f(x)$. Using a centered difference approximation we can rewrite the problem as $Au = f$. Applying Jacobi iteration to the linear system yields

$$u_j^{n+1} = \frac{1}{2}\big(u_{j-1}^n + u_{j+1}^n - h^2 f_i\big). \tag{1}$$

We wish to show that the iteration in Eq. (1) is analogous to a forward Euler applied to the MOL using a centered difference discretization of the heat equation

$$u_t = u_{xx} - f.$$

Applying the MOL to the heat equations yields

$$U_j'(t) = \frac{U_{j+1}(t) - 2U_j(t) + U_{j-1}(t)}{h^2} - f_j.$$

Next, we apply forward Euler

$$\frac{U_j^{n+1}(t) - U_j^n(t)}{k} = \frac{U_{j+1}(t) - 2U_j(t) + U_{j-1}(t)}{h^2} - f_j,$$

and letting the step size be $k = \frac{1}{2}h^2$ gives

$$\begin{aligned}
\frac{U_j^{n+1}(t) - U_j^n(t)}{h^2/2} &= \frac{U_{j+1}^n(t) - 2U_j^n(t) + U_{j-1}^n(t)}{h^2} - f_j, \\
U_j^{n+1}(t) - U_j^n(t) &= \frac{1}{2}\big(U_{j+1}^n(t) - 2U_j^n(t) + U_{j-1}^n(t)\big) - \frac{h^2}{2} f_j \\
U_j^{n+1}(t) &= \frac{1}{2}\big(U_{j+1}^n(t) + U_{j-1}^n(t) - h^2 f_j\big),
\end{aligned}$$

which is analogous to Eq. (1). Thus we have shown Jacobi iteration applied to the linear system Au = f arising from a centered difference approximation of the boundary value problem $u_{xx} = f$ can be interpreted as forward Euler time stepping applied to the MOL equations arising from a centered difference discretization of the heat equation $u_t = u_{xx} - f$ with time step $k = \frac{1}{2}h^2$.

□

Problem 7 *From the previous problem, we see that the Jacobi iteration for solving the boundary value problem $u_{xx}(x) = f(x)$ can be viewed as an explicit time-stepping method for the heat equation $u_t(x,t) = u_{xx}(x,t) - f(x)$ with a time step $k = h^2/2$. Now consider the Gauss-Seidel method for solving the linear system,*

$$U_j^{n+1} = \frac{1}{2}\left(U_{j-1}^{n+1} + U_{j+1}^{n} - h^2 f\left(x_j\right)\right)$$

This can be viewed as a time stepping method for some PDE. Compute the modified equation for this finite difference method and determine what PDE it is consistent with if we let $k = h^2/2$ again. Comment on how this relates to the observation in Section 4.2.1 that Gauss-Seidel takes roughly half as many iterations as Jacobi to converge.

Solution.

Consider the Gauss-Seidel method for solving the linear system

$$U_j^{n+1} = \frac{1}{2}\big(U_{j-1}^{n+1} + U_{j+1}^{n} - h^2 f(x_j)\big).$$

To figure out which PDE this time stepping method solves, we will compute the modified equation by assuming $U_j^n = v(x_j, t_n)$ for a smooth function $v(x,t)$ and plugging this into the method yields

$$2v(x, t+k) = v(x-h, t+k) + v(x+h, t) - h^2 f(x).$$

Now Taylor expanding the left-hand side gives

$$2v + 2kv_t + k^2 v_{tt} + \mathcal{O}(k^3),$$

and Taylor expanding the right-hand sides gives

$$\begin{aligned}
v(x-h,t+k) + v(x+h,t) - h^2 f(x) &= \left(v(x,t+k) - hv_x(x,t+k) + \frac{h^2}{2}v_{xx}(x,t+k)\right) \\
&\quad + \left(v(x,t) + hv_x(x,t) + \frac{h^2}{2}v_{xx}(x,t)\right) - h^2 f + \mathcal{O}(h^3) \\
&= \big(v + kv_t + k^2 v_{tt} + \mathcal{O}(k^3)\big) - h\big(v_x + kv_{xt} + \mathcal{O}(k^2)\big) \\
&\quad + \frac{h^2}{2}(v_{xx} + \mathcal{O}(k)) + v + hv_x + \frac{h^2}{2}v_{xx} - h^2 f(x) + \mathcal{O}(h^3) \\
&= 2v + kv_t + h^2 v_{xx} + k^2 v_{tt} - khv_{xt} - h^2 f(x) \\
&\quad + \mathcal{O}(h^3 + k^2 h + h^2 k + k^3).
\end{aligned}$$

Thus we have the modified equation to be

$$v_t = \frac{h^2}{k}v_{xx} - hv_{xt} - \frac{h^2}{k}f(x) + \mathcal{O}(h^3 + kh + h^2 k + k^2),$$

and letting $k = h^2/2$ gives

$$v_t = 2v_{xx} - hv_{xt} - 2f(x) + \mathcal{O}(h^2) = 2(v_{xx} - f(x)) + \mathcal{O}(h).$$

Thus we see that if we let time step with $k = h^2/2$, then the Gauss-Seidel method for solving the linear system

$$U_j^{n+1} = \frac{1}{2}\big(U_{j-1}^{n+1} + U_{j+1}^{n} - h^2 f(x_j)\big).$$

is consistent with the PDE

$$u_t = 2(u_{xx}(x,t) - f(x)).$$

In Problem 6, we found that Jacobi iteration for the linear system $Au = f$ arising from a centered difference approximation of the boundary value problem $u_{xx}(x) = f(x)$ can be interpreted as solving $u_t(x,t) = u_{xx}(x,t) - f(x)$. Notice that this is 1/2 the RHS as the solution we PDE we found above and thus it makes sense that Gauss-Seidel takes roughly half the time as many iterations as Jacobi to converge.

□