1 Chapter 5: Exercises Solutions

Consider the following finite difference method for the diffusion equation

$$-\frac{1}{2}\mu u_{j-1}^{i+1} + (1+\mu)u_j^{i+1} - \frac{1}{2}\mu u_{j+1}^{i+1} = \frac{1}{2}\mu u_{j-1}^{i} + (1-\mu)u_j^{i} + \frac{1}{2}\mu u_{j+1}^{i}$$

where $\mu = \tau/h^2$.

1. Show that the method has a second-order local truncation error. That is, let $\tilde{u}_j^i = u(x_j, t_i)$, where $x_j = jh$, $t_i = i\tau$, show that the method can be expressed as

$$\frac{u_j^{i+1} - u_j^i}{\tau} = \frac{1}{2}\left(\frac{u_{j+1}^{i+1} - 2u_j^{i+1} + u_{j-1}^{i+1}}{h^2} + \frac{u_{j+1}^{i} - 2u_j^{i} + u_{j-1}^{i}}{h^2}\right)$$

and then show that for $\tau, h \to 0$ (assuming all the relevant partial derivatives are bounded),

$$\frac{\tilde{u}_j^{i+1} - \tilde{u}_j^i}{\tau} - \frac{1}{2}\left(\frac{\tilde{u}_{j+1}^{i+1} - 2\tilde{u}_j^{i+1} + \tilde{u}_{j-1}^{i+1}}{h^2} + \frac{\tilde{u}_{j+1}^{i} - 2\tilde{u}_j^{i} + \tilde{u}_{j-1}^{i}}{h^2}\right) = \mathcal{O}(\tau^2) + \mathcal{O}(h^2).$$

Hint: use Taylor's theorem to expand the exact solution about $(x,t) = (x_j, t_{i+1/2})$, where $t_{i+1/2} = (i+1/2)\tau = t_i + \tau/2$.

Solution The method can be expressed as

$$\frac{u_j^{i+1} - u_j^i}{\tau} = \frac{1}{2}\left(\frac{u_{j+1}^{i+1} - 2u_j^{i+1} + u_{j-1}^{i+1}}{h^2} + \frac{u_{j+1}^{i} - 2u_j^{i} + u_{j-1}^{i}}{h^2}\right).$$

Using Taylor's theorem to expand the solution about $(x_j, t_{i+1/2})$, we have that as $\tau \to 0$,

$$\begin{aligned}
\tilde{u}_j^{i+1} &= u(x_j, t_{i+1}) \\
&= u(x_j, t_{i+1/2} + \tau/2) \\
&= u(x_j, t_{i+1/2}) + \frac{\tau}{2}u_t(x_j, t_{i+1/2}) + \frac{\tau^2}{8}u_{tt}(x_j, t_{i+1/2}) + \mathcal{O}(\tau^3)
\end{aligned}$$

and similarly

$$\begin{aligned}
\tilde{u}_j^{i} &= u(x_j, t_{i}) \\
&= u(x_j, t_{i+1/2} - \tau/2) \\
&= u(x_j, t_{i+1/2}) - \frac{\tau}{2}u_t(x_j, t_{i+1/2}) + \frac{\tau^2}{8}u_{tt}(x_j, t_{i+1/2}) + \mathcal{O}(\tau^3)
\end{aligned}$$

Therefore,

$$\frac{\tilde{u}_j^{i+1} - \tilde{u}_j^i}{\tau} = u_t(x_j, t_{i+1/2}) + \mathcal{O}(\tau^2)$$

We have shown in the lecture notes (using Taylor's theorem) that

$$\frac{\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i}{h^2} = u_{xx}(x_j, t_i) + \mathcal{O}(h^2),$$

therefore, using Taylor's theorem again, but in the time variable,

$$\begin{aligned}
\frac{\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i}{h^2} &= u_{xx}(x_j, t_i) + \mathcal{O}(h^2) \\
&= u_{xx}(x_j, t_{i+1/2} - \tau/2) + \mathcal{O}(h^2) \\
&= u_{xx}(x_j, t_{i+1/2}) - \frac{\tau}{2} u_{txx}(x_j, t_{i+1/2}) + \mathcal{O}(\tau^2) + \mathcal{O}(h^2).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\frac{\tilde{u}_{j+1}^{i+1} - 2\tilde{u}_j^{i+1} + \tilde{u}_{j-1}^{i+1}}{h^2} &= u_{xx}(x_j, t_{i+1}) + \mathcal{O}(h^2) \\
&= u_{xx}(x_j, t_{i+1/2} + \tau/2) + \mathcal{O}(h^2) \\
&= u_{xx}(x_j, t_{i+1/2}) + \frac{\tau}{2} u_{txx}(x_j, t_{i+1/2}) + \mathcal{O}(\tau^2) + \mathcal{O}(h^2)
\end{aligned}$$

and hence

$$\begin{aligned}
&\frac{\tilde{u}_j^{i+1} - \tilde{u}_j^i}{\tau} - \frac{1}{2}\left(\frac{\tilde{u}_{j+1}^{i+1} - 2\tilde{u}_j^{i+1} + \tilde{u}_{j-1}^{i+1}}{h^2} + \frac{\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i}{h^2}\right) \\
&= u_t(x_j, t_{i+1/2}) - u_{xx}(x_j, t_{i+1/2}) + \mathcal{O}(\tau^2) + \mathcal{O}(h^2) \\
&= \mathcal{O}(\tau^2) + \mathcal{O}(h^2)
\end{aligned}$$

2. Use the Von Neumann method to show that the method is unconditionally stable.

Solution Setting $u_j^i = \lambda^i \mathrm{e}^{\mathrm{i}kx_j}$ in

$$-\frac{1}{2}\mu u_{j-1}^{i+1} + (1+\mu)u_j^{i+1} - \frac{1}{2}\mu u_{j+1}^{i+1} = \frac{1}{2}\mu u_{j-1}^i + (1-\mu)u_j^i + \frac{1}{2}\mu u_{j+1}^i$$

we find that

$$\lambda\left[1 - \frac{\mu}{2}\left(\mathrm{e}^{-\mathrm{i}kh} - 2 + \mathrm{e}^{\mathrm{i}kh}\right)\right] = 1 + \frac{\mu}{2}\left(\mathrm{e}^{-\mathrm{i}kh} - 2 + \mathrm{e}^{\mathrm{i}kh}\right)$$

and since $\left(\mathrm{e}^{-\mathrm{i}kh} - 2 + \mathrm{e}^{\mathrm{i}kh}\right) = \left(\mathrm{e}^{-\mathrm{i}kh/2} - \mathrm{e}^{\mathrm{i}kh/2}\right)^2 = -4\sin^2(kh/2)$, it follows that

$$\lambda = \frac{1 - 2\mu\sin^2(kh/2)}{1 + 2\mu\sin^2(kh/2)}$$

Since $|1 - 2\mu\sin^2(kh/2)| \le |1 + 2\mu\sin^2(kh/2)|$ for $\mu > 0$, $k \in \mathbb{Z}$ and $h > 0$, it follows that $|\lambda| \le 1$ for these same parameter values and therefore the method is unconditionally stable.

3. Suppose the finite difference method is used to approximate the solution to the diffusion equation $u_t = u_{xx}$ for $(x,t) \in (0,1) \times (0,T)$ subject to $u(0,t) = \varphi_0(t)$, $u(1,t) = \varphi_1(t)$ for $t \in [0,T]$ and $u(x,0) = f(x)$ for $x \in [0,1]$. Let $u(x_j, t_i) \approx u_j^i$, where $x_j = jh$, $h = 1/(n_x+1)$, $j = 0, \ldots, n_x+1$, $t_i = i\tau$, $\tau = \mu h^2$ and set

$$\mathbf{u}^i = \begin{bmatrix} u_1^i \\ \vdots \\ u_{n_x}^i \end{bmatrix} \in \mathbb{R}^{n_x}.$$

The finite difference method can be expressed as

$$L\mathbf{u}^{i+1} = R\mathbf{u}^i + \mathbf{k}^i,$$

where $L, R \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{k}^i \in \mathbb{R}^{n_x}$. Give the matrices L and R and the vector $\mathbf{k}^i$.

Solution

$$\underbrace{\begin{bmatrix} 1+\mu & -\mu/2 & & & \\ -\mu/2 & 1+\mu & -\mu/2 & & \\ & \ddots & \ddots & \ddots & \\ & & -\mu/2 & 1+\mu & -\mu/2 \\ & & & -\mu/2 & 1+\mu \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} u_1^{i+1} \\ \vdots \\ \vdots \\ \vdots \\ u_{n_x}^{i+1} \end{bmatrix}}_{\mathbf{u}^{i+1}} =$$

$$\underbrace{\begin{bmatrix} 1-\mu & \mu/2 & & & \\ \mu/2 & 1-\mu & \mu/2 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu/2 & 1-\mu & \mu/2 \\ & & & \mu/2 & 1-\mu \end{bmatrix}}_{R} \underbrace{\begin{bmatrix} u_1^{i} \\ \vdots \\ \vdots \\ \vdots \\ u_{n_x}^{i} \end{bmatrix}}_{\mathbf{u}^{i}} + \underbrace{\begin{bmatrix} \mu(\varphi_0(t_i) + \varphi_0(t_{i+1}))/2 \\ 0 \\ \vdots \\ 0 \\ \mu(\varphi_1(t_i) + \varphi_1(t_{i+1}))/2 \end{bmatrix}}_{\mathbf{k}^i}$$

4. What are the eigenvalues of the matrix $A := L^{-1}R$? Find a bound on the spectral radius of A. Hint: consider the eigendecomposition (spectral factorisation) of L and R.

Solution The matrices L and R are TST (Tridiagonal, symmetric, Toeplitz); from the notes we know that an $n_x \times n_x$ TST matrix with α on the main diagonal and β on the super and subdiagonals have eigenvalues

$$\lambda_j = \alpha + 2\beta \cos\left(\frac{\pi j}{n_x + 1}\right), \qquad j = 1, \ldots, n_x.$$

Therefore, for $j = 1, \ldots, n_x$, the eigenvalues of L are

$$\lambda_j^L = 1 + \mu - \mu \cos(\pi x_j) = 1 + \mu - \mu \left(1 - 2\sin^2(\pi x_j/2)\right) = 1 + 2\mu \sin^2(\pi x_j/2)$$

and those of R are

$$\lambda_j^R = 1 - \mu + \mu \cos(\pi x_j) = 1 - \mu + \mu \left(1 - 2\sin^2(\pi x_j/2)\right) = 1 - 2\mu \sin^2(\pi x_j/2)$$

Let

$$\Lambda^L = \begin{bmatrix} \lambda_1^L & & & \\ & \lambda_2^L & & \\ & & \ddots & \\ & & & \lambda_{n_x}^L \end{bmatrix}, \qquad \Lambda^R = \begin{bmatrix} \lambda_1^R & & & \\ & \lambda_2^R & & \\ & & \ddots & \\ & & & \lambda_{n_x}^R \end{bmatrix},$$

then we know from the notes that

$$L = Q\Lambda^L Q^\top, \qquad R = Q\Lambda^R Q^\top$$

where

$$Q = [\mathbf{q}_1|\mathbf{q}_2|\cdots|\mathbf{q}_{n_x}] \in \mathbb{R}^{n_x \times n_x},$$

is an orthogonal matrix and the entries of the eigenvectors $\mathbf{q}_j \in \mathbb{R}^{n_x}$ are

$$q_{j,\ell} = \sqrt{\frac{2}{n_x+1}} \sin\left(\frac{\pi j \ell}{n_x+1}\right), \qquad \ell = 1, \ldots, n_x.$$

Therefore, we have that

$$A = L^{-1}R = Q\left(\Lambda^L\right)^{-1} \Lambda^R Q^\top$$

which implies the eigenvalues of A are

$$\sigma(A) = \left\{\frac{\lambda_j^R}{\lambda_j^L}, j = 1, \ldots, n_x\right\}$$

and since $|\lambda_j^R| \leq |\lambda_j^L|$ for $\mu > 0$, $j = 1, \ldots, n_x$, the spectral radius of A is bounded by 1, i.e., $\rho(A) \leq 1$.

5. Let

$$u(x,0) = f(x) = \sin\frac{1}{2}\pi x + \frac{1}{2}\sin 2\pi x, \qquad 0 \leq x \leq 1,$$

and

$$u(0,t) = \varphi_0(t) = 0, \qquad u(1,t) = \varphi_1(t) = \mathrm{e}^{-\pi^2 t/4}, \qquad t \geq 0,$$

then the exact solution to the diffusion equation is

$$u(x,t) = \mathrm{e}^{-\pi^2 t/4}\sin\frac{1}{2}\pi x + \frac{1}{2}\mathrm{e}^{-4\pi^2 t}\sin 2\pi x, \qquad 0 \leq x \leq 1, \qquad t \geq 0.$$

Using the method in question 3 and the backward Euler method, approximate the exact solution for $T = 1$, $\mu = n_x$ and plot the maximum error of the two methods for $n_x = 2^k$, $k = 4, 5, \ldots, 12$. Comment on your results.

```
using Plots, LinearAlgebra
```

```julia
function BackwardEuler(f,ϕ0,ϕ1,nx,μ,T)

    x = range(0,1,nx + 2)
    h = 1/(nx+1)
    τ = μ*h^2
    t = 0:τ:T
    nt = length(t)-1
    B = SymTridiagonal(fill((1 + 2μ),nx),fill(-μ,nx-1))
    u = zeros(nt+1,nx+2)
    u[:,1] = ϕ0.(t)
    u[:,nx+2] = ϕ1.(t)
    u[1,2:nx+1] = f.(x[2:nx+1])

    for i = 1:nt
        b = u[i,2:nx+1]
        b[1] += μ*ϕ0(t[i+1])
        b[nx] += μ*ϕ1(t[i+1])
        u[i+1,2:nx+1] = B\b
    end

    u, x, t
end
```

```
BackwardEuler (generic function with 1 method)
```

```julia
function CrankNick(f,ϕ0,ϕ1,nx,μ,T)

    x = range(0,1,nx + 2)
    h = 1/(nx+1)
    τ = μ*h^2
    t = 0:τ:T
    nt = length(t)-1
    L = SymTridiagonal(fill((1 + μ),nx),fill(-μ/2,nx-1))
    R = SymTridiagonal(fill((1 - μ),nx),fill(μ/2,nx-1))
    u = zeros(nt+1,nx+2)
    u[:,1] = ϕ0.(t)
    u[:,nx+2] = ϕ1.(t)
    u[1,2:nx+1] = f.(x[2:nx+1])

    for i = 1:nt
        b = R*u[i,2:nx+1]
        b[1] += μ*(ϕ0(t[i]) + ϕ0(t[i+1]))/2
        b[nx] += μ*(ϕ1(t[i]) + ϕ1(t[i+1]))/2
        u[i+1,2:nx+1] = L\b
    end

    u, x, t
end
```

```
CrankNick (generic function with 1 method)
```

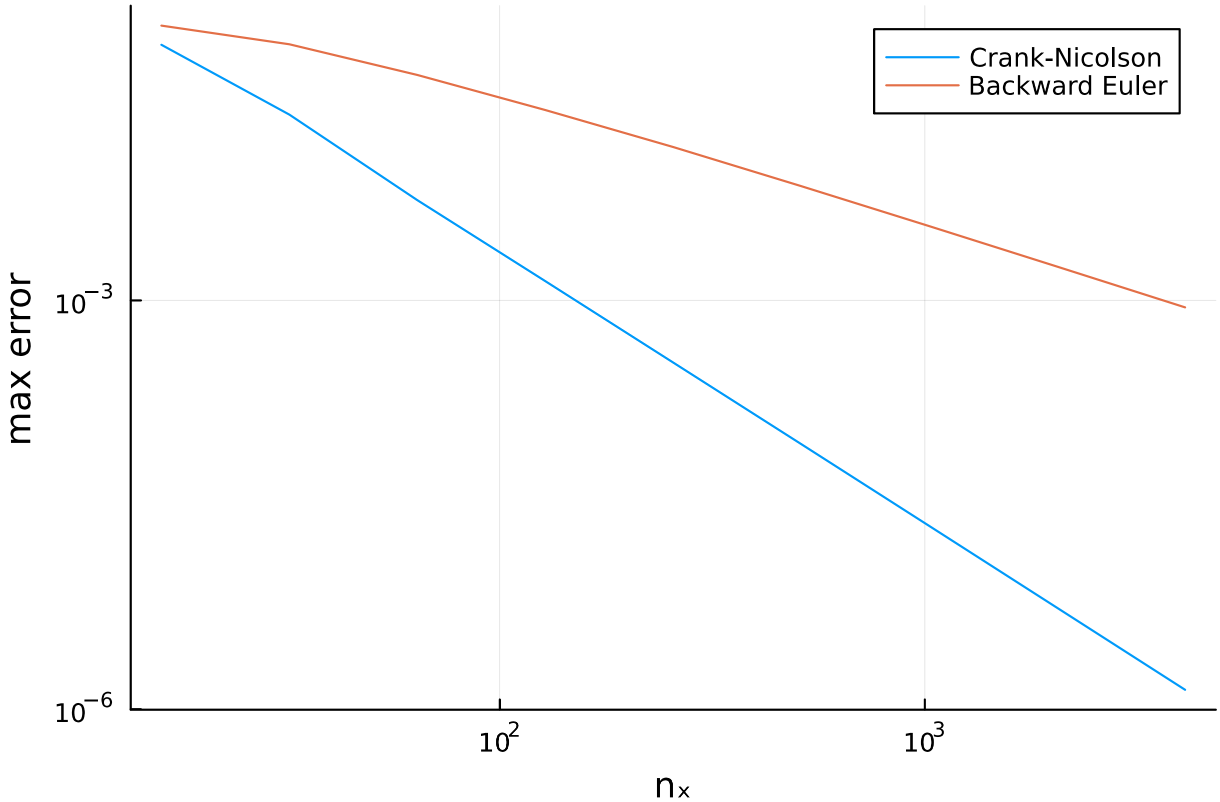
```julia
f = x -> sin(π*x/2) + 0.5*sin(2π*x)
ϕ1 = t -> exp(-π^2*t/4)
ϕ0 = t -> 0
T = 1
ue = (x,t) -> exp(-π^2*t/4)*sin(π*x/2) + 0.5*exp(-4*π^2*t)*sin(2π*x);

cerr = []
berr = []
nv = 2 .^(4:12)
```

```julia
for nx = nv
    global cerr, berr
    u,x,t = @time CrankNick(f,ϕ0,ϕ1,nx,Float64(nx),T)
    nt = length(t) -1
    xx = x' .* ones(nt+1)
    tt = ones(nx+2)' .* t
    error = abs.(ue.(xx,tt) - u)
    cerr = push!(cerr,maximum(error))
    u,~,~ = @time BackwardEuler(f,ϕ0,ϕ1,nx,Float64(nx),T)
    error = abs.(ue.(xx,tt) - u)
    berr = push!(berr,maximum(error))
end
```

```
0.549024 seconds (1.16 M allocations: 58.563 MiB, 4.48% gc time, 99.90% c
ompilation time)
  0.063411 seconds (100.84 k allocations: 4.819 MiB, 99.89% compilation tim
e)
  0.000039 seconds (184 allocations: 66.734 KiB)
  0.000040 seconds (148 allocations: 54.953 KiB)
  0.000135 seconds (345 allocations: 224.609 KiB)
  0.000160 seconds (277 allocations: 186.359 KiB)
  0.000543 seconds (665 allocations: 831.672 KiB)
  0.000507 seconds (533 allocations: 691.422 KiB)
  0.002197 seconds (1.30 k allocations: 3.202 MiB)
  0.003284 seconds (1.04 k allocations: 2.662 MiB)
  0.026735 seconds (2.58 k allocations: 12.401 MiB, 70.68% gc time)
  0.007918 seconds (2.07 k allocations: 10.322 MiB)
  0.045676 seconds (5.14 k allocations: 48.800 MiB, 27.27% gc time)
  0.038093 seconds (4.12 k allocations: 40.643 MiB, 21.34% gc time)
  0.218152 seconds (10.27 k allocations: 193.596 MiB, 16.07% gc time)
  0.168330 seconds (8.21 k allocations: 161.283 MiB, 4.31% gc time)
  1.349141 seconds (41.00 k allocations: 769.626 MiB, 36.63% gc time)
  1.138843 seconds (32.80 k allocations: 641.314 MiB, 33.73% gc time)
```

```julia
plot(nv,cerr;yscale=:log10,xscale=:log10,label="Crank-Nicolson",xlabel="n_x")
plot!(nv,berr;yscale=:log10,xscale=:log10,label="Backward Euler",ylabel="max error")
```



For the Crank-Nicolson method, we estimate the slope of the curve as follows:

```
log(cerr[end]/cerr[end-4])/log(nv[end]/nv[end-4])
```

```
-1.9946564796544979
```

For the backward Euler method, the slope is approximately

```
log(berr[end]/berr[end-3])/log(nv[end]/nv[end-3])
```

```
-0.9854061549915227
```

Hence, for the Crank-Nicolson method, the error decays as $\mathcal{O}(n_x^{-2})$ and for the backward Euler method, the error decays as $\mathcal{O}(n_x^{-1})$ as $n_x \to \infty$.