

1 Problem sheet 2: 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. **[5 marks]** 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 $\tau, h \rightarrow 0$ (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 $\tau \rightarrow 0$,

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

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. **[5 marks]** Use the Von Neumann method to show that the method is unconditionally stable.

Solution Setting $u_j^i = \lambda^i e^{ikx_j}$ 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

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

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

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

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

3. **[5 marks]** 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 $x_j = jh$, $h = 1/(n_x + 1)$, $j = 0, \dots, n_x + 1$, $t_i = i\tau$, $\tau = \mu h^2$ 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. **[5 marks]** 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

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

Therefore, for $j = 1, \dots, n_x$, 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

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

Let

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

then we know from the notes that

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

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

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

Therefore, we have that

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

which implies the eigenvalues of A are

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

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

5. **[10 marks]** Let

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

and

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

then the exact solution to the diffusion equation is

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

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 $n_x = 2^k$, $k = 4, 5, \dots, 12$. Comment on your results.

```
using Plots, LinearAlgebra
```

```
function BackwardEuler(f,  $\phi_0$ ,  $\phi_1$ , nx,  $\mu$ , T)
```

```
    x = range(0, 1, nx + 2)
    h = 1/(nx+1)
     $\tau$  =  $\mu$ *h2
    t = 0: $\tau$ :T
    nt = length(t)-1
    B = SymTridiagonal(fill((1 + 2 $\mu$ ), nx), fill(- $\mu$ , nx-1))
    u = zeros(nt+1, nx+2)
    u[:, 1] =  $\phi_0$ .(t)
    u[:, nx+2] =  $\phi_1$ .(t)
    u[1, 2:nx+1] = f.(x[2:nx+1])
```

```
    for i = 1:nt
        b = u[i, 2:nx+1]
        b[1] +=  $\mu$ * $\phi_0$ (t[i+1])
        b[nx] +=  $\mu$ * $\phi_1$ (t[i+1])
        u[i+1, 2:nx+1] = B\b
    end
```

```
    u, x, t
```

```
end
```

BackwardEuler (generic function with 1 method)

```
function CrankNick(f,  $\phi_0$ ,  $\phi_1$ , nx,  $\mu$ , T)
```

```
    x = range(0, 1, nx + 2)
    h = 1/(nx+1)
     $\tau$  =  $\mu$ *h2
    t = 0: $\tau$ :T
    nt = length(t)-1
    L = SymTridiagonal(fill((1 +  $\mu$ ), nx), fill(- $\mu$ /2, nx-1))
    R = SymTridiagonal(fill((1 -  $\mu$ ), nx), fill( $\mu$ /2, nx-1))
    u = zeros(nt+1, nx+2)
    u[:, 1] =  $\phi_0$ .(t)
    u[:, nx+2] =  $\phi_1$ .(t)
    u[1, 2:nx+1] = f.(x[2:nx+1])
```

```
    for i = 1:nt
        b = R*u[i, 2:nx+1]
        b[1] +=  $\mu$ *( $\phi_0$ (t[i]) +  $\phi_0$ (t[i+1]))/2
        b[nx] +=  $\mu$ *( $\phi_1$ (t[i]) +  $\phi_1$ (t[i+1]))/2
        u[i+1, 2:nx+1] = L\b
    end
```

```
    u, x, t
```

```
end
```

CrankNick (generic function with 1 method)

```
f = x -> sin( $\pi$ *x/2) + 0.5*sin(2 $\pi$ *x)
 $\phi_1$  = t -> exp(- $\pi$ 2*t/4)
 $\phi_0$  = t -> 0
T = 1
ue = (x, t) -> exp(- $\pi$ 2*t/4)*sin( $\pi$ *x/2) + 0.5*exp(-4* $\pi$ 2*t)*sin(2 $\pi$ *x);
```

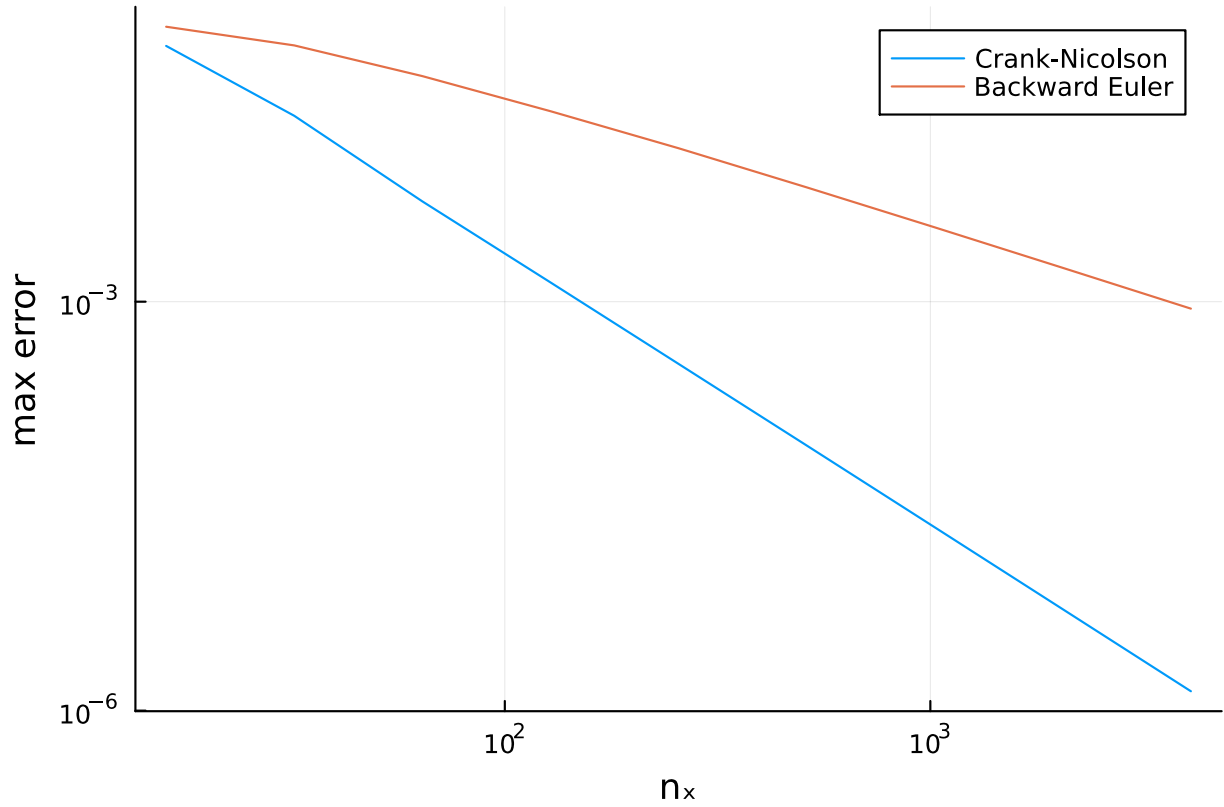
```

cerr = []
berr = []
nv = 2 .^(4:12)
for nx = nv
    global cerr, berr
    u,x,t = @time CrankNicolson(f,phi0,phi1,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,phi0,phi1,nx,Float64(nx),T)
    error = abs.(ue.(xx,tt) - u)
    berr = push!(berr,maximum(error))
end

0.229694 seconds (500.88 k allocations: 26.234 MiB, 14.08% gc time, 99.79
% compilation time)
0.076800 seconds (106.33 k allocations: 5.372 MiB, 99.92% compilation tim
e)
0.000099 seconds (185 allocations: 66.797 KiB)
0.000047 seconds (149 allocations: 55.016 KiB)
0.000145 seconds (346 allocations: 224.672 KiB)
0.000143 seconds (278 allocations: 186.422 KiB)
0.000411 seconds (666 allocations: 831.734 KiB)
0.000495 seconds (534 allocations: 691.484 KiB)
0.002290 seconds (1.31 k allocations: 3.202 MiB)
0.002509 seconds (1.05 k allocations: 2.662 MiB)
0.009629 seconds (2.59 k allocations: 12.401 MiB)
0.008831 seconds (2.07 k allocations: 10.323 MiB)
0.059492 seconds (5.15 k allocations: 48.800 MiB, 33.12% gc time)
0.045780 seconds (4.12 k allocations: 40.643 MiB, 26.55% gc time)
0.244661 seconds (10.27 k allocations: 193.596 MiB, 22.62% gc time)
0.181416 seconds (8.22 k allocations: 161.283 MiB, 4.65% gc time)
1.566246 seconds (41.00 k allocations: 769.626 MiB, 36.94% gc time)
1.271100 seconds (32.80 k allocations: 641.314 MiB, 34.91% gc time)

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 $n_x \rightarrow \infty$.