

1 Chapter 5: Consistency, stability and convergence of methods for evolutionary PDEs

1.0.1 A finite difference method for the diffusion equation

Consider the $(1 + 1)$ -dimensional diffusion equation,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in (0, 1), \quad t \in (0, T]$$

subject to the initial and boundary data

$$u(x, 0) = f(x), \quad 0 \leq x \leq 1, \quad u(0, t) = \varphi_0(t), \quad u(1, t) = \varphi_1(t),$$

We shall analyse a finite difference method for this PDE initial / boundary value problem.

We discretise the spatial variable as follows,

$$x_j = jh, \quad h = \frac{1}{n_x + 1}, \quad j = 0, \dots, n_x + 1,$$

and the temporal variable,

$$t_i = i\tau, \quad \tau = \frac{T}{n_t}, \quad i = 0, 1, 2, \dots, n_t.$$

Let

$$u(x_j, t_i) := \tilde{u}_j^i \approx u_j^i.$$

We set

$$u(x_j, 0) = \tilde{u}_j^0 = f(x_j) = u_j^0, \quad j = 1, \dots, n_x$$

and

$$u(0, t_i) = \tilde{u}_0^i = \varphi_0(t_i) = u_0^i, \quad u(1, t_i) = \tilde{u}_{n_x+1}^i = \varphi_1(t_i) = u_{n_x+1}^i, \quad i \geq 0$$

1.0.2 Order and convergence

Let's replace the time derivative with a (first-order) forward difference and the second spatial derivative with a (second-order) central difference approximation: from Taylor's theorem (see Chapter 1), there exists a $\xi \in [t_i, t_{i+1}]$ such that

$$\frac{u(x_j, t_{i+1}) - u(x_j, t_i)}{\tau} = \frac{\tilde{u}_j^{i+1} - \tilde{u}_j^i}{\tau} = u_t(x_j, t_i) + \frac{\tau}{2} u_{tt}(x_j, \xi).$$

Let's assume that $u_{tt}(x, t)$ exists and is bounded on $(x, t) \in [0, 1] \times [0, T]$, then we have

$$\frac{\tilde{u}_j^{i+1} - \tilde{u}_j^i}{\tau} = u_t(x_j, t_i) + \mathcal{O}(\tau), \quad \tau \rightarrow 0.$$

Similarly, if we assume that $u_{xxxx}(x, t)$ exists and is bounded on $(x, t) \in [0, 1] \times [0, T]$, then we have (see Chapter 1 Exercises)

$$\frac{\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i}{h^2} = u_{xx}(x_j, t_i) + \mathcal{O}(h^2), \quad h \rightarrow 0.$$

We define

$$\mu = \frac{\tau}{h^2},$$

which is known as the Courant number and specify that μ be held constant as $\tau, h \rightarrow 0$. Hence, we have that

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

We shall approximate the solution to the diffusion equation with the finite difference method

$$\frac{u_j^{i+1} - u_j^i}{\tau} - \frac{u_{j+1}^i - 2u_j^i + u_{j-1}^i}{h^2} = 0$$

or

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

This method is known as **Euler's method**. Notice we have shown that if the exact solution is substituted into the finite difference method, then the PDE is recovered exactly and the local error tends to zero as the discretisation parameters tend to zero. The method is **second order** since the local error approaches zero at the rate $\mathcal{O}(h^2)$ as $h \rightarrow 0$. A finite difference method of order $p \geq 1$ is said to be **consistent**. Notice that for consistency, we only require the error at (x_j, t_i) to go to zero as the discretisation parameters tend to zero, hence the order of a method measures the local error. A method is

said to be **convergent** if the *global* error tends to zero as the discretisation parameters tend to zero, i.e., if

$$\lim_{h \rightarrow 0} \left[\lim_{j \rightarrow x/h} \left(\lim_{i \rightarrow t/\tau} u_j^i \right) \right] = u(x, t), \quad x \in [0, 1], \quad t \in [0, T],$$

where $\mu = \tau/h^2$ is kept constant as $h \rightarrow 0$.

1.0.3 Numerical experiments with Euler's method

Before we analyse the convergence or otherwise of Euler's method, let's perform some numerical experiments. First, we note that we can express Euler's method as

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

i.e.,

$$\mathbf{u}^{i+1} = A\mathbf{u}^i + \mathbf{k}^i, \quad i = 0, \dots, n_t - 1.$$

```
using LinearAlgebra, Plots, SparseArrays,
OrdinaryDiffEq, Printf
```

```
function Euler(f, phi0, phi1, nx, mu, T)
```

```
x = range(0,1,nx + 2)
h = 1/(nx+1)
tau = mu*h^2
t = 0:tau:T
```

```

nt = length(t)-1
A = 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
    u[i+1,2:nx+1] = A*u[i,2:nx+1]
    u[i+1,2] += μ*ϕ0(t[i])
    u[i+1,nx+1] += μ*ϕ1(t[i])
end

```

u, x, t

end

Euler (generic function with 1 method)

The diffusion initial / boundary value problem with

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

and

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

has the exact solution

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

```
f = x -> sin(pi*x/2) + 0.5*sin(2pi*x)
```

```
phi1 = t -> exp(-pi^2*t/4)
```

```
phi0 = t -> 0
```

```
nx = 50
```

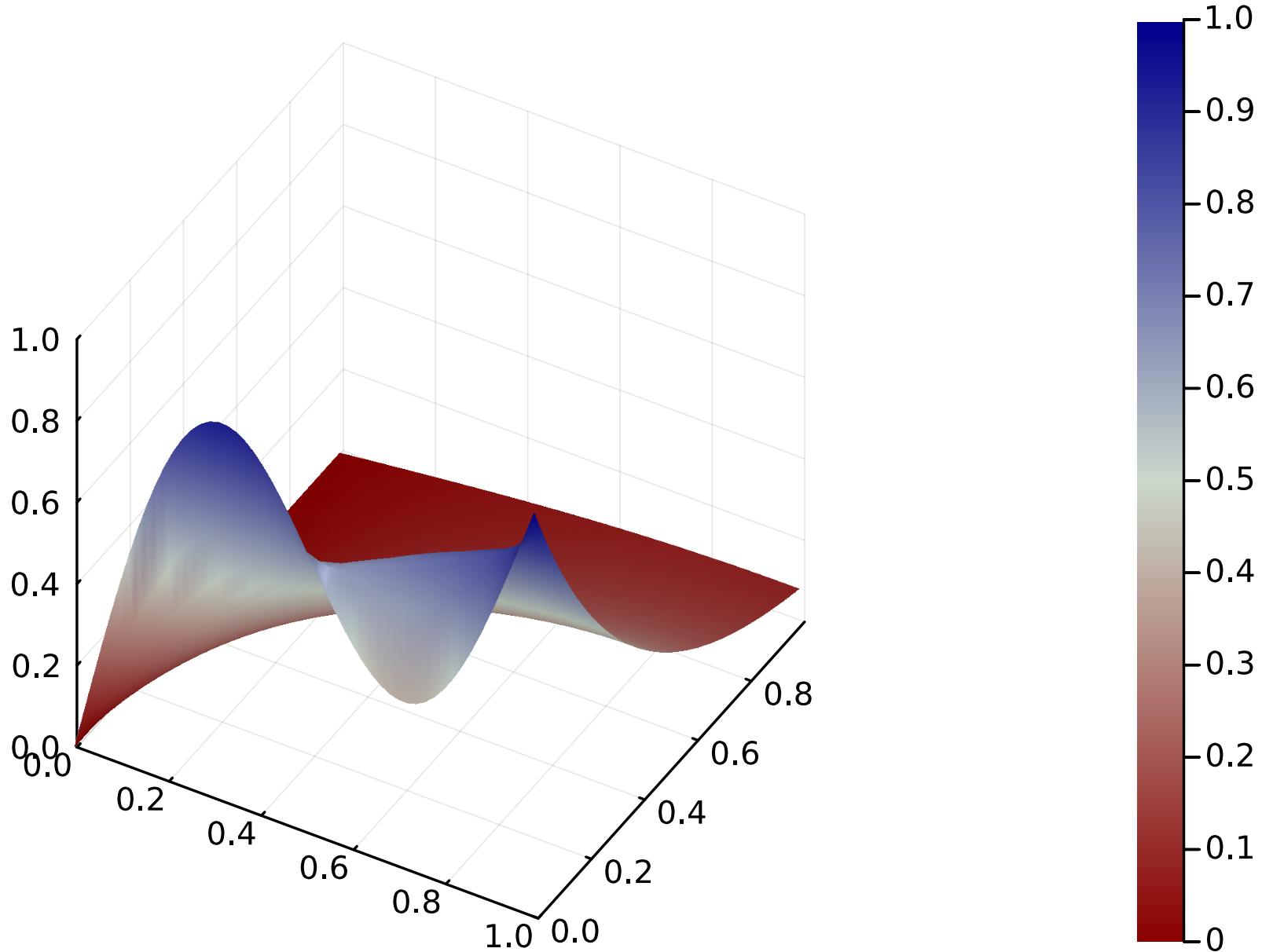
```
mu = 0.50
```

```
T = 1
```

```
u,x,t = Euler(f,phi0,phi1,nx,mu,T)
```

```
nt = length(t) -1
```

```
inc = 100
surface(x,t[1:inc:end],u[1:inc:end,:],seriescolor=:redsblues
camera=(30,40))
```



```
@gif for i = 2:50:nt+1  
    plot(x, u[1,:], size = (500, 150), label = "u0")
```

```

    plot!(x, u[i,:], label = "t = " * (@sprintf("%.2f",
t[i])), legend = :outertopright)
end

Plots.AnimatedGif("C:\\\\Users\\\\mfaso\\\\AppData\\\\Local\\\\Temp
\\\\jl_0BnnflTgTG.gif")

```

```

xx = x' .* ones(nt+1)
tt = ones(nx+2)' .* t
ue = (x,t) -> exp(-π^2*t/4)*sin(π*x/2) +
0.5*exp(-4*π^2*t)*sin(2π*x)
error = ue.(xx,tt) - u
e1 = maximum(error)

```

0.00047001331601359553

If we double n_x , the the error decreases by roughly a factor of 4.

```

nx *= 2
u,x,t = Euler(f,ϕ0,ϕ1,nx,μ,T)
@show nt = length(t) - 1
xx = x' .* ones(nt+1)

```

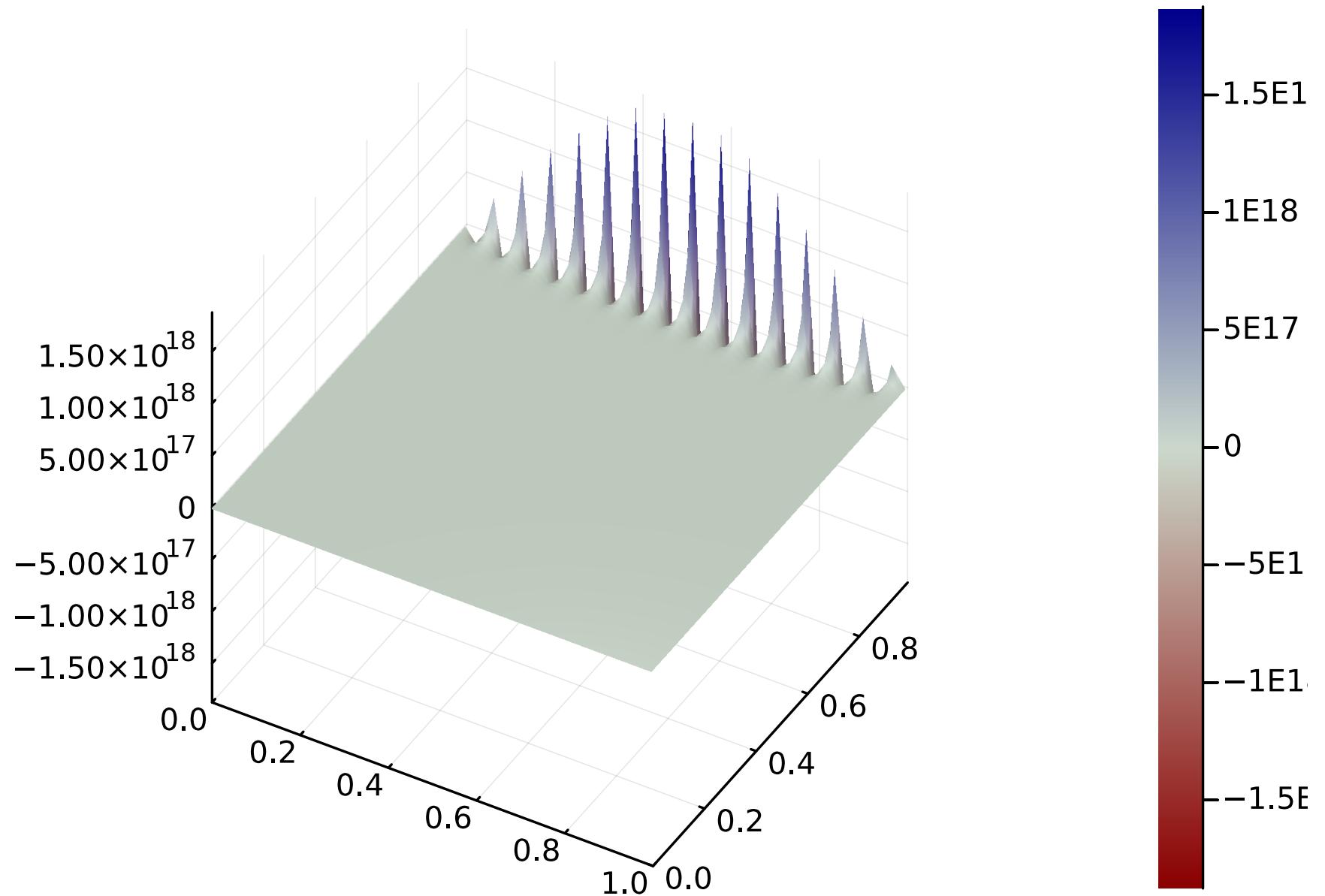
```
tt = ones(nx+2)' .* t  
error = ue.(xx,tt) - u  
e2 = maximum(error)  
e1/e2
```

nt = length(t) - 1 = 20402

3.929469183388294

This suggests the error decays as $\mathcal{O}(n_x^{-2})$, $n_x \rightarrow \infty$.

```
nx = 30  
μ = 0.51  
u,x,t = Euler(f,ϕ0,ϕ1,nx,μ,T);  
  
surface(x,t[1:30:end],u[1:30:end,:],seriescolor=:redsblues,  
camera=(30,40))
```



```
nt = length(t)-1
@gif for i = 2:4:nt÷2
```

```

    plot(x, u[1,:], size = (500, 150), label = "u0")
    plot!(x, u[i,:], label = "t = " * (@sprintf("%.2f",
t[i])), legend = :outertopright)
end

```

```

Plots.AnimatedGif("C:\\\\Users\\\\mfaso\\\\AppData\\\\Local\\\\Temp
\\\\jl_1npospyALm.gi
f")

```

Based on these experiments, we conjecture that Euler's method converges if $\mu \leq \frac{1}{2}$.

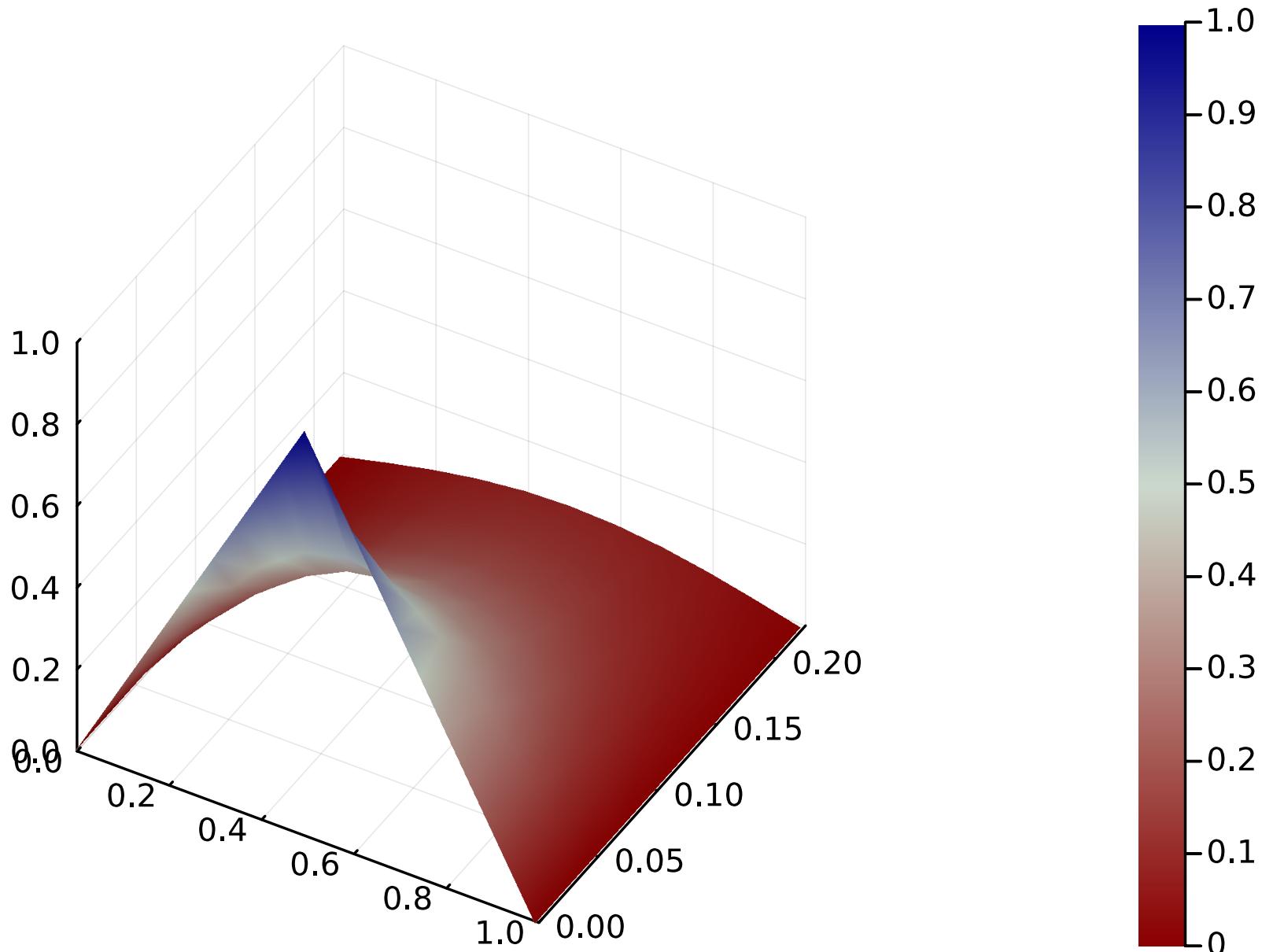
The initial condition is "smoothed out" by the diffusion equation. Here is another example:

```

f = x -> x <= 0.5 ? 2*x : 2-2*x
phi1 = t -> 0
phi0 = t -> 0
nx = 19
mu = 0.50
T = 0.25
u,x,t = Euler(f,phi0,phi1,nx,mu,T);

```

```
surface(x,t[1:30:end],u[1:30:end,:],seriescolor=:redsblues,  
camera=(30,40))
```



```

nt = length(t)-1
@gif for i = 2:nt
    plot(x, u[1,:], size = (500, 150), label = "u0")
    plot!(x, u[i,:], label = "t = " * (@sprintf("%.2f",
t[i])), legend = :outertopright)
end

Plots.AnimatedGif("C:\\\\Users\\\\mfaso\\\\AppData\\\\Local\\\\Temp
\\\\jl_jwZLqzjAkY.gi
f")

```

1.0.4 Convergence of Euler's method

Proposition If $\mu \leq \frac{1}{2}$, then Euler's method converges.

Proof Recall that

$$\frac{\tilde{u}_j^{i+1} - \tilde{u}_j^i}{\tau} - \frac{\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i}{h^2} = \mathcal{O}(h^2), \quad h \rightarrow 0,$$

which implies

$$\tilde{u}_j^{i+1} = \tilde{u}_j^i + \mu \left(\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i \right) + \mathcal{O}(h^4), \quad h \rightarrow 0.$$

Let $e_j^i = \tilde{u}_j^i - u_j^i$, then since

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

we have

$$e_j^{i+1} = e_j^i + \mu \left(e_{j+1}^i - 2e_j^i + e_{j-1}^i \right) + \mathcal{O}(h^4), \quad h \rightarrow 0.$$

This means that for sufficiently small $h > 0$, there exists a constant c such that

$$\left| e_j^{i+1} - e_j^i - \mu \left(e_{j+1}^i - 2e_j^i + e_{j-1}^i \right) \right| \leq ch^4.$$

Hence, for sufficiently small h , we have

$$\begin{aligned}
|e_j^{i+1}| &\leq \left| e_j^i + \mu(e_{j+1}^i - 2e_j^i + e_{j-1}^i) \right| + ch^4 \\
&\leq \mu|e_{j-1}^i| + |1 - 2\mu||e_j^i| + \mu|e_{j+1}^i| + ch^4 \\
&\leq (2\mu + |1 - 2\mu|) \eta^i + ch^4
\end{aligned}$$

where

$$\eta^i = \max_{j=0, \dots, n_x+1} |e_j^i|.$$

The above inequality holds for $|e_j^{i+1}|$, with $j = 0, \dots, n_x + 1$ and therefore it also holds for $|\eta^{i+1}|$. Since $0 < \mu \leq 1/2$ and $u_j^0 = \tilde{u}_j^0$, which implies that η^0 , we have that

$$\eta^{i+1} \leq \eta^i + ch^4 \leq \eta^{i-1} + 2ch^4 \leq \dots \leq \eta^0 + (i+1)ch^4 = (i+1)ch^4.$$

Therefore

$$\eta^{n_t} \leq cn_t h^4 = \frac{c}{\mu} n_t \mu h^4 = \frac{c}{\mu} n_t \tau h^2 = \frac{c}{\mu} Th^2$$

We conclude that $\eta^i \rightarrow 0$ for $i = 0, \dots, n_t$ as $h \rightarrow 0$ and therefore $e_j^i \rightarrow 0$ as $h \rightarrow 0$ for $j = 0, \dots, n_x + 1$ and $i = 0, \dots, n_t$ as $h \rightarrow 0$, which implies that the method converges. ■

Remark: Notice that the maximum error of the Euler method (assuming all relevant partial derivatives exist and are bounded) is bounded by $\frac{c}{\mu} Th^2$ for sufficiently small h . This confirms our numerical observation above that the method converges at the rate $\mathcal{O}(h^2)$ as $h \rightarrow 0$, or equivalently, at the rate $\mathcal{O}(n_x^{-2})$ as $n_x \rightarrow \infty$ (i.e., the method is second order).

1.0.5 Well-posedness and ill-posedness of PDE problems

A PDE intial and / or boundary value problem is **well-posed** if there exists a unique solution that depends continuously on the initial and boundary data. Otherwise, the PDE problem is said to be **ill-posed**.

Example: Suppose $\varphi_0(t) = \varphi_1(t) = 0$ (i.e., zero Dirichlet boundary conditions), then using the method of separation of variables, it can be shown that if

$$u(x, 0) = f(x) = \sum_{m=1}^{\infty} \alpha_m \sin \pi m x, \quad 0 \leq x \leq 1,$$

then the solution to the diffusion equation is

$$u(x, t) = \sum_{m=1}^{\infty} \alpha_m e^{-\pi^2 m^2 t} \sin \pi m x, \quad 0 \leq x \leq 1, \quad t \geq 0.$$

The 2-norm of a function g on $[0, 1]$, or L^2 norm, is defined as

$$\|g\|_2 = \left(\int_0^1 [g(x)]^2 dx \right)^{1/2}$$

Hence,

$$\begin{aligned}\|u(x, t)\|_2^2 &= \int_0^1 [u(x, t)]^2 dx = \int_0^1 \left(\sum_{m=1}^{\infty} \alpha_m e^{-\pi^2 m^2 t} \sin \pi m x \right)^2 dx \\ &= \sum_{m=1}^{\infty} \sum_{j=1}^{\infty} \alpha_m \alpha_j e^{-\pi^2 (m^2 + j^2) t} \int_0^1 \sin \pi m x \sin \pi j x dx\end{aligned}$$

and since

$$\int_0^1 \sin \pi m x \sin \pi j x dx = \begin{cases} \frac{1}{2}, & m = j, \\ 0, & \text{otherwise} \end{cases}$$

we have

$$\|u(x, t)\|_2^2 = \frac{1}{2} \sum_{m=1}^{\infty} \alpha_m^2 e^{-2\pi^2 m^2 t} \leq \frac{1}{2} \sum_{m=1}^{\infty} \alpha_m^2 = \|u(x, 0)\|_2^2.$$

Suppose that $\tilde{u}(x, t)$ is the solution to the diffusion equation with zero boundary conditions and a slightly perturbed initial condition, i.e.,

$$\tilde{u}(x, 0) = \tilde{f}(x) = \sum_{m=1}^{\infty} \tilde{\alpha}_m \sin \pi m x, \quad 0 \leq x \leq 1,$$

where

$$\|\tilde{u}(x, 0) - u(x, 0)\|_2^2 = \|\tilde{f}(x) - f(x)\|_2^2 = \frac{1}{2} \sum_{m=1}^{\infty} (\tilde{\alpha}_m - \alpha_m)^2 \leq \epsilon^2 \ll 1$$

then

$$\tilde{u}(x, t) = \sum_{m=1}^{\infty} \tilde{\alpha}_m e^{-\pi^2 m^2 t} \sin \pi m x, \quad 0 \leq x \leq 1, \quad t \geq 0$$

and therefore

$$\|\tilde{u}(x, t) - u(x, t)\|_2^2 \leq \|\tilde{u}(x, 0) - u(x, 0)\|_2^2 = \epsilon^2.$$

This shows that if $\|\tilde{u}(x, 0) - u(x, 0)\|_2$ is small, then $\|\tilde{u}(x, t) - u(x, t)\|_2$ is also small, which shows that the solution depends continuously on the initial data, i.e., the diffusion equation with zero Dirichlet boundary conditions is

well-posed.

As an example of an ill-posed problem, consider the reversed-time diffusion equation:

$$\frac{\partial u}{\partial t} = -\frac{\partial^2 u}{\partial x^2}, \quad x \in (0, 1), \quad t \in (0, T],$$

subject to zero Dirichlet boundary conditions. If

$$u(x, 0) = f(x) = \sum_{m=1}^{\infty} \alpha_m \sin \pi m x, \quad 0 \leq x \leq 1,$$

then

$$u(x, t) = \sum_{m=1}^{\infty} \alpha_m e^{\pi^2 m^2 t} \sin \pi m x, \quad 0 \leq x \leq 1, \quad t \geq 0.$$

Now suppose we have initial data such that

$$\alpha_m, \tilde{\alpha}_m = 0, \quad m > N$$

and

$$\tilde{\alpha}_m - \alpha_m = \epsilon \sqrt{\frac{2}{N}}$$

so that

$$\|\tilde{u}(x, 0) - u(x, 0)\|_2^2 = \frac{1}{2} \sum_{m=1}^N (\tilde{\alpha}_m - \alpha_m)^2 = \epsilon^2$$

then for $t > 0$

$$\|\tilde{u}(x, t) - u(x, t)\|_2^2 = \frac{1}{2} \sum_{m=1}^N (\tilde{\alpha}_m - \alpha_m)^2 e^{2\pi^2 m^2 t} = \frac{\epsilon^2}{N} \sum_{m=1}^N e^{2\pi^2 m^2 t} \rightarrow \infty,$$

This shows that for an arbitrarily small perturbation of the initial data ($\|\tilde{u}(x, t) - u(x, t)\|_2 \leq \epsilon$) and for any $t > 0$, the difference between the solutions u and \tilde{u} at time t can be made arbitrarily large by taking N large enough. This shows that reversed-time diffusion equation is ill-posed because the solution does not depend continuously on the initial data. In physics, the ill-posedness of the reversed-time diffusion equation is related to the impossibility of deducing the thermal history of an object from its present temperature distribution,

which is studied in thermodynamics.

1.0.6 Stability of numerical methods for PDEs

A method is **stable** if, as $h \rightarrow 0$ and with μ constant,

$$|u_j^i| < \infty, \quad j = 0, \dots, n_x + 1, \quad i = 0, \dots, n_t.$$

The "Fundamental theorem of numerical analysis of differential equations" relates consistency, stability and convergence.

Theorem (Lax equivalence theorem) If a PDE problem is well posed, then a numerical method is convergent if and only if it is stable and consistent.

The von Neumann method for stability analysis The von Neumann method for the stability analysis of numerical methods for PDEs is applicable to linear PDEs and in its simplest form, it ignores boundary conditions. The method is essentially a discretised Fourier analysis. The method proceeds as follows: given a numerical method for a PDE (e.g., a finite difference method), use the ansatz

$$u_j^i = \lambda^i e^{ikx_j},$$

where $k \in \mathbb{Z}$, $x_j = jh$, $j \in \mathbb{Z}$ and $h > 0$.

Note: For the symbol u_j^i , i is an index (a superscript), whereas λ^i means λ raised to the i -th power, where i is a non-negative integer.

Then for stability, we require

$$|\lambda| \leq 1, \quad k \in \mathbb{Z}, \quad h > 0$$

because

$$|u_j^i| = |\lambda^i e^{ikx_j}| = |\lambda^i| = |\lambda|^i.$$

We only need to consider the one wave $u_j^i = \lambda^i e^{ikx_j}$ because we assume the PDE and thus also the discretised equations are linear and hence the principle of superposition holds.

Example Here we illustrate the von Neumann method (or von Neumann stability analysis) for Euler's method:

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

Setting $u_j^i = \lambda^i e^{ikx_j}$, it follows that

$$\begin{aligned}
\lambda &= 1 + \mu \left(e^{ikh} - 2 + e^{-ikh} \right) \\
&= 1 + \mu \left(e^{ikh/2} - e^{-ikh/2} \right)^2 \\
&= 1 - 4\mu \sin^2(kh/2).
\end{aligned}$$

For stability, we require

$$|\lambda| \leq 1 \iff -1 \leq 1 - 4\mu \sin^2(kh/2) \leq 1, \quad h > 0, \quad k \in \mathbb{Z}.$$

Since $0 \leq \sin^2(kh/2) \leq 1$ and $\mu > 0$, we conclude that Euler's method is stable if and only if $\mu \leq \frac{1}{2}$.

Earlier we proved from first principles that if $\mu \leq \frac{1}{2}$, then Euler's method converges. An alternative proof relies on the Lax equivalence theorem: first we show that Euler's method is consistent (which we've done already), then we show (using the von Neumann method) that the method is stable iff $\mu \leq \frac{1}{2}$ and then we can conclude that the method is convergent if $\mu \leq \frac{1}{2}$.

$$\mathbf{u}^{i+1} = A\mathbf{u}^i + \mathbf{k}^i, \quad i = 0, \dots, n_t - 1.$$

where

$$\mathbf{u}^i = \begin{bmatrix} u_1^i \\ \vdots \\ u_{n_x}^i \end{bmatrix}, \quad A = \begin{bmatrix} 1 - 2\mu & \mu & & & \\ \mu & 1 - 2\mu & \mu & & \\ & \ddots & \ddots & \ddots & \\ & & \mu & 1 - 2\mu & \mu \\ & & & \mu & 1 - 2\mu \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}, \quad \mathbf{k}^i =$$

A large class of methods for the diffusion equation (and other linear PDEs) can be put into this form. Before we analyse the stability of these methods using matrix methods, we recall some facts from linear algebra.

Recall the definition of the Euclidean inner product: let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_{n_x} \end{bmatrix} \in \mathbb{R}^{n_x}$$

then

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}$$

If $\mathbf{x}, \mathbf{y} \in \mathbb{C}^{n_x}$, then

$$\langle \mathbf{x}, \mathbf{y} \rangle = \bar{\mathbf{x}}^\top \mathbf{y}.$$

This inner product induces the Euclidean or ℓ_2 norm:

$$\|\mathbf{x}\| = \left(|x_1|^2 + \cdots + |x_{n_x}|^2 \right)^{1/2} = (\langle \mathbf{x}, \mathbf{x} \rangle)^{1/2}.$$

We shall also use a scaled Euclidean inner product and ℓ_2 norm:

$$\langle \mathbf{x}, \mathbf{y} \rangle_h := h \langle \mathbf{x}, \mathbf{y} \rangle$$

and hence

$$\|\mathbf{x}\|_h = (h \langle \mathbf{x}, \mathbf{x} \rangle)^{1/2} = \sqrt{h} \|\mathbf{x}\|.$$

where $h = \frac{1}{n_x+1}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{C}^{n_x}$ or $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n_x}$. One reason for using this inner product is the following: suppose the entries of $\mathbf{f} \in \mathbb{R}^{n_x}$ are the function values $f(x_j)$, $x_j = jh$, $j = 1, \dots, n_x$, then from the Riemann sum

definition of the integral of a function, it follows that as $h \rightarrow 0$

$$\|\mathbf{f}\|_h = \left(h \sum_{j=1}^{n_x} [f(x_j)]^2 \right)^{1/2} \rightarrow \left(\int_0^1 [f(x)]^2 dx \right)^{1/2} = \|f\|_2.$$

i.e., the scaled ℓ_2 vector norm $\|\cdot\|_h$ tends to the L^2 function norm. There is another reason for using the scaled norm $\|\cdot\|_h$: suppose $\mathbf{x} \in \mathbb{R}^{n_x}$ is a vector such that each entry x_j satisfies $x_j = \mathcal{O}(h^p)$, $h \rightarrow 0$, i.e., for sufficiently small h (say $0 < h < h_0 \ll 1$), there's a constant $c > 0$ such that $|x_j| \leq ch^p$ for $j = 1, \dots, n_x$, then for sufficiently small h ,

$$\|\mathbf{x}\|_h = \left(h \sum_{j=1}^{n_x} x_j^2 \right)^{1/2} \leq \left(hn_x c^2 h^{2p} \right)^{1/2} \leq ch^p$$

i.e., $\|\mathbf{x}\|_h = \mathcal{O}(h^p)$, as $h \rightarrow 0$.

Let $A \in \mathbb{R}^{n_x \times n_x}$, then the matrix 2-norm or matrix Euclidean norm induced by the Euclidean or ℓ_2 vector norm is

$$\|A\| = \max_{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|.$$

Since

$$\frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} = \frac{\sqrt{h}\|A\mathbf{x}\|}{\sqrt{h}\|\mathbf{x}\|} = \frac{\|A\mathbf{x}\|_h}{\|\mathbf{x}\|_h},$$

it follows that

$$\|A\|_h = \|A\|.$$

It follows from the definition of the matrix norm that

$$\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|, \quad \|AB\| \leq \|A\|\|B\|$$

and the same is true in the norm $\|\cdot\|_h$.

The eigenvalues or *spectrum* of a square matrix $A \in \mathbb{C}^{n_x \times n_x}$ is

$$\sigma(A) = \{\lambda \in \mathbb{C} : \det(A - \lambda I) = 0\}.$$

The *spectral radius* of a square matrix $A \in \mathbb{C}^{n_x \times n_x}$ is

$$\rho(A) = \max \{ |\lambda| : \lambda \in \sigma(A) \}$$

A square matrix $A \in \mathbb{C}^{n_x \times n_x}$ is **normal** if $AA^* = A^*A$. Recall that A^* is the conjugate transpose of A , hence for $A \in \mathbb{R}^{n_x \times n_x}$, $A^* = A^\top$ and thus for a real matrix is normal if $AA^\top = A^\top A$.

If a matrix is normal, then

$$\|A\|_h = \|A\| = \rho(A).$$

A numerical method for the diffusion equation on $(x, t) \in [0, 1] \times [0, T]$ that takes the form

$$\mathbf{u}^{i+1} = A\mathbf{u}^i + \mathbf{k}^i, \quad i = 0, \dots, n_t - 1.$$

where $\mathbf{u}^i \in \mathbb{R}^{n_x}$ and $A \in \mathbb{R}^{n_x \times n_x}$ is **stable** if there exists a $0 < c < \infty$ such that

$$\lim_{h \rightarrow 0} \left(\max_{n=0,1,\dots,n_t} \|A^n\|_h \right) \leq c$$

where, as before, $n_t\tau = T$, $h = 1/(n_x + 1)$, $\mu = \tau/h^2$ and μ is held

constant.

Theorem (stability for methods with normal matrices) Suppose A is normal, where $A \in \mathbb{R}^{n_x \times n_x}$ or $A \in \mathbb{C}^{n_x \times n_x}$, then the method

$$\mathbf{u}^{i+1} = A\mathbf{u}^i + \mathbf{k}^i, \quad i = 0, \dots, n_t - 1.$$

is stable if there exists a $\nu \geq 0$ such that

$$\rho(A) \leq e^{\nu\tau}, \quad h \rightarrow 0,$$

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

Proof For simplicity, let us assume that $A \in \mathbb{R}^{n_x \times n_x}$. Using the definition of the scaled inner product $\langle \cdot, \cdot \rangle_h$, the normalcy of A and the Cauchy-Schwarz inequality, it follows that for a $\mathbf{w} \in \mathbb{R}^{n_x}$, $\mathbf{w} \neq \mathbf{0}$,

$$\begin{aligned}
\|A^n \mathbf{w}\|_h^2 &= \langle A^n \mathbf{w}, A^n \mathbf{w} \rangle_h \\
&= \langle \mathbf{w}, (A^n)^\top A^n \mathbf{w} \rangle_h \\
&= \langle \mathbf{w}, (A^\top A)^n \mathbf{w} \rangle_h \\
&\leq \|\mathbf{w}\|_h \| (A^\top A)^n \mathbf{w} \|_h \\
&\leq \|\mathbf{w}\|_h^2 \| (A^\top A)^n \|_h \\
&\leq \|\mathbf{w}\|_h^2 \|A\|_h^{2n}.
\end{aligned}$$

Therefore, since A is normal and $0 \leq n \leq n_t$,

$$\frac{\|A^n \mathbf{w}\|_h}{\|\mathbf{w}\|_h} \leq \|A\|_h^n = [\rho(A)]^n \leq e^{\nu n \tau} \leq e^{\nu n_t \tau} = e^{\nu T}.$$

Since

$$\|A^n\|_h = \max_{\mathbf{w} \in \mathbb{R}^{n_x}, \mathbf{w} \neq 0} \frac{\|A^n \mathbf{w}\|_h}{\|\mathbf{w}\|_h},$$

the result follows since we've shown that $\|A^n\|_h \leq c$ for $n = 0, \dots, n_t$ with $c = e^{\nu T}$ as $h \rightarrow 0$. ■

Example (Stability of Euler's method via matrix analysis) For Euler's method,

$$A = \begin{bmatrix} 1 - 2\mu & \mu & & & \\ \mu & 1 - 2\mu & \mu & & \\ & \ddots & \ddots & \ddots & \\ & & \mu & 1 - 2\mu & \mu \\ & & & \mu & 1 - 2\mu \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}.$$

Since A is symmetric, it is a normal matrix. However, A is not just symmetric, it is a TST matrix (tridiagonal, symmetric and Toeplitz) and the eigenvalues of TST matrices are known explicitly:

Lemma (eigenvalues of TST matrices) The eigenvalues of an $n_x \times n_x$ TST matrix with α on the main diagonal and β on the first super- and sub-diagonal are

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

Proof See *A First Course in the Numerical Analysis of Differential Equations* by A. Iserles, 2nd Edition, p. 264.

Returning to Euler's method, since $h = 1/(n_x + 1)$ and $x_j = jh$, the eigenvalues of the matrix A are

$$\lambda_j = 1 - 2\mu + 2\mu \cos(\pi x_j) = 1 - 2\mu + 2\mu(1 - 2\sin^2(\pi x_j/2)) = 1 - 4\mu \sin^2(\pi x_j/2)$$

for $j = 1, \dots, n_x$. If $0 < \mu \leq 1/2$, then $\rho(A) \leq 1$ as $h \rightarrow 0$ and hence the above theorem holds with $\nu = 0$ and we conclude that Euler's method is stable, whereas if $\mu > 1/2$, then $\rho(A) > 1$ as $h \rightarrow 0$ and there isn't a $\nu > 0$ such that the above theorem holds.

```
nx = 11
```

```
μ = 0.5
```

```
A = SymTridiagonal(fill((1 - 2μ), nx), fill(μ, nx-1))
```

```
h = 1/(nx+1)
```

```
x = h:h:1-h
```

```
[eigvals(A) 1 .- 4μ*sin.(π*x[end:-1:1]/2).^2]
```

11×2 Matrix{Float64}:

-0.965926	-0.965926
-0.866025	-0.866025
-0.707107	-0.707107

-0.5	-0.5
-0.258819	-0.258819
-2.68965e-16	2.22045e-16
0.258819	0.258819
0.5	0.5
0.707107	0.707107
0.866025	0.866025
0.965926	0.965926

1.1 Implicit Euler method

Recall that we arrived at Euler's method for the diffusion equation by combining a (first-order) forward difference approximation to the time derivative with a (second-order) central difference approximation to the second spatial derivative. Suppose now that we approximate the time derivative with a (first-order) *backward* difference approximation

$$u_t(x_j, t_i) \approx \frac{u_j^i - u_j^{i-1}}{\tau}$$

and again use a second-order central difference approximation to u_{xx} , then we arrive at the method

$$\frac{u_j^i - u_j^{i-1}}{\tau} - \frac{u_{j+1}^i - 2u_j^i + u_{j-1}^i}{h^2} = 0$$

or

$$u_j^i = u_j^{i+1} - \mu \left(u_{j+1}^{i+1} - 2u_j^{i+1} + u_{j-1}^{i+1} \right), \quad j = 1, \dots, n_x,$$

which is known is the backward Euler method, or the implicit Euler method. In linear algebra notation, the method becomes

$$B\mathbf{u}^{i+1} = \mathbf{u}^i + \mathbf{k}^{i+1}$$

where

$$\mathbf{u}^i = \begin{bmatrix} u_1^i \\ \vdots \\ u_{n_x}^i \end{bmatrix}, \quad B = \begin{bmatrix} 1+2\mu & -\mu & & & \\ -\mu & 1+2\mu & -\mu & & \\ & \ddots & \ddots & \ddots & \\ & & -\mu & 1+2\mu & -\mu \\ & & & -\mu & 1+2\mu \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}, \quad \mathbf{k}^i =$$

As with the (explicit) Euler method, the local truncation error is second order

$$\frac{\tilde{u}_j^i - \tilde{u}_j^{i-1}}{\tau} - \frac{\tilde{u}_{j+1}^i - 2\tilde{u}_j^i + \tilde{u}_{j-1}^i}{h^2} = \mathcal{O}(h^2), \quad h \rightarrow 0,$$

and therefore the method is consistent. There appears to prefer the implicit Euler method over the explicit Euler method because it has the same order and it is computationally more expensive. Now, let's consider the stability of the implicit Euler method.

1.1.1 Von Neumann stability analysis

Setting

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

in the implicit Euler method, we deduce that

$$\begin{aligned} 1 &= \lambda \left[1 - \mu \left(e^{ikh} - 2 + e^{-ikh} \right) \right] \\ &= \lambda \left[1 - \mu \left(e^{ikh/2} - e^{-ikh/2} \right)^2 \right] \\ &= \lambda \left[1 + 4\mu \sin^2(kh/2) \right]. \end{aligned}$$

For stability, we require

$$|\lambda| \leq 1 \iff 1 + 4\mu \sin^2(kh/2) \geq 1, \quad h > 0, \quad k \in \mathbb{Z}.$$

We conclude that the backward Euler method is stable for all $\mu > 0$. We say that the backward Euler method is unconditionally stable. Hence, compared to the explicit Euler method, we can take much larger step sizes while maintaining stability. However, larger step sizes lead to larger errors. The ideal method would be high order and unconditionally stable.

1.1.2 Stability analysis via matrix analysis

The implicit Euler method can be formulated as

$$\mathbf{u}^{i+1} = A\mathbf{u}^i + \mathbf{b}^i$$

where $A = B^{-1}$ and $\mathbf{b}^i = B^{-1}\mathbf{k}^{i+1}$.

The matrix A is normal because

$$AA^\top = B^{-1}B^{-\top} = (B^\top B)^{-1} = (BB^T)^{-1} = B^{-\top}B^{-1} = A^\top A.$$

Since B is a TST matrix, its eigenvalues are

$$\lambda_j = 1 + 2\mu - 2\mu \cos(\pi x_j) = 1 + 2\mu - 2\mu(1 - 2\sin^2(\pi x_j/2)) = 1 + 4\mu \sin^2(\pi x_j/2)$$

Since $\lambda \in \sigma(B)$ iff $\lambda^{-1} \in \sigma(B^{-1})$, we conclude that for any $\mu > 0$ and $h > 0$

$$\rho(A) = \max_{j=1,\dots,n_x} \frac{1}{1 + 4\mu \sin^2(\pi x_j/2)} < 1,$$

and therefore the implicit Euler method is stable for any $\mu > 0$.

`nx = 11`

`μ = 0.5`

```

B = SymTridiagonal(fill((1 + 2μ),nx),fill(-μ,nx-1))
h = 1/(nx+1)
x = h:h:1-h
norm(eigvals(B) - (1 .+ 4μ*sin.(π*x/2).^2),Inf)
1.3322676295501878e-15

```

1.1.3 Numerical experiment

```

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)
```

```
f = x -> sin(π*x/2) + 0.5*sin(2π*x)
```

```
ϕ1 = t -> exp(-π^2*t/4)
```

```
ϕ0 = t -> 0
```

```
nx = 50
```

```
μ = 0.50
```

```
T = 1
```

```
Euler(f,ϕ0,ϕ1,nx,μ,T)
@time Euler(f,ϕ0,ϕ1,nx,μ,T)
BackwardEuler(f,ϕ0,ϕ1,nx,μ,T)
u,x,t = @time BackwardEuler(f,ϕ0,ϕ1,nx,μ,T)
nt = length(t) -1

0.006578 seconds (10.42 k allocations: 7.067 MiB)
  0.012303 seconds (20.82 k allocations: 11.750 MiB)
5202

xx = x' .* ones(nt+1)
tt = ones(nx+2)' .* t
error = ue.(xx,tt) - u
e1 = maximum(error)

0.00091223286366926

μ = 0.50*nx
u,x,t = @time BackwardEuler(f,ϕ0,ϕ1,nx,μ,T)
nt = length(t) -1

0.000282 seconds (429 allocations: 242.906 KiB)
```

```

xx = x' .* ones(nt+1)
tt = ones(nx+2)' .* t
error = ue.(xx,tt) - u
e1 = maximum(error)

```

0.029907965385205237

1.2 Semi-discretisation / Method of lines

We have seen one approach to designing numerical methods for the diffusion equation: replacing derivatives with finite difference approximations. Another approach, as the name semi-discretisation implies, is to discretise only the spatial variable. For example, let

$$u(x_j, t) \approx v_j(t), \quad j = 0, \dots, n_x + 1, \quad x_j = jh, \quad h = \frac{1}{n_x + 1},$$

and suppose we approximate u_{xx} with a second-order central difference:

$$u_{xx}(x_j, t) \approx \frac{1}{h^2} (v_{j+1} - 2v_j + v_{j-1}).$$

The temporal derivative is approximated by setting $u_t(x_j, t) \approx v'_j(t)$, hence we approximate the solution to the diffusion equation by solving

$$v'_j(t) = \frac{1}{h^2} (v_{j+1} - 2v_j + v_{j-1}), \quad j = 1, \dots, n_x.$$

That is, we approximate the solution to the diffusion equation by the solution to a linear system of ODEs. This semi-discretisation approach is also known as the *method of lines* because we are approximating the solution by functions $v_j(t)$ on the lines $(x, t) = (x_j, t)$, $t \geq 0$, $j = 1, \dots, n_x$.

Setting $\tilde{v}_j(t) = u(x_j, t)$ and assuming $\tilde{v}_j(t) = v_j(t)$, it follows that if $u_{xxxx}(x, t)$ is bounded on $(x, t) \in [0, 1] \times [0, T]$, then

$$\tilde{v}'_j(t) = u_t(x_j, t) = \frac{1}{h^2} (\tilde{v}_{j+1} - 2\tilde{v}_j + \tilde{v}_{j-1}) = u_{xx}(x_j, t) + \mathcal{O}(h^2), \quad h \rightarrow 0$$

and therefore

$$\tilde{v}'_j(t) - \frac{1}{h^2} (\tilde{v}_{j+1} - 2\tilde{v}_j + \tilde{v}_{j-1}) = \mathcal{O}(h^2), \quad h \rightarrow 0,$$

hence the semi-discrete method is second order and consistent.

1.2.1 Von Neumann stability analysis of semi-discrete methods

The von Neumann method for the stability analysis of semi-discrete methods is completely analogous to the method for fully discretised methods: we use the ansatz

$$v_j(t) = \lambda(t)e^{ikx_j}$$

ignore boundary conditions and since $|v_j| = |\lambda(t)|$, for stability we require that

$$|\lambda(t)| \leq c < \infty, \quad t \in [0, T], \quad h > 0, \quad k \in \mathbb{Z}.$$

Example For the semi-discrete method above,

$$v'_j(t) = \frac{1}{h^2} (v_{j+1} - 2v_j + v_{j-1}),$$

if we set $v_j(t) = \lambda(t)e^{ikx_j}$, then we obtain

$$\lambda'(t) = \frac{1}{h^2} \left(e^{ikh} - 2 + (e^{-ikh}) \right) \lambda(t) = -\frac{4 \sin^2(kh/2)}{h^2} \lambda(t)$$

and therefore

$$\lambda(t) = e^{-4t \sin^2(kh/2)/h^2} \lambda(0)$$

and the method is stable because for all $h > 0$, $k \in \mathbb{Z}$,

$$|\lambda(t)| \leq |\lambda(0)| < \infty, \quad t \in [0, T].$$

By the Lax equivalence theorem, we conclude the method is convergent because we have showed that it is consistent and stable.

1.2.2 Stability analysis via matrix analysis for semi-discrete methods

All semi-discrete methods for the diffusion equation can be cast in the form

$$\mathbf{v}'(t) = A\mathbf{v}(t) + \mathbf{h}(t), \quad \mathbf{v}(0) = \mathbf{f},$$

where $\mathbf{v}(t), \mathbf{h}(t) \in \mathbb{R}^{n_x}$ and $A \in \mathbb{R}^{n_x \times n_x}$. For example, the method derived above,

$$v'_j(t) = \frac{1}{h^2} (v_{j+1} - 2v_j + v_{j-1}), \quad j = 1, \dots, n_x,$$

can be expressed as

$$\underbrace{\begin{bmatrix} v'_1 \\ \vdots \\ v'_{n_x} \end{bmatrix}}_{\mathbf{v}'} = \frac{1}{h^2} \underbrace{\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}}_A \underbrace{\begin{bmatrix} v_1 \\ \vdots \\ v_{n_x} \end{bmatrix}}_{\mathbf{v}} + \frac{1}{h^2} \underbrace{\begin{bmatrix} \varphi_0(t) \\ 0 \\ \vdots \\ 0 \\ \varphi_1(t) \end{bmatrix}}_{\mathbf{h}(t)}$$

subject to the initial condition

$$v_j(0) = u(x_j, 0) = f(x_j), \quad j = 1, \dots, n_x.$$

The exact solution to the coupled ODE system $\mathbf{v}'(t) = A\mathbf{v} + \mathbf{h}(t)$ is

$$\mathbf{v}(t) = e^{tA} \mathbf{v}(0) + \int_0^t e^{(t-\tau)A} \mathbf{h}(\tau) d\tau$$

where the matrix exponential for a square matrix $B \in \mathbb{R}^{n_x \times n_x}$ or $B \in \mathbb{C}^{n_x \times n_x}$ is defined as

$$e^B = \sum_{k=0}^{\infty} \frac{B^k}{k!}.$$

Therefore we have that a semi-discrete method for the diffusion equation is **stable** if there exists a constant $0 < c < \infty$ such that

$$\lim_{h \rightarrow 0} \left(\max_{t \in [0, T]} \|e^{tA}\|_h \right) \leq c.$$

Example Let's consider the method above, for which

$$A = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n_x \times n_x}.$$

Since A is a TST matrix, its eigenvalues are

$$\lambda_j = \frac{1}{h^2} [-2 + 2 \cos(\pi x_j)] = \frac{1}{h^2} \left[-2 + 2 \left(1 - 2 \sin^2(\pi x_j/2) \right) \right] = -\frac{4}{h^2} \sin^2(\pi x_j/2)$$

Let's check this:

```
nx = 11
h = 1/(nx+1)
A = SymTridiagonal(fill(-2,nx),fill(1,nx-1))/h^2
x = h:h:1-h
[eigvals(A) -4/h^2*sin.(π*x[end:-1:1]/2).^2]
```

11×2 Matrix{Float64}:

-566.187	-566.187
-537.415	-537.415

-491.647	-491.647
-432.0	-432.0
-362.54	-362.54
-288.0	-288.0
-213.46	-213.46
-144.0	-144.0
-84.3532	-84.3532
-38.5847	-38.5847
-9.81336	-9.81336

To make more sense of the meaning of the matrix exponential we'll need the following.

Lemma (eigenvectors of TST matrices) Let λ_j , $j = 1, \dots, n_x$ be the eigenvalues of an $n_x \times n_x$ TST matrix with α on the main diagonal and β on the first super- and sub-diagonal. The entries of the corresponding orthogonal eigenvector $\mathbf{q}_j \in \mathbb{R}^{n_x}$ are

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

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

Proof See *A First Course in the Numerical Analysis of Differential Equations* by A. Iserles, 2nd Edition, p. 264.

Hence, we have

$$A\mathbf{q}_j = \lambda_j \mathbf{q}_j, \quad j = 1, \dots, n_x.$$

Let

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

then

$$AQ = Q\Lambda$$

where

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

and since the vectors \mathbf{q}_j are orthogonal, the matrix Q is orthogonal, i.e., $Q^\top Q = I$. Therefore, the spectral factorisation (eigendecomposition) of A is

$$A = Q\Lambda Q^{-1} = Q\Lambda Q^\top.$$

```
Q = [sqrt(2/(nx+1))*sin(pi*j*l/(nx+1)) for l = 1:nx, j = 1:nx]
```

```
Q'*Q ≈ I
```

```
true
```

```
Λ = diagm(eigvals(A)[end:-1:1])
```

```
Q*Λ*Q' ≈ A
```

```
true
```

We have that

$$\begin{aligned}
e^{tA} &= \sum_{k=0}^{\infty} \frac{(tA)^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{t^k}{k!} Q \Lambda^k Q^\top \\
&= \sum_{k=0}^{\infty} Q \frac{t^k}{k!} \Lambda^k Q^\top \\
&= Q \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} \Lambda^k \right) Q^\top \\
&= Q \begin{bmatrix} \sum_{k=0}^{\infty} \frac{t^k \lambda_1^k}{k!} & & & \\ & \sum_{k=0}^{\infty} \frac{t^k \lambda_2^k}{k!} & & \\ & & \ddots & \\ & & & \sum_{k=0}^{\infty} \frac{t^k \lambda_{nx}^k}{k!} \end{bmatrix} Q^\top,
\end{aligned}$$

$$= Q \begin{bmatrix} e^{t\lambda_1} & & \\ & e^{t\lambda_2} & \\ & & \ddots \end{bmatrix} Q^\top$$

Notice that

$$\sigma(e^{tA}) = \left\{ e^{t\lambda_j} : \lambda_j \in \sigma(A) \right\}$$

The matrix e^{tA} is symmetric

$$(e^{tA})^\top = (Q e^{t\Lambda} Q^\top)^\top = Q e^{t\Lambda} Q^\top = e^{tA}$$

and therefore it's a normal matrix and thus

$$\|e^{tA}\|_h = \|e^{tA}\| = \rho(e^{tA}) = \max \left\{ |e^{t\lambda_j}| : \lambda_j \in \sigma(A) \right\}.$$

Since $\lambda_j < 0$ for $j = 1, \dots, n_x$, it follows that

$$\lim_{h \rightarrow 0} \left(\max_{t \in [0, T]} \|e^{tA}\|_h \right) \leq 1$$

and therefore we have proved that the method is stable.

In the above example, we have proven the stability of a method by finding an upper bound on the spectral radius of A , which implied an upper bound

on the norm of e^{tA} . This method of proving stability can be generalised to all semi-discrete methods with a normal matrix A because normal matrices are *unitarily diagonalisable*, meaning that if $A \in \mathbb{C}^{n_x \times n_x}$ is a normal matrix then,

$$A = U\Lambda U^*$$

where, $U, \Lambda \in \mathbb{C}^{n_x \times n_x}$, Λ is a diagonal matrix containing the eigenvalues of A and U is unitary, i.e., $UU^* = I$.

Theorem (stability of semidiscrete methods for the diffusion equation with normal matrices) The semi-discrete method for the diffusion equation given by

$$\mathbf{v}'(t) = A\mathbf{v}(t) + \mathbf{h}(t), \quad \mathbf{v}(0) = \mathbf{f},$$

where $\mathbf{v}(t), \mathbf{h}(t) \in \mathbb{R}^{n_x}$ and $A \in \mathbb{R}^{n_x \times n_x}$ is a normal matrix is stable if there exists an $\eta \in \mathbb{R}$ such that as $h \rightarrow 0$,

$$\operatorname{Re} \lambda \leq \eta \quad \text{for every } \lambda \in \sigma(A).$$

Proof See *A First Course in the Numerical Analysis of Differential Equations*

by A. Iserles, p. 370.

Remark In the example above, the conditions of this theorem hold for $\eta = 0$.

In the example above, we showed that A had the factorisation

$$A = Q\Lambda Q^\top.$$

This can be used to 'diagonalise' the ODE system

$$\mathbf{v}' = A\mathbf{v} + \mathbf{h}(t),$$

because if we set $\mathbf{w} = Q^\top \mathbf{v}$ and $\mathbf{b}(t) = Q^\top \mathbf{h}(t)$, then

$$\mathbf{w}' = \Lambda\mathbf{w} + \mathbf{b}(t),$$

i.e., if we let $w_j(t)$, $b_j(t)$ be the entries of \mathbf{w} and $\mathbf{b}(t)$ for $j = 1, \dots, n_x$, then

$$w'_j = \lambda_j w_j + b_j(t), \quad j = 1, \dots, n_x.$$

Hence, the coupled system of ODEs is reduced to n_x decoupled scalar ODEs which we know how to solve exactly:

$$w_j(t) = e^{t\lambda_j} w_j(0) + \int_0^t e^{(t-\tau)\lambda_j} b_j(\tau) d\tau.$$

1.2.3 Time discretisation of semi-discrete methods

In general, we won't know the exact solution of the semi-discretised system of equations, so we approximate the solution by discretising time. For example, consider again the semi-discretised method

$$v'_j(t) = \frac{1}{h^2} (v_{j+1} - 2v_j + v_{j-1}).$$

If we approximate the derivative with a (first-order) forward difference by letting $v_j(t_i) \approx u_j^i$, $t_i = i\tau$, $\tau = t/n_t$ and

$$v'_j(t_i) \approx \frac{v_j(t_{i+1}) - v_j(t_i)}{\tau} \approx \frac{u_j^{i+1} - u_j^i}{\tau}, \quad i = 0, \dots, n_t,$$

then we obtain the fully discretised method

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

which is just the (explicit) Euler method. Similarly, if we approximate the time derivative with a first order backward difference, then we obtain the backward, or implicit Euler method.

Alternatively, we can solve the semidiscretised system (the system of ODEs) using a 'black box' ODE integrator / solver.

```
nx = 50
h = 1/(nx+1)
f = x -> sin(π*x/2) + 0.5*sin(2π*x)
ϕ1 = t -> exp(-π^2*t/4)
ϕ0 = t -> 0
T = 1
A = SymTridiagonal(fill(-2,nx),fill(1,nx-1))/h^2
F = (v,p,t) -> A*v + [ϕ0(t);zeros(nx-2);ϕ1(t)]/h^2
x = range(0,1,nx + 2)
prob = ODEProblem(F, f.(x[2:end-1]), (0.0, T));
soln = solve(prob, RK4(), abstol=1e-4);
tv = soln.t
nt = length(tv)
```

3728

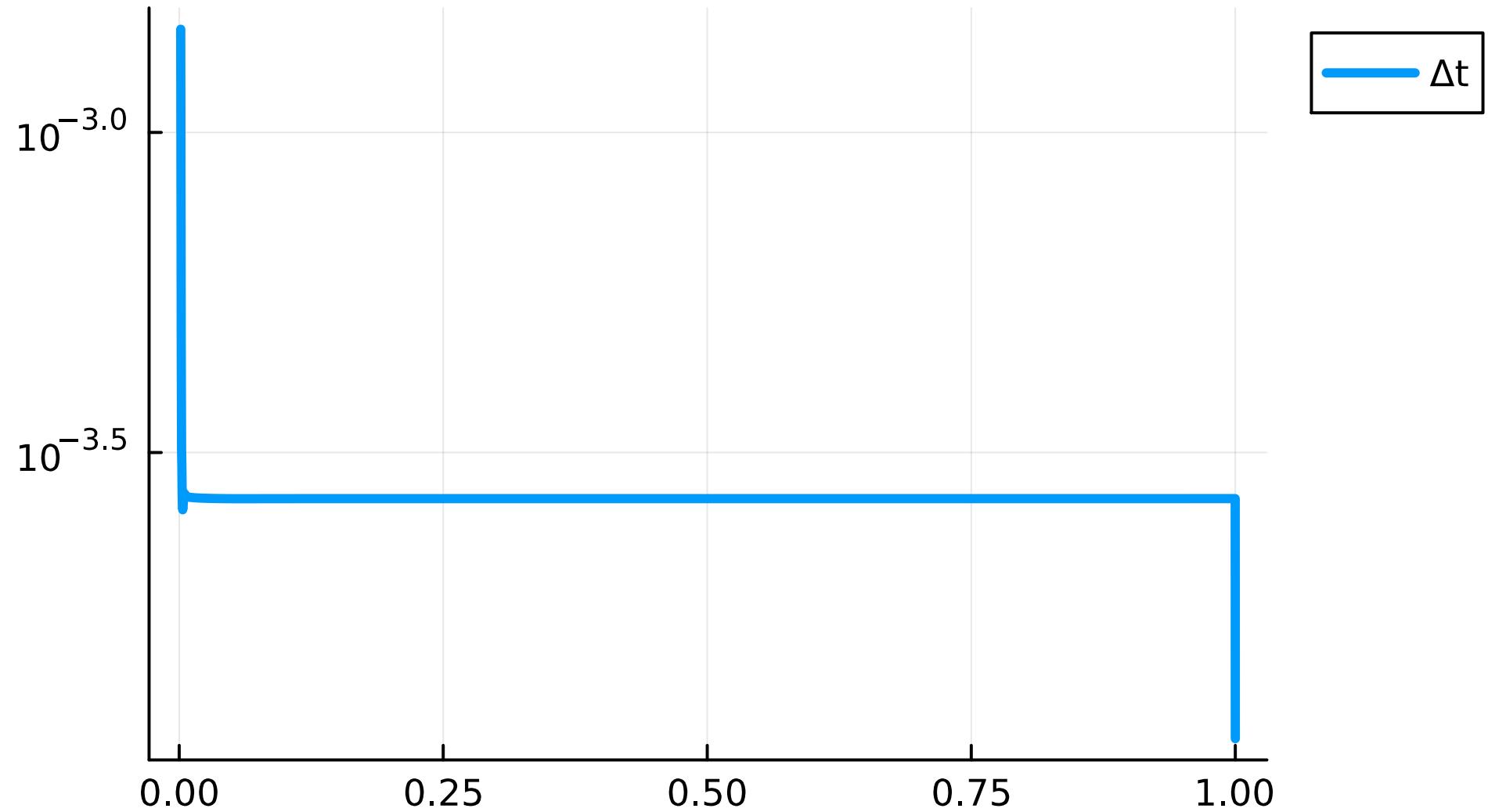
```
errs = zeros(nt-1)
for i = 2:nt
    errs[i-1] = norm(ue.(x[2:nx+1],tv[i]) -
soln.u[i],Inf)
end
norm(errs,Inf)
```

0.0002743087401239075

Let's analyze the numerical solution a bit...

```
dt = diff(soln.t)
plt1 = plot(soln.t[2:end], dt,yscale = :log10, size =
(500, 300), lw = 3, label = " $\Delta t$ ",
legend = :outertright, title = "# steps =
$(length(soln.t))")
```

steps = 3728



```
soln = solve(prob, Rodas4(), abstol=1e-4);
tv = soln.t
nt = length(tv)
```

22

```
errs = zeros(nt-1)
for i = 2:nt
    errs[i-1] = norm(ue.(x[2:nx+1],tv[i]) -
soln.u[i],Inf)
end
norm(errs,Inf)
```

0.00023693854401607428

Let's analyze the numerical solution a bit...

```
dt = diff(soln.t)
plt1 = plot(soln.t[2:end], dt,yscale = :log10, size =
(500, 300), lw = 3, label = " $\Delta t$ ",
legend = :outertright, title = "# steps =
$(length(soln.t))")
```

steps = 22

