Initial Value Problems: ODEs

Simple Numerical Methods

M.Sc. in Mathematical Modelling & Scientific Computing
Practical Numerical Analysis

Michaelmas Term 2026, Lecture 4

The Problem

We wish to solve the first order initial value problem: find $\mathbf{u}(t)$ such that

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{u}),$$

for $t > 0$ with $\mathbf{u}(0) = \mathbf{u}_0$.

The Problem

The study of vector-valued first order problems also allows us to solve (scalar) higher order problems. For example, suppose we wish to solve

$$u^{(n)}(t) = f\left(t, u(t), u'(t), u''(t), \ldots, u^{(n-1)}(t)\right),$$

for $t > 0$ with $u(0)$, $u'(0)$, $u''(0)$, ..., $u^{(n-1)}(0)$ all given.

We can then set $u_1(t) = u(t)$, and $u_k(t) = u^{(k-1)}(t)$ for $k = 2, \ldots, n$. This gives $u_k'(t) = u_{k+1}$ so that we have a system:

$$\begin{aligned} u_1' &= u_2(t) \\ u_2' &= u_3(t) \\ &\vdots \\ u_{n-1}' &= u_n(t) \\ u_n' &= f(t, u_1(t), u_2(t), \ldots, u_{n-1}(t)) \end{aligned}$$

with $u_1(0), u_2(0), \ldots, u_n(0)$ all given.

Scalar Problem

We shall write everything in terms of the scalar problem: find $u(t)$ such that

$$\frac{\mathrm{d}u}{\mathrm{d}t} = f(t, u),$$

for $t > 0$ with $u(0) = u_0$, but all methods are easily generalised to the case where the solution is a vector.

Existence and Uniqueness of Solution to Scalar Problem

Theorem: Picard

Suppose that $f(t,u)$ *is a continuous function of* t *and* u *in a region* $\Omega = [0,T) \times [u_0 - \alpha, u_0 + \alpha]$ *of the* (t,u) *plane and that there exists* $L > 0$ *such that*

$$|f(t,u) - f(t,v)| \leq L|u - v|, \quad \forall (t,u),\ (t,v) \in \Omega.$$

L is called a Lipschitz constant and this a Lipschitz condition. Suppose also that

$$MT \leq \alpha,$$

where $M = \max_\Omega |f|$. *Then there exists a unique continuously differentiable function* $u(t)$ *defined on* $[0,T)$ *satisfying*

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= f(t,u), \quad 0 < t < T, \\ u(0) &= u_0. \end{aligned}$$

Numerical Methods

Suppose we want to solve

$$\begin{aligned} u'(t) &= f(t,u), \quad t>0, \\ u(0) &= u_0. \end{aligned} \tag{1}$$

In order to solve (1) numerically over the time interval $[0,T]$, we define a set of time points at which we wish to approximate the solution. We set $t_n = n\Delta t$ for $n = 0,1,\ldots,N$ where $\Delta t = T/N$.

Then we can integrate (1) to get

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} f(t,u(t))\mathrm{d}t. \tag{2}$$

Using different approximations to the integral in (2) leads to different numerical schemes.

Simplest Methods — Euler Methods

Perhaps the simplest numerical methods are the explicit and implicit Euler methods (also known as forward and backward Euler).

Here we let U_n be the numerical approximation to $u(t_n)$.

For explicit (or forward) Euler we use

$$\int_{t_n}^{t_{n+1}} f(t, u(t))\mathrm{d}t \quad \approx \quad \Delta t f(t_n, u(t_n)).$$

(Recall $t_{n+1} - t_n = \Delta t$.)

This gives the numerical scheme

$$U_{n+1} \quad = \quad U_n + \Delta t f(t_n, U_n),$$

or equivalently

$$\frac{U_{n+1} - U_n}{\Delta t} \quad = \quad f(t_n, U_n)$$

for $n = 0, 1, \ldots, N-1$ and with $U_0 = u_0$.

Simplest Methods — Euler Methods

For implicit Euler we use

$$\int_{t_n}^{t_{n+1}} f(t, u(t))\mathrm{d}t \quad \approx \quad \Delta t f(t_{n+1}, u(t_{n+1})).$$

This gives the numerical scheme

$$U_{n+1} \quad = \quad U_n + \Delta t f(t_{n+1}, U_{n+1}),$$

or equivalently

$$\frac{U_{n+1} - U_n}{\Delta t} \quad = \quad f(t_{n+1}, U_{n+1})$$

for $n = 0, 1, \ldots, N-1$ and with $U_0 = u_0$.

Simplest Methods — Euler Methods

Explicit Euler is particularly simple. Given $U_0 = u_0$ and the function f we compute

$$U_{n+1} \quad = \quad U_n + \Delta t f(t_n, U_n)$$

for $n = 0, 1, \ldots$.

Implicit Euler is more complex in the sense that, if we are given $U_0 = u_0$ and the function f, we compute U_{n+1} as the solution to the *nonlinear* equation

$$U_{n+1} \quad = \quad U_n + \Delta t f(t_{n+1}, U_{n+1})$$

for $n = 0, 1, \ldots$. The solution to this nonlinear equation can be computed by (say) Newton's method. At timestep $n+1$, a good starting guess for Newton's method is U_n.

Trapezium Rule/Crank Nicolson Scheme

Another option is to use the trapezium rule to approximate the integral via

$$\int_{t_n}^{t_{n+1}} f(t, u(t))\mathrm{d}t \approx \frac{\Delta t}{2}\left(f(t_n, u(t_n)) + f(t_{n+1}, u(t_{n+1}))\right),$$

This gives rise to a numerical scheme known as the trapezium rule or the Crank Nicolson scheme

$$U_{n+1} = U_n + \frac{\Delta t}{2}\left(f(t_n, U_n) + f(t_{n+1}, U_{n+1})\right),$$

or equivalently

$$\frac{U_{n+1} - U_n}{\Delta t} = \frac{1}{2}\left(f(t_n, U_n) + f(t_{n+1}, U_{n+1})\right)$$

for $n = 0, 1, \ldots, N-1$ and with $U_0 = u_0$.

Generalisation — θ-Methods

Both the explicit and implicit Euler methods, as well as the Crank Nicolson method, are specific cases of the θ-method which is given by

$$\frac{U_{n+1} - U_n}{\Delta t} = \theta f(t_{n+1}, U_{n+1}) + (1-\theta) f(t_n, U_n) \tag{3}$$

for $n = 0, 1, \ldots$ and with $U_0 = u_0$. Special cases are

- $\theta = 0$ — explicit Euler
- $\theta = 1$ — implicit Euler
- $\theta = 1/2$ — Crank Nicolson method

For all non-zero values of θ, the method is implicit and a nonlinear equation must be solved at each time-step.

Example 1

Consider the problem

$$\begin{aligned} u'(t) &= \lambda u, \quad t > 0, \\ u(0) &= 1. \end{aligned}$$

The numerical schemes for this are:

- Explicit Euler: $U_{n+1} = U_n + \lambda \Delta t U_n = (1 + \lambda \Delta t) U_n$.
- Implicit Euler: $U_{n+1} = U_n + \lambda \Delta t U_{n+1}$, or equivalently

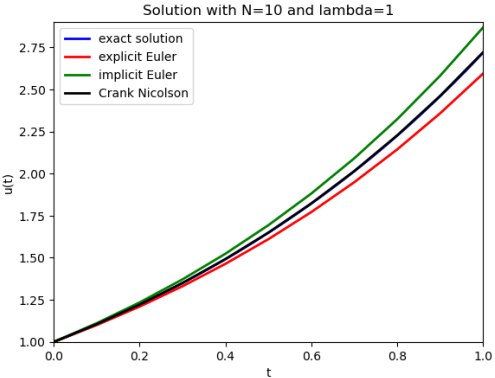
$$U_{n+1} = \frac{U_n}{1 - \lambda \Delta t}.$$

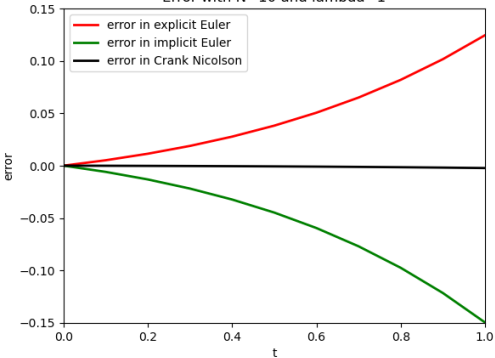
- θ-method: $U_{n+1} = U_n + \lambda \Delta t((1-\theta)U_n + \theta U_{n+1})$, or equivalently

$$U_{n+1} = \frac{1 + (1-\theta)\lambda \Delta t}{1 - \theta \lambda \Delta t} U_n.$$

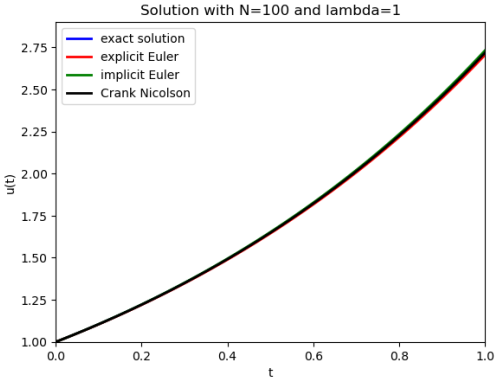
All are for $n = 0, 1, \ldots, N-1$ and with $U_0 = 1$.

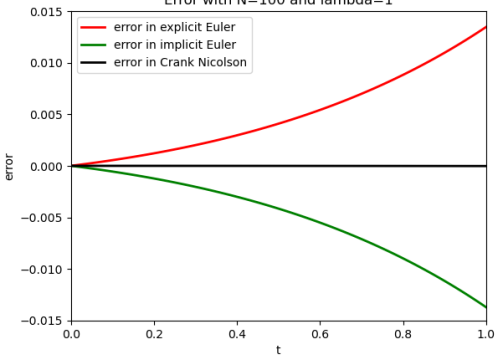
Example 1 Results



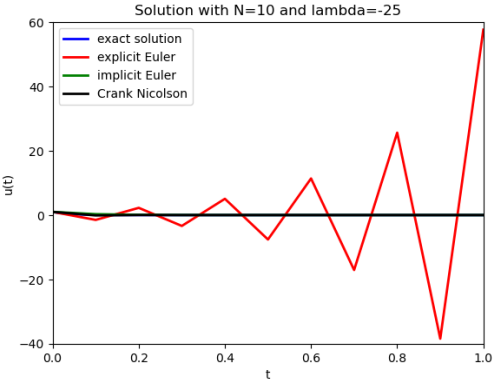


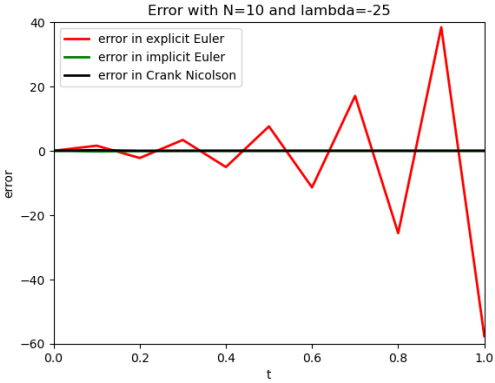
Example 1 Results





Example 1 Results





Example 2

Consider the system

$$\begin{aligned} u'(t) &= -v, \quad u(0) = 1 \\ v'(t) &= u, \quad v(0) = 0 \end{aligned}$$

which has exact solution $u(t) = \cos t$ and $v(t) = \sin t$. The system also has a *conserved* quantity $u^2 + v^2 = 1$.

Let (U_n, V_n) denote the approximation to $(u(t_n), v(t_n))$, then the θ-method is

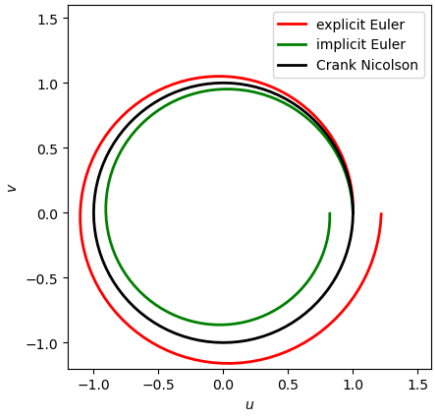
$$\begin{aligned} \frac{U_{n+1} - U_n}{\Delta t} &= -\theta V_{n+1} - (1-\theta) V_n, \\ \frac{V_{n+1} - V_n}{\Delta t} &= \theta U_{n+1} + (1-\theta) U_n, \end{aligned}$$

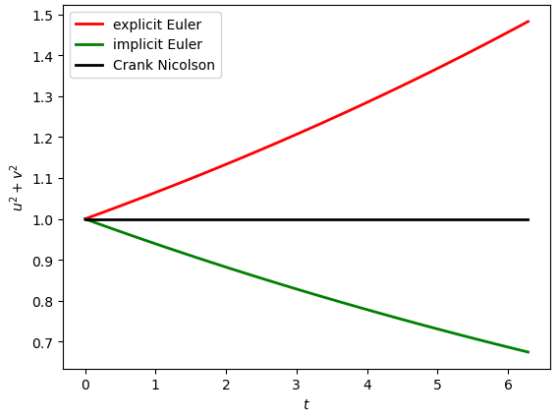
or equivalently, on re-arranging

$$\begin{pmatrix} 1 & \theta \Delta t \\ -\theta \Delta t & 1 \end{pmatrix} \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & -(1-\theta)\Delta t \\ (1-\theta)\Delta t & 1 \end{pmatrix} \begin{pmatrix} U_n \\ V_n \end{pmatrix}$$

for $n = 0, 1, \ldots, N-1$ and with $U_0 = 1$ and $V_0 = 0$.

Example 2 Results





Example 2 Explanation

To explain the results, note that it can be shown that

$$U_{n+1}^2 + V_{n+1}^2 = \left(1 + \frac{(1-2\theta)\Delta t^2}{(1+\theta^2\Delta t^2)}\right)(U_n^2 + V_n^2)$$

Thus if $U_0 = 1$ and $V_0 = 0$ we have

$$U_n^2 + V_n^2 = \left(1 + \frac{(1-2\theta)\Delta t^2}{(1+\theta^2\Delta t^2)}\right)^n$$

and so

- $U_n^2 + V_n^2 > 1$ for $\theta < 1/2$,
- $U_n^2 + V_n^2 = 1$ for $\theta = 1/2$,
- $U_n^2 + V_n^2 < 1$ for $\theta > 1/2$.

(*Symplectic integrators* preserve conserved quantities for Hamiltonian systems.)

Euler Derivations Using Taylor Series

The explicit and implicit Euler schemes can also be motivated using Taylor series expansions. Consider expanding $u(t_{n+1})$ about the point t_n. We have

$$u(t_{n+1}) = u(t_n) + \Delta t u'(t_n) + \mathcal{O}(\Delta t^2).$$

We can rearrange this to get

$$u'(t_n) = \frac{u(t_{n+1}) - u(t_n)}{\Delta t} + \mathcal{O}(\Delta t).$$

Substituting this expression into the differential Equation (1) evaluated at t_n, namely

$$u'(t_n) = f(t_n, u(t_n)),$$

gives

$$\frac{u(t_{n+1}) - u(t_n)}{\Delta t} + \mathcal{O}(\Delta t) = f(t_n, u(t_n)).$$

Euler Derivations Using Taylor Series

If we approximate $u(t_n)$ by U_n and ignore the $\mathcal{O}(\Delta t)$ term in

$$\frac{u(t_{n+1}) - u(t_n)}{\Delta t} + \mathcal{O}(\Delta t) \quad = \quad f(t_n, u(t_n)),$$

we recover the explicit Euler scheme

$$\frac{U_{n+1} - U_n}{\Delta t} \quad = \quad f(t_n, U_n).$$

Similarly, if we expand $u(t_n)$ about t_{n+1}, rearrange, substitute into Equation (1) evaluated at t_{n+1}, and ignore the $\mathcal{O}(\Delta t)$ term, we can recover the implicit Euler scheme.

Truncation Error

As we have just seen, the Euler methods can be derived by truncating Taylor series and the truncation error measures the error committed by doing this. The truncation error for the θ-method is defined as

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - \theta f(t_{n+1}, u_{n+1}) - (1-\theta) f(t_n, u_n), \tag{4}$$

where $u_n = u(t_n)$ is the exact solution at the point t_n. The truncation error can be computed using Taylor series expansions about an appropriately chosen time point.

For $\theta = 0$ (i.e. explicit Euler), the expansions are usually performed about $t = t_n$, while for $\theta = 1$ (i.e. implicit Euler), the expansions are usually performed about $t = t_{n+1}$. For general values of θ it is standard to expand about $t_{n+1/2} = (t_n + t_{n+1})/2 = t_n + \Delta t/2$.

Truncation Error — Explicit Euler Scheme

For the explicit Euler scheme we thus have

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - f(t_n, u_n). \tag{5}$$

We have

$$\begin{aligned} u_{n+1} = u(t_{n+1}) &= u(t_n + \Delta t) \\ &= u(t_n) + \Delta t u'(t_n) + \frac{1}{2}\Delta t^2 u''(\tau_n), \end{aligned} \tag{6}$$

for some $\tau_n \in [t_n, t_{n+1}]$.

Truncation Error — Explicit Euler Scheme

Substituting (6) in (5) gives

$$\begin{aligned} T_n &= \frac{u(t_n) + \Delta t u'(t_n) + \frac{1}{2}\Delta t^2 u''(\tau_n) - u(t_n)}{\Delta t} - f(t_n, u(t_n)) \\ &= u'(t_n) - f(t_n, u(t_n)) + \frac{1}{2}\Delta t u''(\tau_n). \end{aligned}$$

Finally we recall the original ODE was $u'(t) = f(t, u(t))$ so the $\mathcal{O}(1)$ terms cancel and we are left with

$$T_n = \frac{1}{2}\Delta t u''(\tau_n),$$

as the truncation error for the explicit Euler scheme.

Truncation Error — θ-Method

Note that since $u'(t_n) = f(t_n, u(t_n))$, we may re-write the expression for the truncation error

$$\begin{aligned} T_n &= \frac{u_{n+1} - u_n}{\Delta t} - \theta f(t_{n+1}, u_{n+1}) - (1-\theta) f(t_n, u_n) \\ &= \frac{u_{n+1} - u_n}{\Delta t} - \theta u'(t_{n+1}) - (1-\theta) u'(t_n). \end{aligned} \tag{7}$$

We have

$$\begin{aligned} u(t_n) &= u(t_{n+1/2} - \Delta t/2) \\ &= u(t_{n+1/2}) - \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2}\left(\frac{\Delta t}{2}\right)^2 u''(t_{n+1/2}) + \mathcal{O}(\Delta t^3). \end{aligned}$$

Similarly,

$$u(t_{n+1}) = u(t_{n+1/2}) + \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2}\left(\frac{\Delta t}{2}\right)^2 u''(t_{n+1/2}) + \mathcal{O}(\Delta t^3).$$

Truncation Error — θ-Method

We can also expand the first derivatives in Equation (7):

$$\begin{aligned} u'(t_n) &= u'(t_{n+1/2}) - \frac{\Delta t}{2} u''(t_{n+1/2}) + \mathcal{O}(\Delta t^2), \\ u'(t_{n+1}) &= u'(t_{n+1/2}) + \frac{\Delta t}{2} u''(t_{n+1/2}) + \mathcal{O}(\Delta t^2). \end{aligned}$$

Substituting these four expansions into (7) gives

$$\begin{aligned} T_n = \frac{1}{\Delta t} \Bigg\{ & \left(u(t_{n+1/2}) + \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 u''(t_{n+1/2}) \right) \\ & - \left(u(t_{n+1/2}) - \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 u''(t_{n+1/2}) \right) \Bigg\} \\ & - \theta \left(u'(t_{n+1/2}) + \frac{\Delta t}{2} u''(t_{n+1/2}) \right) \\ & - (1-\theta) \left(u'(t_{n+1/2}) - \frac{\Delta t}{2} u''(t_{n+1/2}) \right) + \mathcal{O}(\Delta t^2). \end{aligned} \tag{8}$$

Truncation Error — θ-Method

Many of the terms in (8) cancel so the truncation error simplifies to

$$T_n = \frac{\Delta t}{2}(1-2\theta)u''(t_{n+1/2}) + \mathcal{O}(\Delta t^2).$$

It can be shown by writing out the $\mathcal{O}(\Delta t^2)$ terms in full, that they do not cancel for any value of θ.

Thus we have shown that for constant θ

$$T_n = \begin{cases} \mathcal{O}(\Delta t) & \text{for } \theta \neq 1/2 \\ \mathcal{O}(\Delta t^2) & \text{for } \theta = 1/2 \end{cases}$$

so that the truncation error of the Crank Nicolson scheme converges twice as fast as that of all other θ-methods.

Truncation Error — θ-Method

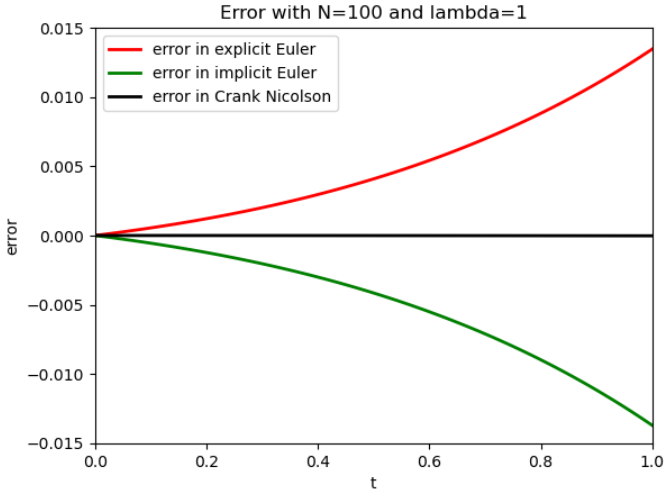
In fact, we can be more precise using the approach we used for the truncation error of the explicit Euler scheme.

We can show that

$$T_n = \begin{cases} \frac{\Delta t}{2} u''(\tau_n^{(1)}) & \text{for } \theta = 0 \\ -\frac{\Delta t^2}{12} u'''(\tau_n^{(2)}) & \text{for } \theta = 1/2 \\ -\frac{\Delta t}{2} u''(\tau_n^{(3)}) & \text{for } \theta = 1 \end{cases}$$

where $\tau_n^{(i)} \in [t_n, t_{n+1}]$ for $i = 1, 2, 3$.

Example 1 Revisited



Order of a Method

The order of a method is defined to be p where p is the largest integer such that $T_n = \mathcal{O}(\Delta t^p)$. Alternatively we may call the method pth order.

We have

- If $\theta \neq 1/2$, the θ-method is 1st order.
- If $\theta = 1/2$, the θ-method is 2nd order.

Pointwise Errors

Recall the definition of the θ-method (3) and the corresponding truncation error (4):

$$\begin{aligned} \frac{U_{n+1} - U_n}{\Delta t} &= \theta f(t_{n+1}, U_{n+1}) + (1-\theta) f(t_n, U_n), \\ T_n &= \frac{u_{n+1} - u_n}{\Delta t} - \theta f(t_{n+1}, u_{n+1}) - (1-\theta) f(t_n, u_n). \end{aligned}$$

We re-arrange both of these to get

$$U_{n+1} = U_n + \Delta t \left(\theta f(t_{n+1}, U_{n+1}) + (1-\theta) f(t_n, U_n)\right) \tag{9}$$

and

$$u_{n+1} = u_n + \Delta t \left(\theta f(t_{n+1}, u_{n+1}) + (1-\theta) f(t_n, u_n)\right) + \Delta t T_n. \tag{10}$$

Now consider subtracting (9) from (10), taking the modulus, and applying the triangle inequality. This gives

$$\begin{aligned} |u_{n+1} - U_{n+1}| \leq\ & |u_n - U_n| + \theta \Delta t |f(t_{n+1}, u_{n+1}) - f(t_{n+1}, U_{n+1})| \\ & + (1-\theta)\Delta t |f(t_n, u_n) - f(t_n, U_n)| + \Delta t |T_n|. \end{aligned} \tag{11}$$

Pointwise Errors

Next suppose that the right-hand-side function $f(t,u)$ satisfies a Lipschitz condition in its second argument, with Lipschitz constant L, so that

$$|f(t,u) - f(t,v)| \leq L|u-v|, \quad \forall (t,u),\ (t,v) \in \Omega.$$

We can use this in (11) to get

$$\begin{aligned} |u_{n+1} - U_{n+1}| &\leq |u_n - U_n| + \theta \Delta t L |u_{n+1} - U_{n+1}| \\ &\quad + (1-\theta)\Delta t L |u_n - U_n| + \Delta t |T_n|. \end{aligned}$$

We can re-arrange this to get (for Δt sufficiently small)

$$\begin{aligned} (1 - L\theta\Delta t)|u_{n+1} - U_{n+1}| &\leq (1 + L(1-\theta)\Delta t)|u_n - U_n| + \Delta t |T_n| \\ &\leq (1 + L(1-\theta)\Delta t)|u_n - U_n| + \Delta t T_{\max}, \end{aligned} \tag{12}$$

where $T_{\max} = \max_{0 \leq n \leq N} |T_n|$ is an upper bound on the absolute value of the truncation error.

Pointwise Errors

Now let $e_n = u_n - U_n$ denote the error at time t_n. Then (12) can be written as

$$|e_{n+1}| \leq \frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}|e_n| + \frac{\Delta t T_{\max}}{1 - L\theta\Delta t}. \tag{13}$$

We can show by induction that

$$\begin{aligned} |e_n| &\leq \left(\frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}\right)^n |e_0| + \frac{\Delta t T_{\max}}{1 - L\theta\Delta t} \sum_{r=1}^{n} \left(\frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}\right)^{r-1} \\ &\leq \left(\frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}\right)^n |e_0| + \frac{T_{\max}}{L}\left[\left(\frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}\right)^n - 1\right], \end{aligned}$$

where the final line comes from evaluating the sum and simplifying. This holds for $n = 0, 1, \ldots, N$.

Pointwise Errors

In practice, we usually set $U_0 = u_0$ which means that $e_0 = 0$.

We also have

$$\begin{aligned} \frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t} &= 1 + \frac{L\Delta t}{1 - L\theta\Delta t} \\ &\leq \exp\left(\frac{L\Delta t}{1 - L\theta\Delta t}\right). \end{aligned}$$

In turn this means

$$\begin{aligned} \left(\frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}\right)^n &\leq \left(\exp\left(\frac{L\Delta t}{1 - L\theta\Delta t}\right)\right)^n \\ &\leq \exp\left(\frac{nL\Delta t}{1 - L\theta\Delta t}\right) \\ &\leq \exp\left(\frac{LT}{1 - L\theta\Delta t}\right). \end{aligned}$$

Pointwise Errors

Thus we have

$$|e_n| \leq \frac{T_{\max}}{L}\left[\exp\left(\frac{LT}{1-L\theta\Delta t}\right)-1\right], \tag{14}$$

for $n = 0, 1, \ldots, N$.

This shows that the pointwise error has the same order as the truncation error.

Summary

- For the initial value problem

$$\begin{aligned} u'(t) &= f(t,u), \quad t \in (0,T], \\ u(0) &= u_0, \end{aligned}$$

the θ-method approximates the solution $u(t)$ at the discrete points $t_n = n\Delta t$, $n = 0, 1, \ldots, N$. Specifically the method approximates $u(t_n)$ by U_n which solves

$$\frac{U_{n+1} - U_n}{\Delta t} = \theta f(t_{n+1}, U_{n+1}) + (1-\theta) f(t_n, U_n)$$

for $n = 0, 1, \ldots, N-1$, with $U_0 = u_0$.

- If $\theta = 0$, the method is explicit.
- If $\theta > 0$, the method is implicit and a nonlinear system must be solved at each timestep.
- If $\theta = 1/2$ the method is second order accurate, otherwise the error is first order accurate.