

Using the θ -method to solve ODEs

Kathryn Gillow

26th October 2022

1 Introduction

- ① In this report we use the θ -method to solve ODE's. We begin by introducing the
② method and deriving it's truncation error. We use this to derive an expression for the
local error. We then show an example to confirm that the correct rates of convergence
③ are achieve.

In what follows we consider initial value problems of the form

④
$$\frac{du}{dt} = f(t, u) \quad (1)$$

- ⑤ for $t > 0$ with an initial condition $u(0) = u_0$. Here, we assume that $f(t, u)$ satisfies a
⑥ Lipschitz condition in its second argument and that $f(u, t)$ is bounded.
⑦ It is also possible to use the θ -method to solve problems with spacial dependence.

For example we could consider the heat equation of the form

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

- ⑧ for $(x, t) \in (-1, 1) \times (0, T]$ with boundary and initial conditions

⑨
$$\begin{aligned} u(x, 0) &= u_0(x) \quad \text{for } -1 < x < 1, \\ u(-1, t) &= g_1(t) \quad \text{for } t > 0, \\ u(1, t) &= g_2(t) \quad \text{for } t > 0. \end{aligned}$$

- ⑩ However, we don't consider such problems here. Instead, we refer the interested reader to Ref. [1].

11 2 The θ -method

12 For a general introduction to the θ method see Ref. [2]. We summarise the key points here.

13 In the θ -method we approximate the solution to Equation (1) at a set of discrete time points $t_n = n\Delta t$ for $n = 0, \dots, N$ where $N \geq 2$ and $N\Delta t = T$, where T is the final time. We let U_n be the numerical approximation to $u(t_n)$.

14 The θ -method for Equation 1 is

$$15 \quad \frac{U_{n+1} - U_n}{\Delta t} = \theta f(t_{n+1}, U_{n+1}) + (1 - \theta)f(t_n, U_n)$$

16 17 Where θ is between 0 and 1. We require this equation to hold for $n = 0, \dots, N - 1$

18 and we apply the initial condition via $U_0 = u_0$. 3 values of θ lead to methods with a specific name:

19 20 • $\theta = 0$ is the explicit Euler scheme(also known as "forward Euler")

21 • $\theta = 1$ is the implicit Euler scheme(also known as "backward Euler")

22 • $\theta = \frac{1}{2}$ is the Crank Nicolson scheme.

23 2.0.1 Truncation Error

The truncation error for the θ -method is defined as

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - \theta f(t_{n+1}, u_{n+1}) - (1 - \theta)f(t_n, u_n), \quad (2)$$

24 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 $t_{n+1/2} = (t_n +$

25 26 $t_{n+1})/2 = t_n + 1/2\Delta t$.

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

$$\begin{aligned} T_n &= \frac{u_{n+1} - u_n}{\Delta t} - \theta f(t_{n+1}, u_{n+1}) - (1 - \theta)f(t_n, u_n) \\ &= \frac{u_{n+1} - u_n}{\Delta t} - \theta u'(t_{n+1}) - (1 - \theta)u'(t_n). \end{aligned} \quad (3)$$

We have

$$\begin{aligned} (27) \quad u(t_n) &= u(t_{n+1/2} - \Delta t/2) \\ (28) \quad &= u(t_{n+1/2}) - \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 u''(t_{n+1/2}) + \mathcal{O}(\Delta t^3) \end{aligned}$$

Similarly,

$$(29) \quad u(t_{n+1}) = u(t_{n+1/2}) + \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 u''(t_{n+1/2}) + \mathcal{O}(\Delta t^3)$$

(29) We can also expand the first derivatives in Equation (3):

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

(30) (31) Substituting these four expansions into (3) gives

$$\begin{aligned} (32) \quad T_n &= \frac{1}{\Delta t} \left((u(t_{n+1/2}) + \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 u''(t_{n+1/2})) \right. \\ &\quad \left. - (u(t_{n+1/2}) - \frac{\Delta t}{2} u'(t_{n+1/2}) + \frac{1}{2} \left(\frac{\Delta t}{2} \right)^2 u''(t_{n+1/2})) \right) \\ (33) \quad &\quad - \theta (u'(t_{n+1/2}) + \frac{\Delta t}{2} u''(t_{n+1/2})) - (1 - \theta) (u'(t_{n+1/2}) - \frac{\Delta t}{2} u''(t_{n+1/2})) + \mathcal{O}(\Delta t^3) \end{aligned}$$

Many of the terms in (4) cancel so the truncation error simplifies to

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

(34) (35) It can be shown by writing out the 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

$$(36) \quad (37) \quad T_n = \begin{cases} \mathcal{O}(\Delta t) & \text{for } \theta \neq 1/2 \\ \mathcal{O}(\Delta t^2) & \text{for } \theta = 1/2 \end{cases}$$

(38) so that the truncation error of the Crank Nicolson scheme converges twice as fast as that of all other *theta*-methods.

2.1 Pointwise Errors

39 Recall the definition of the θ -method (??) and the corresponding truncation error (2):

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

We re-arrange both of these to get

$$40 \quad U_{n+1} = U_n + \Delta t (\theta f(t_{n+1}, U_{n+1}) + (1 - \theta)f(t_n, U_n)) \quad (5)$$

$$u_{n+1} = u_n + \Delta t (\theta f(t_{n+1}, u_{n+1}) + (1 - \theta)f(t_n, u_n)) + \Delta t T_n \quad (6)$$

$$\begin{aligned}\Rightarrow |u_{n+1} - U_{n+1}| &\leq |u_n - U_n| + \theta \Delta t |f(t_{n+1}, u_{n+1}) - f(t_{n+1}, U_{n+1})| \\ &\quad + (1 - \theta) \Delta t |f(t_n, u_n) - f(t_n, U_n)| + \Delta t |T_n|.\end{aligned} \quad (7)$$

41 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:

$$42 \quad |f(t, u) - f(t, v)| \leq L|u - v|, \quad \forall (t, u), (t, v) \in \Omega.$$

We can use this in (7) to get

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

43 We can re-arrange this to get (for $\Delta t \ll 1$)

$$\begin{aligned}(1 - L\theta\Delta t)|u_{n+1} - U_{n+1}| &\leq (1 + L(1 - \theta)\Delta t)|u_n - U_n| + \Delta t |T_n| \\ &\leq (1 + L(1 - \theta)\Delta t)|u_n - U_n| + \Delta t T_{\max},\end{aligned} \quad (8)$$

where

$$44 \quad T_{\max} = \max_{0 \leq n \leq N} |T_n|$$

is an upper bound on the absolute value of the truncation error.

Now let $e_n = u_n - U_n$ denote the error at time t_n . Then (8) can be written as

$$|e_{n+1}| \leq \frac{1 + L(1 - \theta)\Delta t}{1 - L\theta\Delta t} |e_n| + \frac{\Delta t T_{\max}}{1 - L\theta\Delta t}. \quad (9)$$

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 $n = 0, 1, \dots, N$.

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

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

Thus we have

$$|e_n| \leq \frac{T_{\max}}{L} \left[\exp\left(\frac{LT}{1 - L\theta\Delta t}\right) - 1 \right], \quad (10)$$

for $n = 0, 1, \dots, N$. This shows that the pointwise error has the same order as the truncation error.

3 Implementation

Recall that the θ -method is

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

If $\theta \neq 0$ then we have an implicit equation to solve for U_{n+1} at each timestep. We can write this equation as

$$g(U_{n+1}) := U_{n+1} - U_n - dt\theta f(t_{n+1}, U_{n+1}) - dt(1 - \theta)f(t_n, U_n) = 0.$$

50 50

We can solve this using the Newton Rhapson method which is summarized in the code below.

52

```
% file mynewt.m
% this function finds a root of f(x) using Newton's method and a starting
function x=mynewt(f,fprime,xguess,tol)
```

53

```
x=xguess;

while abs(f(x)) > tol
    x=x-f(x)/fprime(x)
end
end
```

4 Numerical Example

Consider the specific problem

54

$$\frac{du}{dt} = \log\log(4 + u^2)$$

55

for $0 < t \leq 1$ and with $u(0) = 1$. The numerical results are shown in Figure (1) below.

56

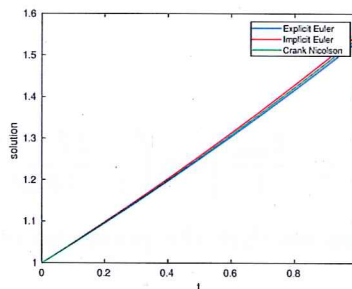


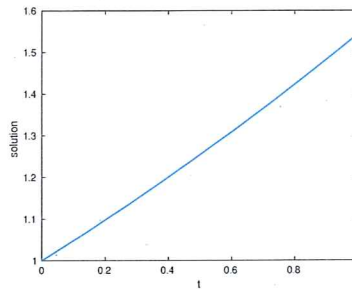
Figure 1: Numerical solution to the example problem.

4.1 Convergence results

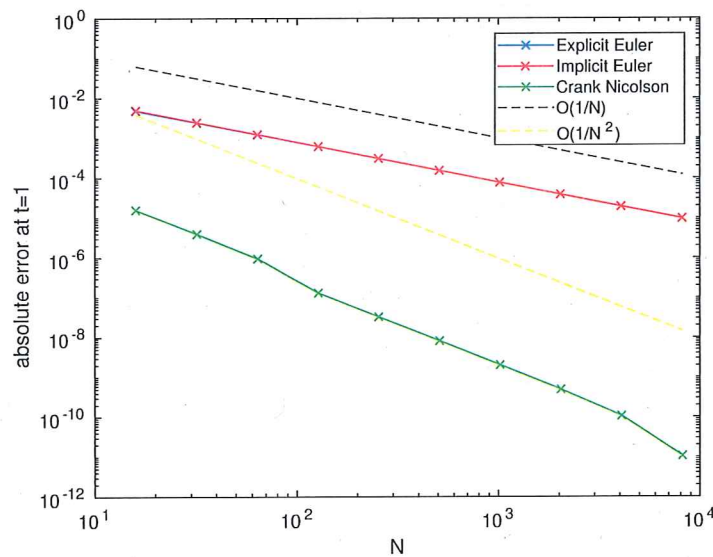
Since the exact solution to this problem is not known, we use a very accurate solution generated using the Crank Nicolson scheme with $N = 10000$ to simulate the exact solution. We then consider the error at time $t = 1$. The results are shown in the figure. We can see that the errors for implicit and explicit Euler are almost the same and converge like $\mathcal{O}(\Delta t)$, whereas the implicit Euler scheme is $\mathcal{O}(\Delta t^2)$.

57

58



60 Figure 2: "Exact" solution to the example problem.



61

62 Figure 3: Convergence to the exact solution of the example problem at time $t = 1$. The errors for implicit and explicit Euler are almost the same and have size $\mathcal{O}(\Delta t)$, whereas the Crank Nicolson error is $\mathcal{O}(\Delta t^2)$.

5 Conclusion

63

We have looked at the θ -method for solving initial value ordinary differential equation problems. The parameter θ is chosen to lie in the interval $[0, 1]$. If $\theta = 0$ then the numerical method is explicit, otherwise it is implicit and a nonlinear equation must

64

be solved at each timestep. If $\theta = \frac{1}{2}$ the method is second order accurate, other-

65

wise the method is first order accurate. These convergence rates were demonstrated numerically.

References

66

- [1] K.W. Morton and D.F. Mayers. Numerical Solution of Partial Differential Equations. Cambridge University Press, 1994.
- [2] Süli, E. & Mayers, D. F. An Introduction to numerical analysis.