Using the θ-method to Solve ODEs

Kathryn Gillow

29th October 2025

1 Introduction

In this report, we study the θ-method for solving ordinary differential equations (ODEs). We begin by introducing the method and deriving its truncation error, which we use to obtain an expression for the local error. Finally, we illustrate the method's convergence properties with a numerical example.

We consider initial value problems of the form

$$\frac{du}{dt} = f(t, u), \quad t > 0, \quad u(0) = u_0. \tag{1}$$

We assume that $f(t, u)$ satisfies a Lipschitz condition in its second argument and that it is bounded. While the θ-method can also be extended to problems with spatial dependence (e.g., the heat equation), this report focuses solely on temporal discretisation. For spatially dependent problems, see Ref. [1].

2 The θ-method

The θ-method approximates the solution to Equation (1) at discrete time points $t_n = n\Delta t$ for $n = 0, 1, \ldots, N$, where $N\Delta t = T$. Let U_n denote the numerical approximation to $u(t_n)$. The method is defined by

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

The parameter $\theta \in [0, 1]$ determines the form of the method: $\theta = 0$ gives the explicit Euler scheme, $\theta = 1$ gives the implicit Euler scheme, and $\theta = \frac{1}{2}$ yields the Crank–Nicolson method.

2.1 Truncation Error

The local truncation error for the θ-method is given by

$$T_n = \frac{u_{n+1} - u_n}{\Delta t} - \left[\theta f(t_{n+1}, u_{n+1}) + (1 - \theta) f(t_n, u_n)\right]. \tag{3}$$

Expanding $u(t)$ and its derivatives about the midpoint $t_{n+\frac{1}{2}}$ yields

$$T_n = \frac{\Delta t}{2}(1-2\theta)u''(t_{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^2). \tag{4}$$

Thus, the method has first-order accuracy when $\theta \neq \frac{1}{2}$ and second-order accuracy for $\theta = \frac{1}{2}$ (Crank–Nicolson).

2.2 Pointwise Errors

Let $e_n = u_n - U_n$ denote the error at time t_n. Assuming that $f(t,u)$ satisfies a Lipschitz condition with constant L, the error satisfies

$$|e_{n+1}| \leq \frac{1 + L(1-\theta)\Delta t}{1 - L\theta\Delta t}|e_n| + \frac{\Delta t}{1 - L\theta\Delta t}T_{\max}, \tag{5}$$

where $T_{\max} = \max|T_n|$. Solving this recurrence relation yields

$$|e_n| \leq \frac{T_{\max}}{L}\left[\exp\left(\frac{LT}{1 - L\theta\Delta t}\right) - 1\right]. \tag{6}$$

Hence, the global error has the same order as the truncation error, confirming the expected convergence behaviour.

3 Implementation

For $\theta \neq 0$, the θ-method requires solving a nonlinear equation for U_{n+1} at each timestep. This can be achieved using the Newton–Raphson method. A robust implementation in Python is shown below:

Listing 1: Newton-Raphson root finder implementation.

```python
def newton_raphson(f, fprime, x0, tol=1e-10, max_iter=100):
    """Find a root of f(x) = 0 using Newton-Raphson iteration."""
    x = x0
    for i in range(max_iter):
        fx = f(x)
        dfx = fprime(x)
        if abs(dfx) < 1e-14:
            raise ValueError("Derivative␣too␣small;␣no␣convergence.")
        x_new = x - fx / dfx
        if abs(x_new - x) < tol:
            return x_new
        x = x_new
    raise ValueError("Newton-Raphson␣did␣not␣converge␣within␣max_iter.")
```

4 Numerical Example

Consider the problem

$$\frac{du}{dt} = \log(\log(4 + u^2)), \quad 0 < t \leq 1, \quad u(0) = 1. \tag{7}$$

The numerical solutions obtained using explicit Euler, implicit Euler, and Crank–Nicolson schemes are compared below. Figure 1 shows the approximate solutions, and Figure 2 presents the reference solution computed with $N = 10000$ using Crank–Nicolson.

Figure 1: Numerical solutions obtained with explicit, implicit, and Crank–Nicolson schemes.

Figure 2: Reference solution computed using Crank–Nicolson with $N = 10000$ (serving as the "exact" solution).

4.1 Convergence Results

The absolute error at $t = 1$ is plotted against the number of timesteps N. As expected, the explicit and implicit Euler methods exhibit first-order convergence $\mathcal{O}(\Delta t)$, while the Crank–Nicolson scheme achieves second-order convergence $\mathcal{O}(\Delta t^2)$.

Figure 3: Convergence of numerical schemes to the reference solution at $t = 1$.

5 Conclusion

We have examined the θ-method for solving initial value problems for ODEs. The parameter $\theta \in [0, 1]$ determines both accuracy and stability: explicit methods ($\theta = 0$) are simple but conditionally stable, while implicit methods ($\theta > 0$) are unconditionally stable but require solving nonlinear equations. The Crank–Nicolson scheme ($\theta = \frac{1}{2}$) achieves second-order accuracy and good stability properties, making it a robust choice for stiff problems.

References

[1] K.W. Morton and D.F. Mayers, *Numerical Solution of Partial Differential Equations*, Cambridge University Press, 1994.

[2] E. Suli and D.F. Mayers, *An Introduction to Numerical Analysis*, Oxford University Press, 2003.

[3] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*, Wiley, 2016.