Elements of Calculus

Prof. Sören Petrat, Constructor University

Lecture notes from Spring 2025

Session 17
(Week 9A)

5. ODEs

Topic for Week 9 A: Finite Difference Methods

How do we solve ODEs $\frac{dy}{dt} = f(y(t), t)$ on a computer?

We need to discretize our variable t: Consider the interval $[t_0, T]$, and discretize it into N equally spaced points $t_n = t_0 + n\Delta t$, with $\Delta t = \frac{T - t_0}{N}$. Note that then $t_N = T$.
Correspondingly we define a discretized function/sequence y_n, $n = 0, \dots, N$.

Next: How do we discretize the ODE, i.e., $\frac{dy}{dt}$?

This can be done in different ways:

- Forward difference quotient: $D_n^+ y = \dfrac{y(t_n + \Delta t) - y(t_n)}{\Delta t}$.

What is the error between this approximation and the derivative $\frac{dy}{dt}$? Let us use a Taylor expansion: $y(t_n + \Delta t) = y(t_n) + \Delta t \underbrace{y'(t_n)}_{\frac{dy}{dt}(t_n)} + \frac{(\Delta t)^2}{2} \underbrace{y''(m)}_{\frac{d^2y}{dt^2}(m)}$ for some $m \in (t_n, t_n + \Delta t)$.

$$\Rightarrow \frac{dy}{dt}(t_n) - \left(\frac{y(t_n + \Delta t) - y(t_n)}{\Delta t} \right) = \frac{dy}{dt}(t_n) - \left(\frac{y(t_n) + \Delta t\, y'(t_n) + \frac{(\Delta t)^2}{2} y''(m) - y(t_n)}{\Delta t} \right)$$

$$= \frac{\Delta t}{2} y''(m)$$

- $\boxed{\text{Backward difference quotient}}$: $D_n^- y = \dfrac{y(t_n) - y(t_n - \Delta t)}{\Delta t}$

A Taylor expansion yields

$$y(t_n - \Delta t) = y(t_n) - \Delta t\, y'(t_n) + \frac{(\Delta t)^2}{2} y''(m) \text{ for some } m \in (t_n, t_n + \Delta t).$$

$$\Rightarrow \frac{dy}{dt}(t_n) - \left(\frac{y(t_n) - y(t_n - \Delta t)}{\Delta t} \right) = \frac{dy}{dt}(t_n) - \left(\frac{y(t_n) - \left[y(t_n) - \Delta t\, y'(t_n) + \frac{(\Delta t)^2}{2} y''(m) \right]}{\Delta t} \right)$$

$$= \frac{\Delta t}{2} y''(m)$$

- $\boxed{\text{Central difference quotient}}$: $D_n^0 y = \dfrac{y(t_n + \Delta t) - y(t_n - \Delta t)}{2 \Delta t}$.

Here, a third order Taylor expansion gives the following error:

$$\frac{dy}{dt}(t_n) - \left(\frac{y(t_n + \Delta t) - y(t_n - \Delta t)}{2 \Delta t} \right)$$

$$= \frac{dy}{dt}(t_n) - \left(\frac{y(t_n) + \Delta t\, y'(t_n) + \frac{(\Delta t)^2}{2} y''(t_n) + \frac{(\Delta t)^3}{6} y'''(m) - \left[y(t_n) - \Delta t\, y'(t_n) + \frac{(\Delta t)^2}{2} y''(t_n) - \frac{(\Delta t)^3}{6} y'''(\hat{m}) \right]}{2 \Delta t} \right)$$

$$= \frac{(\Delta t)^2}{12} \left(y'''(m) + y'''(\hat{m}) \right) \quad \text{for some } m \in (t_n, t_n + \Delta t),\ \hat{m} \in (t_n - \Delta t, t_n).$$

$\Rightarrow$ The central difference quotient has a smaller error term.

However, it can lead to inaccuracies when applied to oscillating functions.

Accordingly, there are different methods of approximating the ODE $\frac{dy}{dt} = f(y(t), t)$:

- $\boxed{\text{Forward/Explicit Euler Method}}$: We discretize the ODE as $\dfrac{y_{n+1} - y_n}{\Delta t} = f(y_n, t_n)$.

Need to solve the (explicitly given) iteration $y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$.

The total error between the solution $y(t)$ of $\frac{dy}{dt} = f(y(t), t)$ and the solution y_n of the discretized ODE $y_{n+1} = y_n + \Delta t\, f(y_n, t_n)$ is $\frac{(\Delta t)^2}{2} y''(m)$ (i.e., Δt times the error from before)

in each time step, hence we would expect $|y(T) - y_N| \approx \sum_{n=1}^{N} \frac{(\Delta t)^2}{2} y''(m_n) \underset{\Delta t = \frac{T - t_0}{N}}{\approx} \frac{c(T)}{N}$.

- $\boxed{\text{Backward/Implicit Euler Method:}}$ We discretize the ODE as $\frac{y_{n+1} - y_n}{\Delta t} = \underbrace{f(y_{n+1}, t_{n+1})}_{\text{Compared to before, we evaluate } f \text{ at } n+1 \text{ here.}}$.

This cannot be written down as an iteration immediately. Here, to compute y_{n+1}, we need to solve an equation, which might or might not be possible depending on f. (y_{n+1} is implicitly given.)

- There are many more methods, a more complete discussion is part of the Numerical Methods class.

Example:

- Exponential decay: $\frac{dy}{dt} = -\lambda y$ with $\lambda > 0$.

Here, we know the exact solution $y(t) = y_0 e^{-\lambda t}$, with $y(0) = y_0$ (i.e., $t_0 = 0$ here).

↳ Explicit Euler Method: $\frac{y_{n+1} - y_n}{\Delta t} = -\lambda y_n \Rightarrow y_{n+1} = y_n - \lambda y_n \Delta t = y_n (1 - \lambda \Delta t)$.

This iteration has solution $y_N = (1 - \lambda \Delta t)^N y_0$.

In the limit $N \to \infty$ the discretized solution becomes $y_\infty = \lim_{N \to \infty} y_N = \underbrace{\lim_{N \to \infty} \left(1 - \lambda \frac{T}{N}\right)^N}_{= e^{-\lambda T}} y_0 = e^{-\lambda T} y_0$, i.e., it converges to the exact solution.

If N is some finite value, we want the solution to decrease, because otherwise it would move away from $e^{-\lambda t} y_0$.

$\Rightarrow$ Need $|1-\lambda\Delta t| < 1 \Rightarrow \begin{cases} 1-\lambda\Delta t < 1 \Rightarrow \lambda\Delta t > 0 \text{ holds always for } \lambda>0. \\ -(1-\lambda\Delta t) < 1 \Rightarrow \lambda\Delta t < 2 \Rightarrow \Delta t < \frac{2}{\lambda}. \end{cases}$

$\Rightarrow$ We need to choose N large enough, namely such that $\Delta t = \frac{T}{N} < \frac{2}{\lambda}$ $\left(N > \frac{T\lambda}{2}\right)$.

This is called a $\boxed{\text{stability condition.}}$

↳ Implicit Euler Method: $\frac{y_{n+1}-y_n}{\Delta t} = -\lambda y_{n+1}$

$\Rightarrow y_{n+1} = y_n - \lambda\Delta t\, y_{n+1} \Rightarrow (1+\lambda\Delta t)\, y_{n+1} = y_n \Rightarrow y_{n+1} = \frac{1}{1+\lambda\Delta t}\, y_n,$

i.e., for this example we can solve the iteration.

The solution is $y_N = \left(\frac{1}{1+\lambda\Delta t}\right)^N y_0$.

Here, the stability condition reads $\left|\frac{1}{1+\lambda\Delta t}\right| < 1$, which always holds (since $\lambda, \Delta t > 0$).

$\Rightarrow$ The implicit method is $\boxed{\text{unconditionally stable.}}$

Conclusion:
- Explicit methods lead to explicitly given iterations, which can then for example be solved numerically. However, there are usually stability conditions, i.e., the time step Δt has to be chosen small enough.
- Implicit Methods lead to implicit equations in each iteration step, and it is hence more computationally expensive to solve them. But they are usually unconditionally stable.