

Chapter

Differential Equations

..1– Example: Below are three differential equations (of increasing complexity) followed by a fourth equation that is not a differential equation. Here $t \in \mathbb{R}$, $\mathbf{t} := [t_1, t_2] \in \mathbb{R}^2$ and $\gamma_i \in \mathbb{R}$ and $\partial y = y'$ in one variable and otherwise $\partial y = (\partial_1 y, \dots, \partial_m y)$.

$$\gamma_2 \frac{\partial^2 y(t)}{\partial t^2} + \gamma_1 \frac{\partial y(t)}{\partial t} + \gamma_0 y(t) = 0 \quad (1)$$

$$\left(\frac{\partial y(t)}{\partial t} \right)^2 - t y(t)^4 \frac{\partial y(t)}{\partial t} + \sin y(t) = 0 \quad (2)$$

$$\frac{\partial^2 y(\mathbf{t})}{\partial t_1 \partial t_2} = t_1 t_2 y(\mathbf{t}) \frac{\partial y(\mathbf{t})}{\partial t_1} \frac{\partial y(\mathbf{t})}{\partial t_2} \quad (3)$$

$$\partial y = y(y(t)). \quad (4)$$

Solving a differential equation means to find a function y that satisfies the equation.

I Ordinary differential equations

An ODE is an equation of the form

$$F(t, y(t), \partial y(t), \dots, \partial^r y(t)) = 0, \quad t \in \mathbb{R}$$

to be solved for $y : \mathbb{R} \rightarrow \mathbb{R}^n$.

A.I.2– Exercise [2]: What is an explicit, what an implicit ODE?

The *Cauchy Problem* or IVP (Initial value problem)

$$\begin{aligned}\partial^n y &= F(t, \partial^1 y, \dots, \partial^{n-1} y) \\ (\partial^i y)(a) &= y_{i0}, \quad i = 0..n - 1,\end{aligned}$$

is equivalent to the first order system

$$\begin{aligned}\partial y_i &= f_i(t, \mathbf{y}), \quad i = 1..m \\ \mathbf{y}(a) &= \mathbf{y}_0, \quad t \in [a, b],\end{aligned}$$

where $\mathbf{y} := (y_1, \dots, y_m)$

A.I.3– Example: Consider the system (from $\partial^2 u = -u + t$)

$$\begin{aligned}\partial u &= v \\ \partial v &= -u + t\end{aligned}\tag{I.1}$$

With $y := \begin{bmatrix} u \\ v \end{bmatrix}$, we have

$$\partial y = F(t, y) = \begin{bmatrix} v \\ -u \end{bmatrix} + \begin{bmatrix} 0 \\ t \end{bmatrix}.$$

The IVP is to determine a function (on a manifold) given that its slope (a direction field) and an starting point are known.

A.I.4– Exercise [10]: Draw the direction field of

$$\partial y = \frac{y}{1-t}$$

in a neighborhood of $t = 1$.

Cauchy, (Picard, Lindelöf) existence and uniqueness theorem

If $f_i, i = 1..m$ are bounded and uniformly Lipschitz continuous on

$$G := \{t, \mathbf{y} : |t - t_0| < a, |y_i - y_{i,0}| < b\},$$

ie

$$|f_i(t, \bar{\mathbf{y}}) - f_i(t, \mathbf{y})| < K \|\bar{\mathbf{y}} - \mathbf{y}\|_1$$

then the Cauchy problem has a *unique solution* for $|t - t_0| < \min\{a, b/\max\{f_i\}\}$.

The proof is via *Picard iteration*

$$y_n = y_0 + \int_a^t f(\tau, y_{n-1}(\tau))d\tau, \quad y_0(t) = y_0.$$

A.I.5– Exercise [5]: Iterate three times to approximately solve $\partial y = t^2 + y^2$, $y_0 = y(0) = 1$.

A.I.6– Exercise [10]: Are the conditions of Cauchy’s Theorem satisfied for

$$\partial y = \frac{y}{1-t}$$

in a neighborhood of $t = 1$?

There are a number of classical methods to solve classes of ODEs explicitly. But we skip them for now in favor of numerical techniques.

II Euler’s Method

yields generally inaccurate approximations to the function and should be avoided in practice – but it nicely motivates the strategy. Consider the following Taylor expansion and note the substitution $\partial y = f$:

$$\begin{aligned} y(t+h) &= y(t) + h\partial y(t) + O(h^2) \\ &= y(t) + hf(t, y(t)) + O(h^2). \end{aligned}$$

The substitution is smart because f is known for all values of t and y . Accepting an error of $O(h^2)$, Euler decided to start with $Y_0 = y(a) = \alpha$ and compute the approximation Y to y by the *algorithm*

$$\begin{aligned} Y_{i+1} &= Y_i + hf(a + ih, Y_i), \\ i &= 0, 1, \dots, N := (b-a)/h, \end{aligned}$$

for a fixed step size h .

B.II.7– Example: The IVP

$$\partial y = -y + t + 1, \quad t \in [0, 1] \quad y(0) = 1. \quad (\text{II.1})$$

has the exact solution $y = t + e^{-t}$. Choose $h = \frac{1}{10}$ and $Y_0 = 1$. Then

$$\begin{aligned} Y_1 &= 1 + \frac{1}{10}(-1 + (0 + 0\frac{1}{10}) + 1) = 1 \\ Y_2 &= 1 + \frac{1}{10}(-1 + (0 + 1\frac{1}{10}) + 1) = 1.01 \\ &\vdots \end{aligned}$$

Note that the algorithm makes sense when f and y are vectors. Then ∂f is a matrix.

a Error analysis of Euler's method

The *local discretization error* is defined as

$$e_n := y(t_n) - Y_n.$$

Subtracting Euler's rule, $Y_{n+1} = Y_n + hf(t_n, Y_n)$, from the Taylor expansion of $y(t_{n+1}) = y(t_n) + h\partial y(t_n) + \frac{h^2}{2}D^2y(\tau)$ yields

$$\begin{aligned} e_{n+1} &= e_n + h[f(t_n, y(t_n)) - f(t_n, Y_n)] + \frac{h^2}{2}D^2y(\xi) \\ &= e_n + h\frac{f(t_n, y(t_n)) - f(t_n, Y_n)}{y(t_n) - Y_n}e_n + \frac{h^2}{2}D^2y(\xi) \\ &\leq e_n + hLe_n + \frac{h^2M}{2} \end{aligned}$$

for $\partial_2 f \leq L$ and $D^2y \leq M$ on the interval of interest.

b Upper bound on the local discretization error

B.II.8– Exercise [5]: Check that the difference equation

$$\xi_0 = 0, \quad \xi_{n+1} = (1 + hL)\xi_n + \frac{h^2M}{2}$$

has the solution $\xi_n := c(1 + hL)^n$, $c := \frac{hM}{2L}$. Then $\xi_n \geq e_n$ is an upper bound on the *local discretization error*. Start the induction with $\xi_0 = e_0 = 0$ and check that $\xi_{n+1} = (1 + hL)\xi_n + cLh \geq (1 + hL)e_n + cLh \geq e_{n+1}$.

To get a *global discretization error* use $1 + x \leq e^x$, to estimate

$$\xi_n \leq c(e^{(nh)L} - 1) = c(e^{(t_n - x_0)L} - 1).$$

local error

== *local truncation error*

== *local discretization error*

≈ *accuracy* (when only the differential operator and not the right hand side is considered).

Adding all the N local errors yields the *total discretization error*

== *accumulated error*

== *global discretization error*.

B.II.9– Example: The local discretization error of Euler’s method is $O(h^2) = O(N^{-2})$.

The total discretization error of Euler’s method is $NO(N^{-2}) = O(N^{-1})$. This is consistent with the error $h\frac{M}{2L}(e^{t_n - x_0} - 1)$ computed earlier.

III Taylor-expansion methods.

Euler’s method is also called a first degree (second order) Taylor method (why?). A second degree Taylor method is based on the expansion

$$y(t + h) = y(t) + \partial y(t)h + \partial^2 y(t)\frac{h^2}{2} + O(h^3)$$

Since $\partial y(t) = f(t, y(t)) = f(g(t))$ where $g : \mathbb{R} \rightarrow \mathbb{R}^2$, $t \mapsto g(t) := (t, y(t))$,

$$\begin{aligned} \partial^2 y &= \partial(f(g)) = (\partial f)(g)\partial g = \begin{bmatrix} \partial_1 f & \partial_2 f \end{bmatrix} \begin{bmatrix} \partial g_1 \\ \partial g_2 \end{bmatrix} = \begin{bmatrix} \partial_1 f & \partial_2 f \end{bmatrix} \begin{bmatrix} 1 \\ f \end{bmatrix} \\ &= \partial_2 f f + \partial_1 f. \end{aligned} \tag{III.1}$$

C.III.10– Exercise [5]: Verify the expansion by applying the chain rule of differentiation.

C.III.11– Example: If $f = y^2 + 3t$, then $\partial_2 f = 2y$ and $\partial_1 f = 3$.

a Systems of ODEs

Recall that systems of ODEs arise for example from writing a higher order ODE as a system of first order equations.

$$\partial^2 y = f(t, y(t))$$

becomes

$$\partial \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} z \\ f(t, y(t)) \end{bmatrix}$$

C.III.12– Example: We continue (I.1), $\partial u = v$, $\partial v = -u + t$, and apply a second degree Taylor method to these two ODEs. Set $\partial y = F(t, y)$ and, according to (III.1), to get the approximation

$$\begin{bmatrix} U_{i+1} \\ V_{i+1} \end{bmatrix} = \begin{bmatrix} U_i \\ V_i \end{bmatrix} + \begin{bmatrix} V_i \\ -U_i + t_i \end{bmatrix} h + \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} V_i \\ -U_i + t_i \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \frac{h^2}{2}.$$

We can derive *higher order Taylor methods* from the expansion

$$y(t+h) = \sum_{j=0}^k \frac{h^j}{j!} D^j y(t) + O(h^{k+1})$$

and substituting.

$$\begin{aligned} \partial y &= f \\ \partial^2 y &= \partial_2 f * \partial y + \partial_1 f = \partial_2 f * f + \partial_1 f \\ \partial^3 y &= \partial_2^2 f * \partial y * f + \partial_1 \partial_2 f * f + \dots \end{aligned}$$

IV Runge-Kutta Methods

Replacing derivatives by difference approximations and hence evaluations leads to the Runge-Kutta (RK) methods. An s -stage explicit Runge Kutta method is

defined by

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + c_2h, y_n + ha_{21}k_1) \\ &\vdots \\ k_s &= f(t_n + c_sh, y_n + h \sum_{j=1}^{s-1} a_{js}k_j) \\ y_{n+1} &= y_n + h \sum_{j=1}^s b_jk_j \end{aligned}$$

where usually $c_i = \sum_{j=1}^{i-1} a_{ij}$.

For a RK method of order p the local discretization error is

$$\|y(t_0 + h) - y_1\| \leq Kh^{p+1}.$$

D.IV.13– Example: Check the matlab file rk43.m.

The basic idea is illustrated by a 1-stage Runge-Kutta method. If f is independent of y then

$$\partial y = f(t), \quad y(t_0) = y_0$$

has the solution $y(t) = y_0 + \int_{t_0}^t f(\tau)d\tau$. The integral $\int_{t_i}^{t_{i+1}} f(\tau)d\tau$ can be approximated (with an error bounded by Ch^2) by the composite midpoint rule $hf(t_i + h/2)$. In analogy, we approximate

$$y(t_i + h) \approx y_i + hf(t_i + h/2, y(t_i + h/2))$$

using an Euler step (whose error is scaled by h) to get

$$\begin{aligned} k_1 &:= f(t_i, y_i). \\ y(t_i + h/2) &\approx y_i + k_1h/2, \end{aligned}$$

The resulting rule is

$$\begin{aligned} k_1 &:= f(t_i, Y_i). \\ Y_{i+1} &= Y_i + k_1h/2, \end{aligned}$$

In general the approach is to choose constants a_i and b_i such that

$$\partial y - f(t, y(t)) - y_{\text{RK}} = O(h^r)$$

or

$$y(t_n) - Y_n = O(h^{r+1})$$

a Error analysis of a Runge Kutta method

The *local discretization error* is defined as

$$e_n := y(t_n) - Y_n.$$

A more general 1-stage method allows us to choose a_1, a_2, α_2 and δ_2 such that

$$\begin{aligned} & \partial y(t, y) + \frac{h}{2} \partial^2 y(t, y) + \frac{h^2}{3!} \partial^3 y(t, y) \\ & - (a_1 f(t, y) + a_2 f(t + \alpha_2, y + \delta_2 f(t, y))) \\ & = O(h^2) \end{aligned}$$

The choice $a_1 = a_2 = 1/2, \alpha_2 = \delta_2 = h$ yields

$$\begin{aligned} \frac{1}{2} f(t, y) + \frac{1}{2} f(t + h, y + hf) &= \frac{1}{2} f(t, y) + \frac{1}{2} (f(t, y + hf) + h \partial_1 f(t, y + hf) + O(h^2)) \\ &= \frac{1}{2} f(t, y) + \frac{1}{2} (f(t, y) + hf(t, y) \partial_2 f(t, y) + h \partial_1 f(t, y)) + O(h^2) \\ &= f(t, y) + \frac{h}{2} (f \partial_2 f + \partial_1 f) + O(h^2) \\ &= \partial y + \partial^2 y + O(h^2). \end{aligned}$$

Hence the local discretization error is $r = 2$.

In general the RK methods are analyzed by looking at the Taylor-expansion

$$\begin{aligned} e(h) &= y(t_n + h) - Y_{n+1} \\ &= y(t_n + h) - (Y_{n+1} + \sum_{i=1}^s b_i k_i) \\ &\leq Ch^{p+1} \end{aligned}$$

V Multistep methods

can improve on the earlier methods by using previous values Y_j, \dots, Y_{j-k} to compute Y_{k+1} , the approximation to $y(t_{k+1})$ (I just switched notation, $Y_j := Y(j)$). The general form of a $k + 1$ -step method is

$$Y_{j+1} = \sum_{l=0}^k a_l Y_{j-l} + h \sum_{l=-1}^k b_l F_{j-l}$$

The starting values Y_1, \dots, Y_k have to be generated by a different method (e.g. Runge-Kutta, Taylor, etc.). For more details see for example [CdB], Chapter 8. Multistep methods can be viewed as Taylor methods with derivatives replaced by differences. For example, the **Adams-Bashforth method** is based on the approximation

$$\begin{aligned} y(t_{j+1}) &= y(t_j) + \int_{t_j}^{t_{j+1}} f(y(\tau), \tau) d\tau \\ &\approx y(t_j) + \int p_k(f(t), t; \tau) \end{aligned}$$

where $t = [t_j, \dots, t_{j-k}]$ and $t_l := lh$, h is the step size. Choosing the interpolating polynomial p_k in Newton form, we get

$$Y_{j+1} = Y_j + h \sum_{l=0}^k \gamma_l \nabla^l F_j$$

where ∇^l is a backward difference and

$$\gamma_k = 1 - \sum_{l=0}^{k-1} \frac{\gamma_l}{k+1-l}.$$

This method has order of accuracy $k+1$. Another example is the **Adams-Moulton method** which also has a term $F_{j+1} = f(Y_{j+1})$ on the right hand side so that an equation has to be solved. Instead of directly solving one iterates, combining the two methods in the *predictor-corrector algorithm*:

1. predict with A-B: $Y_{j+1} = Y_j + h \sum_{l=0}^k \gamma_l \nabla^l F_j$.
2. correct with A-M: until $|Y_{j+1}^{new} - Y_{j+1}^{old}| = O(h^{k+3})$ do

$$Y_{j+1} = Y_j + h \sum_{l=-1}^k \gamma'_l \nabla^{l+1} F_{j+1}.$$

a Error analysis of a multistep methods

The *local discretization error* is defined as

$$e_n := y(t_n) - Y_n.$$

The quality of a multistep method is judged by *accuracy* and *stability*. Given an ODE $p(\partial)y = f$, where p is a polynomial, we say a multistep method $P(y, h) = R(f, h)$ is accurate of order r if for any sufficiently smooth function y

$$R(p(\partial)y, h) - P(y, h) = O(h^{r+1}).$$

E.V.14– Example: Consider the multistep method

$$Y_{j+1} = \sum_{l=0}^k a_l Y_{j-l} + \frac{h^2}{2} f(Y_j)$$

for approximating the second order equation

$$\partial^2 y = f(y).$$

Under what conditions on the parameters a_l is the method second order accurate? For this we compute the a_l so that

$$\partial^2 y - f(y) - [Y_{j+1} - \sum_{l=0}^k a_l Y_{j-l} + \frac{h^2}{2} f(Y_j)] = O(h^3)$$

Here $P(y, h) = Y_{j+1} - \sum_{l=0}^k a_l Y_{j-l}$, and $R(f, h) = \frac{h^2}{2} f(Y_j)$, hence $R(p(\partial)y, h) = R(\partial^2 y, h) = \frac{h^2}{2} \partial^2 y$. First expand

$$y(t+h) = y(t) + h * \partial y(t) + \frac{h^2}{2} \partial^2 y(t) + O(h^3) \quad (\text{V.1})$$

The term $\partial^2 y(t)$ can be replaced by $f(Y_j)$, $y(t+h)$ by Y_{i+1} etc. To replace $\partial y(t)$ use

$$y(t-h) = y(t) - h * \partial y(t) + \frac{h^2}{2} \partial^2 y(t) + O(h^3)$$

$$y(t-2h) = y(t) - 2h * \partial y(t) + \frac{4h^2}{2} \partial^2 y(t) + O(h^3)$$

Multiplying the first by -4 and adding (the multiresolution trick) yields

$$y(t - 2h) - 4y(t - h) + 3y(t) = 2h\partial y(t) + O(h^3),$$

or $h\partial y(t) = \frac{y(t-2h) - 4y(t-h) + 3y(t)}{2} + O(h^3)$. Therefore

$$Y_{i+1} = Y_i + \frac{Y_{i-2} - 4Y_{i-1} + 3Y_i}{2} + \frac{h^2}{2}f(Y_i) + O(h^3)$$

and $a_0 = \frac{5}{2}$, $a_1 = -2$, $a_2 = \frac{1}{2}$.

VI Alternative techniques

(1) Integration by expansion, $y' = f(t, y)$

$$y(t) = \sum_{i=0}^{\infty} a_i(t - t_0)^i, \quad a_i = \frac{1}{i!}y^{(i)}(t_0)$$

$$y' = t^2 + y^2$$

(2) Symbolic substitution methods