Revision of one-step methods for ODEs

In this course we consider initial value problems (IVPs) defined by autonomous Ordinary Differential Equations (ODEs) of the form

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathcal{F}(\mathbf{x}(t)) \in \mathbb{R}^d \\ \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^d. \end{cases} \tag{1}$$

We recall that non-autonomous case can be reduced to this case by introducing the additional equation $\dot{t} = 1$. We will refer to the vector field $\mathcal{F}$ interchangeably as a function $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ and as a smooth vector field, denoting it as $\mathcal{F} \in \mathfrak{X}(\mathbb{R}^d)$. $\mathcal{F}$ is supposed to be Lipschitz continuous, so that we can guarantee the existence and uniqueness of the solution to (1).

Notation: I will try to stick as much as possible to the following notation conventions. Vectors are represented with bold symbols, like $\mathbf{x}$. Matrices are not, but I write them with capital letters. For vector fields and sets I will use calligraphic letters, such as $\mathcal{F}$. To denote the exact flow map of a vector field $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ I will use interchangeably $\phi^t_{\mathcal{F}} : \mathbb{R}^d \to \mathbb{R}^d$ and $\phi_{\mathcal{F}} : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$, where $\phi^t_{\mathcal{F}}(\mathbf{x}) := \phi_{\mathcal{F}}(t, \mathbf{x})$.

Let us consider the time domain $[0, T]$, $T > 0$, and introduce a uniform grid over it defined as $t_i = ih$, $i = 0, ..., N$, $h = T/N$. A one-step numerical method $\varphi^h_{\mathcal{F}} : \mathbb{R}^d \to \mathbb{R}^d$ aims to provide an approximation of the exact flow map $\phi^h_{\mathcal{F}} : \mathbb{R}^d \to \mathbb{R}^d$ of the vector field $\mathcal{F}$ to which it is applied. Whenever it will be clear from the context which vector field we are working with, we will omit the subscript $\mathcal{F}$, and write φ^h.

Definition 1 (Method of order p)**.** *A one-step numerical method $\varphi^h : \mathbb{R}^d \to \mathbb{R}^d$ has order p if, whenever applied to a smooth enough vector field $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$, it satisfies*

$$\varphi^h_{\mathcal{F}} = \phi^h_{\mathcal{F}} + \mathcal{O}(h^{p+1}).$$

1 Runge–Kutta methods

The simplest numerical method one can consider is the explicit Euler method, defined as

$$\varphi^h_{\mathcal{F}}(\mathbf{x}) = \mathbf{x} + h\mathcal{F}(\mathbf{x}).$$

By Taylor expanding the exact solution at $t = 0$, we see that

$$\phi_{\mathcal{F}}^h(\mathbf{x}) = \mathbf{x} + h\mathcal{F}(\mathbf{x}) + \mathcal{O}(h^2),$$

hence telling us that the explicit Euler method is first-order accurate. A generalisation of this method is provided by the very popular family of Runge–Kutta methods, with which we will work quite a lot. We provide the definition of these methods for non-autonomous vector fields so it is presented in full generality.

Definition 2 (Runge–Kutta method)**.** *Let us consider the non-autonomous differential equation $\dot{\mathbf{x}}(t) = \mathcal{F}(t, \mathbf{x}(t))$. A Runge–Kutta method of s stages based on the tableau $(A, \mathbf{b}, \mathbf{c})$, $A \in \mathbb{R}^{s\times s}$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^s$, is a one-step method defined as*

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{s} b_i \mathcal{F}(t_n + c_i h, \mathbf{k}_{n,i})$$

$$\mathbf{k}_{n,i} = \mathbf{x}_n + h\sum_{j=1}^{s} a_{ij} \mathcal{F}(t_n + c_j h, \mathbf{k}_{n,j}).$$

We call $\mathbf{k}_{n,i}$, $i = 1, ..., s$, the hidden stages of the method.

We will generally drop the subscript n in the stages. A Runge–Kutta method is explicit if the matrix A in the tableau is stricly lower triangular. In this case there is no need to solve a non-linear algebraic equation at every step of the method. If the method is implicit, one can approximate the solution of the non-linear algebraic equation associated to one step by using iterative methods like Newton, or quasi-Newton schemes.

The study of the order conditions of Runge–Kutta methods is very well developed, but it is out of the scope of this course. For the interested reader see [1, 3]. In practice, the order conditions will amount to restrictions over the tableau $(A, \mathbf{b}, \mathbf{c})$.

Even though the order conditions for a generic Runge–Kutta method are not simple to derive, we can do that for an important subfamily of these methods: collocation methods. We dedicate the last part of this introduction of one-step methods to collocation methods because they have an interesting interpretation, and they will be perfect examples of structure-preserving numerical methods.

1.1 Collocation methods

Let us now consider the differential equation $\dot{\mathbf{x}} = \mathcal{F}(\mathbf{x})$, and suppose that we have just obtained an approximate solution $\mathbf{x}_n \approx \mathbf{x}(t_n)$. We want to approximate the solution $\mathbf{x}(t)$ at time $t_{n+1} = t_n + h$, so after one time step. To do so, we make the assumption that in-between time $t = t_n$ and $t_{n+1} = t_n + h$ we can approximate the solution with a polynomial of degree s. Let us call $\widetilde{\mathbf{x}} \in \mathbb{P}^s(\mathbb{R})$ this polynomial.

To characterise it, we enforce that $\widetilde{\mathbf{x}}(t_n) = \mathbf{x}_n$, and that

$$\dot{\widetilde{\mathbf{x}}}(t_n + c_i h) = \mathcal{F}(\widetilde{\mathbf{x}}(t_n + c_i h)),\ i = 1, ..., s,$$

for a choice of s distinct numbers $0 \leq c_1 < c_2 < ... < c_s \leq 1$. These $s+1$ conditions suffice to uniquely characterise a polynomial of degree s. We now write explicitly the form of $\dot{\widetilde{\mathbf{x}}}(t)$:

$$\dot{\widetilde{\mathbf{x}}}(t) = \sum_{i=1}^{s} \mathcal{F}(\widetilde{\mathbf{x}}(t_n + c_i h))\ell_i\left(\frac{t-t_n}{h}\right), \tag{2}$$

where $\ell_i(t) \in \mathbb{P}^{s-1}(\mathbb{R})$ are the elementary Lagrange polynomials such that

$$\ell_i(c_j) = \delta_{ij} = \begin{cases} 1, \ i = j, \\ 0, \ i \neq j. \end{cases}$$

We recall that these polynomials are defined as

$$\ell_i(t) = \prod_{\substack{j=1 \\ j\neq i}}^{s} \frac{t-c_j}{c_i - c_j}.$$

We now integrate both sides of (2) over the interval $[t_n, t_n + c_i h]$ to get

$$\widetilde{\mathbf{x}}(t_n + c_i h) = \mathbf{x}_n + \sum_{j=1}^{s} \mathcal{F}(\widetilde{\mathbf{x}}(t_n + c_j h)) \int_{t_n}^{t_n + c_i h} \ell_j\left(\frac{t-t_n}{h}\right) dt.$$

The change of variables $t = t_n + sh$, allows us to rewrite the condition as

$$\widetilde{\mathbf{x}}(t_n + c_i h) = \mathbf{x}_n + h\sum_{j=1}^{s} \mathcal{F}(\widetilde{\mathbf{x}}(t_n + c_j h)) \int_{0}^{c_i} \ell_j(s) ds,$$

which can be rewritten as

$$\mathbf{k}_i = \mathbf{x}_n + h\sum_{j=1}^{s} a_{ij}\mathcal{F}(\mathbf{k}_j)$$

if we set

$$\mathbf{k}_i := \widetilde{\mathbf{x}}(t_n + c_i h), \ a_{ij} := \int_0^{c_i} \ell_j(s) ds.$$

Integrating (2) over $[t_n, t_n + h]$, we instead find

$$\widetilde{\mathbf{x}}(t_n + h) = \mathbf{x}_n + h\sum_{i=1}^{s} b_i \mathcal{F}(\mathbf{k}_i),$$

where we set $b_i := \int_0^1 \ell_i(s) ds$. We thus see that as long as we define $\mathbf{x}_1 = \varphi_{\mathcal{F}}^h(\mathbf{x}_0) := \widetilde{\mathbf{x}}(t_0 + h)$, we recover a one-step method of the Runge–Kutta type with a suitably constrained tableau $(A, \mathbf{b}, \mathbf{c})$.

An important and common choice for the coefficients $c_1, ..., c_s$ is the one provided by Gauss-Legendre quadrature nodes. We call these methods Gauss-Legendre collocation methods.

Theorem 1 (Gauss-Legendre collocation methods)**.** *The Gauss-Legendre collocation methods based on s collocation nodes are of order $2s$ when applied to a smooth enough vector field $\mathcal{F}:\mathbb{R}^d\to\mathbb{R}^d$.*

To prove this theorem, we need a very formula which is very useful also for other estimates. This is The Gröbner-Alekseev formula:

Proposition 1 (Gröbner-Alekseev Formula [1])**.** *Let us consider the two autonomous initial value problems*

$$\begin{cases}\dot{\mathbf{x}}(t)=\mathcal{F}(\mathbf{x}(t))\\ \mathbf{x}(0)=\mathbf{x}_0\end{cases},\qquad \begin{cases}\dot{\mathbf{y}}(t)=\mathcal{F}(\mathbf{y}(t))+\mathcal{G}(\mathbf{y}(t))\\ \mathbf{y}(0)\end{cases}$$

with $\mathcal{F}\in\mathcal{C}^1(\mathbb{R}^d,\mathbb{R}^d)$, and supposing they both admit a unique solution. Then

$$\mathbf{y}(t)-\mathbf{x}(t)=\int_0^t \left.\frac{\partial\phi_{\mathcal{F}}^{\tau,t}(\mathbf{z}_0)}{\partial\mathbf{z}_0}\right|_{\mathbf{z}_0=\mathbf{y}(\tau)}\mathcal{G}(\mathbf{y}(\tau))d\tau \tag{3}$$

for every $\tau\in[0,t]$.

In (3), by $\phi_{\mathcal{F}}^{\tau,t}(\mathbf{z}_0)$ we refer to the time-t solution of the initial value problem

$$\begin{cases}\dot{\mathbf{x}}(t)=\mathcal{F}(\mathbf{x}(t))\\ \mathbf{x}(\tau)=\mathbf{z}_0.\end{cases}$$

Proof. Since $\widetilde{\mathbf{x}}(t)$, the polynomial approximation, is differentiable, it solves the initial value problem

$$\begin{cases}\dot{\widetilde{\mathbf{x}}}(t)=\mathcal{F}(\widetilde{\mathbf{x}}(t))+\left(\dot{\widetilde{\mathbf{x}}}(t)-\mathcal{F}(\widetilde{\mathbf{x}}(t))\right)\\ \dot{\widetilde{\mathbf{x}}}(t_n)=\mathbf{x}_n\end{cases}$$

over the time interval $[t_n,t_{n+1}]$. Let us consider the initial value problem with the same initial condition $\mathbf{x}_n$ as above, but with the correct vector field $\mathcal{F}$. Call $\mathbf{x}$ the exact solution of this second problem. Using (3) with $\mathcal{G}(\widetilde{\mathbf{x}}(t)):=\dot{\widetilde{\mathbf{x}}}(t)-\mathcal{F}(\widetilde{\mathbf{x}}(t))$, we can say that

$$\begin{aligned}\mathbf{x}(t_{n+1})-\widetilde{\mathbf{x}}(t_{n+1})&=\int_{t_n}^{t_{n+1}}\left.\frac{\partial\phi_{\mathcal{F}}^{t,t_{n+1}}(\mathbf{z}_0)}{\partial\mathbf{z}_0}\right|_{\mathbf{z}_0=\widetilde{\mathbf{x}}(t)}\left(\dot{\widetilde{\mathbf{x}}}(t)-\mathcal{F}(\widetilde{\mathbf{x}}(t))\right)dt\\ &=h\int_0^1\left.\frac{\partial\phi_{\mathcal{F}}^{t_n+sh,t_{n+1}}(\mathbf{z}_0)}{\partial\mathbf{z}_0}\right|_{\mathbf{z}_0=\widetilde{\mathbf{x}}(t_n+sh)}\left(\dot{\widetilde{\mathbf{x}}}(t_n+sh)-\mathcal{F}(\widetilde{\mathbf{x}}(t_n+sh))\right)dt.\end{aligned}$$

The integral on the right-hand side can be approximated with the Gauss-Legendre quadrature rule associated to the s nodes $c_1,\dots,c_s$, so to get

$$\mathbf{x}(t_{n+1})-\widetilde{\mathbf{x}}(t_{n+1})=\underbrace{h\sum_{i=1}^{s}\omega_i\left.\frac{\partial\phi_{\mathcal{F}}^{t_{n,i},t_{n+1}}(\mathbf{z}_0)}{\partial\mathbf{z}_0}\right|_{\mathbf{z}_0=\widetilde{\mathbf{x}}(t_{n,i})}\mathcal{G}(\widetilde{\mathbf{x}}(t_{n,i}))}_{(\#)}+\mathcal{O}(h^{2s+1}),$$

where $\omega_1, ..., \omega_s$ are the Gauss-Legendre quadrature weights, and $t_{n,i} = t_n + c_i h$. By the characterisation of the collocation polynomial $\widetilde{\mathbf{x}}$ in (2), we see that the term (#) vanishes, and hence we can conclude the proof. □

An example of Gauss-Legendre collocation method is the implicit midpoint method, with tableau

$$A = 1/2,\ b = 1,\ c = 1/2,$$

and which writes

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathcal{F}\left(\frac{\mathbf{x}_n + \mathbf{x}_{n+1}}{2}\right).$$

See [2, Chapter 3] for more details about collocation methods.

References

[1] Ernst Hairer, Syvert P Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*, volume 8. Springer, 1993.

[2] Arieh Iserles. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.

[3] Gerhard Wanner and Ernst Hairer. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, volume 375. Springer Berlin Heidelberg New York, 1996.