Numerical calculus

1. | Initial value problems

Definition 1. An initial-value problem is said to be *well-posed in the Hadamard sense* (or simply *well-posed*) if it has existence and uniqueness of solutions, and if it has continuous dependence on initial conditions and parameters.

One-step methods

Consider the ivp

$$\begin{cases} \mathbf{x}' = \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \tag{1}$$

For $n \in \mathbb{N} \cup \{0\}$ let $t_{n+1} := t_n + h$, where $h > 0$ is called *step size*. We would like to create a sequence $(\tilde{\mathbf{x}}_n)$ (*mesh-points*) that approximates (in some sense) $\mathbf{x}_n := \mathbf{x}(t_n)$ from a first iterate $\tilde{\mathbf{x}}_0 := \mathbf{x}_0$. In this section we will describe several algorithms that intend to do so. We will denote $\mathbf{f}_n := \mathbf{f}(t_n, \mathbf{x}_n)$ and $\tilde{\mathbf{f}}_n := \mathbf{f}(t_n, \tilde{\mathbf{x}}_n)$. Note that solving Eq. (1) is equivalent to solve the integral problem:

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(t, \mathbf{x}(s))\, \mathrm{d}s$$

Choosing different numerical-integration methods for approximating this latter integral will lead to different methods for solving the ivp.

Definition 2. A numerical method is called *explicit* if the n-th iterate can be computed directly in terms of some previous iterates. A method is called *implicit* if the n-th iterate depends implicitly on itself.

Definition 3. A *one-step method* $\boldsymbol{\Phi}$ for the approximation of Eq. (1) can be cast in the concise form

$$\tilde{\mathbf{x}}_{n+1} = \boldsymbol{\Phi}(t_n, \tilde{\mathbf{x}}_n, \tilde{\mathbf{x}}_{n+1}, \mathbf{f}, h) = \tilde{\mathbf{x}}_n + h\boldsymbol{\phi}(t_n, \tilde{\mathbf{x}}_n, \tilde{\mathbf{x}}_{n+1}, \mathbf{f}, h) \tag{2}$$

The remarkable fact is that the n-th iterate only depends on the previous one. The function $\boldsymbol{\phi}$ is called *incremental function*. From here we can define the *local truncation errors* as

$$\boldsymbol{\tau}_n(h) = \frac{\mathbf{x}_{n+1} - \mathbf{x}_n - h\boldsymbol{\phi}(t_n, \mathbf{x}_n, \mathbf{x}_{n+1}, \mathbf{f}, h)}{h}$$

We define $\tau(h)$ as:

$$\tau(h) = \sup_{n \geq 1} \|\boldsymbol{\tau}_n(h)\|$$

Finally, we define the *global truncation error* as:

$$\mathbf{e}_n = \mathbf{x}_n - \tilde{\mathbf{x}}_n$$

We can also define the iterates $\tilde{\mathbf{x}}_n^*$ as defined by:

$$\tilde{\mathbf{x}}_n^* = \mathbf{x}_n + h\boldsymbol{\phi}(t_n, \mathbf{x}_n, \mathbf{x}_{n+1}, \mathbf{f}, h)$$

Remark. In reality in Eq. (2) we should add a term of the form $h^q \boldsymbol{\varepsilon}_n K$ with $K > 0$, $q \in \mathbb{N}$ and $\|\boldsymbol{\varepsilon}\| \leq 1$ on account of the approximation errors due to the float-precision arithmetic. But from here on, we should omit it in order to simplify the notation.

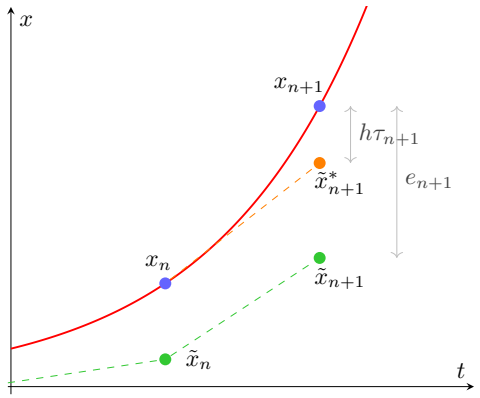

Figure 1: Geometrical interpretation of the local and global truncation errors

Definition 4 (Euler method). Consider the ivp of Eq. (1). The *forward Euler method* or *explicit Euler method* is defined as:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + h\tilde{\mathbf{f}}_n$$

The *backward Euler method* or *implicit Euler method* is defined as:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + h\tilde{\mathbf{f}}_{n+1}$$

Note that the forward method is explicit, whereas the backward method is *implicit*.

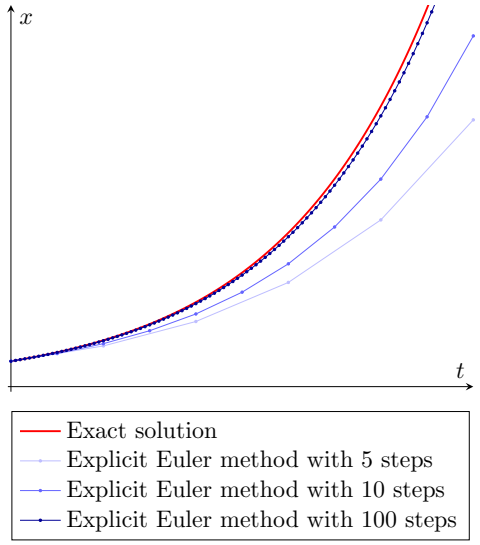

Figure 2: Explicit Euler method for approximating the ivp $\{x' = x, x(0) = 1\}$ with different number of steps.

Definition 5 (Trapezoidal method). Consider the ivp of Eq. (1). The *Trapezoidal method* is defined as:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + \frac{h}{2}\left(\tilde{\mathbf{f}}_n + \tilde{\mathbf{f}}_{n+1}\right)$$

Definition 6 (Heun method). Consider the ivp of Eq. (1). The *Heun method* is defined as:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + \frac{h}{2}\left(\tilde{\mathbf{f}}_n + \mathbf{f}(t_{n+1}, \tilde{\mathbf{x}}_n + h\tilde{\mathbf{f}}_n)\right)$$

Definition 7 (Taylor method). Consider the ivp of Eq. (1) and suppose that $\mathbf{f} \in \mathcal{C}^r(\mathbb{R} \times \mathbb{R}^d)$. The *Taylor method of order* r is the method constructed from the Taylor series of the solution $\mathbf{x}(t)$. Thus, the Taylor method of order r is:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + \sum_{k=1}^{r} \frac{h^k}{k!}\mathbf{x}_n^{(k)}$$

We should then substitute each unknown derivative $\mathbf{x}_n^{(k)}$ by a function of $\mathbf{f}_n$. For example the Taylor method of order 2 would be:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + h\tilde{\mathbf{f}}_n + \frac{h^2}{2}\left(\mathbf{f}_t(t_n, \tilde{\mathbf{x}}_n) + \mathbf{D}_2\mathbf{f}(\tilde{\mathbf{f}}_n)\right)$$

Note that the Taylor method of order 1 is precisely the 4 Euler method.

Definition 8. A one-step method for the approximation of Eq. (1) is said to be *consistent* if $\lim_{h\to 0} \tau(h) = 0$. Moreover, we say that the algorithm has *order of consistency* (or *order of accuracy*, or simply *order*) p if $\tau(h) = \mathrm{O}(h^p)$.

Definition 9. A one-step method for the approximation of Eq. (1) is *convergent* if

$$\lim_{h\to 0} \sup_{n\geq 1} \|\mathbf{e}_n\| = 0$$

Moreover, we say that the algorithm has *order of convergence* p if $\|\mathbf{e}_n\| = \mathrm{O}(h^p)$.

Remark. Note that in a consistent method the difference equation for the method approaches the ODE as the step size goes to zero, whereas in a convergent method is the solution to the difference equation that approaches the solution to the ODE as the step size goes to zero.

Theorem 10. Consider a consistent one-step explicit method such that its incremental function $\boldsymbol{\phi}$ is Lipschitz continuous (with constant L) with respect to $\mathbf{x}$. Then:

$$\|\mathbf{e}_{n+1}\| \leq \frac{\mathrm{e}^{L(t_{n+1}-t_0)} - 1}{L}\tau(h)$$

Proof.

$$\begin{aligned}
\|\mathbf{e}_{n+1}\| &\leq \left\|\mathbf{x}_{n+1} - \tilde{\mathbf{x}}^*_{n+1}\right\| + \left\|\tilde{\mathbf{x}}^*_{n+1} - \tilde{\mathbf{x}}_{n+1}\right\| \\
&\leq h\left\|\boldsymbol{\tau}_{n+1}(h)\right\| + \|\mathbf{e}_n\| + \\
&\qquad + h\left\|\boldsymbol{\phi}(t_n, \mathbf{x}_n, \mathbf{f}, h) - \boldsymbol{\phi}(t_n, \tilde{\mathbf{x}}_n, \mathbf{f}, h)\right\| \\
&\leq h\left\|\boldsymbol{\tau}_{n+1}(h)\right\| + (1 + hL)\|\mathbf{e}_n\|
\end{aligned}$$

Iterating the process (note that $\mathbf{e}_0 = \mathbf{0}$) we have:

$$\begin{aligned}
\|\mathbf{e}_{n+1}\| &\leq h[1 + (1 + hL) + \cdots + (1 + hL)^n]\tau(h) \\
&= \frac{(1 + hL)^{n+1} - 1}{L}\tau(h) \\
&\leq \frac{\mathrm{e}^{L(t_{n+1}-t_0)} - 1}{L}\tau(h)
\end{aligned}$$

where the last inequality follows from $1 + x \leq \mathrm{e}^x$. □

Corollary 11. Consider a one-step method with order of consistency p such that its incremental function $\boldsymbol{\phi}$ is Lipschitz continuous with respect to $\mathbf{x}$. Then, if $t_n \leq T$ for a fixed $T \in \mathbb{R}$, the convergence of the method has also order p.

Lemma 12. Euler method has order of consistency 1, whereas Heun method has order of consistency 2.

Proof. Using the Taylor series expansion of $\mathbf{x}(t)$ we have that:

$$\frac{\mathbf{x}(t+h) - \mathbf{x}(t) - h\mathbf{f}(t, \mathbf{x})}{h} = \frac{h\mathbf{x}''(t)}{2}$$

Hence, Euler method has order 1. For the Heun method we will describe a general procedure for constructing methods of arbitrary order. Let

$$\begin{gathered}
\mathbf{k}_1 = \mathbf{f}_n \quad \mathbf{k}_2 = \mathbf{f}(t_n + c_2 h, \mathbf{x}_n + h a_{21}\mathbf{k}_1) \\
\mathbf{x}_{n+1} = \mathbf{x}_n + h(b_1\mathbf{k}_1 + b_2\mathbf{k}_2) + \mathrm{O}\left(h^3\right)
\end{gathered}$$

Expanding $\mathbf{k}_2$ we have that:

$$\mathbf{k}_2 = \mathbf{f} + c_2 h\mathbf{f}_t + a_{21}h\mathbf{D}_2\mathbf{f}(\mathbf{k}_1) + \mathrm{O}\left(h^2\right)$$

So:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + (b_1 + b_2)h\mathbf{f} + h^2(b_2 c_2 \mathbf{f}_t + b_2 a_{21}\mathbf{D}_2\mathbf{f}(\mathbf{f})) + \mathrm{O}\left(h^3\right) \tag{3}$$

But from $\mathbf{x}' = \mathbf{f}(t, \mathbf{x})$ we have:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f} + \frac{h^2}{2}(\mathbf{f}_t + \mathbf{D}_2\mathbf{f}(\mathbf{f})) + \mathrm{O}\left(h^3\right) \tag{4}$$

Matching coefficients from Eqs. (3) and (4), we get the desired result. □

Remark. For a method of order s (see Theorem 14), just start with $r \geq s$ values $\mathbf{k}_1, \ldots, \mathbf{k}_r$ of the form:

$$\mathbf{k}_i = \mathbf{f}(t_n + c_s h, \mathbf{x}_n + h(a_{s1}\mathbf{k}_1 + \cdots + a_{i(i-1)}\mathbf{k}_{i-1}))$$

for $i \geq 2$ and $\mathbf{k}_1 = \mathbf{f}_n$, and impose:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{r} b_i\mathbf{k}_i + \mathrm{O}\left(h^{s+1}\right)$$

There are tables that determine the smallest r necessary for a given order s (see Table 1).

Runge-Kutta methods

Definition 13. The family of *s-stage Runge-Kutta methods* (or *RK methods*) is defined by

$$\boldsymbol{\phi}(t, \mathbf{x}, \mathbf{f}, h) = \mathbf{x} + h \sum_{i=1}^{s} b_i \mathbf{k}_i$$

where the stages $\mathbf{k}_i \in \mathbb{R}^d$ are the solutions to the coupled system of (generally nonlinear) equations

$$\mathbf{k}_i = \mathbf{f}\left(t + c_i h, \mathbf{x} + h \sum_{j=1}^{s} a_{ij} \mathbf{k}_j\right) \quad i = 1, \dots, s$$

where $c_i := \sum_{j=1}^{s} a_{ij}$ for $i = 1, \dots, s$. Denoting $\mathbf{c} = (c_i)$, $\mathbf{b} = (b_i)$ and $\mathbf{A} = (a_{ij})$ we can construct the *Butcher tableau*

$\mathbf{c}$	$\mathbf{A}$
	$\mathbf{b}^{\mathrm{T}}$

to summarize the information about the method. The method is explicit if $a_{ij} = 0 \ \forall j \geq i$. Otherwise, it is implicit.

Lemma 14. A Runge-Kutta method is consistent if and only if $\sum_{i=1}^{s} b_i = 1$. If moreover, $\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$, then it has order of consistency 2. And if the conditions $\sum_{i=1}^{s} b_i {c_i}^2 = \frac{1}{3}$ and $\sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} c_j = \frac{1}{6}$ are also satisfied, then the consistency is of order 3.

Proof. In the following equations we omit the evaluation at $(t_n, \mathbf{x}_n)$. On the one hand we have:

$$\begin{aligned}
\mathbf{x}' &= \mathbf{f} \\
\mathbf{x}'' &= \mathbf{f}_t + \mathbf{f}_{\mathbf{x}}\mathbf{f} =: \mathbf{F} \\
\mathbf{x}''' &= \mathbf{f}_{tt} + 2\mathbf{f}_{\mathbf{x}t}\mathbf{f} + \mathbf{f}_{\mathbf{x}\mathbf{x}}\mathbf{f}^2 + \mathbf{f}_{\mathbf{x}}(\mathbf{f}_t + \mathbf{f}_{\mathbf{x}}\mathbf{f}) =: \mathbf{G} + \mathbf{f}_{\mathbf{x}}\mathbf{F}
\end{aligned}$$

Note that here $\mathbf{f}_{\mathbf{x}\mathbf{x}} \in \mathcal{L}(\mathbb{R}^d, \mathcal{L}(\mathbb{R}^d, \mathbb{R}^d))$. That is, is a vector of matrices. And the vector product $\mathbf{f}_{\mathbf{x}\mathbf{x}}\mathbf{f}^2$ is done as follows: $(\mathbf{f}_{\mathbf{x}\mathbf{x}})_i \mathbf{f}$, for each $i = 1, \dots, d$, which result in d-column vectors that form a matrix that gets multiplied by $\mathbf{f}$. And on the other hand:

$$\begin{aligned}
\mathbf{k}_i &= \mathbf{f} + c_i h \mathbf{f}_t + \mathbf{f}_{\mathbf{x}}\left(h \sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right) + \frac{{c_i}^2 h^2}{2}\mathbf{f}_{tt} + \\
&+ c_i h^2 \mathbf{f}_{\mathbf{x}t}\left(\sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right) + \frac{h^2}{2}\mathbf{f}_{\mathbf{x}\mathbf{x}}\left(\sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right)^2 + \mathrm{O}\left(h^3\right) \\
&= \mathbf{f} + c_i h \mathbf{F} + \frac{{c_i}^2}{2} h^2 \mathbf{G} + h^2 \left(\sum_{j=1}^{s} a_{ij} c_j\right) \mathbf{f}_{\mathbf{x}}\mathbf{F} + \mathrm{O}\left(h^3\right)
\end{aligned}$$

Therefore:

$$\boldsymbol{\tau}_n(h) = \mathbf{f} + \frac{1}{2}h\mathbf{F} + \frac{1}{6}h^2\left(\mathbf{G} + \mathbf{f}_{\mathbf{x}}\mathbf{F}\right) + \mathrm{O}\left(h^3\right) - \sum_{i=1}^{s} b_i \mathbf{k}_i$$

Matching coefficients we get the desired result. □

Lemma 15. The consistency order p of an s-stage Runge-Kutta method is bounded by $p \leq 2s$. If the Runge-Kutta method is explicit, then $p \leq s$.

Remark. Looking at Table 1 we see why the RK4, i.e. the RK method with 4 stages, is so widely known.

Order	1	2	3	4	5	6	7	8
s	1	2	3	4	6	7	9	11

Table 1: Number of stages of an explicit RK method needed for a given order of consistency

Step-size control for Runge-Kutta methods

Theorem 16. Let $\mathbf{f} : [t_0, t_n] \times \mathbb{R}^d \to \mathbb{R}^d$ be a function of class $\mathcal{C}^{N+1}$ with respect to the second variable and let $\tilde{\mathbf{x}}(t)$ be the numerical solution to the ivp Eq. (1) obtained by a one-step method of order $p \leq N$ with step-size h. Then, $\tilde{\mathbf{x}}(t)$ has an asymptotic expansion of:

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) + \mathbf{e}_p(t)h^p + \cdots + \mathbf{e}_N(t)h^N + E_{N+1}(t,h)h^{N+1}$$

with $\mathbf{e}_k(t_0) = 0 \ \forall k \geq p$. This is valid $\forall t \in [t_0, t_n]$ and all $h > 0$. Moreover, the functions $\mathbf{e}_k$ are differentiable and independent of h and $\|E_{N+1}(t, \cdot)\|_\infty < \infty \ \forall t \in [t_0, t_n]$.

Theorem 17 (Richardson extrapolation). Consider the ivp of Eq. (1) and let $\tilde{\mathbf{x}}(t; h)$ be the numerical solution obtained by a one-step method of order $p \leq N$ with step-size h. Then:

$$\mathbf{x}(t) = \tilde{\mathbf{x}}(t; h/2) - \frac{\tilde{\mathbf{x}}(t; h) - \tilde{\mathbf{x}}(t; h/2)}{2^p - 1} + \mathrm{O}\left(h^{p+1}\right)$$

Proof. By Theorem 16 we have that:

$$\begin{aligned}
\mathbf{x}(t; h) &= \mathbf{x}(t) + \mathbf{e}_p(t)h^p + \mathrm{O}\left(h^{p+1}\right) \\
\mathbf{x}(t; h/2) &= \mathbf{x}(t) + \mathbf{e}_p(t)\left(\frac{h}{2}\right)^p + \mathrm{O}\left(h^{p+1}\right)
\end{aligned}$$

Subtracting the two equations we have:

$$\mathbf{e}_p(t)\left(\frac{h}{2}\right)^p = \frac{\tilde{\mathbf{x}}(t; h) - \tilde{\mathbf{x}}(t; h/2)}{2^p - 1} + \mathrm{O}\left(h^{p+1}\right)$$

□

Theorem 18 (Runge-Kutta-Fehlberg method). Consider two explicit RK methods of orders p and $p+1$ with incremental functions $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$ respectively such that their Butcher tableaus have the same (a_{ij}) coefficients (and therefore the same (c_{ij}) coefficients):

$\mathbf{c}$	$\mathbf{A}$
	$\mathbf{b}^{\mathrm{T}}$
	$\hat{\mathbf{b}}^{\mathrm{T}}$

These methods $\boldsymbol{\phi}$ and $\hat{\boldsymbol{\phi}}$ are called *embedded methods*[1]. Denote by $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ the numerical solutions using the respective incremental functions. Then, given a tolerance ϵ and an older step-size h, we would like to choose a new

[1] Usually the notation RK$p(q)s$ is used to refer for a method of order p with an embedded method of order $q < p$ and a total of s stages.

step size h_{new} for which our approximate solutions differ no more than ϵ between them. At the time t_n we have:

$$\tilde{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + h\boldsymbol{\phi}(t_n, \tilde{\mathbf{x}}_n, \mathbf{f}, h)$$
$$\hat{\mathbf{x}}_{n+1} = \tilde{\mathbf{x}}_n + h\hat{\boldsymbol{\phi}}(t_n, \hat{\mathbf{x}}_n, \mathbf{f}, h)$$

To obtain this we have to choose the new step-size

$$h_{\text{new}} \simeq h \sqrt[p+1]{\frac{\epsilon}{\|\hat{\mathbf{x}}(t_n + h) - \tilde{\mathbf{x}}(t_n + h)\|}} \tag{5}$$

If this new step-size was not successful, i.e. we have

$$\|\hat{\mathbf{x}}(t_{n+1} + h_{\text{new}}) - \tilde{\mathbf{x}}(t_{n+1} + h_{\text{new}})\| > \epsilon$$

we will have to repeat the last step with another step-size $h^*_{\text{new}} < h_{\text{new}}$.

Proof. We assume that the n-th iteration was successful, i.e.:

$$\|\hat{\mathbf{x}}(t_n + h) - \tilde{\mathbf{x}}(t_n + h)\| \leq \epsilon$$

From the hypothesis and Theorem 16 we have:

$$\|\hat{\mathbf{x}}(t_n + h) - \tilde{\mathbf{x}}(t_n + h)\| \leq \mathbf{c}(\mathbf{x}(t_n))h^{p+1}$$

Moreover, up to errors of first order, we have $\mathbf{c}(\mathbf{x}(t_n)) \approx \mathbf{c}(\mathbf{x}(t_n + h))$. Finally, imposing

$$\left\|\mathbf{c}(\mathbf{x}(t_n + h)){h_{\text{new}}}^{p+1}\right\| \lesssim \epsilon$$

yields to:

$$\|\hat{\mathbf{x}}(t_n + h) - \tilde{\mathbf{x}}(t_n + h)\| \left(\frac{h_{\text{new}}}{h}\right)^{p+1} \lesssim \epsilon$$

□

Remark. Note that there exist RK embedded methods by the following argument. Start with a RK method of order $p+1$ that has s stages. Then we can construct a RK method of order p with s stages by copying the coefficients $\mathbf{A}$ and $\mathbf{c}$ and adjusting the coefficients $\mathbf{b}$ properly to make it "less" consistent.

Remark. In practice in order to avoid many unsuccessful steps, instead of the new step in Eq. (5) we use the following:

$$h_{\text{new}} \simeq \alpha h \sqrt[p+1]{\frac{\epsilon}{\|\hat{\mathbf{x}}(t_n + h) - \tilde{\mathbf{x}}(t_n + h)\|}}$$

with $\alpha \simeq 0.9$. Furthermore, in order to avoid rapid oscillations of the step-size, h should not, however, be changed by more than a factor of 2 to 5 from one step to the next.

Stability of Runge-Kutta methods

Definition 19. Consider a RK method applied to the ivp $y' = \lambda y$. We can express it as:

$$\tilde{y}_{n+1} = g(h\lambda)\tilde{y}_n$$

for some function $g : \mathbb{R} \to \mathbb{R}$. This function is called *stability function*. Given h and λ, the method is said to be *stable* if $|g(h\lambda)| \leq 1$ and *absolutely stable* if $|g(h\lambda)| < 1$.

Definition 20. Consider a RK method with stability function g. We define the *stability region* of the method as the set:

$$\mathcal{A} := \{z \in \mathbb{C} : |g(z)| < 1\}$$

We say that the method is *A-stable* (or *unconditionally absolutely stable*) if $\{\text{Re}(z) < 0\} \subseteq \mathcal{A}$. Otherwise, we say that the method is *conditionally absolutely stable*

Remark. The motivation behind this definition of stability is on the *stiff equations*, which are differential equations for which certain numerical methods for solving the equation are numerically unstable, unless the step size is taken extremely small. A-stable methods do not exhibit these instability problems.

Theorem 21 (Lax theorem). Consider a consistent and stable RK method. Then, the method is convergent.

Proof. We will prove it only for the *test problem* $y' = \lambda y$. We have:

$$e_{n+1} = y_{n+1} - \tilde{y}_{n+1} = y_{n+1} - g(h\lambda)\tilde{y}_n =$$
$$= y_{n+1} - g(h\lambda)(y_n - e_n) = h\tau_n(h) + g(h\lambda)e_n$$

Iterating the process and taking norms we have:

$$|e_n| \leq \sum_{j=0}^{n} |g(h\lambda)|^j h|\tau_{n-j}(h)| \leq nh\tau(h) \xrightarrow{h\to 0} 0$$

because $nh = t_n - t_0$ is bounded. □

Multistep method

Definition 22. A k-step method is a method that uses the previous k steps to compute the next step.

Definition 23 (Linear multistep method). Consider the ivp of Eq. (1). A *linear k-step method* is a method of the form:

$$\sum_{j=0}^{k} \alpha_j \tilde{\mathbf{y}}_{n+j} = h \sum_{j=0}^{k} \beta_j \tilde{\mathbf{f}}_{n+j} \tag{6}$$

that computes the $(n+k)$-th iterate from the previous k iterates. Here $\alpha_j, \beta_j \in \mathbb{R}$ are such that $\alpha_k \neq 0$ and ${\alpha_0}^2 + {\beta_0}^2 \neq 0$. Observe that we need k initial values in order to use the method. Finally, note that if $\beta_k = 0$ then the method is explicit.

Remark. Since in practice we only have one initial value, we can use a one-step method to compute the first k iterates and then use the k-step method.

Definition 24. Consider the multistep method of Eq. (6). We define the *first and second characteristic polynomials* of the method as:

$$\rho(z) = \sum_{j=0}^{k} \alpha_j z^j \qquad \sigma(z) = \sum_{j=0}^{k} \beta_j z^j$$

Definition 25. A linear k-step method is said to be *zero-stable* if there is a constant $C > 0$ such that for every $N \in \mathbb{N}$ sufficiently large and for any two different sets

of initial data $\tilde{\mathbf{y}}_0, \dots, \tilde{\mathbf{y}}_{k-1}$ and $\hat{\mathbf{y}}_0, \dots, \hat{\mathbf{y}}_{k-1}$, the two respective sequences $(\tilde{\mathbf{y}}_n)_{0\le n\le N}$ and $(\hat{\mathbf{y}}_n)_{0\le n\le N}$ of iterates satisfy

$$\max_{0\le n\le N} \|\tilde{\mathbf{y}}_n - \hat{\mathbf{y}}_n\| \le C \max_{0\le n\le k-1} \|\tilde{\mathbf{y}}_n - \hat{\mathbf{y}}_n\|$$

as $h \to 0$.

Definition 26. A linear k-step method satisfies the *root condition* if all zeros of its first characteristic polynomial $\rho(z)$ lie inside the closed unit disc, and every zero that lies on the unit circle is simple.

Theorem 27. Consider the ivp of Eq. (1) and suppose that $\mathbf{f}$ is Lipschitz continuous. Then, a linear k-step method is zero-stable if and only if it satisfies the root condition.

Remark. This theorem implies that zero-stability of a multistep method can be determined by merely considering its behavior when applied to the trivial differential equation $y' = 0$. It is for this reason that it is called *zero*-stability.

Definition 28 (Adams method). Consider the ivp of Eq. (1). The *Adams method* is a linear multistep method of the form:

$$\tilde{\mathbf{y}}_{n+k} = \tilde{\mathbf{y}}_{n+k-1} + h\sum_{j=0}^{k} \beta_j \tilde{\mathbf{f}}_{n+j} \tag{7}$$

If $\beta_k = 0$ then the method is explicit, and it is called *Adams-Bashforth method*. If $\beta_k \neq 0$ then the method is implicit, and it is called *Adams-Moulton method*. The coefficients β_j are found by integrating the Lagrange interpolating polynomial between t_{n+k-1} and t_{n+k} constructed from the nodes $(t_{n+j}, \tilde{\mathbf{f}}_{n+j})$ for $j = 0, \dots, k-1$, for the Adams-Bashforth method, and for $j = 0, \dots, k$ for the Adams-Moulton method. That is, the respective incremental functions are given by:

$$\phi_{\text{AB}} = \int_{t_{n+k-1}}^{t_{n+k}} \sum_{j=0}^{k-1} \mathbf{f}_{n+j} \prod_{\substack{i=0\\ i\neq j}}^{k-1} \frac{t - t_{n+i}}{t_{n+j} - t_{n+i}}\, \mathrm{d}t$$

$$\phi_{\text{AM}} = \int_{t_{n+k-1}}^{t_{n+k}} \sum_{j=0}^{k} \mathbf{f}_{n+j} \prod_{\substack{i=0\\ i\neq j}}^{k} \frac{t - t_{n+i}}{t_{n+j} - t_{n+i}}\, \mathrm{d}t$$

In the following table we expose the first three Adams' incremental functions for the explicit and implicit methods:

k	Adams-Bashforth	Adams-Moulton
1	$\mathbf{f}_n$	$\mathbf{f}_{n+1}$
2	$\frac{3\mathbf{f}_{n+1} - \mathbf{f}_n}{2}$	$\frac{\mathbf{f}_{n+1} + \mathbf{f}_n}{2}$
3	$\frac{23\mathbf{f}_{n+1} - 16\mathbf{f}_n + 5\mathbf{f}_{n-1}}{12}$	$\frac{5\mathbf{f}_{n+2} + 8\mathbf{f}_{n+1} - \mathbf{f}_n}{12}$

Definition 29. Consider the k-step method of Eq. (6). We define the *local truncation error* of the method as:

$$\boldsymbol{\tau}_n(h) = \frac{\sum_{j=0}^{k}[\alpha_j \mathbf{y}_{n+j} - h\beta_j \mathbf{f}_{n+j}]}{h}$$

We define $\tau(h)$ as:

$$\tau(h) = \sup_{n\ge 1} \|\boldsymbol{\tau}_n(h)\|$$

We say that the method is *consistent* if $\lim_{h\to 0} \tau(h) = 0$. Moreover, we say that the algorithm has *order of consistency* or *order of accuracy* p if $\tau(h) = \mathrm{O}(h^p)$.

Remark. The global error of the method and the convergence of it are the same as in the one-step case.

Proposition 30. The Adams-Bashforth k-step method has order of consistency k, whereas the Adams-Moulton method has order of consistency $k+1$.

Theorem 31. A necessary condition for the convergence of the linear multistep method of Eq. (6) is that it has to be zero-stable and consistent.

Theorem 32 (Dahlquist's theorem). Let $\mathbf{f}$ be a Lipschitz continuous function and consider the multistep method of Eq. (6). Suppose the method is consistent. Then, the method is zero-stable if and only if it is convergent. Moreover, if the solution $\mathbf{y}$ is of class $\mathcal{C}^{p+1}$ and the consistency error is $\mathrm{O}(h^p)$, then the global error is $\mathrm{O}(h^p)$.

2. | Nonlinear systems of equations

Newton method

Definition 33 (Newton method). Let $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^d$ be a differentiable field. We would like to find the solutions of $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. The *Newton method* is a recurrence of the form:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{DF}^{-1}(\mathbf{x}_n)\mathbf{F}(\mathbf{x}_n)$$

which starts with an initial guess $\mathbf{x}_0$. Rather than actually computing the inverse of the Jacobian matrix, one may save time and increase numerical stability by solving the system of linear equations

$$\mathbf{DF}(\mathbf{x}_n)(\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{F}(\mathbf{x}_n)$$

for the unknown $\mathbf{x}_{n+1} - \mathbf{x}_n$.

Lemma 34. Let $C \subseteq \mathbb{R}^d$ be an open convex set and $\mathbf{F} \in \mathcal{C}^0(C)$ be such that $\mathbf{DF}(\mathbf{x})$ exists and satisfies:

$$\|\mathbf{DF}(\mathbf{x}) - \mathbf{DF}(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|$$

for some $L > 0$ and for all $\mathbf{x}, \mathbf{y} \in C$. Then, $\forall \mathbf{x}, \mathbf{y} \in C$ we have:

$$\|\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{y}) - \mathbf{DF}(\mathbf{y})(\mathbf{x} - \mathbf{y})\| \le \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

Proof. Consider $\boldsymbol{\varphi} : [0,1] \to \mathbb{R}^d$ defined by $\boldsymbol{\varphi}(t) = \mathbf{F}(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))$. Then, $\boldsymbol{\varphi}$ is differentiable,

$$\boldsymbol{\varphi}'(t) = \mathbf{DF}(\mathbf{y} + t(\mathbf{x} - \mathbf{y}))(\mathbf{x} - \mathbf{y})$$

and satisfies:

$$\begin{aligned}\|\boldsymbol{\varphi}'(t) - \boldsymbol{\varphi}(0)\| &\le \|\mathbf{DF}(\mathbf{y} + t(\mathbf{x} - \mathbf{y})) - \mathbf{DF}(\mathbf{y})\|\,\|\mathbf{x} - \mathbf{y}\| \\ &\le Lt\|\mathbf{x} - \mathbf{y}\|^2\end{aligned}$$

Moreover:

$$\Delta := \mathbf{F}(x) - \mathbf{F}(y) - \mathbf{DF}(\mathbf{y})(\mathbf{x}-\mathbf{y}) =$$
$$= \varphi(1) - \varphi(0) - \varphi'(0) = \int_0^1 \varphi'(t) - \varphi'(0)\,\mathrm{d}t$$

Therefore:

$$\|\Delta\| \leq \int_0^1 Lt\,\|\mathbf{x}-\mathbf{y}\|^2\,\mathrm{d}t = \frac{L}{2}\,\|\mathbf{x}-\mathbf{y}\|^2$$

□

Theorem 35. Let $C \subseteq \mathbb{R}^d$ be an open convex set and $\mathbf{F} \in \mathcal{C}^1(C)$ with a zero $\mathbf{x}^* \in C$. Suppose that $\mathbf{DF}(\mathbf{x}^*)$ is invertible and satisfies:

1. $\left\|\mathbf{DF}^{-1}(\mathbf{x}^*)\right\| \leq M$ for some $M > 0$.
2. $\|\mathbf{DF}(\mathbf{x}) - \mathbf{DF}(\mathbf{y})\| \leq L\,\|\mathbf{x}-\mathbf{y}\|$ for some $L > 0$ and for all $\mathbf{x}, \mathbf{y} \in B(\mathbf{x}^*, R)$, for some R.

Then, $\exists r > 0$ such that $\forall \mathbf{x}_0 \in B(\mathbf{x}^*, r)$, the Newton method sequence is well-defined and converges quadratically (at least) to $\mathbf{x}^*$.

Proof. Let $r := \min\left(R, \frac{1}{2ML}\right)$. We prove first that if $\mathbf{y} \in B(\mathbf{x}^*, r)$, then $\mathbf{DF}(\mathbf{y})$ is invertible. Recall that from **??**, if $\|\mathbf{A}\| < 1$, then $\mathbf{I}+\mathbf{A}$ is invertible and $\left\|(\mathbf{I}+\mathbf{A})^{-1}\right\| \leq \frac{1}{1-\|\mathbf{A}\|}$. With that in mind, if $\mathbf{A} := \mathbf{DF}^{-1}(\mathbf{x}^*)\mathbf{DF}(\mathbf{y}) - \mathbf{I}_d$, then:

$$\|\mathbf{A}\| \leq \left\|\mathbf{DF}^{-1}(\mathbf{x}^*)\right\|\,\|\mathbf{DF}(\mathbf{y}) - \mathbf{DF}(\mathbf{x}^*)\| \leq MLr \leq \frac{1}{2}$$

Hence, $\mathbf{A} + \mathbf{I}_d = \mathbf{DF}^{-1}(\mathbf{x}^*)\mathbf{DF}(\mathbf{y})$ is invertible and therefore so is $\mathbf{DF}(\mathbf{y})$. Furthermore:

$$\left\|\mathbf{DF}^{-1}(\mathbf{y})\right\| \leq \frac{\left\|\mathbf{DF}^{-1}(\mathbf{x}^*)\right\|}{1-\|\mathbf{A}\|} \leq 2M \tag{8}$$

Now we prove by induction that any term $\mathbf{x}_n$ on the sequence is well-defined and satisfies:

$$\|\mathbf{x}_{n+1} - \mathbf{x}^*\| \leq ML\,\|\mathbf{x}_n - \mathbf{x}^*\|^2$$

By hypothesis, we know that $\mathbf{x}_0 \in B(\mathbf{x}^*, r)$, so $\mathbf{DF}(\mathbf{x}_0)$ is invertible and $\mathbf{x}_1 = \mathbf{x}_0 - \mathbf{DF}^{-1}(\mathbf{x}_0)\mathbf{F}(\mathbf{x}_0)$ is well-defined. Moreover:

$$\begin{aligned}
\|\mathbf{x}_1 - \mathbf{x}^*\| &= \left\|\mathbf{x}_0 - \mathbf{x}^* - \mathbf{DF}^{-1}(\mathbf{x}_0)[\mathbf{F}(\mathbf{x}_0) - \mathbf{F}(\mathbf{x}^*)]\right\| \\
&= \left\|\mathbf{DF}^{-1}(\mathbf{x}_0)[\mathbf{F}(\mathbf{x}^*) - \mathbf{F}(\mathbf{x}_0) - \right. \\
&\qquad \left. - \mathbf{DF}(\mathbf{x}_0)(\mathbf{x}^* - \mathbf{x}_0)]\right\| \\
&\leq 2M\frac{L}{2}\,\|\mathbf{x}_0 - \mathbf{x}^*\|^2 \\
&\leq \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|}{2}
\end{aligned}$$

where in the penultimate step we used Eq. (8) and Theorem 34. Thus, $\mathbf{x}_1 \in B(\mathbf{x}^*, r)$ and so $\mathbf{x}_2$ is well-defined. Recursively, we prove that $\mathbf{x}_{n+1}$ is well-defined and:

$$\|\mathbf{x}_{n+1} - \mathbf{x}^*\| \leq \frac{\|\mathbf{x}_n - \mathbf{x}^*\|}{2} \leq \cdots \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|}{2^{n+1}} \xrightarrow{n\to\infty} 0$$

So the limit exists, and it is $\mathbf{x}^*$. □

Quasi-Newton methods

Definition 36. A *quasi-Newton method* for finding the zero of a function $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^d$ is a method that uses an approximation $\mathbf{DF}_n$ of the Jacobian $\mathbf{DF}(\mathbf{x}_n)$ to compute the next iterate $\mathbf{x}_{n+1}$, instead of computing the Jacobian directly, as in Newton's method. Once solved the system $\mathbf{DF}_n\mathbf{y} = -\mathbf{F}(x_n)$, the next iterate in a quasi-Newton method is given by $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n\mathbf{y}$, where α_n is a *damping parameter*.

Remark. Computing the Jacobian is a difficult and expensive operation. Broyden proposed a method that computes the whole Jacobian only at the first iteration and does rank-one updates at other iterations.

Definition 37 (Broyden's method). Let $\mathbf{F} : \mathbb{R}^d \to \mathbb{R}^d$. The *Broyden's method* is *secant-like method* which uses a recurrence for the Jacobian matrix of the form:

$$\mathbf{DF}_{n+1} = \mathbf{DF}_n + \frac{\mathbf{F}_n - \mathbf{DF}_n\Delta\mathbf{x}_n}{\|\Delta\mathbf{x}_n\|^2}(\Delta\mathbf{x}_n)^{\mathrm{T}}$$

where $\Delta\mathbf{x}_n = \mathbf{x}_{n+1} - \mathbf{x}_n$ and $\mathbf{F}_n = \mathbf{F}(\mathbf{x}_n)$. We then proceed with the Newton method:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - {\mathbf{DF}_n}^{-1}\mathbf{F}_n$$

A variant to the Broyen's method for computing directly the inverse of the Jacobian matrix is:

$${\mathbf{DF}_{n+1}}^{-1} = {\mathbf{DF}_n}^{-1} + \frac{\Delta\mathbf{x}_n - {\mathbf{DF}_n}^{-1}\Delta\mathbf{F}_n}{(\Delta\mathbf{x}_n)^{\mathrm{T}}{\mathbf{DF}_n}^{-1}\Delta\mathbf{F}_n}(\Delta\mathbf{x}_n)^{\mathrm{T}}{\mathbf{DF}_n}^{-1}$$

where $\Delta\mathbf{F}_n = \mathbf{F}_{n+1} - \mathbf{F}_n$.

Remark. Another alternative for the computation of the Jacobian matrix would be to use the finite difference method for each column of the matrix:

$$\mathbf{D}_j\mathbf{F}(\mathbf{x}_n) \simeq \frac{\mathbf{F}(\mathbf{x}_n + h\mathbf{e}_j) - \mathbf{F}(\mathbf{x}_n)}{h}$$

Optimization

Definition 38 (Descent method). Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. We want to find the minimum of f, or equivalently, the zeros of ∇f. A *descent method* is a method for finding the minimum of f using the following iteration:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n\mathbf{d}_n$$

where α_n is a *step size* and $\mathbf{d}_n$ is a *descent direction*, i.e. satisfying ${\mathbf{d}_n}^{\mathrm{T}}\nabla f(\mathbf{x}_n) < 0$ whenever $\nabla f(\mathbf{x}_n) \neq \mathbf{0}$ and $\mathbf{d}_n = \mathbf{0}$ otherwise. Some of the most common descent methods are:

- *Newton method*: $\mathbf{d}_n = -(\mathbf{H}f)^{-1}(\mathbf{x}_n)\nabla f(\mathbf{x}_n)$
- *Inexact Newton method*: $\mathbf{d}_n = -{\mathbf{B}_n}^{-1}(\mathbf{x}_n)\nabla f(\mathbf{x}_n)$, where $\mathbf{B}_n$ is an approximation of the Hessian matrix $\mathbf{H}f(\mathbf{x}_n)$.

- *Steepest descent* or *Gradient descent*: $\mathbf{d}_n = -\boldsymbol{\nabla} f(\mathbf{x}_n)$.
- *Conjugate gradient method*: $\mathbf{d}_n = -\boldsymbol{\nabla} f(\mathbf{x}_n) + \beta_n \mathbf{d}_{n-1}$, where β_n is a parameter chosen such that the directions $(\mathbf{d}_n)$ are pairwise conjugate by the Hessian matrix $\mathbf{H}f(\mathbf{x}_n)$, that is, ${\mathbf{d}_\ell}^{\mathrm{T}} \mathbf{H} f(\mathbf{x}_n) \mathbf{d}_k = 0$ for $\ell, k \leq n$.

In order to find a suitable step size α_n, we need to solve the following optimization problem:

$$\text{minimize } \phi(\alpha) = f(\mathbf{x}_n + \alpha \mathbf{d}_n)$$

Remark. In practice, this latter problem is solved using a *line search method*. For $n \geq 1$, if $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$, then α_n is accepted, and we choose $\alpha_{n+1} = \alpha_n$. Otherwise, $\alpha_n \leftarrow \alpha_n/2$ and we repeat the proces until the difference between $f(\mathbf{x}_{n+1})$ and $f(\mathbf{x}_n)$ is sufficiently small. We shall start with α_0 sufficiently large.

Definition 39 (Broyden-Fletcher-Goldfarb-Shanno method). Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function. The *Broyden-Fletcher-Goldfarb-Shanno method* (*BFGS method*) is a quasi-Newton descent method for finding the minimum of f using an approximation of the Hessian matrix. The algorithm is as follows. Start with an approximation $\mathbf{B}_0$ of $\mathbf{H}f(\mathbf{x}_0)$. Then, for $n \geq 0$:

1. Solve $\mathbf{B}_n \mathbf{d}_n = -\boldsymbol{\nabla} f(\mathbf{x}_n)$.
2. Perform a line search to find α_n.
3. Update $\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n \mathbf{d}_n$.
4. Define $\mathbf{y}_n = \boldsymbol{\nabla} f(\mathbf{x}_{n+1}) - \boldsymbol{\nabla} f(\mathbf{x}_n)$.
5. Update $\mathbf{B}_n$:

$$\mathbf{B}_{n+1} = \mathbf{B}_n + \frac{\mathbf{y}_n {\mathbf{y}_n}^{\mathrm{T}}}{\alpha_n {\mathbf{y}_n}^{\mathrm{T}} \mathbf{d}_n} - \frac{\mathbf{B}_n \mathbf{d}_n {\mathbf{d}_n}^{\mathrm{T}} {\mathbf{B}_n}^{\mathrm{T}}}{{\mathbf{d}_n}^{\mathrm{T}} \mathbf{B}_n \mathbf{d}_n}$$

Remark. There is also a variant of the BFGS method that computes recursively an approximation of the inverse of the Hessian matrix.

3. | Boundary value problems

Shooting method

Definition 40 (Shooting method). Suppose we want to solve the boundary value problem:

$$\begin{cases} \mathbf{x}' = \mathbf{f}(t, \mathbf{x}) \\ \mathbf{r}(\mathbf{x}(t_0), \mathbf{x}(t_1)) = \mathbf{0} \end{cases} \tag{9}$$

where $\mathbf{r} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a function that defines the boundary conditions. Let $\mathbf{x}(t; t_0, \mathbf{s})$ be the flow of the ivp:

$$\begin{cases} \mathbf{x}' = \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x}(t_0) = \mathbf{s} \end{cases}$$

If

$$\mathbf{r}(\mathbf{s}, \mathbf{x}(t_1; t_0, \mathbf{s})) = 0 \tag{10}$$

then $\mathbf{x}(t; t_0, \mathbf{s})$ will also be a solution to the boundary value problem. The *shooting method* is the process of solving the initial value problem for many values of $\mathbf{s}$ until one finds the solution $\mathbf{x}(t; t_0, \mathbf{s})$ that satisfies the desired boundary conditions of Eq. (10). That is, the solutions $\mathbf{s}$ correspond to roots of:

$$\mathbf{F}(s) := \mathbf{r}(\mathbf{s}, \mathbf{x}(t_1; t_0, \mathbf{s}))$$

Remark. Given several initial guesses, we can use interpolation with these nodes, find the root of the interpolating polynomial and use it as the new guess. Alternatively, one can use the Newton's method or a quasi-Newton method for finding a root of $\mathbf{F}$.

Multiple shooting method

Definition 41 (Multiple shooting method). Suppose we want to solve the problem of Eq. (9) and consider the partition of the time-interval of integration $t_0 < t_1 < \cdots < t_N$. The *multiple shooting method* starts by guessing the solution $\mathbf{s}_k$ of the BVP at t_k, $k = 0, \ldots, N$. Now let, $\mathbf{x}(t; t_k, \mathbf{x}_k)$ be the flow of the ivp:

$$\begin{cases} \mathbf{x}' = \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x}(t_k) = \mathbf{x}_k \end{cases}$$

All these solutions can be pieced together to form a continuous trajectory if the functions $\mathbf{x}(t; t_k, \mathbf{s}_k)$ match at the grid points $t_1, \ldots, t_{N-1}$. Thus, solutions of the boundary value problem correspond to solutions of the following system of N equations:

$$\begin{cases} \mathbf{x}(t_1; t_0, \mathbf{s}_0) = \mathbf{s}_1 \\ \vdots \\ \mathbf{x}(t_N; t_{N-1}, \mathbf{s}_{N-1}) = \mathbf{s}_N \end{cases}$$

Finite difference method

Definition 42. Suppose we want to solve the BVP:

$$\begin{cases} x'' + \lambda(t) x' + \mu(t) x = f(t, x) \\ x(a) = \alpha \\ x(b) = \beta \end{cases} \tag{11}$$

Consider an equally-spaced partition of the time-interval of integration $t_n = t_0 + kn$, $k = 0, \ldots, N$, with $t_0 = a$ and $t_N = b$. The *finite difference method* starts by approximating the derivatives of x by finite differences (usually centered derivatives). For example, the centered derivatives of orders 1 and 2 are:

$$x'(t_n) \simeq \frac{x_{n+1} - x_{n-1}}{2h} \qquad x''(t_n) \simeq \frac{x_{n+1} - 2x_n + x_{n-1}}{h^2}$$

with error terms $\left\| x^{(3)} \right\|_\infty \frac{h^2}{6}$ and $\left\| x^{(4)} \right\|_\infty \frac{h^2}{12}$, respectively, where $x_n := x(t_n)$. We then solve the resulting iterative system of equations

$$\begin{cases} \dfrac{x_{n+1} - 2x_n + x_{n-1}}{h^2} + \lambda_n \dfrac{x_{n+1} - x_{n-1}}{2h} + \mu_n x_n = f(t_n, x_n) \\ x_0 = \alpha \\ x_N = \beta \end{cases}$$

which can be concise in a matrix form:

$$\begin{pmatrix} -2+M & 1+L & 0 & \cdots & 0 \\ 1-L & -2+M & 1+L & \ddots & \vdots \\ 0 & 1-L & -2+M & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1+L \\ 0 & \cdots & 0 & 1-L & -2+M \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} h^2 f(t_1, x_1) - \alpha(1-L) \\ h^2 f(t_2, x_2) \\ \vdots \\ h^2 f(t_{N-2}, x_{N-2}) \\ h^2 f(t_{N-1}, x_{N-1}) - \beta(1+L) \end{pmatrix}$$

where $L := \frac{\lambda_n h}{2}$, $M := \mu_n h^2$, $\lambda_n := \lambda(t_n)$ and $\mu_n := \mu(t_n)$.

4. | Numerical linear algebra

Singular value decomposition

Lemma 43. Let $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$. Then, all the eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ are non-negative.

Proof. Assume $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, with $\mathbf{v}$ unitary. Then:

$$\lambda = \langle \lambda\mathbf{v}, \mathbf{v}\rangle = \left\langle \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{v}, \mathbf{v}\right\rangle = \langle \mathbf{A}\mathbf{v}, \mathbf{A}\mathbf{v}\rangle = \|\mathbf{A}\mathbf{v}\|^2 \geq 0$$

□

Definition 44 (Singular value). Let $m \geq n$ and $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$. The *singular values* of $\mathbf{A}$ are the square roots of the eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, which are real and non-negative by 43.

Theorem 45 (Singular value decomposition). Let $m \geq n$ and $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$. Then there exist orthogonal matrices $\mathbf{V} \in \mathcal{O}_n(\mathbb{R})$, $\mathbf{U} \in \mathcal{M}_{m\times n}(\mathbb{R})$ and a diagonal matrix $\mathbf{\Sigma} \in \mathcal{M}_n(\mathbb{R})$ such that:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \dots, \sigma_n)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ are the singular values of $\mathbf{A}$. A decomposition of this form is called *singular value decomposition* (*SVD*) of $\mathbf{A}$. The columns of $\mathbf{U}$ are called *left singular vectors*, while the columns of $\mathbf{V}$ are called *right singular vectors*. Writing $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ and $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$, we have another expression for the singular value decomposition:

$$\mathbf{A} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i{}^{\mathrm{T}} \tag{12}$$

Proof. For simplicity we assume $\sigma_n > 0$ and $m \geq n$. From linear algebra (check **??**) we know that since $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is symmetric, it admits a decomposition of the form $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{\mathrm{T}}$, where $\mathbf{\Lambda}$ is diagonal and $\mathbf{V} \in \mathcal{O}_n(\mathbb{R})$. Thus, we have that $(\mathbf{A}\mathbf{V})^{\mathrm{T}}(\mathbf{A}\mathbf{V}) = \mathbf{\Lambda} =: \mathbf{\Sigma}^2$ is diagonal and so $\mathbf{U} := \mathbf{A}\mathbf{V}\mathbf{\Sigma}^{-1}$ is orthonormal. □

Remark. From here one, we will assume that the singular values are ordered in decreasing order. Thus, σ_1 will always be the largest singular value and σ_n the smallest.

Remark. Note that the SVD is not unique, even though having the same ordering of the singular values. For example, we can replace $\mathbf{u}_i$ and $\mathbf{v}_i$ by $-\mathbf{u}_i$ and $-\mathbf{v}_i$ in Eq. (12). The singular values, on the other hand, are unique.

Corollary 46. Let $m \geq n$, $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$ and consider a SVD $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$. Then:

1. The columns of V are the eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.
2. The columns of U are the eigenvectors of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$.
3. If $\mathbf{A}$ is symmetric, then $\sigma_i = |\lambda_i|$, $\forall \lambda_i \in \sigma(\mathbf{A})$.

Proof. The third property is easy, and the first and second one are similar, so we only prove the second one. Note that from the identity $\mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$ we have $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$, for $i = 1, \dots, n$. Similarly, from $\mathbf{A}^{\mathrm{T}}\mathbf{U} = \mathbf{V}\mathbf{\Sigma}$ we have $\mathbf{A}^{\mathrm{T}}\mathbf{u}_i = \sigma_i \mathbf{v}_i$, for $i = 1, \dots, n$. Thus:

$$\mathbf{A}\mathbf{A}^{\mathrm{T}}\mathbf{u}_i = \sigma_i \mathbf{A}\mathbf{v}_i = {\sigma_i}^2 \mathbf{u}_i$$

□

Proposition 47. Let $m \geq n$ and $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$. Then, $\|\mathbf{A}\|_2 = \sigma_1$ and $\left\|\mathbf{A}^{-1}\right\|_2 = \frac{1}{\sigma_n}$.

Proof. Let $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ be a SVD of $\mathbf{A}$ and $\mathbf{x} \in \mathbb{R}^n$ be such that $\|\mathbf{x}\|_2 = 1$. We know that if $\mathbf{y} = \mathbf{V}^{\mathrm{T}}\mathbf{x}$, then $\|\mathbf{y}\|_2 = 1$. Thus:

$$\|\mathbf{A}\mathbf{x}\|_2{}^2 = \left\|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{x}\right\|_2^2 = \left\|\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}\mathbf{x}\right\|_2^2 = \|\mathbf{\Sigma}\mathbf{y}\|_2{}^2 = \\ = \sum_{i=1}^{n} {\sigma_i}^2 {y_i}^2 \leq {\sigma_1}^2 \|\mathbf{y}\|_2{}^2 = {\sigma_1}^2$$

And this value is reachable by taking $\mathbf{x} = \mathbf{v}_1$. The second part is analogous. □

Remark. Note that similarly to Eq. (12) we can write $\mathbf{A}^{-1} = \sum_{i=1}^{n} \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i{}^{\mathrm{T}}$, and therefore if we want to solve the system $\mathbf{A}\mathbf{x} = \mathbf{b}$, we can write:

$$\mathbf{x} = \sum_{i=1}^{n} \frac{\mathbf{u}_i{}^{\mathrm{T}}\mathbf{b}}{\sigma_i} \mathbf{v}_i$$

Theorem 48. Let $\mathbf{A} \in \mathrm{GL}_n(\mathbb{R})$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ be such that $\mathbf{A}\mathbf{x} = \mathbf{b}$. Then, the error $\Delta\mathbf{x}$ in the equation $\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$, $\Delta\mathbf{b} \in \mathbb{R}^n$, can be controlled by:

$$\frac{1}{\kappa(\mathbf{A})} \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} \leq \frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

where $\kappa(\mathbf{A}) = \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\|$ is the condition number of $\mathbf{A}$.

Proof. The second inequality is a consequence of **??**. For the first one, note that from $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ and $\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b}$ we have:

$$\|\mathbf{x}\| \leq \left\|\mathbf{A}^{-1}\right\| \|\mathbf{b}\| \qquad \|\Delta\mathbf{b}\| \leq \|\mathbf{A}\| \|\Delta\mathbf{x}\|$$

Hence, $\|\Delta\mathbf{b}\| \|\mathbf{x}\| \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \|\Delta\mathbf{x}\| \|\mathbf{b}\|$. □

Proposition 49. Let $\mathbf{A}, \mathbf{C} \in \mathcal{M}_n(\mathbb{R})$ and let $\mathbf{R} := \mathbf{I}_n - \mathbf{A}\mathbf{C}$. If $\|\mathbf{R}\| < 1$, then $\mathbf{A}$ and $\mathbf{C}$ are non-singular and:

$$\mathbf{A}^{-1} = \frac{\|\mathbf{C}\|}{1 - \|\mathbf{R}\|} \qquad \frac{\|\mathbf{R}\|}{\|\mathbf{A}\|} \leq \left\|\mathbf{C} - \mathbf{A}^{-1}\right\| \leq \frac{\|\mathbf{C}\| \|\mathbf{R}\|}{1 - \|\mathbf{R}\|}$$

Proof. If $\|\mathbf{R}\| < 1$, then $\rho(\mathbf{R}) < 1$, and so $\mathbf{I}_n - \mathbf{R}$ is invertible and so are $\mathbf{A}$ and $\mathbf{C}$ (taking the determinant). Moreover, $\left\|\mathbf{A}^{-1}\right\| \leq \|\mathbf{C}\| \left\|(\mathbf{I}_n + \mathbf{R})^{-1}\right\|$ from which the first inequality follows. For the second one, note that:

$$\|\mathbf{R}\| \leq \|\mathbf{A}\| \left\|\mathbf{C} - \mathbf{A}^{-1}\right\|$$

ANd using the previous one, we get the last one: $\left\|\mathbf{C} - \mathbf{A}^{-1}\right\| = \left\|\mathbf{A}^{-1}\mathbf{R}\right\| \leq \left\|\mathbf{A}^{-1}\right\| \|\mathbf{R}\|$. □

Truncated singular value decomposition

Remark. In practice however, doing a full SVD is not always possible, since it requires a lot of memory and time. A *truncated* version of it is often used, where we only keep the k largest singular values and their associated singular vectors.

Definition 50. Let $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$. The *truncated singular value decomposition* (*TSVD*) of $\mathbf{A}$ is an inexact decomposition of $\mathbf{A}$ of the form:

$$\tilde{\mathbf{A}} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^{\mathrm{T}}$$

where $\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_k)$, $\mathbf{U}_k \in \mathcal{M}_{m\times k}(\mathbb{R})$ is created selecting the k left singular vectors of $\mathbf{A}$ associated with $\sigma_1, \dots, \sigma_k$, and $\mathbf{V}_k \in \mathcal{M}_{n\times k}(\mathbb{R})$ is created selecting the k right singular vectors of $\mathbf{A}$ associated with $\sigma_1, \dots, \sigma_k$.

Remark. In this case the general solution of $\mathbf{Ax} \simeq \mathbf{A_k x} = \mathbf{b}$ is given by:

$$\mathbf{x} = \sum_{i=1}^{k} \frac{\mathbf{u}_i{}^{\mathrm{T}}\mathbf{b}}{\sigma_i}\mathbf{v}_i + \sum_{i=k+1}^{n} \xi_i \mathbf{v}_i$$

with $\xi_i \in \mathbb{R}$ arbitrary.

Proposition 51. Let $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$ and $\mathbf{b} \in \mathbb{R}^m$. Suppose that

$$\mathbf{x}_\lambda = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|_2{}^2 + \lambda \|\mathbf{x}\|_2{}^2$$

where $\lambda > 0$ is a regularization parameter. Then:

$$\mathbf{x}_\lambda = \sum_{i=1}^{n} \frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2} \frac{\mathbf{u}_i{}^{\mathrm{T}}\mathbf{b}}{\sigma_i}\mathbf{v}_i$$

Proof. First note that we can express $\mathbf{x}_\lambda$ as:

$$\mathbf{x}_\lambda = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\| \begin{pmatrix} \mathbf{A} \\ \lambda \mathbf{I}_n \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \right\|_2{}^2$$

Suppose a SVD of $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$. Then, from **?? ??** we know that this solution is given by:

$$\begin{aligned}
\mathbf{x}_\lambda &= \left(\begin{pmatrix} \mathbf{A} \\ \lambda \mathbf{I}_n \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \mathbf{A} \\ \lambda \mathbf{I}_n \end{pmatrix} \right)^{-1} \begin{pmatrix} \mathbf{A} \\ \lambda \mathbf{I}_n \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \\
&= \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda^2 \mathbf{I}_n\right)^{-1} \mathbf{A}^{\mathrm{T}}\mathbf{b} \\
&= \left(\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\mathrm{T}} + \lambda^2 \mathbf{V}\mathbf{V}^{\mathrm{T}}\right)^{-1} \mathbf{V}\mathbf{\Sigma}\mathbf{U}^{\mathrm{T}}\mathbf{b} \\
&= \mathbf{V}\left(\mathbf{\Sigma}^2 + \lambda^2 \mathbf{I}_n\right)^{-1} \mathbf{\Sigma}\mathbf{U}^{\mathrm{T}}\mathbf{b} \\
&= \sum_{i=1}^{n} \frac{{\sigma_i}^2}{{\sigma_i}^2 + \lambda^2} \frac{\mathbf{u}_i{}^{\mathrm{T}}\mathbf{b}}{\sigma_i}\mathbf{v}_i
\end{aligned}$$

□

QR decomposition

Lemma 52. Let $\mathbf{Q} \in \mathcal{O}_n(\mathbb{R})$. Then, $\forall \lambda \in \sigma(\mathbf{Q})$, $|\lambda| = 1$. Moreover, $\sigma_i = 1$, $i = 1, \dots, n$.

Proof. Let $\mathbf{v}$ be a unitary eigenvector of $\mathbf{Q}$ associated to λ. Then:

$$\lambda^2 = \langle \lambda\mathbf{v}, \lambda\mathbf{v}\rangle = \langle \mathbf{Qv}, \mathbf{Qv}\rangle = \left\langle \mathbf{v}, \mathbf{Q}^{\mathrm{T}}\mathbf{Qv}\right\rangle = \langle \mathbf{v}, \mathbf{v}\rangle = 1$$

To see that all the singular values are 1, note that if $\lambda \in \sigma(\mathbf{Q})$ with eigenvector $\mathbf{v}$, then $\lambda^{-1} \in \sigma(\mathbf{Q}^{\mathrm{T}})$ with eigenvector $\mathbf{v}$. Thus, $\forall \lambda \in \sigma(\mathbf{Q})$ with associated eigenvector $\mathbf{v}$, we have:

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Qv} = \mathbf{Q}^{\mathrm{T}}\lambda\mathbf{v} = \mathbf{v}$$

□

Lemma 53. Let $\mathbf{Q} \in \mathcal{O}_n(\mathbb{R})$. Then:

1. $\det \mathbf{Q} = \pm 1$.
2. $\|\mathbf{Q}\|_2 = 1$.

Proof. Note that $\mathbf{QQ}^{\mathrm{T}} = \mathbf{I}_n$. Taking determinants, we obtain the first equality. The second equality, follows from the preservation of the norm by orthogonal matrices: $\|\mathbf{Qv}\|_2 = \|\mathbf{v}\|_2$, $\forall \mathbf{v} \in \mathbb{R}^n$. □

Definition 54 (QR descompostion). Let $\mathbf{A} \in \mathcal{M}_n(\mathbb{R})$ be a matrix. A *QR decomposition* of $\mathbf{A}$ is an expression $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q} \in \mathcal{O}_n(\mathbb{R})$ and $\mathbf{R} \in \mathcal{M}_n(\mathbb{R})$ is upper triangular.

Proposition 55. Let $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$ be a full-rank matrix with $m \geq n$. Then, there exist matrices $\mathbf{Q} \in \mathcal{M}_{m\times n}(\mathbb{R})$ and $\mathbf{R} \in \mathcal{M}_n(\mathbb{R})$ such that $\mathbf{Q}$ is orthogonal, $\mathbf{R}$ is upper triangular and $\mathbf{A} = \mathbf{QR}$.

Proof. We use the **?? ??**. Assume $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$. We define $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2}$ and $\mathbf{q}_j = \frac{\mathbf{a}_j - \sum_{k=1}^{j-1} \langle \mathbf{a}_j, \mathbf{q}_k\rangle \mathbf{q}_k}{\left\|\mathbf{a}_j - \sum_{k=1}^{j-1} \langle \mathbf{a}_j, \mathbf{q}_k\rangle \mathbf{q}_k\right\|_2}$ for $j = 2, \dots, n$. Then, $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_n)$ is orthonormal and $\mathbf{R} := \mathbf{Q}^{\mathrm{T}}\mathbf{A}$ is upper triangular because if $\mathbf{R} = (r_{ij})$, then $r_{ij} := \langle \mathbf{q}_i, \mathbf{a}_j\rangle$ and from the above expression of $\mathbf{q}_j$, we have:

$$\mathbf{a}_j = \sum_{k=1}^{j-1} r_{kj}\mathbf{q}_k + r_j \mathbf{q}_j$$

for $j = 1, \dots, n$, with $r_j := \left\|\mathbf{a}_j - \sum_{k=1}^{j-1} \langle \mathbf{a}_j, \mathbf{q}_k\rangle \mathbf{q}_k\right\|_2$. □

Remark. In the literature, this latter decomposition is sometimes called *thin QR decomposition* to distinguish it from the *full QR decomposition* where $\mathbf{Q}$ and $\mathbf{R}$ are square matrices. In the thin QR decomposition, we can write:

$$\mathbf{A} = \mathbf{QR} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{pmatrix} = \mathbf{Q}_1\mathbf{R}_1$$

with $\mathbf{Q}_1 \in \mathcal{M}_{m\times n}(\mathbb{R})$, $\mathbf{Q}_2 \in \mathcal{M}_{m\times(m-n)}(\mathbb{R})$ and $\mathbf{R}_1 \in \mathcal{M}_n(\mathbb{R})$. Note that both $\mathbf{Q}_1$ and $\mathbf{Q}_2$ have orthogonal columns, and $\mathbf{R}_1$ is upper triangular.

Lemma 56. Let $\mathbf{A} \in \mathrm{GL}_n(\mathbb{R})$, $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{Ax} = \mathbf{b}$ be a system of linear equations. Suppose $\mathbf{A} = \mathbf{QR}$ for some orthogonal matrix $\mathbf{Q}$ and some upper triangular matrix $\mathbf{R}$, both of size n. Then, solving the system $\mathbf{Ax} = \mathbf{b}$ is equivalent to solving the triangular system $\mathbf{Rx} = \mathbf{Q}^{\mathrm{T}}\mathbf{b}$.

Proposition 57. Let $\mathbf{A} \in \mathcal{M}_{m\times n}(\mathbb{R})$ be a full-rank matrix with $m > n$. The least-squares problem

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \|\mathbf{Ax} - \mathbf{b}\|_2$$

has a solution $\mathbf{x}^*$ given by:

$$\mathbf{x}^* = \mathbf{R}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{b}$$

Proof. Let $\mathbf{A} = (\mathbf{Q}, \mathbf{Q}_\perp)\begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix} = \mathbf{QR}$ be the full QR decomposition of $\mathbf{A}$. Then:

$$\begin{aligned}\|\mathbf{Ax} - \mathbf{b}\|_2{}^2 &= \left\| \begin{pmatrix}\mathbf{Q}^{\mathrm{T}}\\ \mathbf{Q}_\perp{}^{\mathrm{T}}\end{pmatrix} (\mathbf{Ax} - \mathbf{b}) \right\|_2^2 \\ &= \left\| \begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix} \mathbf{x} - \begin{pmatrix}\mathbf{Q}^{\mathrm{T}}\mathbf{b}\\ \mathbf{Q}_\perp{}^{\mathrm{T}}\mathbf{b}\end{pmatrix} \right\|_2^2\end{aligned}$$

And so, by **??** **??** we have:

$$\mathbf{x}^* = \left(\begin{pmatrix}\mathbf{R} & \mathbf{0}\end{pmatrix} \begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix} \right)^{-1} \begin{pmatrix}\mathbf{R} & \mathbf{0}\end{pmatrix} \begin{pmatrix}\mathbf{Q}^{\mathrm{T}}\mathbf{b}\\ \mathbf{Q}_\perp{}^{\mathrm{T}}\mathbf{b}\end{pmatrix} = \mathbf{R}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{b}$$

□