Chapter 1

What is Numerical Analysis about ?

Case 1. Minimizer

Theorem 1. *Suppose $f : [a, b] \to \mathbb{R}$ is continuous on $[a, b]$. Let $m := \inf\{f(x) \mid x \in [a, b]\}$. Then there exists a minimizer $x_* \in [a, b]$ such that $f(x_*) = m$.*

Note: we have used that a continuous function f on $[a, b]$ is bounded.

Proof. Step 1. For each $n = 1, 2, 3, \cdots$ let $x_n \in [a, b]$ be any element such that

$$f(x_n) < m + \frac{1}{n}.$$

For each n, such an element must exist: Otherwise, i.e., if $f(x) \geq m + \frac{1}{n}$ for every $x \in [a, b]$, then $m + \frac{1}{n}$ is a lower bound of the range set that is strictly bigger than m, contradicting to the definition of m.
Step 2. We have defined the sequence (x_n), and let $y_n := f(x_n)$. Then $y_n \to m$ as $n \to \infty$ because

$$0 \leq |y_n - m| < \frac{1}{n}.$$

Step 3. (x_n) is a sequence of real numbers in $[a, b]$, and thus (x_n) is a bounded sequence. Therefore, there exists a convergent subsequence $(x_{n_k})_{k=1}^{\infty}$ to a limit x_*. Furthermore, because $[a, b]$ is closed, the limit point $x_* \in [a, b]$. We also know that the subsequence (y_{n_k}) of a convergent sequence (y_n) also is convergent to the same limit m.
Step 4. Now, we observe

$$m = \lim_{k\to\infty} y_{n_k} = \lim_{k\to\infty} f(x_{n_k}) = f(x_*),$$

where in the last equality, we used the fact that f is continuous on $[a, b]$. □

Case 2 : Riemann Integral or Quadrature

Let us consider high-school series calculations. Let n be a fixed natural number. For each i, we let $x_i = a + i\frac{b-a}{n}$. We let t_i be any point belogs to $[x_{i-1}, x_i]$ for $i = 1, \cdots n$. The definition of the Riemann Integral of f over $[a, b]$ is justified by the existence of a real number I such that

$$\lim_{n\to\infty} \sum_{i=1}^{n} f(t_i)\ \frac{b-a}{n} = I.$$

regardless of choice (t_i) at each n. We examine the proof of the assertion.

Proof. Step 1. By extreme value theorem on each compact interval $[x_{i-1}, x_i]$, there exist the minimizer t_{i*} and t_i^* in $[x_{i-1}, x_i]$. Then we have for each n

$$\sum_{i=1}^{n} f(t_{i*})\ \frac{b-a}{n} \le \sum_{i=1}^{n} f(t_i)\ \frac{b-a}{n} \le \sum_{i=1}^{n} f(t_i^*)\ \frac{b-a}{n}.$$

Step 2. We omit the proof that $\left(\sum_{i=1}^{n} f(t_{i*})\ \frac{b-a}{n}\right)_{n=1}^{\infty}$ and $\left(\sum_{i=1}^{n} f(t_i^*)\ \frac{b-a}{n}\right)_{n=1}^{\infty}$ are bounded monotone sequences so that their respective limits $\alpha, \beta \in \mathbb{R}$, $\alpha \le \beta$ exist.
Step 3. That f is continuous in $[a, b]$ implies the following:

$$\begin{aligned} &\text{For every } \epsilon > 0, \text{ there exists } \delta > 0 \text{ such that} \\ &x, x' \in [a, b], |x - x'| < \delta \quad \Longrightarrow \quad |f(x) - f(x')| < \frac{\epsilon}{b-a}. \end{aligned}$$

that is to say, f is uniformly continuous in $[a, b]$. For a given ϵ, we take $K \in \mathbb{N}$ so that $\frac{b-a}{K} < \delta$ for each δ. This gives that

for every ϵ, there exists K such that for $n \ge K$,

$$\Big|\sum_{i=1}^{n} f(t_i^*)\ \frac{b-a}{n} - \sum_{i=1}^{n} f(t_{i*})\ \frac{b-a}{n}\Big| \le \sum_{i=1}^{n} \Big|f(t_i^*) - f(t_{i*})\Big|\ \frac{b-a}{n} \le \sum_{i=1}^{n} \frac{\epsilon}{b-a}\ \frac{b-a}{n} = \epsilon.$$

This implies that the limits $\alpha = \beta$. By the Squeeze theorem, $\left(\sum_{i=1}^{n} f(t_i)\ \frac{b-a}{n}\right)_{n=1}^{\infty}$ is convergent, to the same limit. The limit is defiend to be I. □

We will use quite much better working numerical methods (midpoint rule, simpson rule, gaussian quadrature, $\cdots$) than the one suggested in the proof.

Case 3 : rank of a matrix

Let A be an $n \times k$ real matrix of $\mathbb{R}^{n\times k}$. That the very first fundamental notion for A is the rank of A, the number being defined by the fact that

the number of linearly independent columns = the number of linearly independent rows.

Denote the $(LHS) = r_{col}(A)$ and $(RHS) = r_{row}(A)$. This fundamental equality is a consequence of the inequality that

$$\text{for any matrix } A, \quad r_{row}(A) \leq r_{col}(A)$$

because if that is true then $\Big(r_{col}(A) = \Big)r_{row}(A^T) \leq r_{col}(A^T)\Big(= r_{row}(A)\Big)$.
The inequality follows from the assertion

$$\begin{aligned}&\text{if } r_{i_1}, r_{i_2}, \cdots r_{i_\rho} \text{ are linearly independent rows of } A\\ &\text{then } Ar_{i_1}, Ar_{i_2}, \cdots Ar_{i_\rho} \text{ are linearly independent vectors in } \mathbb{R}^n.\end{aligned}$$

Here we denote the rows of A from top to bottom by r_1, r_2, $\cdots$, r_n. We examine the proof of the assertion.

Proof. Suppose not, i.e., we have $(\lambda_1, \lambda_2, \cdots, \lambda_\rho) \neq 0$ such that $\sum_{\alpha=1}^{\rho} \lambda_\alpha \big(Ar_{i_\alpha}\big) = 0$. Then,

$$0 = \sum_{i=\alpha}^{\rho} \lambda_\alpha (Ar_{i_\alpha}) = \sum_{\alpha=1}^{\rho} A(\lambda_\alpha r_{i_\alpha}) = A\Big(\sum_{\alpha=1}^{\rho} \lambda_\alpha r_{i_\alpha}\Big).$$

But,

$$Ax = 0 \quad \Longleftrightarrow \quad x \perp \mathsf{span}\left\langle r_1, r_2, \cdots, r_n \right\rangle.$$

Combining the two observations,

$$\sum_{\alpha=1}^{\rho} \lambda_\alpha r_{i_\alpha} \in \mathsf{span}\left\langle r_1, r_2, \cdots, r_n \right\rangle \cap \mathsf{span}\left\langle r_1, r_2, \cdots, r_n \right\rangle^{\perp} = \{0\}.$$

Thus, we conclude that $\sum_{\alpha=1}^{\rho} \lambda_\alpha r_{i_\alpha}$ is a zero vector. Recalling that $r_{i_1}, r_{i_2}, \cdots r_{i_\rho}$ are linearly independent, we conclude $\lambda_1 = \lambda_2 = \cdots = \lambda_\rho = 0$. Contradiction. □

We have to wait until we prove that any A admits a Singular Value Decomposition to compute the rank of A.

Case 4 : Jordan factorization of a square matrix T.

Another striking example where we do not have an algorithm for it is the Jordan factorization.

We will not prove the statement here, that any square matrix T admits a Jordan factorization $T = PJP^{-1}$. But you will see in its proof that it does not provide a constructive method. There is a fundamental obstacle that prevents us to compute it in general case.

Any pair of (theory,method) from variety of mathematical areas.

What will we treat in this course ?

1. Existence of interpolating / approximating polynomials for a class of functions. Topics: spline, Tchebyshev polynomials, $\cdots$. 1d, multi-d.
2. Existence of fixed point and Newton's method on nonlinear system.
3. Existence of solution of initial value problem of nonlinear ode / Runge-Kutta methods, $\cdots$
4. For all of above, error analysis methodology.
5. (2nd semester) Linear algebra theory
6. (2nd semester) more on (partial) differential equations.

Chapter 2

How does my computer throw $\sin(x)$ for an input x?

Case 1. A function from $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ to $\mathbb{R}$

We can define a function $a : \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \mapsto \mathbb{R}$, using the table.

n	1	2	3	4	5	6	7	8	9	10
a_n										

Case 2. A function from $\mathbb{N}$ to $\mathbb{R}$

Are there ways to define a function from countably infinite set to $\mathbb{R}$?

Case 3. A function from $[0, 1] \subset \mathbb{R}$ to $\mathbb{R}$

Are there ways to define a function from this uncountably infinite set $[0, 1]$ to $\mathbb{R}$?

A function on $E \subset \mathbb{R}^n$?

A function $f : X \mapsto Y$ is a correspondence $G \subset X \times Y$ satisfying the rule that

For every $x \in X$, there exists unique $y \in Y$ so that $(x, y) \in G$.

Is a function a black box ?
Is a function a table ?
Can you come up with an example of a function that you can build as a machine ?
How does my computer know which value to return for $\sin(x)$?

I wanted you to see the importance of a class of functions on $\mathbb{R}^n$ that are

(1) piecewise polynomial, and

(2) each piece's domain (the support more precisely) is elementary in the sense that you can finish the membership check of a given x to the domain in a finite procedures.

Examples of sets $E \subset \mathbb{R}^n$, where the membership check is finitely done.

We will call those functions an elementary pp function or a good pp function.

Remark 2.1. The literal definition of the function on an uncountable set, which might suggest the tabularized mapping, indeed is beyond our intuition. If a function is continuous or measurable, we have excuses.

Excuses of not doing Uncountable Table

1. $f(x)$, the value to return, is calculated "on the fly":
   (a) $id(x)$ is identified on the fly, and a few arithmetic calculations involving it are processed afterwards.
   (b) The power p, for the term x^p, is limited to the Natural numbers (or Integers) so that we can operate
   $$id(x) \times id(x) \times \cdots \times id(x) \quad p \text{ times.}$$
   n-th root, in general, cannot be computed in a finite procedure.
2. irrational input, irrational coefficient are not processed.

Excuses by "Limit"

1. For irrational input, irrational coefficient cases, non-elementary piece's domain (support), the value to return can be reached by the limit procedure.

2. More importantaly,

   A continuous function on $\mathbb{R}^n$ is **a limit of continuous elementary piecewise polynomial functions**, where the convergence is uniform in every compact subset of $\mathbb{R}^n$.

   A measurable function on $\mathbb{R}^n$ is **a limit of elementary piecewise polynomial functions**, where the convergence is pointwise, for almost everywhere $x \in \mathbb{R}^n$.

3. Therefore, in principle, a suitable quantity can be computed as accurate as we want.

Objective of our study

For a given function f, we come up with a pp function φ and seek for a formula of the remainder

$$R(x) = f(x) - \varphi(x).$$

1. How do we choose a pp function φ? $\rightarrow$ theory and method
2. How accurate is it ? $\rightarrow$ The remainder formula.

For these two questions,

First, we will take one piece, out of piecewise polynomial function, and seek for the remainder formular, restricted on the piece's support. (Local problem)

Second, we assemble pieces into one pp function. (Global problem)

(Local) polynomial interpolation

Let $I = [a, b]$. We consider

$$C^{n+1}([a,b]) = \big\{ f : [a,b] \to \mathbb{R} \mid f \text{ is } n+1 \text{ times differentiable and } \\ f,\, f',\, f'',\, \cdots,\, f^{(n+1)} \text{ are continuous in } [a,b] \big\}.$$

Note: At the endpoint a and b, the differentibility is the right-differentiability and the left-differentiability respectively.

We consider the following interpolation problem:

1. Suppose we sample the data $(x_0, f(x_0))$, $\cdots$, $(x_n, f(x_n))$ for $f \in C^{n+1}([a,b])$ and $x_0, \cdots, x_n \in [a,b]$.
2. We look for a polynomial p on $[a,b]$ with
$$p(x_i) = f(x_i) \quad i = 0, 1, 2, \cdots n.$$
3. We seek for a formula of the remainder $R(x)$,
$$R(x) = f(x) - p(x), \quad x \in [a,b]$$

In fact, the term *interpolation* does not necessarily assume the function f and does not assume that the data are collected by sampling. Nevertheless, it is convenient to speak of the function f.

We are going to prove the far-reaching remainder theorem in the next lecture, which is much stronger than the Taylor's theorem. We recall first the fundamental theorem of calculus:

Theorem 1. *Let $f \in C^1([a,b])$ and $x_0 \in [a,b]$. Then for $x \in [a,b]$,*

$$R(x) = \int_{x_0}^{x} \frac{df}{dx}(x')\,dx' = (x-x_0)\int_{\tau=0}^{1} \frac{df}{dx}\big((1-\tau)x_0+\tau x\big)\,d\tau = f(x)-f(x_0).$$

Theorem 2 (1d remainder theorem)**.** *Let $f \in C^{n+1}([a,b])$ and $x_0,\ x_1,\ \cdots,\ x_n \in [a,b]$. Then, for $x \in [a,b]$,*

$$\begin{aligned} R(x) &= \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})(x-x_n)}{\sqrt{n+2}} \int_{\Lambda^{n+1}\subset\mathbb{R}^{n+2}} \frac{d^{n+1}f}{dx^{n+1}}\big(\lambda_0 x_0+\cdots\lambda_n x_n+\lambda_{n+1}x\big)\,dS(\lambda) \\ &= f(x) - \Big(f[x_0] + f[x_0,x_1](x-x_0) + f[x_0,x_1,x_2](x-x_0)(x-x_1)+\cdots \\ &\quad + f[x_0,x_1,\cdots,x_n](x-x_0)(x-x_1)(x-x_2)\cdots(x-x_{n-1})\Big) \end{aligned}$$

Remark 2.2. We will introduce the definition, which is the integral expression, of

$$f[x_0,x_1,\cdots,x_k],\Lambda^k,\cdots$$

Remark 2.3. We will see that the Taylor's theorem is a special case where $x_0 = x_1 = \cdots = x_n = \bar{x}$.

Theorem 3 (Taylor's theorem)**.** *Let $f \in C^{n+1}([a,b])$ and $x_0 \in [a,b]$. Then for $x \in [a,b]$,*

$$\begin{aligned} R(x) &= (x-x_0)^{n+1} \times \frac{1}{n!}\int_{\tau=0}^{1} (1-\tau)^n \frac{d^{n+1}f}{dx^{n+1}}\Big((1-\tau)x_0+\tau x\Big)\,d\tau \\ &= f(x) - \Big(f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n\Big). \end{aligned}$$

Chapter 3

Divided Differences

Let $I = [a, b]$, $n \in \mathbb{N}$, and consider $k+1$ points ($k \in \{0, 1, 2, \cdots, n+1\}$)

$$x_0, x_1, x_2, \cdots, x_k \in [a, b] \quad \text{(they may be repeated)}.$$

We will denote the choice of $(k+1)$ points by

$$[x_0, x_1, \cdots, x_k].$$

A choice $[x_0, \cdots, x_k]$ acts on the class $C^{n+1}([a, b])$: For $f \in C^{n+1}([a, b])$,

$$\text{if } k = 0 \qquad \big([x_0], f\big) = f(x_0),$$

$$\text{if } k \geq 1 \qquad \big([x_0, x_1, \cdots, x_k], f\big) := \frac{1}{\sqrt{k+1}} \int_{\Lambda^k \subset \mathbb{R}^{k+1}} \frac{d^k f}{dx^k}\big(\lambda_0 x_0 + \cdots \lambda_k x_k\big)\, dS(\lambda).$$

$$\text{The set } \Lambda^k = \Big\{(\lambda_0, \lambda_1, \cdots, \lambda_k) \in \mathbb{R}^{k+1} \mid \forall e \;\; \lambda_e \geq 0, \quad \sum_{e=0}^{k} \lambda_e = 1\Big\} \subset \mathbb{R}^{k+1}.$$

The integral is denoted by $\big([x_0, x_1, \cdots, x_k], f\big)$, or $[x_0, \cdots, x_k]f$, or $f[x_0, x_1, \cdots, x_k]$.

Properties of the Integral $f[x_0, x_1, \cdots, x_k]$

(1) The integral is independent of orders in the $[x_0, x_1, \cdots, x_k]$.

(2) $k = 1$ cases:

$$\begin{aligned} &\text{if } x_0 \neq x_1, && f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0) - f(x_1)}{x_0 - x_1}, \\ &\text{if } x_0 = x_1 = \bar{x}, && f[\bar{x}, \bar{x}] = f'(\bar{x}). \end{aligned}$$

(3) For $k \geq 1$, the equality always holds:

$$f[x_0, x_1, \cdots, x_k]\ (x_k - x_0) = f[x_1, x_2, \cdots, x_k] - f[x_0, x_1, \cdots, x_{k-1}].$$

Indeed, above two $k = 1$ cases lead to that

$$f[x_0, x_1](x_1 - x_0) = f[x_1] - f[x_0].$$

Change of variables formula for the integral

For $k \geq 1$,

$$\text{The set } \tau^k = \big\{(\tau_1, \tau_2, \cdots, \tau_k) \in \mathbb{R}^k \mid 0 \leq \tau_1 + \cdots + \tau_k \leq 1\big\}.$$

For $k \geq 1$,

$$\begin{aligned}
f[x_0, x_1, \cdots, x_k] =& \frac{1}{\sqrt{k+1}} \int_{\Lambda^k \subset \mathbb{R}^{k+1}} \frac{d^k f}{dx^k}\big(\lambda_0 x_0 + \cdots + \lambda_k x_k\big)\, dS(\lambda) \\
\text{(change of variable)} \quad =& \int_{\tau^k \subset \mathbb{R}^k} \frac{d^k f}{dx^k}\big((1 - \tau_1 - \tau_2 - \cdots - \tau_k)x_0 + \tau_1 x_1 + \cdots + \tau_k x_k\big)\, d\mathcal{L}^k(\tau) \\
\text{(Fubini)} \quad =& \int_{\tau_1=0}^{1} \int_{\tau_2=0}^{1-\tau_1} \int_{\tau_3=0}^{1-\tau_1-\tau_2} \cdots \int_{\tau_k=0}^{1-\tau_1-\cdots-\tau_{k-1}} \\
& \frac{d^k f}{dx^k}\big((1 - \tau_1 - \tau_2 - \cdots - \tau_k)x_0 + \tau_1 x_1 \cdots \tau_k x_k\big)\, d\tau_k d\tau_{k-1} \cdots d\tau_1
\end{aligned}$$

For the change of variable, the mapping from τ^k to Λ^k is

$$(\tau_1, \tau_2, \cdots, \tau_k) \quad \mapsto \quad \big(\lambda_0, \lambda_1, \cdots, \lambda_k\big) = \big((1 - \tau_1 - \tau_2 - \cdots - \tau_k), \tau_1, \cdots, \tau_k\big).$$

The linear mapping from Λ^k to $[a, b]$ is

$$x(\lambda) = \lambda_0 x_0 + \lambda_1 x_1 + \cdots + \lambda_k x_k \in [a, b].$$

The linear mapping from τ^k to $[a, b]$ is

$$\tilde{x}(\tau) = (1 - \tau_1 - \tau_2 - \cdots - \tau_k)x_0 + \tau_1 x_1 + \cdots + \tau_k x_k \in [a, b].$$

We can also compute

$$\int_{\tau_1=0}^{1} \int_{\tau_2=0}^{1-\tau_1} \int_{\tau_3=0}^{1-\tau_1-\tau_2} \cdots \int_{\tau_k=0}^{1-\tau_1-\cdots-\tau_{k-1}} 1\, d\tau_k d\tau_{k-1} \cdots d\tau_1 = \frac{1}{k!}.$$

Proof of the Key equality

Lemma 1. *Let $f \in C^{n+1}([a,b])$. Then for $k \in \{1,2,\cdots,n+1\}$ and x_0, x_1, $\cdots$, x_k $\in [a,b]$,*

$$f[x_0,x_1,\cdots,x_k]\ (x_k - x_0) = f[x_1,x_2,\cdots,x_k] - f[x_0,x_1,\cdots,x_{k-1}].$$

Proof. Thanks to the order independence,

$$\begin{aligned}(RHS) &= f[x_k,x_1,x_2,\cdots,x_{k-1}] - f[x_0,x_1,x_2,\cdots,x_{k-1}]\\ &= \int_{\tau_1=0}^{1}\int_{\tau_2=0}^{1-\tau_1}\int_{\tau_3=0}^{1-\tau_1-\tau_2}\cdots\int_{\tau_{k-1}=0}^{1-\tau_1-\cdots-\tau_{k-2}}\\ &\quad \frac{d^{k-1}f}{dx^{k-1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{k-1})x_k+\tau_1x_1+\tau_2x_2+\cdots+\tau_{k-1}x_{k-1}\big)\\ &\quad - \frac{d^{k-1}f}{dx^{k-1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{k-1})x_0+\tau_1x_1+\tau_2x_2+\cdots+\tau_{k-1}x_{k-1}\big)\,d\tau_{k-1}\cdots d\tau_1,\\ &=: A.\end{aligned}$$

The integrand is

$$\left[\frac{d^{k-1}f}{dx^{k-1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{k-1}-\tau_k)x_0+\tau_1x_1+\tau_2x_2+\cdots+\tau_{k-1}x_{k-1}+\tau_kx_k\big)\right]_{\tau_k=0}^{1-\tau_1-\tau_2-\cdots-\tau_{k-1}}$$

For each fixed $\tau_1,\tau_2,\cdots,\tau_{k-1}$, the map

$$\tau_k \mapsto (1-\tau_1-\tau_2-\cdots-\tau_{k-1}-\tau_k)x_0+\tau_1x_1+\tau_2x_2+\cdots+\tau_{k-1}x_{k-1}+\tau_kx_k$$

from the interval $[0,1-\tau_1-\tau_2-\cdots-\tau_{k-1}]$ to $[a,b]$ is a linear map, and $f \in C^{n+1}[a,b]$. Thus, for $k \in \{1,2,\cdots,n+1\}$ the composition function

$$\tau_k \mapsto \frac{d^{k-1}f}{dx^{k-1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{k-1}-\tau_k)x_0+\tau_1x_1+\tau_2x_2+\cdots+\tau_{k-1}x_{k-1}+\tau_kx_k\big)$$

is a continuously differentiable function in the interval $[0,1-\tau_1-\tau_2-\cdots-\tau_{k-1}]$, fulfilling the assumption of the Fundamental Theorem of Calculus. Therefore, the integrand is

$$\begin{aligned}&\int_{\tau_k=0}^{1-\tau_1-\tau_2-\cdots-\tau_{k-1}} \frac{d}{d\tau_k}\frac{d^{k-1}f}{dx^{k-1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{k-1}-\tau_k)x_0+\tau_1x_1+\cdots+\tau_kx_k\big)\,d\tau_k\\ =&\int_{\tau_k=0}^{1-\tau_1-\tau_2-\cdots-\tau_{k-1}} \frac{d}{dx}\frac{d^{k-1}f}{dx^{k-1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{k-1}-\tau_k)x_0+\tau_1x_1+\cdots+\tau_kx_k\big)\,d\tau_k(x_k-x_0).\end{aligned}$$

Substituting this expression in place of the integrand in A, we see that

$$A = f[x_0,x_1,\cdots,x_k](x_k-x_0).$$

□

Proof of 1d remainder theorem

All we need is the key equality

$$f[x_0, x_1, \cdots, x_k]\ (x_k - x_0) = f[x_1, x_2, \cdots, x_k] - f[x_0, x_1, \cdots, x_{k-1}].$$

For $k \in \{0, 1, 2, \cdots, n\}$, we let

$$\begin{aligned} p_k(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ + f[x_0, x_1, \cdots, x_k](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{k-1}). \end{aligned}$$

Theorem 2 (1d remainder theorem)**.** *Let $f \in C^{n+1}([a, b])$ and x_0, x_1, $\cdots$, $x_n \in [a, b]$. Then, for $x \in [a, b]$,*

$$\begin{aligned} R(x) &= f[x_0, x_1, x_2, \cdots, x_n, x](x - x_0)(x - x_1) \cdots (x - x_{n-1})(x - x_n) \\ &= f(x) - \Big(f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ &+ f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{n-1}) \Big). \end{aligned}$$

Proof. We assert that for $k \in \{0, 1, 2, \cdots, n\}$,

$$f(x) - p_k(x) = f[x_0, x_1, \cdots, x_k, x](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_k).$$

At $k = 0$, by the equality

$$f(x) - p_0(x) = f(x) - f(x_0) = f[x_0, x](x - x_0).$$

Now, if the assertion is true for $0, 1, 2, \cdots, k - 1$,

$$\begin{aligned} f(x) - p_k(x) &= f(x) - p_{k-1}(x) - f[x_0, x_1, \cdots, x_k](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{k-1}) \\ &= f[x_0, x_1, \cdots, x_{k-1}, x](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{k-1}) \\ &- f[x_k, x_0, x_1, \cdots, x_{k-1}](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{k-1}) \\ &= f[x_0, x_1, \cdots, x_{k-1}, x_k, x] \Big((x - x_0)(x - x_1)(x - x_2) \cdots (x - x_{k-1})(x - x_k) \Big). \end{aligned}$$

□

Mean Value type Theorem

Although we will not use this much later, it is worth to know that the mean value of the integrand is attained.

Theorem 3. *Let $f \in C^{n+1}([a,b])$, $k \in \{0,1,2,\cdots,n+1\}$, and $x_0, x_1, x_2, \cdots, x_k \in [a,b]$. Then there exists $\xi \in [a,b]$ such that*

$$f[x_0, x_1, \cdots, x_k] = \frac{1}{k!}\frac{d^k f}{dx^k}(\xi).$$

Remark 3.1. In other words, the attained value $\beta = \frac{d^k f}{dx^k}(\xi)$ at $\xi \in [a,b]$ is the mean value so that

$$\int_{\tau^k \subset \mathbb{R}^k} \frac{d^k f}{dx^k}\big(\tilde{x}(\tau)\big)\, d\mathcal{L}^k(\tau) = \left(\int_{\tau^k \subset \mathbb{R}^k} 1\, d\mathcal{L}^k(\tau)\right) \times \beta.$$

Proof. For $\lambda \in \Lambda^k$, the linear function

$$x(\lambda) = \lambda_0 x_0 + \lambda_1 x_1 + \cdots + \lambda_k x_k \in [a,b]$$

is continuous and thus $\lambda \mapsto \frac{d^k f}{dx^k} \circ x(\lambda)$ is a continuous function on the compact set Λ^k. Hence, there exist λ^*, $\lambda_* \in \Lambda^k$, the maximizer and the minimizer.
If $\lambda_* = \lambda^*$, then $\frac{d^k f}{dx^k} \circ x$ must be a constant function over Λ^k. Thus, at $\xi = x(\bar{\lambda})$ for any $\bar{\lambda} \in \Lambda^k$, the integral

$$(LHS) = \frac{d^k f}{dx^k}(\xi)\frac{1}{\sqrt{k+1}}\int_{\Lambda^k} 1\, dS(\lambda) = \frac{1}{k!}\frac{d^k f}{dx^k}(\xi)$$

and we are done.
Now, we assume $\lambda_* \neq \lambda^*$. Note that the convex set Λ^k contains the line segment joining λ_* and λ^*. Therefore, we can define a function

$$h(t) : [0,1] \ni t \mapsto \frac{d^k f}{dx^k} \circ x\big((1-t)\lambda_* + t\lambda^*\big).$$

h is a continuous function on $[0,1]$. By intermediate value theorem, every values in the range $[h(0), h(1)] = \big[\frac{d^k f}{dx^k}(\lambda_*), \frac{d^k f}{dx^k}(\lambda^*)\big]$ is attained at some $t \in [0,1]$.
Certainly,

$h(0) = \frac{d^k f}{dx^k}\big(x(\lambda_*)\big) \leq \frac{d^k f}{dx^k}\big(x(\lambda)\big) \leq \frac{d^k f}{dx^k}\big(x(\lambda^*)\big) = h(1)$. This implies

$$\frac{k!}{\sqrt{k+1}}\int_{\Lambda^k} \frac{d^k f}{dx^k}\big(x(\lambda_*)\big)\, dS(\lambda) \leq \frac{k!}{\sqrt{k+1}}\int_{\Lambda^k} \frac{d^k f}{dx^k}\big(x(\lambda)\big)\, dS(\lambda) \leq \frac{k!}{\sqrt{k+1}}\int_{\Lambda^k} \frac{d^k f}{dx^k}\big(x(\lambda^*)\big)\, dS(\lambda).$$

But note the leftmost term is $h(0)$ and the rightmost term is $h(1)$. Therefore, for some $\bar{t} \in [0,1]$

$$\frac{k!}{\sqrt{k+1}}\int_{\Lambda^k \subset \mathbb{R}^{k+1}} \frac{d^k f}{dx^k}\big(x(\lambda)\big)\, dS(\lambda) = h(\bar{t}) = \frac{d^k f}{dx^k}(\xi) \quad \text{at } \xi = x(\bar{\lambda}),\ \bar{\lambda} = (1-\bar{t})\lambda_* + \bar{t}\lambda^*.$$

□

Reduction to the Taylor's Theorem

Theorem 4 (Taylor's theorem)**.** *Let $f \in C^{n+1}([a,b])$ and $x_0 \in [a,b]$. Then for $x \in [a,b]$,*

$$\begin{aligned}
R(x) &= \int_{x_0}^{x} \frac{(x-x')^n}{n!} \frac{d^{n+1}f}{dx^{n+1}}(x')\, dx' \\
&= (x-x_0)^{n+1} \times \int_{\tau=0}^{1} \frac{(1-\tau)^n}{n!} \frac{d^{n+1}f}{dx^{n+1}}\Big((1-\tau)x_0 + \tau x\Big)\, d\tau \\
&= (x-x_0)^{n+1} \times \frac{1}{(n+1)!} \frac{d^{n+1}f}{dx^{n+1}}(\xi) \quad \textit{for some } \xi \in [a,b] \\
&= f(x) - \Big(f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n\Big).
\end{aligned}$$

The reduction is justified by following calculations: If $x_0 = x_1 = \cdots = x_n$,

1. We have $f[x_0, \cdots, x_k] = \frac{1}{k!} f^{(k)}(x_0)$ for each k.
2. We have

$$\begin{aligned}
R(x) =& (x-x_0)(x-x_1)\cdots(x-x_n) f[x_0, x_1, \cdots, x_n, x] \\
=& (x-x_0)^{n+1} \times \int_{\tau^{n+1} \subset \mathbb{R}^{n+1}} \frac{d^{n+1}f}{dx^{n+1}}\big((1-\tau_1-\tau_2-\cdots-\tau_{n+1})x_0 + \tau_1 x_1 \cdots \tau_n x_n + \tau_{n+1} x\big)\, d\mathcal{L}^{n+1}(\tau) \\
=& (x-x_0)^{n+1} \times \int_{\tau^{n+1} \subset \mathbb{R}^{n+1}} \frac{d^{n+1}f}{dx^{n+1}}\big((1-\tau_{n+1})x_0 + \tau_{n+1} x\big)\, d\mathcal{L}^{n+1}(\tau) \\
=& (x-x_0)^{n+1} \times \int_{\tau_{n+1}=0}^{1} \frac{d^{n+1}f}{dx^{n+1}}\big((1-\tau_{n+1})x_0 + \tau_{n+1} x\big) \times \\
& \int_{\tau_n=0}^{1-\tau_{n+1}} \int_{\tau_{n-1}=0}^{1-\tau_{n+1}-\tau_n} \cdots \int_{\tau_1=0}^{1-\tau_{n+1}-\cdots-\tau_2} 1\, d\tau_1 d\tau_2 \cdots d\tau_{n+1} \\
=& (x-x_0)^{n+1} \times \int_{\tau_{n+1}=0}^{1} \frac{d^{n+1}f}{dx^{n+1}}\big((1-\tau_{n+1})x_0 + \tau_{n+1} x\big) \times \frac{(1-\tau_{n+1})^n}{n!}\, d\tau_{n+1}.
\end{aligned}$$

   Finally, by the Mean Value type Theorem

$$R(x) = (x-x_0)^{n+1} \times \frac{1}{(n+1)!} \frac{d^{n+1}f}{dx^{n+1}}(\xi) \quad \text{for some } \xi \in [a,b].$$

Chapter 4

(Local) Polynomial Interpolation

Case 1. Data $\big(x_0, f(x_0)\big), \cdots, \big(x_n, f(x_n)\big)$, x_i all distinct.

We definde $\mathcal{P}^{<n+1}([a,b]) \subset C^{n+1}([a,b])$ be the set of all polynomials of order less than $n+1$, tautologically, the set of all polynomials of order at most n.

Theorem 1 (existence)**.** *For any $f \in C^{n+1}([a,b])$ and distinct points $x_0, x_1, \cdots, x_n \in [a,b]$, there exists a polynomial $p \in \mathcal{P}^{<n+1}([a,b])$ such that*

$$p(x_i) = f(x_i) \quad \textit{for every } i = 0, 1, 2, \cdots, n.$$

Proof. It suffices to recall that

$$f(x) - p_n(x) = f[x_0, x_1, \cdots, x_n, x](x - x_0)(x - x_1)\cdots(x - x_n),$$

and $p_n \in \mathcal{P}^{<n+1}([a,b])$. By assumption that $\frac{d^{n+1}f}{dx^{n+1}}$ is continuous in $[a,b]$, $f[x_0, x_1, \cdots, x_n, x]$ is a continuous function of x in $[a,b]$ and thus it is bounded in $[a,b]$. The conclusion follows by plugging in x_i in place of x. □

Theorem 2 (uniqueness)**.** *Under the same assumptions, the interpolating polynomial in $\mathcal{P}^{<n+1}([a,b])$ is unique.*

Proof. Suppose $p, q \in \mathcal{P}^{<n+1}([a,b])$ be the two interpolating polynomials. Then $r = p - q \in \mathcal{P}^{<n+1}([a,b])$ too and

$$r(x_i) = 0 \quad \text{for every } i = 0, 1, 2, \cdots, n.$$

Since $x_0, \cdots, x_{n-1}$ are roots of r,

$$r(x) = A(x - x_0)(x - x_1)\cdots(x - x_{n-1}). \quad \text{But}$$

$$r(x_n) = A(x_n - x_0)(x_n - x_1)\cdots(x_n - x_{n-1}) = 0.$$

Since $x_0, x_1, \cdots, x_n$ are all distinct, we conclude $A = 0$, or $p = q$. □

Case 2. Data $\big(x_1, f(x_1)\big), \cdots, \big(x_n, f(x_n)\big)$, $\big(x_1, f'(x_1)\big), \cdots, \big(x_n, f'(x_n)\big)$, x_i all distinct.

Theorem 3 (existence). *For any $f \in C^{2n}([a,b])$ and distinct points $x_1, \cdots, x_n \in [a,b]$, there exists a polynomial $p \in \mathcal{P}^{<2n}([a,b])$ such that*

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i) \quad \textit{for every } i = 1, 2, \cdots, n.$$

Proof. We consider a choice of $2n$ points in $[a,b]$ that is

$$[\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_{2n-1}] = [x_1, x_1, x_2, x_2, x_3, x_3, \cdots, x_n, x_n].$$

It suffices to recall that

$$\begin{aligned} f(x) - p_{2n-1}(x) &= f[\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_{2n-1}, x](x - \tilde{x}_0)(x - \tilde{x}_1)\cdots(x - \tilde{x}_{2n-1}) \\ &= f[x_1, x_1, x_2, x_2, \cdots, x_n, x_n, x](x - x_1)^2(x - x_2)^2 \cdots (x - x_n)^2 = R(x), \end{aligned}$$

where $f[x_1, x_1, x_2, x_2, \cdots, x_n, x_n, x]$ is continuous function of x in $[a,b]$ and

$$\begin{aligned} p_{2n-1}(x) = f[x_1] + f[x_1, x_1](x - x_1) + f[x_1, x_1, x_2](x - x_1)^2 + f[x_1, x_1, x_2, x_2](x - x_1)^2(x - x_2) + \cdots \\ + f[x_1, x_1, x_2, x_2, x_3, x_3, \cdots, x_n, x_n](x - x_1)^2(x - x_2)^2 \cdots (x - x_{n-1})^2(x - x_n) \end{aligned}$$

in $\mathcal{P}^{<2n}([a,b])$. Certainly, $R(x_i) = 0$ for $i = 1, \cdots, n$. It is also certain that $R(x) = f(x) - p_{2n-1}(x)$ is differentiable at x_i. We directly compute the derivative at x_1: By writing $g(x) = f[x_1, x_1, x_2, x_2, \cdots, x_n, x_n, x](x - x_2)^2 \cdots (x - x_n)^2$, continuous in $[a,b]$,

$$\frac{R(x_1 + h) - R(x_1)}{h} = \frac{R(x_1 + h)}{h} = \frac{h^2 g(x_1 + h)}{h} = h g(x_1 + h) \to 0 \quad \text{as } h \to 0.$$

Similar calculations can be done at $x = x_i$, to conclude that

$$R'(x_i) = 0 \quad \text{for } i = 1, \cdots, n.$$

□

Theorem 4 (uniqueness). *Under the same assumptions, the interpolating polynomial in $\mathcal{P}^{<2n}([a,b])$ is unique.*

Proof. Suppose $p, q \in \mathcal{P}^{<2n}([a,b])$ be the two interpolating polynomials. Then $r = p - q \in \mathcal{P}^{<2n}([a,b])$ too and x_i, $i = 1, 2, 3, \cdots, n$ are all double roots of r. We write

$$r(x) = A(x - x_1)^2(x - x_2)^2 \cdots (x - x_{n-1})^2(x - x_n). \quad \text{But}$$

$$r'(x_n) = A(x_n - x_1)^2(x_n - x_1)^2 \cdots (x_n - x_{n-1})^2 = 0.$$

Since $x_1, x_2, \cdots, x_n$ are all distinct, we conclude $A = 0$, or $p = q$. □

Remark 4.1. Note that we can in general achieve interpolation with a choice

$$[x_0, x_1, x_2, \cdots, x_r] \quad \text{(points can be repeated in any certain way)}$$

in $\mathcal{P}^{<r+1}([a,b])$.

Chapter 5

Methods for the Lagrange Interpolation

Case 1. Data $\big(x_0, f(x_0)\big), \cdots, \big(x_n, f(x_n)\big)$, x_i all distinct.

1. The same polynomial $p \in \mathcal{P}^{<n+1}([a,b])$ of order at most n can be expanded in many different ways. For instance, the polynomial

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots \\ + f[x_0, x_1, \cdots, x_n](x - x_0)(x - x_1)(x - x_2)\cdots(x - x_{n-1}),$$

   which is expanded with polynomials of increasing orders with respect to the sampling point x_i, of course can be re-arranged in the form

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

2. The former is said to be expanded in Newton basis, while the latter is said to be expanded in power basis. Newton basis and power basis both are bad.

Remark 5.1. Although the newton basis expansion of p is bad, at least one can compute it because the coefficient $f[x_0, x_1, \cdots, x_k]$ can be computed by the key recursive equality: because $x_0, x_1, \cdots, x_n$ are distinct

$$\begin{aligned} f[x_0] &= f(x_0) \\ f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ f[x_0, x_1, x_2] &= \frac{\frac{f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0} \\ &\vdots \end{aligned}$$

Remark 5.2. A basis what is called Lagrange basis and another basis what is called the Bernstein basis are in general said to better behave. Here we implement polynomial construction expanded in Lagrange basis.

Definition 1. *Let $x_0, x_1, \cdots, x_n \in [a, b]$ be distinct points. We define for each $i = 0, 1, \cdots, n$,*

$$L_i(x) = \beta_i(x - x_0)(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n) \in \mathcal{P}^{<n+1}([a, b]),$$

where $\beta_i = \Big((x_i - x_0)(x_i - x_1)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)\Big)^{-1} = (\gamma_i)^{-1}$.

A few observations:

1. By assumption that $x_0, x_1, \cdots, x_n$ are distinct, $\gamma_i \neq 0$ and thus β_i is well-defined.
2. $L_i(x_i) = 1$ and $L_i(x_j) = 0$ if $i \neq j$.
3. For every i, the order of $L_i(x)$ is n.

Remark 5.3. While $L_i(x)$ is of order n for every i, for a combination $\sum_{i=0}^{n} f(x_i)L_i(x)$ terms may cancel so that the resultant polynomial is of order less than n.

Proposition 2. *The interpolating polynomial $p \in \mathcal{P}^{<n+1}([a, b])$ obtained uniquely by Theorem 1,2 in the preceding section must be*

$$\sum_{i=0}^{n} f(x_i)L_i(x).$$

Proof. We check that $q(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$ is a member of $\mathcal{P}^{<n+1}([a, b])$, and satisfies

$$q(x_i) = f(x_i) \quad i = 0, 1, \cdots, n.$$

Hence q is an interpolating polynomial in $\mathcal{P}^{<n+1}([a, b])$. Since the interpolating polynomial is unique, q coincides with the one obtained by Theorem 1,2. □

Here, we devise three implementations of the function that returns the value $q(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$.

The First Implementation

We consider the literal implementation of

$$q(\bar{x}) = \sum_{i=0}^{n} f(x_i)L_i(\bar{x}).$$

1. We assume the sampling points $x_s = (x_0, x_1, \cdots, x_n)$ and $f_s = (f(x_0), f(x_1), \cdots, f(x_n))$ are given as $n+1$ size vectors.

2. Recalling that

$$L_i(x) = \beta_i(x - x_0)(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n),$$

   we compute

$$\beta_i \quad \text{and} \quad m_i(x) = (x - x_0)(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n).$$

   (blank)

---

3. We implement a function returning the value $q(\bar{x})$ for a given $\bar{x}$.

   (blank)

---

4. This works perfectly fine.

Remark 5.4. We inserted codes block that are in Julia programming language. The labeling of x_i in Julia was adapted so that $i = 1, 2, \cdots, n+1$.

The second Implementation

We consider a few intersting implementational considerations.

1. In computing $L_i(\bar{x}) = \beta_i m_i(\bar{x})$, it is tempting we compute $m_i(\bar{x})$ not making n products, but making one division with $w(\bar{x}) = (\bar{x} - x_0)(\bar{x} - x_1)\cdots(\bar{x} - x_n)$.

$$\text{If } \bar{x} \notin \{x_0, x_1, \cdots, x_n\}, \quad L_i(\bar{x}) = \frac{w(\bar{x})\beta_i}{\bar{x} - x_i},$$

   $w(\bar{x})$ is computed once and is shared afterwards. This is to say that the first implementation is an $O(n^2)$ algorithm while the second implementation is an $O(n)$ algorithm. The difference in order by 1 reflects a very large difference for n large. (I am not sure though if we use such a large n.)

2. Here we are arranging $q(\bar{x})$ in the form

$$q(\bar{x}) = \begin{cases} f(x_i) & \text{if } \bar{x} \in \{x_0, x_1, \cdots, x_n\} \text{ and } \bar{x} = x_i, \\ \displaystyle\sum_{i=0}^{n} f(x_i)\frac{w(\bar{x})\beta_i}{\bar{x} - x_i} & \text{if } \bar{x} \notin \{x_0, x_1, \cdots, x_n\}. \end{cases}$$

   (blank)

---

3. One may worry when $\bar{x}$ is very close to some x_j but $\bar{x} - x_j$ is not declared to be 0 by a computer.

4. Practically, this way of computing weights does not cause a problem. Justification?

The third Implementation

We consider another intersting implementation that uses idea to avoid the overflow.

1. Lagrange basis $\big(L_i(x)\big)_{i=0}^n$ has a nice property that

$$\text{for any } x \in [a,b], \quad L_0(x) + L_1(x) + \cdots + L_n(x) = 1$$

   i.e., $L_i(x)$, interpreted as weights, add up to 1. (But they are signed weights !).

2. This is because, if we interpolate the function $f(x) \equiv 1$, then $f(x) \in \mathcal{P}^{n+1}([a,b])$ and of course f itself is interpolating the data. By uniqueness, the lagrange basis interpolating polynomial $q(x) = \sum_{i=0}^n f(x_i)L_i(x)$ of 1 must be 1.

3. This in particular gives that

$$\text{If } \bar{x} \notin \{x_0, x_1, \cdots x_n\}, \quad 1 = (\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)\sum_{i=0}^n \frac{\beta_i}{\bar{x}-x_i}.$$

4. This is in fact to conduct the partial fraction. If $\bar{x} \notin \{x_0, x_1, \cdots x_n\}$

$$\frac{1}{(\bar{x}-x_0)(\bar{x}-x_1)\cdots(\bar{x}-x_n)} = \frac{\beta_0}{\bar{x}-x_0} + \frac{\beta_1}{\bar{x}-x_1} + \cdots + \frac{\beta_n}{\bar{x}-x_n}.$$

5. Therefore,

$$\text{If } \bar{x} \notin \{x_0, x_1, \cdots x_n\}, \quad L_i(\bar{x}) = \frac{w(\bar{x})\beta_i}{\bar{x}-x_i} = \frac{\frac{\beta_i}{\bar{x}-x_i}}{\sum_{k=0}^n \frac{\beta_k}{\bar{x}-x_k}}.$$

6. If $\bar{x} \notin \{x_0, x_1, \cdots x_n\}$, define

$$t_k := \frac{\beta_k}{\bar{x}-x_k}, \quad M := \max_{k=0,1,\cdots,n} |t_k|, \quad r_k := \frac{t_k}{M} \in [-1,1].$$

7. Since $L_i(\bar{x}) = \dfrac{t_i}{\sum_k t_k} = \dfrac{t_i/M}{\sum_k (t_k/M)}$, we compute

$$\text{If } \bar{x} \notin \{x_0, x_1, \cdots x_n\}, \quad L_i(\bar{x}) = \frac{r_i}{\sum_k r_k}.$$

8. What will happen for μ_i if $\bar{x}$ is very close to some x_j ?

(blank)

Chapter 6

The remainder $R(x)$

1. The 1d remainder theorem, stating

$$f(x) - p(x) = f[x_0, x_2, \cdots, x_n, x](x - x_0)(x - x_2) \cdots (x - x_n),$$

asserts the equality of the left-hand-side and the right-hand-side. (It does not assert the right-hand-side is small.)

2. Of course, we will use the procedure of (sampling $\rightarrow$ interpolation) for the purpose of approximation. How big $|R(x)|$ can be ?

Two general principles

We examine cases where $|R(x)|$ can be so big that (sampling $\rightarrow$ interpolation) procedure fails. Inspecting examples should lead us to the following conclusion to avoid this failure.

Two lessons.

1. For a fixed number of sampling data, use low order polynomials with many small pieces, instead of using high order polynomial with 1 piece.
2. For each small such an interval piece, if possible, choose chebychev zeroes as sampling points, instead of easy equi-distance points.

The Polynomial w of one higher order

1. In our examples,
$$w(x) = (x - x_0)(x - x_2)(x - x_3) \cdots (x - x_n) \notin \mathcal{P}^{<n+1}$$
will play an important role.

2. Note that for this w, the sampling data will be $(x_i, 0)$ for $i = 0, 1, 2, \cdots n$. Hence, the interpolating polynomial p in $\mathcal{P}^{<n+1}$ is the zero function $p \equiv 0$.

3. The 1d remainder theorem is telling that
$$w(x) - 0 = R(x) = w(x).$$
Indeed, in $R(x) = w[x_0, x_1, \cdots, x_n, x](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_n)$, check that $w[x_0, x_1, \cdots, x_n, x]$ is an integral of $(n + 1)$-th derivative of w but
$$\frac{d^{n+1}}{dx^n}(x - x_0)(x - x_1)(x - x_2) \cdots (x - x_n) = (n + 1)! \quad (const.)$$
and thus
$$\int_{\tau^{n+1}} (n + 1)!\, d\mathcal{L}(\tau) = \frac{(n + 1)!}{(n + 1)!} = 1.$$
The remainder $w[x_0, x_1, \cdots, x_n, x](x - x_0)(x - x_1)(x - x_2) \cdots (x - x_n)$ is indeed $w(x)$ itself.

The constant $M := \sup_{x \in [a,b]} |w(x)|$

1. We define $M := \sup_{x \in [a,b]} |w(x)|$. M is dependent on the $[a, b]$, and the choice $(x_i)_{i=0}^n$.

2. The use of w in remainder analysis for a general f can be put in the following way. The remainder
$$f[x_0, x_1, \cdots, x_n, x](x - x_0)(x - x_1) \cdots (x - x_n)$$
is the product of two expressions $f[x_0, x_1, \cdots, x_n, x]$ and $w(x)$.
For given $x \in [a, b]$, $|w(x)|$ depends only on the choice of sampling points while $\left|f[x_0, x_1, \cdots, x_n, x]\right|$ depends on $(n+1)$-th derivative of f as well. We may estimate size of the remainder separately. In case of $f = w$,
$$|f[x_0, x_1, \cdots, x_n, x]| \equiv 1, \quad |w(x)| \leq M.$$

Problems on M when not following the two lessons before.

How does the polynomial $w(x)$ look like in $[a, b]$ for a particular choice $(x_i)_{i=0}^n$?

Calculations of M of for two cases.

1. We adjust the domain in the form $[-a, a]$ and we directly estimate two constants

   $$M_1 := \sup_{x\in[-a,a]} \left|(x-x_0)(x-x_1)\cdots(x-x_n)\right| \quad M_2 := \sup_{x\in[-a,a]} \left|(x-r_0)(x-r_1)\cdots(x-r_n)\right|$$

   where in the first case we use equi-distanced points $x_i = -a + i\frac{2a}{n}$, and in the second case we use r_i the zeroes of $(n+1)$-th order *chebychev polynomial.*

2. The former is a bad case scenario, while the latter is a good case scenario. The estimation of M_1 and M_2 will reveal how big $|w|$ can be, for good and bad cases.

Calculations of M_2: case of chebychev zeroes

A Minimal introduction to Chebychev polynomials.

Let $\tilde{I} = [0, \pi]$. We consider its parametrization by a bijection from $[-a, a]$:

$$\theta : [-a, a] \to [0, \pi], \quad \theta(x) = \arccos\Big(\frac{x}{a}\Big).$$

Definition 1. *For each* $k = 0, 1, 2, \cdots$

$$T_k(x) := \cos\big(k\theta(x)\big).$$

Proposition 2. *For each* $k = 0, 1, 2, \cdots,$

1. T_k *has* k *zeroes in* $[-a, a]$.
2. *For* $k \geq 1$, T_k *attains* -1, *the minimum, and* $+1$, *the maximum, alternatingly so that* $|T_k(x)|$ *hits* 1 *exactly* $k + 1$ *times in* $[-a, a]$.
3. T_k *is a* k*-th order polynomial.*
4. *For* $k \geq 1$, *the coefficient of the highest order term of* T_k *is* $\frac{1}{2}\Big(\frac{2}{a}\Big)^k$.

Proof. 1. In case $k = 0$, $T_0(x) \equiv 1$ and it has no zero. For $k \geq 1$, T_k attains zero only at x such that

$$k\theta(x) = \frac{\pi}{2} + m\pi$$

for m integer. Those integers so that $\phi_m = \frac{1}{k}\Big(\frac{\pi}{2} + m\pi\Big) \in [0, \pi]$, i.e. those integers so that $r_m = \theta^{-1}(\phi_m) \in [-a, a]$, are $m = 0, 1, 2, \cdots, k-1$. Note (ϕ_m) is strictly increasing in m and $\theta(x)$ is a bijection. Thus, $r_0, r_1, \cdots, r_{k-1}$ are distinct k points in $[-a, a]$.

2. $T_k(x) \in [-1, 1]$, the range of the cosine function. For $k \geq 1$, $|T_k(x)|$ hits 1 only at x such that

$$k\theta(x) = m\pi$$

for m integer. Those integers so that $\tilde{\phi}_m = \frac{m\pi}{k} \in [0, \pi]$, i.e. those integers so that $t_m = \theta^{-1}(\tilde{\phi}_m) \in [-a, a]$, are $m = 0, 1, 2, \cdots, k$. Note $(\tilde{\phi}_m)$ is strictly increasing in m and $\theta(x)$ is a decreasing bijection. Thus, $t_0, t_1, \cdots, t_k$ are distinct $k + 1$ points such that

$$t_0 > t_1 > \cdots > t_k$$

and $T_k(t_m) = \cos(m\pi) = (-1)^m$.

3. For $k = 0$, $T_0(x) \equiv 1$ is 0-th order polynomial. For $k = 1$, $T_1(x) = \frac{x}{a}$. Now, we use the trigonometric identity for $k \geq 1$. For notational convenience write $\theta = \theta(x)$, then

$$\begin{aligned}
T_{k+1}(x) &= \cos\Big(k\theta + \theta\Big) = \cos k\theta \cos\theta - \sin k\theta \sin\theta \\
T_{k-1}(x) &= \cos\Big(k\theta - \theta\Big) = \cos k\theta \cos\theta + \sin k\theta \sin\theta \\
&\implies T_{k+1}(x) + T_{k-1}(x) = 2T_k(x)\cos(\theta(x)) = 2T_k(x)\frac{x}{a} \\
&\implies T_{k+1}(x) = 2T_k(x)\frac{x}{a} - T_{k-1}(x).
\end{aligned}$$

Therefore, if T_0, T_1, $\cdots$, T_{k-1} are polynomials of order $0, 1, \cdots, k-1$ respectively, then $T_k(x)$ is a k-th order polynomial.

4. The coefficient of the highest order term of $T_1(x)$ is $\frac{1}{a}$, and from the last recursive formula, that of $T_k(x)$ for $k \geq 1$ is

$$\frac{1}{a}\Big(\frac{2}{a}\Big)^{k-1} = \frac{1}{2}\Big(\frac{2}{a}\Big)^{k}.$$

□

Now, we can compute $M_2 := \sup_{x\in[-a,a]} \big|(x-r_0)(x-r_1)\cdots(x-r_n)\big|$. Since r_i are $(n+1)$ distinctive zeroes of $(n+1)$-th order polynomials T_{n+1} and w, and the coefficient of the highest order of $w(x)$ is 1, $w(x)$ must be

$$w(x) = 2\Big(\frac{a}{2}\Big)^{n+1} T_{n+1}(x).$$

Since the maximum of $|T_{n+1}(x)| = 1$, we conclude that $M_2 = 2\Big(\frac{a}{2}\Big)^{n+1}$.

Corollary 3. *The constant* $M_2 = 2\Big(\frac{a}{2}\Big)^{n+1}$.

Calculations of $\left|w\left(\frac{x_0+x_1}{2}\right)\right|$: equi-distance sampling

Instead of computing the supremum M_1, we go for easier calculations of $K = \left|w\left(\frac{x_0+x_1}{2}\right)\right|$. Note the equi-distance is $\frac{2a}{n}$.

$$\begin{aligned} K &= \frac{a}{n} \times \frac{a}{n} \times \frac{3a}{n} \times \cdots \times \frac{\big(2n-1\big)a}{n} \\ &= \frac{a^{n+1}}{n^{n+1}}\ 1 \times 3 \times \cdots \times \big(2n-1\big). \end{aligned}$$

Because

$$\begin{aligned} 1 \times 2 \times 4 \times \cdots \times 2n-2 \ &< \ 1 \times 3 \times \cdots \times \big(2n-1\big) \ < \ 2 \times 4 \times \cdots \times 2n, \\ (2n-1)! \ &< \ \Big(1 \times 3 \times \cdots \times \big(2n-1\big)\Big)^2 \ < \ (2n)! \end{aligned}$$

Thus,

$$\frac{a^{n+1}}{n^{n+1}}\sqrt{(2n-1)!} \ < \ K \ < \ \frac{a^{n+1}}{n^{n+1}}\sqrt{(2n)!}.$$

It is known that for $n \geq 1$

$$\sqrt{2\pi}\sqrt{2n}\Big(\frac{2n}{e}\Big)^{2n} \ < \ (2n)! \ < \ e\sqrt{2\pi}\sqrt{2n}\Big(\frac{2n}{e}\Big)^{2n}.$$

This gives that

$$\frac{a^{n+1}}{n^{n+1}}\frac{1}{\sqrt{2n}} \times \sqrt{\sqrt{2\pi}\sqrt{2n}\Big(\frac{2n}{e}\Big)^{2n}} \ < \ K \ < \ \frac{a^{n+1}}{n^{n+1}} \times \sqrt{e\sqrt{2\pi}\sqrt{2n}\Big(\frac{2n}{e}\Big)^{2n}}$$

This simplifies to that

$$\Big(\frac{2a}{e}\Big)^{n+1} \times n^{-5/4} \times \frac{e}{2\sqrt{2}}(4\pi)^{\frac{1}{4}} \ < \ K \ < \ \Big(\frac{2a}{e}\Big)^{n+1} \times n^{-3/4} \times \frac{e^{3/2}}{2}(4\pi)^{\frac{1}{4}}.$$

Remark 6.1. The common ratio respectively for two cases:

$$\rho_1 = \frac{2a}{e}, \quad \rho_2 = \frac{2a}{4}, \quad \text{for the interval length } 2a.$$

Thus, if the interval length $2a$ is not smaller than the denominator, the supremum will increases geometrically for both cases.

Runge's example and Hilbert space techniques

Suppose $a = 1$. The interval lenth $2a = 2$. For this interval:

1. One can prove in the chebychev zero case that the interpolating polynomial p_n at chebychev zeroes does converge uniformly to a given function f, if f has a certain differentiability, say $f \in C^2([-1,1])$. This proof needs Hilbert space techniques.

2. By contrast, in the equi-distance case, Runge found an example of infinitely smooth function on $[-1,1]$, where the contribution from $\big|f[x_0, x_2, \cdots, x_n, x]\big|$ increases so fast in n that it overwhelms the geometric decrement of the contribution from $|w(x)|$.

$$
\begin{aligned}
f(x) &= \frac{1}{1+(5x)^2} = \frac{1}{2i}\Big(\frac{1}{5x-i} - \frac{1}{5x+i}\Big) = \frac{1}{5\times 2i}\Big(\frac{1}{x-i/5} - \frac{1}{x+i/5}\Big) \\
\frac{d^{n+1}f}{dx^{n+1}}(x) &= \frac{1}{5\times 2i}(-1)^{n+1}(n+1)!\Big(\frac{1}{(x-i/5)^{n+2}} - \frac{1}{(x+i/5)^{n+2}}\Big) \\
&= \frac{1}{5\times 2i}(-1)^{n+1}(n+1)!5^{n+2}\Big(\frac{1}{(5x-i)^{n+2}} - \frac{1}{(5x+i)^{n+2}}\Big) \\
&= \frac{(-1)^{n+1}(n+1)!5^{n+1}}{(1+(5x)^2)^{n+2}}\frac{1}{2i}\big(z^{n+2} - \bar{z}^{n+2}\big), \quad z = 5x+i, \quad x \in \mathbb{R} \\
&= \frac{(-1)^{n+1}(n+1)!5^{n+1}}{(1+(5x)^2)^{n+2}}\rho(x)^{n+2}\frac{1}{2i}\big(e^{i(n+2)\theta(x)} - e^{-i(n+2)\theta(x)}\big), \quad z = 5x+i = \rho(x)e^{i\theta(x)} \\
&= \frac{(-1)^{n+1}(n+1)!5^{n+1}}{\big(\sqrt{1+(5x)^2}\big)^{n+2}}\sin\big((n+2)\theta(x)\big).
\end{aligned}
$$

3. This derivative formula alone cannot conclude that $\sup\limits_{x\in[-1,1]} |R(x)|$ diverge as n increases, but the factor 5^{n+1} is a conceivable danger: it overwhems the factor $\big(\frac{2}{e}\big)^{n+1}$, i.e., $5 > e$.

At this stage, we wrap it up here, and move on to the topic of piecewise polynomials.

Chapter 7

Nonlinear system of equations and Newton method

We are to solve a system of n equations on n unknowns: for instance

$$\begin{aligned}
\sin(x_1 + x_3^2) + \exp(x_5)\cos(x_2 x_4) &= 3,\\
x_3^2 + x_4^2 &= 1,\\
x_1^4 - x_2^3 - 4x_3 + x_2 &= -7,\\
\sinh(x_3 - x_1) + \cosh(x_2 + x_5) &= 3,\\
x_5 + \log(x_1) &= -2.
\end{aligned}$$

For $\mathbf{f} : E \subset \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ we solve

$$\mathbf{f}(\mathbf{x}) = \mathbf{y}, \quad \mathbf{x} \in \mathbb{R}^n$$

that is a system of equations

$$\begin{aligned}
f_1(x_1, x_2, \cdots, x_n) &= y_1,\\
f_2(x_1, x_2, \cdots, x_n) &= y_2,\\
f_3(x_1, x_2, \cdots, x_n) &= y_3,\\
&\vdots\\
f_n(x_1, x_2, \cdots, x_n) &= y_n.
\end{aligned}$$

We may of course let $\tilde{\mathbf{f}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \mathbf{y}$ and solve $\tilde{\mathbf{f}}(\mathbf{x}) = 0$.

Nonlinear system of equations

For a given nonlinear system of equations, we do not know in advance in general,

1. whether any solution exists or not,
2. if exists, whether there are many solutions or not.

One very crude but reasonable idea is to make this nonlinear problem into an artificial linear problem.

Suppose we want to compute the $\sqrt{5}$, or we want to solve

$$x^2 = 5.$$

Since we can solve $ax = 5$ ($a \neq 0$), namely $x = \frac{5}{a}$, we artificially pose a problem

$$\tilde{x} \times x = 5,$$

where $\tilde{x}$ is to be regarded as a *coefficient*. Now, for given $\tilde{x}$, define

$$\frac{5}{\tilde{x}} =: \eta(\tilde{x}).$$

We now hope that we are very lucky that it happen to be $x = \eta(\tilde{x}) = \tilde{x}$. More rational expectation is to hope the iteration

$$x_{n+1} = \eta(x_n)$$

converges. If so, assuming η is continuous, this implies that at the limit value $\bar{x}$, $\bar{x} = \eta(\bar{x})$ and we found the solution.

The art of choosing $\tilde{x}$ part and x part in nonlinear equations.

The first objective of this crude procedure can be the following.

1. We choose $\tilde{x}$ part and x part in nonlinearities in such a way so that the solution $x = \eta(\tilde{x})$ of the artificial solvable problem falls into a bounded controlled region.
2. More specifically, this first objective is to come up with choice of $\tilde{x}$, x, and the candidate set $K \subset \mathbb{R}^n$ that is compact and convex such that
$$\eta(K) \subset K.$$
3. our example $\eta(\tilde{x}) = \frac{5}{\tilde{x}}$?

Theorem 1. *Suppose $K \subset \mathbb{R}^n$ is a nonempty compact and convex set, η is continuous on K and $\eta(K) \subset K$. Then there exists $\bar{x} \in K$ such that $\eta(\bar{x}) = \bar{x}$.*

We have a few remarks:

1. This theorem gives the existence of the solution.
2. The theorem works merely under the assumption of continuity.
3. Other than the case $n = 1$, it is very difficult to construct an algorithm.

A better iteration?

We return to the problem of solving $x^2 = 5$. For some reason, instead of $\frac{5}{\tilde{x}}$, we set

$$\eta(\tilde{x}) = \frac{1}{2}\Big(\tilde{x} + \frac{5}{\tilde{x}}\Big)$$

and we run the iteration

$$x_{n+1} = \eta(x_n),$$

hoping the generated sequence converges.

In fact, we ran the newton method,

$$x_{n+1} = x_n - \big(f'(x_n)\big)^{-1}(f(x_n)), \quad f(x) = x^2 - 5.$$

When can we come up with a better iteration?

Let us write

$$\mathbf{y}_* - \mathbf{f}(\mathbf{x}) - D\mathbf{f}(\mathbf{x})\big(\mathbf{x} - \mathbf{x}\big) = 0.$$

and consider the artificial problem

$$\mathbf{y}_* - \mathbf{f}(\tilde{\mathbf{x}}) - D\mathbf{f}(\tilde{\mathbf{x}})\big(\mathbf{x} - \tilde{\mathbf{x}}\big) = 0$$

that is a linear problem for a given $\tilde{\mathbf{x}}$. In case $D\mathbf{f}(\tilde{\mathbf{x}})$ is invertible, the linear solver

$$\eta(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}} - D\mathbf{f}(\tilde{\mathbf{x}})^{-1}\big(\mathbf{f}(\tilde{\mathbf{x}}) - \mathbf{y}_*\big).$$

By any chance, will the iteration $\mathbf{x}_{n+1} = \eta(\mathbf{x}_n)$ generate a convergent sequence?

Let us suppose the followings:

1. The nonlinearity $\mathbf{f} \in C^2(U)$. $U \subset \mathbb{R}^n$ is a nonempty open set.
2. There exists a solution $\mathbf{x}_* \in U$.
3. $\mathbf{x}_*$ is a simple root, i.e., $D\mathbf{f}(\mathbf{x}_*)$ is invertible.

   Then, the Inverse Function Theorem says that there exist a neighborhood V of $\mathbf{x}_*$ and a neighborhood W of $\mathbf{y}_*$ such that $\mathbf{f}$ is a bijection between V and W.
4. Suppose we do know what V and W is, and where $\mathbf{y}_*$ is. We do not know where $\mathbf{x}_*$ is.

What method can capture the solution $\mathbf{x}_*$?

Define

$$\mathbf{f}(\mathbf{x}_*) - \mathbf{f}(\mathbf{x}) - D\mathbf{f}(\mathbf{x})\big(\mathbf{x}_* - \mathbf{x}\big) =: R(\mathbf{x}).$$

By the Taylor's theorem, $|R(\mathbf{x})|$ converges to 0 quadratically as $\mathbf{x} \to \mathbf{x}_*$:

$$\lim_{\mathbf{x}\to\mathbf{x}_*} \frac{|R(\mathbf{x})|}{|\mathbf{x} - \mathbf{x}_*|^2} \le M.$$

5. Suppose $\bar{\mathbf{x}} \in V$ is a reasonably good approximation of $\mathbf{x}_*$, so that $|\bar{\mathbf{x}} - \mathbf{x}_*|^2$ is much smaller than $|\bar{\mathbf{x}} - \mathbf{x}_*|$.

Then, for L, an $n \times n$ invertible matrix,

$$\begin{aligned}
& L^{-1}\big(\mathbf{f}(\mathbf{x}_*) - \mathbf{f}(\bar{\mathbf{x}})\big) - L^{-1}D\mathbf{f}(\bar{\mathbf{x}})\big(\mathbf{x}_* - \mathbf{x}\big) = L^{-1}R(\bar{\mathbf{x}}) \\
\iff\ & L^{-1}\big(\mathbf{f}(\mathbf{x}_*) - \mathbf{f}(\bar{\mathbf{x}})\big) - \big(\mathbf{x}_* - \bar{\mathbf{x}}\big) + \big(\mathrm{I} - L^{-1}D\mathbf{f}(\bar{\mathbf{x}})\big)\big(\mathbf{x}_* - \bar{\mathbf{x}}\big) = L^{-1}R(\bar{\mathbf{x}}) \\
\iff\ & \Big[\bar{\mathbf{x}} - L^{-1}\big(\mathbf{f}(\bar{\mathbf{x}}) - \mathbf{f}(\mathbf{x}_*)\big)\Big] - \mathbf{x}_* = D\mathbf{f}(\mathbf{x})^{-1}R(\bar{\mathbf{x}}) - \big(\mathrm{I} - L^{-1}D\mathbf{f}(\bar{\mathbf{x}})\big)\big(\mathbf{x}_* - \bar{\mathbf{x}}\big)
\end{aligned}$$

If we can take L so that $L^{-1}D\mathbf{f}(\bar{\mathbf{x}})$ is close to the identity, and if we can justify the $(RHS) \le C_0|\bar{\mathbf{x}} - \mathbf{x}_*|^2$, possibly much smaller than $|\bar{\mathbf{x}} - \mathbf{x}_*|$, then the expression

$$\bar{\mathbf{x}} - L^{-1}\big(\mathbf{f}(\bar{\mathbf{x}}) - \mathbf{y}_*\big)$$

could become a better approximation of $\mathbf{x}_*$ than $\bar{\mathbf{x}}$. Thus, we try

$$\eta(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}} - L^{-1}\big(\mathbf{f}(\tilde{\mathbf{x}}) - \mathbf{y}_*\big),$$

expecting the convergence in the iteration $\mathbf{x}_{n+1} = \eta(\mathbf{x}_n)$.

We will justify this convergence in theorem-proof style.

It is important to note that

1. we need to start with a good approximation $\bar{\mathbf{x}}$.
2. $\mathbf{x}_*$ needs to be a simple root. (We will consider later the case $\mathbf{x}_*$ is a multiple root)
3. $\mathbf{f}$ needs to be C^2 on some neighborhood of $\mathbf{x}_*$.

Chapter 8

Convergence of Newton Method in 1d

1. For given $\mathbf{f}$ and $\mathbf{y}_*$, let $\tilde{\mathbf{f}}(x) = \mathbf{f}(x) - \mathbf{y}_*$ and solve zero finding problem

$$\tilde{\mathbf{f}}(\mathbf{x}) = 0.$$

2. We examined that the newton iteration may possibly converge under conditions where $\mathbf{f} \in C^2(U)$ and the zero $\mathbf{x}_*$ has multiplicity 1.

3. Since the Newton Method is convergent under the quite general assumption of multiplicity, we formulate our assumptions with respect to the multiplicity below.

Assumptions with respect to the multiplicity $p \geq 1$ are formulated in (M_p):

(i) if $p = 1$, $f \in C^1\big(I(r)\big)$ for some $r > 0$, where we denote $I(r) = (x_* - r, x_* + r)$.

$(i)'$ if $p > 1$, $f \in C^1\big(I(r)\big)$ for some $r > 0$, where we denote $I(r) = (x_*, x_* + r)$ in this case.

(ii) for $x \in I(r)$, $\quad f(x) = c_p(x - x_*)^p + R(x) \quad$ for some $c_p \neq 0$ and

$$\lim_{x \to x_*} \frac{|R(x)|}{|x - x_*|^p} = 0, \quad \lim_{x \to x_*} \frac{|R'(x)|}{|x - x_*|^{p-1}} = 0, \quad \text{where the limit is taken in } I(r) \setminus \{x_*\}. \tag{M_p}$$

Theorem 1. *Suppose f and $p \geq 1$ satisfy (M_p). Then there exists $0 < r' \leq r$ so that followings are true.*

1. *If $x \in I(r') \setminus x_*$ then $f'(x) \neq 0$.*
2. *If $x \in I(r') \setminus x_*$, define $\eta(x) = x - \frac{f(x)}{f'(x)}$. Then,*

$$\begin{aligned} &\text{for some } 0 < \lambda < 1, \quad |\eta(x) - x_*| < \lambda |x - x_*|, \\ &\text{in particular if } p > 1, \quad 0 \leq \eta(x) - x_* < \lambda (x - x_*). \end{aligned}$$

Remark 8.1. Because of the item 2, $\eta(x)$ is a member of $I(r')$. Hence, either $\eta(x)$ happen to be the solution x_* or the interation can be continued: If $x_0 \in I(r')$, the newton iteration either finds solution in finite steps or can be continued to converge geometrically.

Proof. Let $p \geq 1$. We have that

$$\begin{aligned}|f'(x)| = \big|pc_p(x-x_*)^{p-1} + R'(x)\big| &\geq |pc_p||x-x_*|^{p-1} - |R'(x)| \\ &= |x-x_*|^{p-1}\left(|pc_p| - \frac{|R'(x)|}{|x-x_*|^{p-1}}\right).\end{aligned}$$

By assumption (M_p), $\frac{|R'(x)|}{|x-x_*|^{p-1}} \to 0$ in $I(r)\backslash\{x_*\}$ as $x \to x_*$. Therefore, for $\epsilon = \frac{|pc_p|}{2} > 0$, there exists $0 < r'' \leq r$ such that

$$x \in I(r'') \setminus \{x_*\} \quad \Longrightarrow \quad \frac{|R'(x)|}{|x-x_*|^{p-1}} < \epsilon.$$

This implies that if $x \in I(r'') \setminus \{x_*\}$ then $f'(x) \neq 0$.

Now, assume $x \in I(r'') \setminus \{x_*\}$ and define $\eta(x) = x - \frac{f(x)}{f'(x)}$. Then,

$$\begin{aligned}\eta(x) - x_* &= x - x_* - \frac{f(x)}{f'(x)} \\ &= (x-x_*)\left(1 - \frac{f(x)}{f'(x)(x-x_*)}\right) = (x-x_*)\lambda(x).\end{aligned}$$

Now,

$$\begin{aligned}\lambda(x) &= 1 - \frac{c_p(x-x_*)^p + R(x)}{pc_p(x-x_*)^p + R'(x)(x-x_*)} \\ &= 1 - \frac{c_p + \frac{R(x)}{(x-x_*)^p}}{pc_p + \frac{R'(x)}{(x-x_*)^{p-1}}}.\end{aligned}$$

Note that $\lambda(x) \to 1 - \frac{1}{p}$ as $x \to x_*$ in $I(r'') \setminus \{x_*\}$. Therefore, for any $\epsilon > 0$, there exists $0 < r' \leq r''$ such that

$$x \in I(r') \setminus \{x_*\} \quad \Longrightarrow \quad \big|\lambda(x) - (1 - \frac{1}{p})\big| < \epsilon.$$

In case $p = 1$, $1 - \frac{1}{p} = 0$ and we choose $\epsilon = \frac{1}{2}$ to conclude

$$x \in I(r') \setminus \{x_*\} \quad \Longrightarrow \quad |\eta(x) - x_*| \leq |x-x_*|\ |\lambda(x)| < \frac{1}{2}|x-x_*|.$$

In case $p > 1$, we can take $\epsilon > 0$ so small that

$$0 \leq 1 - \frac{1}{p} - \epsilon \quad \text{and} \quad 1 - \frac{1}{p} + \epsilon < 1.$$

We conclude that with this choice,

$$x \in I(r') \setminus \{x_*\} \quad \Longrightarrow \quad 0 \leq \eta(x) - x_* = \lambda(x)(x-x_*) < \Big(1 - \frac{1}{p} + \epsilon\Big)(x-x_*).$$

□

1. If the multiplicity $p \geq 1$ is in advance known, one can have hypergeometric convergence by using a variant of newton method:

$$\eta(x) = x - p\frac{f(x)}{f'(x)}.$$

2. Here, we consider this hypergeometric convergence for the case $p = 1$ under the conditions

$$f \in C^2(I(r)), \quad \text{for some } M > 0 \quad \lim_{x \to x_*} \frac{|R(x)|}{|x - x_*|^2} \leq M, \quad \lim_{x \to x_*} \frac{|R'(x)|}{|x - x_*|} \leq M.$$

Chapter 9

Error Analysis framework

1. Many of numerical analysis problem can be formulated as an equation solving problem
$$F(x) = y_*,$$
where $F : X \to Y$ for some nonempty sets X and Y. (Say, metric spaces)
2. For example, consider a problem of finding QR decomposition of a matrix $A \in \mathbb{R}^{n\times k}$.
   (a) We take the set $X = X_1 \times X_2$, where X_1 is the set of $n \times k$ orthogonal matrices, X_2 is the set of $k \times k$ upper triangular matrices.
   (b) We take Y as the vector space of $n \times k$ square matrices.
   (c) We take the nonlinear function by the matrix multiplication $F(Q, R) = QR$. Entrywisely, $F_{i,j} = \sum_k Q_{i,k}R_{k,j}$.
   (d) The problem is to solve $F(Q, R) = A$.

To be able to numerically solve the equation, a few minimum requirements should be met.

1. F must be continuous. Otherwise, even if we happen to find a good approximate solution $\tilde{x}$ with $|x - \tilde{x}|$ small, we will not be able to check if $F(\tilde{x}) - y_*$ is close to 0 or not.
2. One might not want to assume the existence of solution for each $y \in Y$. However, we take the existence as granted here: The excuse is that we are working on designing an algorithm for a known theory of existence, which has characterized the set Y such that for $y \in Y$, there exists a solution $F(x) = y$.

We thus assume that
$$F \text{ is continuous and surjective.}$$
We develop an error analysis framework under this assumption.

One typical way of making algorithm.

1. Of course, to obtain the solution is assumed to be more difficult than evaluating F at x.

   For example, if we consider a partial differential equation

   $$\begin{aligned} -\Delta u(x) &= \rho(x), \quad x \in U, \\ u(x) &= 0, \quad x \in \partial U, \end{aligned}$$

   evaluating the Laplacian for given u is easy, but taking inverse from ρ is difficult.

2. Hence, as we have seen in previous class, we change the problem into easier one that we can solve, namely

   $$\tilde{F}(x) = y.$$

   Here, let us take the solvability of $\tilde{F}$, as we have a unique solution for this problem, and we know howto solve it to have

   $$\tilde{x} = \tilde{M}(y).$$

   For example, a pde problem can be changed to a numerical problem in finite dimensions.

   But have in mind that, in case $F(x) = y$ has many solutions, different algorithm can pick different solution uniquely.

3. In many cases, we have a sequence of modification $\tilde{F}_j$, $j = 1, 2, \cdots$, and thereby a sequence of approximate solver $\tilde{M}_j$.

4. Even though we do not intend to have such parametrized solver $\tilde{M}_j$, our method $\tilde{M}$ will be implemented in a machine, and is affected by rounding-off due to the finite digits of our machine. In this case, the number of digits will do the role of parameter.

a posteriari error analysis

A consistent algorithm

We consider the definition of the consistency of methods $(\tilde{M}_j)$. We first define the minimal consistency as requirement that $\tilde{M}_j(y) \in X$ at $y \in Y$.

Definition 1 (consistency of $(\tilde{M}_j)$ and $\tilde{M}$)**.** .

1. *We say that $(\tilde{M}_j)$ is consistent at y, if $\tilde{M}_j(y) \in X$ and*
$$\left|y - F(\tilde{M}_j(y))\right| \to 0 \quad \text{as } j \to \infty.$$
2. *We say that $\tilde{M}$ is η-consistent at y for $\eta > 0$ if $\tilde{M}(y) \in X$ and*
$$\left|y - F(\tilde{M}(y))\right| < \eta.$$

We will call $\left|y - F(\tilde{M}_j(y))\right|$ the testing error. This is distinguished from the solution error $\left|x_* - \tilde{M}_j(y)\right|$ for x_* a solution.

1. In many of numerical analysis problem, devising a consistent algorithm can be sufficient, or can be optimal.
2. In that case, we are not aware of that the solution error is small or not.

A convergent algorithm

Definition 2 (convergence of $(\tilde{M}_j)$ and accuracy of $\tilde{M}$)**.** .

1. *We say that a consistent algorithm $(\tilde{M}_j)$ at y is convergent if $\tilde{M}_j(y)$ is convergent to some $\bar{x} \in X$ as $j \to \infty$.*
2. *We say that an η-consistent algorithm $\tilde{M}$ at y is ϵ-accurate for $\epsilon > 0$ if there exists a solution $\bar{x} \in X$ such that*
$$|\bar{x} - \tilde{M}(y)| < \epsilon.$$

a posteriari error analysis

For simpler exposition,

Assumption (A1)-(A2):

A1. F is injective, thus it is bijective between X and Y, and

A2. The inverse R is continuous. For every $y' \in Y$, and every $\epsilon > 0$, there exists $\eta > 0$ such that

$$|y - y'| < \eta \quad \Longrightarrow \quad |x - x'| < \epsilon, \quad \text{where } x = R(y),\ x' = R(y').$$

For more general exposition,

Assumption (A):

A. F is an open map. For every $x' \in X$, and every $\epsilon > 0$, there exists $\eta > 0$ such that

$$|y - F(x')| < \eta \quad \Longrightarrow \quad \exists x \quad \text{with} \quad F(x) = y \quad \text{and} \quad |x - x'| < \epsilon.$$

In both cases the ϵ is a function of η and y. We write $\epsilon(\eta, y)$. (More precisely, $\eta(\epsilon, y)$ for a fixed y can be arranged as a monotone function of $\epsilon > 0$ and we invert the function.)

a posteriari error analysis:

1. Suppose an algorithm $\tilde{M}$ is η-consistent at y_*, i.e.,

$$\left|y_* - F(\tilde{M}(y_*))\right| < \eta. \quad \text{Define} \quad \tilde{y} := F(\tilde{M}(y_*)).$$

2. Suppose $\eta > 0$ and $\epsilon(\eta, \tilde{y}) > 0$ are chosen as in the above.

3. By (A1)-(A2),

$$|x_* - \tilde{M}(y_*)| < \epsilon(\eta, \tilde{y}).$$

3′ By (A), denoting $x' = \tilde{M}(y_*)$,

$$\text{there exists } x_* \in X \text{ with } F(x_*) = y_* \text{ and } |x_* - \tilde{M}(y_*)| < \epsilon(\eta, \tilde{y}).$$

a priori error analysis

Definition 3 (consistency of $(\tilde{F}_j)$ and $\tilde{F}$)**.** .

1. *We say that* $(\tilde{F}_j)$ *is consistent at* x *if*
$$|\tilde{F}_j(x) - F(x)| \to 0 \quad \text{as } j \to \infty.$$
2. *We say that* $\tilde{F}$ *is* η*-consistent at* x *for* $\eta > 0$ *if*
$$|\tilde{F}(x) - F(x)| < \eta.$$

Assumption (B):

B. The inverse $\tilde{M}$ of $\tilde{F}$ is continuous. For every $y' \in Y$, and every $\epsilon > 0$, there exists $\eta > 0$ such that
$$|y - y'| < \eta \quad \Longrightarrow \quad |x - x'| < \epsilon, \quad \text{where } x = \tilde{M}(y),\ x' = \tilde{M}(y').$$

B′. For a parameterized case, we assume the $\eta > 0$ in the above statement for $\tilde{M}_j$ is independent of j.

a priori error analysis:

1. Suppose $\tilde{F}$ is η-consistent at x_*, i.e.,
$$\left|\tilde{F}(x_*) - F(x_*)\right| < \eta.$$
2. Suppose $\eta > 0$ and $\epsilon(\eta, y_*) > 0$ are chosen as in the above.
3. Note that $F(x_*) = y_* = \tilde{F}\big(\tilde{M}(y_*)\big)$. Therefore, we have $\left|\tilde{F}(x_*) - \tilde{F}\big(\tilde{M}(y_*)\big)\right| < \eta$. This implies that
$$|x_* - \tilde{M}(y_*)| < \epsilon(\eta, y_*).$$

1. In a priori error analysis,
   (a) We do not know what x_* is. Therefore, the consistency level η, which depends on x, assumes the worst case scenario in X.
   (b) Since $\tilde{M}$ is the one we built, the map $\epsilon(\eta, y_*)$ for $\tilde{M}$ can be accurately estimated.
   (c) The solution error level $\epsilon(\eta, y_*)$ is the error that is given in advance we run the algorithm.
2. In a posteriari error analysis,
   (a) We do know what y_* is, and after we run the algorithm we know what $\tilde{y} = F\big(\tilde{M}(y_*)\big)$ is. Therefore, the consistency level η, which depends on y, assumes the adapted scenario that suffices to work only in the neighborhood of $\tilde{y}$.
   (b) The map $\epsilon(\eta, \tilde{y})$ for R has to be known by theoretical analysis.
   (c) The solution error level $\epsilon(\eta, \tilde{y})$ is the error that is given after we run the algorithm.

In many of problems, the solution error ϵ may not be directly estimated (although it looks so in some algorithms mathematically). Rather,

1. The consistency level $\eta > 0$ is first set.
2. The ϵ-η relation is inherent to the problem, decided either by $R = F^{-1}$ or $\tilde{M}$.
3. The solution error ϵ is given accordingly to the above two.

In newton's method, $\eta > 0$ is set by the termination condition $|F(\tilde{x})| < \eta$ in the iteration. The multiplicity $p \geq 1$ gives rise to $\varepsilon(\eta) \simeq c_p \eta^{\frac{1}{p}}$ in the neighborhood of 0. The solution error level may not be small for large p.

Chapter 10

Vector space of pp functions

1. In 1d, we recall that the set of all polynomials of order at most n on $[a, b]$ is the vector space $\mathcal{P}^{<n+1}([a, b])$ of dimensions $n + 1$.
2. In the similar way, we consider the set of piecewise polynomials on $[a, b]$ in $1d$.

A few notions:

1. We let $[a, b] = I$ be the domain of pp function. We call I the global domain.
2. We consider its partition by *break points*

$$a = t_0 < t_1 < t_2 < \cdots < t_L = b.$$

3. This partition induces local intervals

$$I_\ell = [t_{\ell-1}, t_\ell], \quad \ell = 1, 2, \cdots, L.$$

4. Intersections of such local intervals are interfaces. Here interfaces are simply

$$(t_k)_{k=1}^K, \quad K = L - 1.$$

Assembling a pp function φ by polynomials on I_ℓ.

The first example

1. Suppose for each local interval I_ℓ, the maximum order of polynomial piece of $\varphi\big|_{I_\ell}$ on I_ℓ we want is $n(\ell)$.
2. Then, we are to fix each piece $p_\ell \in \mathcal{P}^{<n(\ell)+1}(I_\ell)$, and this defines a pp function φ:

$$\varphi(x) = p_\ell(x) \quad \text{if } x \in \text{int}(I_\ell).$$

Remark 10.1. We are careless if the local domain is $(t_{\ell-1}, t_\ell)$ or $[t_{\ell-1}, t_\ell)$ or $[t_{\ell-1}, t_\ell]$ because in I_ℓ, $p_\ell(x)$ gives rise to the sided limit from inside, i.e., p_ℓ is uniformly continuous on I_ℓ.

1. If $\varphi(x)$ and $\psi(x)$ are two such functions, then so is $\alpha\varphi(x) + \beta\psi(x)$ for $\alpha, \beta \in \mathbb{R}$. Thus, we collect all such pp functions to form a vector space denoted by $\Pi([a, b])$.
2. Total degrees of freedom, or the dimensions of $\Pi([a, b]) = n(1) + 1 + n(2) + 1 + \cdots + n(L) + 1 = N$.
3. Concerning interpolation, we conduct for each local domain I_ℓ (for example) the Lagrange interpolation. There, of course we need in total N data.

Polynomial order $n(\ell)$ and number of matching conditions $m(k)$.

1. Suppose on top of previous considerations, we want to impose matching conditions on each interface t_k $k = 1, 2, \cdots, K$.
2. Let us restrict ourselves to the types of matching conditions that are

$$\begin{aligned} \varphi(t_k-) &= \varphi(t_k+) \\ \varphi'(t_k-) &= \varphi'(t_k+) \\ &\vdots \\ \frac{d^{m-1}\varphi}{dx^{m-1}}(t_k-) &= \frac{d^{m-1}\varphi}{dx^{m-1}}(t_k+) \end{aligned}$$

   up to some derivatives of order $m(k) - 1$ for each interface t_k.

1. If $\varphi(x)$ and $\psi(x)$ are two such functions, then so is $\alpha\varphi(x) + \beta\psi(x)$ for $\alpha, \beta \in \mathbb{R}$. Thus, we collect all such pp functions to form a vector space denoted by $\Pi([a, b])$.
2. Global (total) degrees of freedom without imposing the matching conditions was $\sum_{\ell=1}^{L} n(\ell) + 1 = N$.
3. Since we imposed constraints in total $\sum_{k=1}^{K} m(k) = M$, degrees of freedom are reduced by M, and thus the dimensions of $\Pi([a, b])$ is $N - M$. Of course, the case $N - M < 0$ is meaningless.
4. Concerning interpolation, we conduct for each local domain I_ℓ the interpolation using data including derivatives. We have to make sure to share common data on the interface.

The second example of piecewise linear function

Remainder inequality

1. Let f be a given function on $[a,b]$, and $\varphi \in \Pi([a,b])$.
2. The local remainder for each ℓ, the local remainder is
$$R_\ell(x) = f(x) - p_\ell(x), \quad \mathcal{E}_\ell = \sup_{x \in I_\ell} |R_\ell(x)|.$$
3. Then, the global remainder inequality is
$$|f(x) - \varphi(x)| \leq \max_{\ell=1,2,\cdots,L} \mathcal{E}_\ell.$$
4. The point is that the local domain length can be kept small.

Chapter 11

Splines of order r

1. Particularly handy choice of $\Pi[a,b]$ is the case where for $m \in \{0\} \cup \mathbb{N}$

$$m(1) = m(2) = \cdots = m(K) = m, \quad n(1) = n(2) = \cdots = n(L) = m.$$

   (a) $m = 0$: piecewise constant function without matching conditions.
   (b) $m = 1$: piecewise linear function that is continuous,
   (c) $m = 2$: piecewise quadratic that is C^1 function,
   (d) $m = 3$: piecewise cubic that is C^2 function,

$$\vdots$$

2. We denote such spaces for each m by $\Pi^r([a,b])$ with $r = m + 1$. A member of $\Pi^r([a,b])$ is called a spline function of order r.

Remark 11.1.

$$\begin{aligned} m &: \text{order of polynomials} = \text{number of matching conditions} \\ m+1 = r &: \text{order of splines} = \text{local degrees of freedom} \\ m-1 = \alpha &: C^\alpha \text{ differentiability} \end{aligned}$$

Let $a = t_0 < t_1 < \cdots < t_L = b$ be the break points.

The vector spaces sharing break points in $[a,b]$

$$\Pi^1([a,b]) \supset \Pi^2([a,b]) \supset \Pi^3([a,b]) \supset \cdots$$

are respectively subsets of

$$B([a,b]) \supset C^0([a,b]) \supset C^1([a,b]) \supset \cdots$$

To fix one spline in $\Pi^r([a,b])$

1. As in the interpolation in $\mathcal{P}^{<n+1}([a,b])$, we consider a problem to fix a spline in $\Pi^r([a,b])$ (with the above break points), that interpolates the given sample data.

2. Total degrees of freedom is

$$\Big(\sum_{\ell=1}^{L} n(\ell)+1\Big)-\Big(\sum_{k=1}^{K} m(k)\Big)=\Big(\sum_{\ell=1}^{L} m+1\Big)-\Big(\sum_{k=1}^{L-1} m\Big)=m+L=r+L-1=N$$

   (The symbol N is redefined here.)

3. As in the interpolation in $\mathcal{P}^{<n+1}([a,b])$, once we select the basis functions $b_1(x)$, $b_2(x)$, $\cdots$, $b_N(x)$ of $\Pi^r([a,b])$, any member of $\Pi^r([a,b])$ is

$$\varphi(x)=c_1b_1(x)+c_2b_2(x)+\cdots c_Nb_N(x).$$

4. Hence, for instance if we are given N sample data $\big(x_i,f(x_i)\big)$, we solve

$$c_1b_1(x_i)+c_2b_2(x_i)+\cdots+c_Nb_N(x_i)=f(x_i)\quad i=1,2,\cdots,N,$$

   i.e., we solve

$$\begin{pmatrix} b_1(x_1) & b_2(x_1) & \cdots & b_N(x_1)\\ b_1(x_2) & b_2(x_2) & \cdots & b_N(x_2)\\ b_1(x_3) & b_2(x_3) & \cdots & b_N(x_3)\\ \vdots & \vdots & \vdots & \vdots\\ b_1(x_N) & b_2(x_N) & \cdots & b_N(x_N)\end{pmatrix}\begin{pmatrix} c_1\\ c_2\\ c_3\\ \vdots\\ c_N\end{pmatrix}=\begin{pmatrix} f(x_1)\\ f(x_2)\\ f(x_2)\\ \vdots\\ f(x_N)\end{pmatrix}.$$

5. Hence, now we choose basis functions of $\Pi^r([a,b])$.

1. The same discrimination applies here, by which we took Lagrange Basis or Bernstein Basis as good ones for $\mathcal{P}^{<n+1}([a,b])$ and Power Basis or Newton Basis as bad ones.

2. We assemble polynomial to make bump-shaped functions for $i = 1, 2, \cdots, N$ with bumps distributed *evenly* over the global domain $[a, b]$.

Example: $r = 2$

Example: $r = 3$

Here is a systematic way of generating those nice bump-shaped functions for each of spline order $r = 1, 2, 3, \cdots$

1. It is convenient to take a strictly monotone increasing sequence

$$(t_i)_{i=-\infty}^{\infty} \quad t_i \to \pm\infty \text{ as } i \to \pm\infty$$

and embeds our break points $(t_i)_{i=0}^{L}$ as a part of the sequence.

2. We give doubly parametrized sequence of function definitions of

$$B_i^r(x) : \mathbb{R} \to \mathbb{R}, \quad \text{for } r \in \mathbb{N} \text{ and } i \in \mathbb{Z}.$$

We give definitions by induction.

1. At $r = 1$. For each $i \in \mathbb{Z}$, define

$$B_i^1(x) = \chi_{(t_i, t_{i+1})}(x) = \begin{cases} 1 & \text{if } x \in (t_i, t_{i+1}) \\ 0 & \text{otherwise} \end{cases}$$

2. At $r = 2$. For each $i \in \mathbb{Z}$,
   (a) We first define the normalization of order 1 splines,

$$C_i^1(x) = \frac{B_i^1(x)}{\int_{\mathbb{R}} B_i^1(x')\, dx'} = \frac{B_i^1(x)}{\int_{t_i}^{t_{i+1}} B_i^1(x')\, dx'} = \frac{B_i^1(x)}{t_{i+1} - t_i}.$$

   (b) Then, $D_i^1(x) = C_i^1(x) - C_{i+1}^1(x)$ is a function with zero integral with

$$\text{supp}(D_i^1) = [t_i, t_{i+2}].$$

   (c) We define

$$B_i^2(x) = \int_{-\infty}^{x} C_i^1(x') - C_{i+1}^1(x')\, dx'.$$

3. If $(B_i^1)_{i\in\mathbb{Z}}$, $(B_i^2)_{i\in\mathbb{Z}}$, $\cdots$, $(B_i^{r-1})_{i\in\mathbb{Z}}$ are defined,
   (a) We define the normalization of order $r-1$ splines,

$$C_i^{r-1}(x) = \frac{B_i^{r-1}(x)}{\int_{\mathbb{R}} B_i^{r-1}(x')\, dx'} = \frac{B_i^{r-1}(x)}{\int_{t_i}^{t_{i+r-1}} B_i^{r-1}(x')\, dx'}.$$

   (b) Then, $D_i^{r-1}(x) = C_i^{r-1}(x) - C_{i+1}^{r-1}(x)$ is a function with zero integral.
   (c) We define

$$B_i^r(x) = \int_{-\infty}^{x} C_i^{r-1}(x') - C_{i+1}^{r-1}(x')\, dx'.$$

Remark 11.2. We omit the proofs but a few observations can be checked:

1. $B_i^r(x)$ is nonnegative and bump-shaped.
2. The support of B_i^r is $[t_i, t_{i+r}]$.
3. Let $m = r - 1$. $B_i^r\big|_{(t_j, t_{j+1})}$ for each j is at most m-th order polynomial.
4. Let $\alpha = r - 2$. $B_i^r \in C^\alpha(\mathbb{R})$ for $r \geq 2$. B_i^1 is a bounded function.
5. In summary, B_i^r is a spline of order r, that is nonnegative, bump-shaped, and supported in $[t_i, t_{i+r}]$.
6. For $r \geq 1$, the derivative $\frac{d}{dx} B_i^r(x) = D_i^{r-1}(x)$ that is a spline of order $r - 1$.

Remark 11.3. One another property that are not readily seen is that

$$\text{for every } x \in \mathbb{R}, \quad \sum_j B_j^r(x) = 1.$$

This is because of the following.

1. Fix $x \in \mathbb{R}$, and suppose $x \in [t_i, t_{i+1}]$.
2. Since the support of $B_j^r(x)$ is a compact set $[t_j, t_{j+r}]$, we may let j runs in sufficiently but finitely many indices from N_0 to N_1. We assume that $N_0 + r < i < i + 1 < N_1$.
3. Then for this $x \in [t_i, t_{i+1}]$,

$$\begin{aligned}
\sum_{j=N_0}^{N_1} B_j^r(x) &= \int_{-\infty}^{x} \left(C_{N_0}^{r-1} - C_{N_0+1}^{r-1}\right) + \left(C_{N_0+1}^{r-1} - C_{N_0+2}^{r-1}\right) + \cdots + \left(C_{N_1-1}^{r-1} - C_{N_1}^{r-1}\right) dx' \\
&= \int_{-\infty}^{x} C_{N_0}^{r-1}(x') - C_{N_1}^{r-1}(x')\, dx' \\
&= \int_{-\infty}^{x} C_{N_0}^{r-1}(x')\, dx' = 1.
\end{aligned}$$

Summary

1. We fix a strictly monotone increasing sequence

$$(t_j)_{j=-\infty}^{\infty} \quad t_j \to \pm\infty \text{ as } j \to \pm\infty.$$

2. We define $(B_j^r(x))_{j\in\mathbb{Z},r\in\mathbb{N}}$.

3. We define the vector space of r-splines in $\mathbb{R}$

$$\Pi^r(\mathbb{R}) := \mathsf{Span}\langle \big(B_j^r(x)\big)_{j\in\mathbb{Z}} \rangle.$$

Remark 11.4. Generalized r-bumps.

1. In fact, more general definition for $B_j^r(x)$ can be given by

$$B_j^r(x) = (t_{j+r} - t_j) f_{r,x}[t_j, t_{j+1}, \cdots, t_{j+r}],$$

   where $f_{r,x}(t) = \big(\max\{t-x, 0\}\big)^{r-1}$.

2. With this more general definition, the sequence $(t_j)_{j\in\mathbb{Z}}$ needs not be strictly monotone. We leave discussion on this as further study: Carl de Boor, A practical guide to splines.

Chapter 12

Interpolation by cubic splines

1. Let us be given the data $(x_i, f(x_i))$ for $i = 0, 1, \cdots, n$.
   Assume data points (x_i) are pairwise distinct, and we let
   $$a = x_0 < x_1 < x_2 < \cdots < x_n = b.$$
2. For simpler exposition, we take a primitive strictly monotone increasing sequence
   $$(\tilde{t}_j)_{j=-\infty}^{\infty} \quad \tilde{t}_j \to \pm\infty \text{ as } j \to \pm\infty$$
   and adapt the data points $x_i = \tilde{t}_i$ for $i = 0, 1, \cdots, n$.
3. Definitions (B_j^r) will be by a subsequence (t_j) of the primitive one.

First interpolation example: by splines of order $r = 2$. (piecewise linear)

Two layers of implementation of interpolation

Let us given $(t_j)_{j\in\mathbb{Z}}$.

Our implementation of the interpolation is of two layers:

1. Implementing $x \mapsto B_j^r(x)$,
2. Implementing interpolation using $B_j^r(x)$ as basis.

In our class, we do not discuss the first layer.

The functions $(B_j^r(x))$

1. Treat $B_j^r(x)$ as functions listed with names in our dictionary of functions, not diffrerently from those functions of $\sin(x)$, $\log(x)$, $\exp(x)$, $\cdots$.
2. Scientific Library implementing them is available in Matlab, Python, Julia, $\cdots$.

Why is the matrix good ?

The graph of $B_j^4(x)$

1. This is because (B_j^r) is nonnegative, bump-shaped, and also
$$\sum_j B_j^r(x) = 1 \quad \text{for all } x.$$
2. $B_j^4(x)$ is supported in the interval $[t_j, t_{j+4}]$ over the 5 points
$$t_j < t_{j+1} < t_{j+2} < t_{j+3} < t_{j+4}.$$
The peak is near the point t_{j+2}.
3. Let us shift the index
$$\hat{B}_j^4(x) = B_{j-2}^4(x)$$
so that $\hat{B}_j^4(x)$ is supported in $[t_{j-2}, t_{j+2}]$ with a peak near t_j.
4. $\hat{B}_j^4(t_k) \neq 0$ only when $k \in \{j-1, j, j+1\}$. Similarly, $\frac{d}{dx}\hat{B}_j^4(t_k) \neq 0$ only when $k \in \{j-1, j, j+1\}$, and $\frac{d^2}{dx^2}\hat{B}_j^4(t_k) \neq 0$ only when $k \in \{j-1, j, j+1\}$.

In principle, the collocation matrix

$$\begin{pmatrix} \hat{B}_{J_0}^4(t_{J_0}) & \hat{B}_{J_0+1}^4(t_{J_0}) & \cdots & \hat{B}_{J_1}^4(t_{J_0}) \\ \hat{B}_{J_0}^4(t_{J_0+1}) & \hat{B}_{J_0+1}^4(t_{J_0+1}) & \cdots & \hat{B}_{J_1}^4(t_{J_0+1}) \\ \hat{B}_{J_0}^4(t_{J_0+2}) & \hat{B}_{J_0+1}^4(t_{J_0+2}) & \cdots & \hat{B}_{J_1}^4(t_{J_0+2}) \\ \hat{B}_{J_0}^4(t_{J_0+3}) & \hat{B}_{J_0+1}^4(t_{J_0+3}) & \cdots & \hat{B}_{J_1}^4(t_{J_0+3}) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{B}_{J_0}^4(t_{J_1}) & \hat{B}_{J_0+1}^4(t_{J_1}) & \cdots & \hat{B}_{J_1}^4(t_{J_1}) \end{pmatrix}$$

is tri-banded, diagonal dominant, $\cdots$.

The first instance of cubic spline interpolation

1. Our choice of subsequence (t_j) is such that
$$\cdots, \tilde{t}_{-1}, \tilde{t}_0, \tilde{t}_2, \tilde{t}_3, \tilde{t}_4, \cdots, \tilde{t}_{n-3}, \tilde{t}_{n-2}, \tilde{t}_n, t_{n+1}, \cdots$$
to be
$$\cdots, t_{-1}, t_0, t_1, t_2, t_3, \cdots, t_{L-2}, t_{L-1}, t_L, t_{L+1}, \cdots$$
and $L = n - 2$.
2. This is to discard x_1 and x_{n-1} from $(\tilde{t}_j)$
3. We take a vector space
$$\Pi([a, b]) = \mathsf{Span}\langle(\hat{B}^r)_{j=-1}^{L+1}\rangle$$
that coincides with the vector space of splines of order 4 with break points $(t_\ell)_{\ell=0}^{L}$.
4. Its dimension $\dim \Pi([a, b]) = n + 1 = m + L = 3 + (n - 2)$.
5. This is known as "not a knot" cubic spline interpolation of sample data $(x_i, f(x_i))_{i=0}^{n}$.
6. The matrix is
$$\begin{pmatrix} \hat{B}^4_{-1}(x_0) & \hat{B}^4_0(x_0) & \cdots & \hat{B}^4_{L+1}(x_0) \\ \hat{B}^4_{-1}(x_1) & \hat{B}^4_0(x_1) & \cdots & \hat{B}^4_{L+1}(x_1) \\ \hat{B}^4_{-1}(x_2) & \hat{B}^4_0(x_2) & \cdots & \hat{B}^4_{L+1}(x_2) \\ \hat{B}^4_{-1}(x_3) & \hat{B}^4_0(x_3) & \cdots & \hat{B}^4_{L+1}(x_3) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{B}^4_{-1}(x_{n-1}) & \hat{B}^4_0(x_{n-1}) & \cdots & \hat{B}^4_{L+1}(x_{n-1}) \\ \hat{B}^4_{-1}(x_n) & \hat{B}^4_0(x_n) & \cdots & \hat{B}^4_{L+1}(x_n) \end{pmatrix}$$

The second instance of cubic spline interpolation

1. Our choice of subsequence (t_j) is same as $(\tilde{t}_j)$,

$$\cdots, \tilde{t}_{-1}, \tilde{t}_0, \tilde{t}_1, \tilde{t}_2, \tilde{t}_3, \cdots, \tilde{t}_{n-2}, \tilde{t}_{n-1}, \tilde{t}_n, t_{n+1}, \cdots$$

   to be

$$\cdots, t_{-1}, t_0, t_1, t_2, t_3, \cdots, t_{L-2}, t_{L-1}, t_L, t_{L+1}, \cdots$$

   and $L = n$.

2. We take a vector space

$$\Pi([a,b]) = \mathsf{Span}\langle (\hat{B}^r_j)_{j=-1}^{L+1} \rangle$$

   that coincides with the vector space of splines of order 4 with break points $(t_\ell)_{\ell=0}^{L}$.

3. Its dimension $\dim \Pi([a,b]) = n + 3 = m + L$.

4. To the given sample data $(x_i, f(x_i))_{i=0}^n$, we append $(x_0, f''(x_0)) = (x_0, 0)$ and $(x_n, f''(x_n)) = (x_n, 0)$.

   This makes the spline end linearly near the boundary.

5. This is known as "natural" cubic spline interpolation of sample data $(x_i, f(x_i))_{i=0}^n$.

6. The matrix is

$$\begin{pmatrix}
\hat{B}^{4''}_{-1}(x_0) & \hat{B}^{4''}_{0}(x_0) & \cdots & \hat{B}^{4''}_{L+1}(x_0) \\
\hat{B}^{4}_{-1}(x_0) & \hat{B}^{4}_{0}(x_0) & \cdots & \hat{B}^{4}_{L+1}(x_0) \\
\hat{B}^{4}_{-1}(x_1) & \hat{B}^{4}_{0}(x_1) & \cdots & \hat{B}^{4}_{L+1}(x_1) \\
\hat{B}^{4}_{-1}(x_2) & \hat{B}^{4}_{0}(x_2) & \cdots & \hat{B}^{4}_{L+1}(x_2) \\
\hat{B}^{4}_{-1}(x_3) & \hat{B}^{4}_{0}(x_3) & \cdots & \hat{B}^{4}_{L+1}(x_3) \\
\vdots & \vdots & \vdots & \vdots \\
\hat{B}^{4}_{-1}(x_n) & \hat{B}^{4}_{0}(x_n) & \cdots & \hat{B}^{4}_{L+1}(x_n) \\
\hat{B}^{4''}_{-1}(x_n) & \hat{B}^{4''}_{0}(x_n) & \cdots & \hat{B}^{4''}_{L+1}(x_n)
\end{pmatrix}$$

Remark 12.1. Why not using $\mathsf{Span}\langle (\hat{B}^r)_{j=0}^n \rangle$?

Chapter 13

Initial Value Problem of Nonlinear ODE

We consider a system whose dynamics is described by a state vector function of time t, $t \mapsto u(t) \in \mathbb{R}^n$. The function $t \mapsto u(t)$ obeys the differential equation:

$$\begin{aligned} u'(t) + f(t, u(t)) &= g(t) \quad t \in (0, T), \\ u(0) &= \alpha. \end{aligned}$$

Here,

1. The function $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ is the nonlinearity. We assume sufficient differentibility of f.
2. The function $g(t)$ is the right-hand-side or the source term.
3. α is the initial condition.

Error analysis framework:

1. We let X be the set where those functions $u(t)$ lie in. We will choose this precisely later.
2. We let $Y = Y_1 \times Y_2$. Y_1 is the set where those functions $g(t)$ lie in and we will choose this precisely later. Y_2 is the set where initial conditions lie in. $Y_2 = \mathbb{R}^n$.
3. We let $F : X \to Y$,
$$F(u) = \begin{pmatrix} u'(t) + f(t, u(t)) \\ u(0) \end{pmatrix}$$
4. We analyze the solution finding problem
$$F(u) = \begin{pmatrix} g(t) \\ \alpha \end{pmatrix}.$$

Choice of X and Y

Theory of ODE can be formulated in many different suitable ways. We find it convenient to use space of functions of Lipschitz class.

1. Define $B([0,T])$ to be the set of all bounded functions in the interval $[0,T]$.
2. Define

$$C^{0,1}([0,T]) = \big\{u \in B([0,T]) \;\big|\; \exists L > 0 \quad |u(t) - u(t')| \le L|t-t'| \quad \text{for all } t,t' \in [0,T].\big\}.$$

3. Define for $k \ge 1$

$$C^{k,1}([0,T]) = \big\{u \in C^k([0,T]) \;\big|\; \exists L > 0 \quad |u^{(k)}(t) - u^{(k)}(t')| \le L|t-t'| \quad \text{for all } t,t' \in [0,T].\big\}.$$

Considering the differential equation, provided f is sufficiently smooth, for $u \in C^{k,1}$,

$$u'(t) + f(t,u(t)) \in C^{k-1,1}.$$

For the choice of the pair of sets (X, Y_1),

$$(C^{0,1}, B), \quad (C^{1,1}, C^{0,1}), \quad (C^{2,1}, C^{1,1}), \cdots$$

will be considered.

It is also noteworthy that spaces of r splines are suitable subspaces of the choice.

$$\begin{aligned} C^{0,1} \times B \quad &\supset \Pi^2 \times \Pi^1, \\ C^{1,1} \times C^{0,1} &\supset \Pi^3 \times \Pi^2, \\ C^{2,1} \times C^{1,1} &\supset \Pi^4 \times \Pi^3, \\ &\vdots \end{aligned}$$

Methods we design

Let $n = 1$. Below, we illustrate how an approximate solver can be designed.

The map F is modified by two stages:

(i) interpolation/sampling (ii) modified problem formulation using (i).

(i) **Discretization**

1. We consider the partition of the interval $[0, T]$,

$$0 = t_0 < t_1 < t_2 < \cdots < t_L = T, \quad k_i = t_i - t_{i-1}, \quad k = \max_i k_i,$$

   where $i = 1, 2, \cdots, L$.

2. Assume $u \in \Pi^2$, $g \in \Pi^1$.

   (a) We let the sampling procedure

$$\Pi^2 \ni u \quad \mapsto \quad (u(t_i))_{i=0}^L \in \mathbb{R}^{L+1}$$

   be denoted by $P_2(u)$. Its inverse P_2^{-1} is the map

$$\mathbb{R}^{L+1} \ni (u(t_i))_{i=0}^L \quad \mapsto \quad \sum_{i=0}^{L} u(t_i) B_{i-1}^2(t) \in \Pi^2.$$

   (b) Similarly, (if we use the right value sampling)

$$\Pi^1 \ni g \quad \mapsto \quad (g(t_i))_{i=1}^L \in \mathbb{R}^L$$

   be denoted by $P_1(g)$. Its inverse P_1^{-1} is the map

$$\mathbb{R}^{L} \ni (g(t_i))_{i=1}^L \quad \mapsto \quad \sum_{i=1}^{L} g(t_i) B_{i-1}^1(t) \in \Pi^1.$$

Remark 13.1. P_2 is defined on X in the same manner for a choice X in the previous discussion. Similarly, P_1 is defined on Y_1.

(*ii*) **Modified problem**

We pose a modified problem to find an approximate solution U at t_i. More specifically,

1. The modified problem consists of

$$F_k : u(t) \quad \overset{P_2}{\longmapsto} \quad (u(t_i)) \quad \overset{F_{*k}}{\longmapsto} \quad \Big(\big(G(t_i)\big), u(0)\Big) \quad \overset{(P_1^{-1}, id)}{\longmapsto} (G(t), u(0)).$$

2. We devise a nonlinear map F_{*k}

$$F_{*k} : \mathbb{R}^{L+1} \to \mathbb{R}^L \times \mathbb{R},$$

   so that F_k is a suitable approximation of F.

3. We solve the F_{*k} map equation for given $\Big(\big(g(t_i)\big), \alpha\Big) \in R^{L+1}$, producing $U(t_i)_{i=0,1,\cdots,L} \in \mathbb{R}^{L+1}$, an approximate solution. This is the solver M_{*k}.

4. Out of data $(t_i, U(t_i))$, we reconstruct the spline of order 2, the function $t \mapsto U(t) \in \Pi^2$.

5. This is the approximate solver M_k, parametrized by the time spacing size k.

Remark 13.2. We let

$$F_* : u(t) \mapsto \Big(\big(u'(t_i) + f(t_i, u(t_i))\big)_{i=1}^{L} \ , \ u(0)\Big).$$

In many occasions, error analysis is done between F_* and $F_{*k} \circ P_2$.

First method: Explicit Euler

We will examine several methods in the following orders:

1. Explicit Euler,
2. Implicit Euler,
3. a few more methods,
4. methods in the framework of Runge-Kutta methods.

Let $k = \frac{T}{L} > 0$ be fixed. Explicit Euler is to pose F_{*k} that is the system of equations on $(u(t_i))$,

$$\begin{aligned}
u(t_0) &= \alpha \\
\frac{u(t_1) - u(t_0)}{k} + f(t_0, u(t_0)) &= g(t_0) \\
\frac{u(t_2) - u(t_1)}{k} + f(t_1, u(t_1)) &= g(t_1) \\
\frac{u(t_3) - u(t_2)}{k} + f(t_2, u(t_2)) &= g(t_2) \\
&\vdots \\
\frac{u(t_{i+1}) - u(t_i)}{k} + f(t_i, u(t_i)) &= g(t_i) \\
&\vdots \\
\frac{u(t_L) - u(t_{L-1})}{k} + f(t_{L-1}, u(t_{L-1})) &= g(t_{L-1}).
\end{aligned}$$

(The order in the right-hand-side is flipped: $\big(\alpha, g(t_0), \cdots\big)$.)

1. Note that equations are easily solvable from the top to bottom.
2. The solver is denoted by M_{*k} and thus

$$M_{*k} : \big(\alpha, g(t_0), g(t_1), \cdots\big) \quad \mapsto \quad \Big(U(t_0), U(t_1), \cdots U(t_L)\Big)$$

a priori error analysis of Explicit Euler

Testing error

1. Omitting the intermediate discretization procedure, we work with F_* and $F_{*k} \circ P_2$.
2. Let $\hat{u}(t)$ be the true solution, which we do not know. We assume X is with sufficient differentiability.
3. We let the sampling data $\hat{U} = (\hat{u}(t_i)) = P_2(\hat{u})$.
4. We compute the consistency level of F_{*k} at the true solution $\hat{u}(t)$.

$$\left|F_{*k}(\hat{U}) - F_*(\hat{u})\right| = \left| \begin{pmatrix} \hat{U}(0) - \hat{u}(0) \\ \frac{\hat{U}(t_1)-\hat{U}(t_0)}{k} + f(t_0, \hat{U}(t_0)) - \big(u'(t_0) + f(t_0, \hat{u}(t_0)\big) \\ \frac{\hat{U}(t_2)-\hat{U}(t_1)}{k} + f(t_1, \hat{U}(t_1)) - \big(u'(t_1) + f(t_1, \hat{u}(t_1)\big) \\ \vdots \\ \frac{\hat{U}(t_L)-\hat{U}(t_{L-1})}{k} + f(t_{L-1}, \hat{U}(t_{L-1})) - \big(u'(t_{L-1}) + f(t_{L-1}, \hat{u}(t_{L-1}))\big) \end{pmatrix} \right|$$

5. Let us use the length $|(a_1, a_2, \cdots, a_m)| = \max\limits_{i=1,\cdots,m} |a_i|$.
6. From the second row, the difference is denoted by

$$\tau_{i+1} = \frac{\hat{U}(t_{i+1}) - \hat{U}(t_i)}{k} + f(t_i, \hat{U}(t_i)) - \big(u'(t_i) + f(t_i, \hat{u}(t_i))\big), \quad i = 0, 1, \cdots, L-1.$$

7. We can write

$$\tau_{i+1} = \hat{u}[t_i, t_{i+1}] - \hat{u}[t_i, t_i] = (t_{i+1} - t_i)\hat{u}[t_i, t_i, t_{i+1}],$$

   provided $\hat{u}$ is sufficiently differentiable.

8. Using that

$$\left|\hat{u}[t_i, t_i, t_{i+1}]\right| \leq \frac{1}{2}\|\hat{u}''(t)\|_\infty$$

   we can conclude that the testing error (consistency level) is bounded by

$$\frac{k}{2}\|\hat{u}''(t)\|_\infty.$$

9. Standard terminology of $k \times$ consistency level is the Local Truncation Error.
10. We have checked that

$$\left|F_{*k}(\hat{U}) - F_*(\hat{u})\right| \leq \frac{k}{2}\|\hat{u}''(t)\|_\infty.$$

Solution error

For simpler exposition, we assume the global Lipschitz condition for the nonlinearity

$\exists \Lambda > 0$ such that for every $t \in [0,T]$ and every a, b $|f(t,a) - f(t,b)| \le \Lambda |a-b|$.

Continuity property of the approximate solver $M_{*k} = F_{*k}^{-1}$.

1. For given two (RHS) $\big(\alpha, g(t_0), g(t_1), \cdots\big)$ and $\big(\beta, h(t_0), h(t_1), \cdots\big)$, we let
$$\Big(U(t_0), U(t_1), \cdots U(t_L)\Big) = M_{*k}\big(\alpha, g(t_0), g(t_1), \cdots\big),$$
$$\Big(V(t_0), V(t_1), \cdots V(t_L)\Big) = M_{*k}\big(\beta, h(t_0), h(t_1), \cdots\big),$$
$$E_i = U(t_i) - V(t_i) \quad i = 0, 1, \cdots, L.$$

2. We also let
$$k\tau_0 = \alpha - \beta$$
$$\Big[\frac{U(t_1)-U(t_0)}{k} + f(t_0, U(t_0))\Big] - \Big[\frac{V(t_1)-V(t_0)}{k} + f(t_0, V(t_0))\Big] = \tau_1 = g(t_0) - h(t_0)$$
$$\Big[\frac{U(t_2)-U(t_1)}{k} + f(t_1, U(t_1))\Big] - \Big[\frac{V(t_2)-V(t_1)}{k} + f(t_1, V(t_1))\Big] = \tau_2 = g(t_1) - h(t_1)$$
$$\vdots$$
$$\Big[\frac{U(t_L)-U(t_{L-1})}{k} + f(t_{L-1}, U(t_{L-1}))\Big] - \Big[\frac{V(t_L)-V(t_{L-1})}{k} + f(t_{L-1}, V(t_{L-1}))\Big] = \tau_L = g(t_{L-1}) - h(t_{L-1}).$$

3. We deduce that
$$|E_0| = k|\tau_0| \le k \max_i |\tau_i| = k|\tau|$$
and for $i = 1, 2, \cdots, L$
$$|E_i| \le |E_{i-1}| + k\big(|\tau_i| + \Lambda |E_{i-1}|\big) \le (1 + k\Lambda)|E_{i-1}| + k \max_i |\tau_i| \le e^{k\Lambda}|E_{i-1}| + k|\tau|$$

4. We claim that for $i = 0, 1, 2, \cdots, L$,
$$|E_i| \le k|\tau|\Big(1 + e^{k\Lambda} + e^{2k\Lambda} + e^{3k\Lambda} + \cdots + e^{ik\Lambda}\Big)$$
Certainly, $|E_0| \le k|\tau|$. If $|E_{i-1}| \le k|\tau|\Big(1 + e^{k\Lambda} + e^{2k\Lambda} + e^{3k\Lambda} + \cdots + e^{(i-1)k\Lambda}\Big)$,
$$|E_i| \le e^{k\Lambda}|E_{i-1}| + k|\tau| \le k|\tau|\Big(1 + e^{k\Lambda} + e^{2k\Lambda} + e^{3k\Lambda} + \cdots + e^{ik\Lambda}\Big).$$

5. Now,
$$\begin{aligned}
|E| &\le k|\tau|\Big(1 + e^{k\Lambda} + e^{2k\Lambda} + e^{3k\Lambda} + \cdots + e^{Lk\Lambda}\Big) \\
&= k|\tau|e^{Lk\Lambda}\Big(1 + e^{-k\Lambda} + e^{-2k\Lambda} + \cdots + e^{-Lk\Lambda}\Big) \\
&= k|\tau|e^{T\Lambda}\Big(1 + e^{-k\Lambda} + e^{-2k\Lambda} + \cdots + e^{-Lk\Lambda}\Big) \\
&\le k|\tau|e^{T\Lambda}\frac{1}{1 - e^{-k\Lambda}} = \frac{k}{1 - e^{-k\Lambda}}e^{T\Lambda}|\tau| \le \frac{e^{2T\Lambda}}{\Lambda}|\tau|
\end{aligned}$$
From the inspection of the continuity of M_{*k}, we have that
$$|\hat{U} - U| = \Big|M_{*k}\Big(F_{*k}(\hat{U})\Big) - M_{*k}\Big(F_*(\hat{u})\Big|\Big| \le \Big(\frac{e^{2T\Lambda}}{2\Lambda}\|\hat{u}''(t)\|_\infty\Big)k$$

Second method: Implicit Euler

Let $k = \frac{T}{L} > 0$ be fixed. Implicit Euler is to pose F_{*k} that is the system of equations on $(u(t_i))$,

$$\begin{aligned}
u(t_0) &= \alpha \\
\frac{u(t_1) - u(t_0)}{k} + f(t_1, u(t_1)) &= g(t_1) \\
\frac{u(t_2) - u(t_1)}{k} + f(t_2, u(t_2)) &= g(t_2) \\
\frac{u(t_3) - u(t_2)}{k} + f(t_3, u(t_3)) &= g(t_3) \\
&\vdots \\
\frac{u(t_{i+1}) - u(t_i)}{k} + f(t_{i+1}, u(t_{i+1})) &= g(t_{i+1}) \\
&\vdots \\
\frac{u(t_L) - u(t_{L-1})}{k} + f(t_L, u(t_L)) &= g(t_L).
\end{aligned}$$

1. Note that this is a nonlinear system of equations.
2. Practically, newton method can be considered to solve this system.
3. The solver is denoted by M_{*k} and thus

$$M_{*k} : \big(\alpha, g(t_1), g(t_2), \cdots\big) \quad \mapsto \quad \Big(U(t_0), U(t_1), \cdots U(t_L)\Big)$$

a priori error analysis of Implicit Euler

Testing error

1. Omitting the intermediate discretization procedure, we work with F_* and $F_{*k} \circ P_2$.
2. Let $\hat{u}(t)$ be the true solution, which we do not know. We assume X is with sufficient differentiability.
3. We let the sampling data $\hat{U} = (\hat{u}(t_i)) = P_2(\hat{u})$.
4. We compute the consistency level of F_{*k} at the true solution $\hat{u}(t)$.

$$\left|F_{*k}(\hat{U}) - F_*(\hat{u})\right| = \left| \begin{pmatrix} \hat{U}(0) - \hat{u}(0) \\ \frac{\hat{U}(t_1)-\hat{U}(t_0)}{k} + f(t_1, \hat{U}(t_1)) - \big(u'(t_1) + f(t_1, \hat{u}(t_1)\big) \\ \frac{\hat{U}(t_2)-\hat{U}(t_1)}{k} + f(t_2, \hat{U}(t_2)) - \big(u'(t_2) + f(t_2, \hat{u}(t_2)\big) \\ \vdots \\ \frac{\hat{U}(t_L)-\hat{U}(t_{L-1})}{k} + f(t_L, \hat{U}(t_L)) - \big(u'(t_L) + f(t_L, \hat{u}(t_L))\big) \end{pmatrix} \right|$$

5. From the second row, the difference is denoted by

$$\tau_{i+1} = \frac{\hat{U}(t_{i+1}) - \hat{U}(t_i)}{k} + f(t_{i+1}, \hat{U}(t_{i+1})) - \big(u'(t_{i+1}) + f(t_{i+1}, \hat{u}(t_{i+1}))\big), \quad i = 0, 1, \cdots, L-1.$$

6. We can write

$$\tau_{i+1} = \hat{u}[t_i, t_{i+1}] - \hat{u}[t_{i+1}, t_{i+1}] = -(t_{i+1} - t_i)\hat{u}[t_i, t_{i+1}, t_{i+1}],$$

   provided $\hat{u}$ is sufficiently differentiable.

7. Using that

$$\left|\hat{u}[t_i, t_{i+1}, t_{i+1}]\right| \leq \frac{1}{2}\|\hat{u}''(t)\|_\infty$$

   we can conclude that the testing error (consistency level) is bounded by

$$\frac{k}{2}\|\hat{u}''(t)\|_\infty.$$

8. We have checked that

$$\left|F_{*k}(\hat{U}) - F_*(\hat{u})\right| \leq \frac{k}{2}\|\hat{u}''(t)\|_\infty.$$

Solution error

Continuity property of the approximate solver $M_{*k} = F_{*k}^{-1}$.

1. For given two (RHS) $\big(\alpha, g(t_1), g(t_2), \cdots\big)$ and $\big(\beta, h(t_1), h(t_2), \cdots\big)$, we let
$$\Big(U(t_0), U(t_1), \cdots U(t_L)\Big) = M_{*k}\big(\alpha, g(t_1), g(t_2), \cdots\big),$$
$$\Big(V(t_0), V(t_1), \cdots V(t_L)\Big) = M_{*k}\big(\beta, h(t_1), h(t_2), \cdots\big),$$
$$E_i = |U(t_i) - V(t_i)| \quad i = 0, 1, \cdots, L.$$

2. We also let
$$k\tau_0 = \alpha - \beta$$
$$\Big[\frac{U(t_1) - U(t_0)}{k} + f(t_1, U(t_1))\Big] - \Big[\frac{V(t_1) - V(t_0)}{k} + f(t_1, V(t_1))\Big] = \tau_1 = g(t_1) - h(t_1)$$
$$\Big[\frac{U(t_2) - U(t_1)}{k} + f(t_2, U(t_2))\Big] - \Big[\frac{V(t_2) - V(t_1)}{k} + f(t_2, V(t_2))\Big] = \tau_2 = g(t_2) - h(t_2)$$
$$\vdots$$
$$\Big[\frac{U(t_L) - U(t_{L-1})}{k} + f(t_L, U(t_L))\Big] - \Big[\frac{V(t_L) - V(t_{L-1})}{k} + f(t_L, V(t_L))\Big] = \tau_L = g(t_L) - h(t_L).$$

3. We deduce that
$$|E_0| = k|\tau_0| \le k \max_i |\tau_i| = k|\tau|$$
and for $i = 1, 2, \cdots, L$
$$|E_i|(1 - k\Lambda) \le |E_{i-1}| + k|\tau_i| \le |E_{i-1}| + k|\tau|.$$
Provided that $k\Lambda < \frac{1}{2}$,
$$|E_i| \le \frac{1}{1 - k\Lambda}\Big(|E_{i-1}| + k|\tau|\Big) \le (1 + 2k\Lambda)\Big(|E_{i-1}| + k|\tau|\Big) \le e^{2k\Lambda}\Big(|E_{i-1}| + k|\tau|\Big).$$

4. We claim that for $i = 1, 2, \cdots, L$,
$$E_i \le 2k|\tau|\Big(e^{2k\Lambda} + e^{4k\Lambda} + e^{6k\Lambda} + \cdots + e^{2ik\Lambda}\Big)$$
Certainly, $E_1 \le e^{2k\Lambda}\Big(|E_0| + k|\tau|\Big) \le 2k|\tau|e^{2k\Lambda}$.
If $E_{i-1} \le 2k|\tau|\Big(e^{2k\Lambda} + e^{4k\Lambda} + e^{6k\Lambda} + \cdots + e^{2(i-1)k\Lambda}\Big)$
$$E_i \le e^{2k\Lambda}\Big(E_{i-1} + 2k|\tau|\Big) \le 2k|\tau|\Big(e^{2k\Lambda} + e^{4k\Lambda} + e^{6k\Lambda} + \cdots + e^{2ik\Lambda}\Big).$$

5. Now,
$$|E| \le 2k|\tau|e^{2Lk\Lambda}\Big(1 + e^{-2k\Lambda} + e^{-4k\Lambda} + e^{-6k\Lambda} + \cdots + e^{(-2Lk+2k)\Lambda}\Big)$$
$$\le 2k|\tau|e^{2Lk\Lambda}\frac{1}{1 - e^{-2k\Lambda}} = \frac{2k}{1 - e^{-2k\Lambda}}e^{2T\Lambda}|\tau| \le \frac{e^{3T\Lambda}}{\Lambda}|\tau|.$$

From the inspection of the continuity of M_{*k}, we have that
$$|\hat{U} - U| = \Big|M_{*k}\Big(F_{*k}(\hat{U})\Big) - M_{*k}\Big(F_*(\hat{u})\Big)\Big| \le \Big(\frac{e^{3T\Lambda}}{2\Lambda}\|\hat{u}''(t)\|_\infty\Big)k$$

Chapter 14

True Solver continuity of an Initial Value Problems

Consider an initial value problem

$$\begin{aligned} u'(t) + f(t,u) &= g(t) \quad t \in (0,T) \\ u(0) &= \alpha \end{aligned}$$

Let us denote the true solver $M^f : (\alpha, g) \mapsto u$.

1. Let us first agree that it does not make much senses to expect a *well-conditioned* numerical algorithm $\tilde{M}$ while true solver M is *ill-conditioned*.
2. We have not discussed the theoretical counter parts on IVP: existence, true solver continuity, $\cdots$
3. We classify IVP, or classify M^f into ones ill-conditioned and the others well-conditioned.

To simplify our exposition, let us consider an IVP

$$\begin{aligned} u'(t) + f(u) &= 0 \quad t \in (0,T) \\ u(0) &= \alpha, \end{aligned}$$

where $g \equiv 0$, and $f = f(u)$.

The solver is thus a map $M^f : \alpha \mapsto u$.

The 1st example

Let $u'(t) = u^2(t), \quad u(0) = 1.$

We can solve the equation by hands,

Note that as long as solution $u(t)$ exists until time $t > 0$, $u(t) \geq 1 \neq 0$.

$$u^{-2}(t)u'(t) = 1 \quad \Longleftrightarrow \quad \Big(-\frac{u^{-3}(t)}{3} \Big)' = 1.$$

Integrating from 0 to $t > 0$ $\quad -\dfrac{1}{3u^3(t)} + \dfrac{1}{3u^3(0)} = t \quad \Longrightarrow 3u^3(t) = \dfrac{1}{\frac{1}{3} - t}.$

We conclude that $u(t) \to \infty$ as $t \nearrow \frac{1}{3}$.

Globally Lipschitz f in relatively short time

Let us examine the continuity of $M^f = M^f(\alpha)$, when f is globally Lipschitz. Suppose that

$$\begin{aligned} u' + f(u) = 0, \quad u(0) = \alpha, \\ v' + f(v) = 0, \quad v(0) = \beta. \end{aligned}$$

1. Subtracting the equations,

$$(u - v)' + f(u) - f(v) = 0, \quad u(0) - v(0) = \alpha - \beta.$$

2. Multiply $(u - v)$ both sides,

$$\Big(\frac{(u - v)^2}{2}\Big)' = -f[u, v](u - v)^2 \leq \Lambda(u - v)^2.$$

3. With $e(t) = (u(t) - v(t))^2$,

$$e' \leq 2\Lambda e, \quad e(0) = (\alpha - \beta)^2$$

4. We conclude that

$$(u(T) - v(T))^2 \leq e^{2\Lambda T}(\alpha - \beta)^2.$$

To make the solution error small, we see that in case f is globally Lipschitz, $e^{\Lambda T}$ needs to be kept not too big.

A dissipative system

Let $\lambda > 0$,

$$u' = -\lambda u, \quad u(0) = \alpha, \quad v' = -\lambda v, \quad v(0) = \beta.$$

Then

$$u(t) = \alpha e^{-\lambda t}, \quad v(t) = \beta e^{-\lambda t}, \quad |u(T) - v(T)| \leq e^{-\lambda T}|\alpha - \beta| \leq |\alpha - \beta|.$$

Typical Lifespan of an orbit

Transient phase vs dissipating phase vs blowing-up phase

Remark 14.1 (Warning). The exposition here is not saying that above three behaviors comprise every possible scenario. It is saying that we typically encounter such a scenario generically.

Let us consider

$$u' = u(1-u), \quad u(0) = \alpha.$$

Blowing-up phase

1. Suppose $\alpha = -1$. Then $u \searrow -\infty$ as t increases.

Transient phase

1. Suppose $\alpha = 0.1$. Then u increases as t increases, and soon, u surpasses the value 0.5.

Dissipating phase

1. Suppose $\alpha = 0.7$. Then u increases as t increases, converging to 1, with the following details.
   (a) As a matter of fact, it takes infinite time to convere to 1.
   (b) It gets close to 1 exponentially fast, however.

Rough statement for an IVP of nonlinear ODE:

1. If initial data α is taken, in short time it undergoes some transient phase.
2. This may end up with entering into blowing-up phase. The blowing-up may occur in finite time or in infinite time.
3. If the orbit does not enter the blowing-up phase, it may dissipate to a certain stable state, being kept in dissipative phase.
4. It is qualitatively possible that the lifespan of the orbit experiences several of transient-dissipative phases, or the orbit may exhibit periodic behaviors.
5. Or otherwise, some chaotic behavior is possible, but let us not discuss this ODE theoretical subject.

Conclusion

Our algorithm is designed for the two generic purposes, (i) to capture relatively short time transient behavior, and (ii) to capture relatively long time dissipative behavior.

Indeed, those two cases are when the True solver $M^f : \alpha \mapsto u$ is well-conditioned.

1. $(u(T) - v(T))^2 \leq e^{\Lambda T}(\alpha - \beta)^2$ and we assume $e^{\Lambda T}$ is not too big.
2. To take an example of dissipative phase, let $u' = -u$. Then

$$u(t) = \alpha e^{-t}, \quad v(t) = \beta e^{-t}, \quad |u(T) - v(T)| \leq e^{-T}|\alpha - \beta|.$$

Explicit vs. Implicit

1. Explicit Euler and Implicit Euler both can capture the relatively short time transient behavior, say when $e^{\Lambda T}$ is not too big.
2. Explicit Euler may not be able to capture the dissipative behavior, if dissipation occurs very fast. Let us be precise on this statement below.

Suppose $u'(t) = -3000u$, $u(0) = 1$ and let us run the Explicit Euler:

$$\begin{aligned}
u(t_0) &= 1 \\
\frac{u(t_1) - u(t_0)}{k} + 3000u(t_0) &= 0 \\
\frac{u(t_2) - u(t_1)}{k} + 3000u(t_1) &= 0 \\
\frac{u(t_3) - u(t_2)}{k} + 3000u(t_2) &= 0 \\
&\vdots \\
\frac{u(t_{i+1}) - u(t_i)}{k} + 3000u(t_i) &= 0 \\
&\vdots \\
\frac{u(t_L) - u(t_{L-1})}{k} + 3000u(t_{L-1}) &= 0.
\end{aligned}$$

or simply,

$$u(t_{i+1}) = (1 - 3000k)u(t_i), \quad i = 0, 1, \cdots, L-1, \quad u(t_0) = 1.$$

1. We observe that unless $|1 - 3000k| < 1$, $u(t)$ does not dissipate to the zero state. Not only that, it even geometrically grows.
2. $k < \frac{1}{3000}$ may not look serious restriction. This is because Explicit Euler is a 1st order algorithm. Higher order explicit algorithm may have a significant restriction.

This calculation is contrasted to the Implicit Euler calculations

$$\begin{aligned}
u(t_0) &= 1 \\
\frac{u(t_1)-u(t_0)}{k} + 3000u(t_1) &= 0 \\
\frac{u(t_2)-u(t_1)}{k} + 3000u(t_2) &= 0 \\
\frac{u(t_3)-u(t_2)}{k} + 3000u(t_3) &= 0 \\
&\vdots \\
\frac{u(t_{i+1})-u(t_i)}{k} + 3000u(t_{i+1}) &= 0 \\
&\vdots \\
\frac{u(t_L)-u(t_{L-1})}{k} + 3000u(t_L) &= 0.
\end{aligned}$$

or simply,

$$u(t_{i+1})\big(1+3000k\big) = u(t_i) \iff u(t_{i+1}) = \frac{1}{1+3000k}u(t_i), \quad i = 0, 1, \cdots, L-1, \quad u(t_0) = 1.$$

We observe that for any $k > 0$, u will dissipate to the zero state geometrically fast, which should be the case.

Remark 14.2. One notices that if sign is reversed, say $u' = 3000u$, then behavior of Implicit Euler goes weird. But from the first place we do not (in general) attempt to solve this IVP to capture the exponential blowing-up.

We formulate this abnormal behavior of Explicit Euler for the fast dissipative problem as an inconsistency of an approximate solver M_{*k}

We recall that the consistency of an approximate solver M_{*k} (parametrized by k here) consists of two conditions,

1. $M_{*k}(y) \in X$,
2. $|y - F\big(M_{*k}(y)\big)| \leq \eta_k$

We take the first condition here in a narrower sense for dissipative phase solving, that (if we know what u_* is)

$$\hat{X} = \big\{u \mid |u(t) - u_*| \text{ does not grow}\big\}, \quad M^f_{*k}(y) \in \hat{X}.$$

To take an example, if $f(u) = \lambda u$, $\lambda > 0$ (i.e., $u' = -\lambda u$) we may take

$$\hat{X} = \big\{u \mid |u(t)| \text{ does not grow}\big\}.$$

In this sense, Explicit Euler with some k cannot pass the consistency condition until k gets unreasonably small.

Implicit Euler is always consistent for a dissipative phase solving.

Remark 14.3. .

1. Consistency is formulated as a necessary condition for an algorithm, and thus if an algorithm fails to be consistent, further solution error analysis does not make much senses.
2. In the ODE literature, this failure is formulated as a failure of "stability". Since stability in numerical analysis means too many different things, we kept using the terminology "consistency".

The region of stability (non-growth) for $u' = \lambda u$.

1. It is very difficult to know in advance whether an algorithm passes the consistency condition for each of a nonlinear ODE problem.
2. For a given algorithm, we at least check if it passes the non-growth condition for a linear problem $u' = \lambda u$.

Explicit Euler

$$u(t_{i+1}) = (1 + \lambda k)u(t_i).$$

Implicit Euler

$$u(t_{1+1}) = \frac{1}{1 - \lambda k}u(t_i).$$

The quantity λk, where $k > 0$ and $\lambda \in \mathbb{C}$, is denoted by a symbol $z \in \mathbb{C}$.

We identify the region $K \ni z$ so that $|u(t_{i+1})| \leq |u(t_i)|$. (to see if z with negative real part does gives the decay.)

Explicit Euler

$$K = \{z \in \mathbb{C} \mid |1 + z| \leq 1.\}$$

Implicit Euler

$$K = \{z \in \mathbb{C} \mid |1 - z| \geq 1.\}$$

Chapter 15

High order (in testing error) methods

1. we look for a method so that the testing error is of order k^r:

$$\text{The true solution } \hat{u} \in C^\infty \quad \Longrightarrow \quad \max_i |F_{*k}(\hat{u})(t_i) - F_*(\hat{u})(t_i)| \leq \eta_k \leq Ck^r$$

for some constant $C > 0$ and $r \geq 1$.

Examples

Suppose we design F_{*k} such that i-th equation reads

$$\frac{1}{k}\begin{pmatrix}\alpha_{i+q} & \cdots & \alpha_{i+1} & \alpha_i & \alpha_{i-1} & \cdots & \alpha_{i-q'}\end{pmatrix}\begin{pmatrix}u(t_{i+q})\\ \cdots\\ u(t_{i+1})\\ u(t_i)\\ u(t_{i-1})\\ \cdots\\ u(t_{i-q'})\end{pmatrix}$$

$$+\begin{pmatrix}\beta_{i+q} & \cdots & \beta_{i+1} & \beta_i & \beta_{i-1} & \cdots & \beta_{i-q'}\end{pmatrix}\begin{pmatrix}f\big(t_{i+q},u(t_{i+q})\big)\\ \cdots\\ f\big(t_{i+1},u(t_{i+1})\big)\\ f\big(t_i\quad,u(t_i)\quad\big)\\ f\big(t_{i-1},u(t_{i-1})\big)\\ \cdots\\ f\big(t_{i-q'},u(t_{i-q'})\big)\end{pmatrix}=g(t_i).$$

For the true solution $\hat{u}$, $\hat{u}'(t_i)+f\big(t_i,u(t_i)\big)=g(t_i)$.

We want to have for instance the both terms

$$\frac{1}{k}\begin{pmatrix}\alpha_{i+q} & \cdots & \alpha_{i+1} & \alpha_i & \alpha_{i-1} & \cdots & \alpha_{i-q'}\end{pmatrix}\begin{pmatrix}u(t_{i+q})\\ \cdots\\ u(t_{i+1})\\ u(t_i)\\ u(t_{i-1})\\ \cdots\\ u(t_{i-q'})\end{pmatrix}-u'(t_i)=O(k^r)$$

$$\begin{pmatrix}\beta_{i+q} & \cdots & \beta_{i+1} & \beta_i & \beta_{i-1} & \cdots & \beta_{i-q'}\end{pmatrix}\begin{pmatrix}f\big(t_{i+q},u(t_{i+q})\big)\\ \cdots\\ f\big(t_{i+1},u(t_{i+1})\big)\\ f\big(t_i\quad,u(t_i)\quad\big)\\ f\big(t_{i-1},u(t_{i-1})\big)\\ \cdots\\ f\big(t_{i-q'},u(t_{i-q'})\big)\end{pmatrix}-f(t_i,u(t_i))=O(k^r).$$

as $k\to 0$.

BDF2 methods: The i-th equation reads

$$\frac{1}{k}\begin{pmatrix}\frac{3}{2} & -\frac{4}{2} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}u(t_i)\\ u(t_{i-1})\\ u(t_{i-2})\end{pmatrix}+\begin{pmatrix}1 & 0 & 0\end{pmatrix}\begin{pmatrix}f\big(t_i \quad , u(t_i)\quad\big)\\ f\big(t_{i-1},u(t_{i-1})\big)\\ f\big(t_{i-2},u(t_{i-2})\big)\end{pmatrix}=g(t_i).$$

Then,

$$\begin{aligned}
&\frac{1}{k}\begin{pmatrix}\frac{3}{2} & -\frac{4}{2} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}u(t_i)\\ u(t_{i-1})\\ u(t_{i-2})\end{pmatrix}+\begin{pmatrix}1 & 0 & 0\end{pmatrix}\begin{pmatrix}f\big(t_i \quad , u(t_i)\quad\big)\\ f\big(t_{i-1},u(t_{i-1})\big)\\ f\big(t_{i-2},u(t_{i-2})\big)\end{pmatrix}-\Big(u'(t_i)+f(t_i,u(t_i))\Big)\\
=&\frac{1}{k}\begin{pmatrix}\frac{3}{2} & -\frac{4}{2} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}u(t_i)\\ u(t_{i-1})\\ u(t_{i-2})\end{pmatrix}-u[t_i,t_i]\\
=&\frac{3}{2}u[t_{i-1},t_i]-\frac{1}{2}u[t_{i-1},t_{i-2}]-u[t_i,t_i]\\
=&\frac{1}{2}\Big(u[t_{i-1},t_i]-u[t_{i-1},t_{i-2}]\Big)+\Big(u[t_{i-1},t_i]-u[t_i,t_i]\Big)\\
=&k\Big(u[t_{i-2},t_{i-1},t_i]-u[t_{i-1},t_i,t_i]\Big)\\
=&-2k^2u[t_{i-2},t_{i-1},t_i,t_i],
\end{aligned}$$

and

$$\Big|-2k^2u[t_{i-2},t_{i-1},t_i,t_i]\Big|\le 2k^2\frac{\|u'''\|_\infty}{6}.$$

Chapter 16

Runge-Kutta Methods

1. We seek for one-step high order method.

We absorb $g(t)$ term into $\tilde{f}(t,u(t)) = g(t) - f(t,u(t))$, to make our notations on RK agree to the other materials.

$$\begin{aligned} u'(t) &= f\big(t,u(t)\big) \quad t \in (0,T) \\ u(0) &= \alpha \end{aligned}$$

We let the interval $[t_i, t_{i+1}]$ be partitioned by

$$t_i = s_0 \le s_1 \le s_2 \le \cdots \le s_r = t_{i+1}$$

and consider a format

$$\begin{aligned} &\begin{pmatrix} \alpha_r & \alpha_{r-1} & \cdots & \alpha_0 \end{pmatrix} \begin{pmatrix} u(s_r) \\ u(s_{r-1}) \\ \vdots \\ u(s_0) \end{pmatrix} \\ &- \begin{pmatrix} \beta_r & \beta_{r-1} & \cdots & \beta_0 \end{pmatrix} \begin{pmatrix} f\big(s_r \quad , u(s_r) \quad \big) \\ f\big(s_{r-1}, u(s_{r-1})\big) \\ \vdots \\ f\big(s_0 \quad , u(s_0) \quad \big) \end{pmatrix} = 0, \end{aligned}$$

as if we knew what $u(s_a)$ $a = 0, 1, \cdots r$ are. (Here $\alpha_a = O(k)$ scale)

In particular, we simply consider

$$\frac{u(s_r) - u(s_0)}{t_{i+1} - t_i} - \begin{pmatrix} \beta_r & \beta_{r-1} & \cdots & \beta_0 \end{pmatrix} \begin{pmatrix} f\big(s_r \quad , u(s_r) \quad \big) \\ f\big(s_{r-1}, u(s_{r-1})\big) \\ \vdots \\ f\big(s_0 \quad , u(s_0) \quad \big) \end{pmatrix} = 0$$

One Interpretation of Runge-Kutta

1. Let us seek for the approximate solution

$$u(s_0), u(s_1), u(s_2), \cdots, u(s_r) \quad \text{and} \quad v(s_0), v(s_1), \cdots, v(s_r), \quad v(s_a) = u'(s_a)$$

2. Let us continue as if we knew $v(s_0)$, $v(s_1)$, $v(s_2)$, $\cdots$, to write

$$\frac{u(t_{i+1}) - u(t_i)}{t_{i+1} - t_i} = \sum_{a=1}^{r} \beta_a v(s_a).$$

3. Thus the i-th equation requires that the divided difference of the approximate solution with respect to t_i and t_{i+1} equals to the (RHS), the numerical quadrature of the slope v.

   (a) Suppose we did know, what $v(s_0), v(s_1), \cdots, v(s_r)$ are, i.e., slopes $u'(s_0), u'(s_1), \cdots, u'(s_r)$. We look for high order numerical quadrature so that

$$\left| \frac{1}{t_{i+1} - t_i} \int_{t_i}^{t_{i+1}} \varphi(s)\, ds - \sum_{a=1}^{r} \beta_a \varphi(s_a) \right| = O(k^p), \quad p \geq r.$$

   Then

$$\frac{u(t_{i+1}) - u(t_i)}{t_{i+1} - t_i} - \sum_{a=1}^{r} \beta_a v(s_a) = O(k^p).$$

   (b) In fact, we do not know what $v(s_0)$, $v(s_1)$, $\cdots$, $v(s_r)$. Note that it suffices to have replacements of slopes $V_0, V_1, \cdots V_r$, as long as the quadrature

$$\sum_{a=1}^{r} \beta_a \big(v(s_a) - V(s_a)\big) = O(k^q), \quad q \geq r.$$

   If this is the case, we let F_{*k} is to achieve

$$\frac{u(t_{i+1}) - u(t_i)}{t_{i+1} - t_i} = \sum_{a=1}^{r} \beta_a V(s_a).$$

   (c) At the true solution $\hat{u}$, we note that

$$\begin{aligned}
&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=1}^{r} \beta_a \hat{V}(s_a) - \Big(\hat{u}'(t_i) - f\big(t_i, \hat{u}(t_i)\big)\Big) \\
=&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=1}^{r} \beta_a \hat{V}(s_a) \\
=&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=1}^{r} \beta_a \hat{v}(s_a) + \sum_{a=1}^{r} \beta_a \big((\hat{v}(s_a) - \hat{V}(s_a)\big) \\
=&O(k^p) + O(k^q) = O(k^r).
\end{aligned}$$

We look for

1. Break points $s_0 \leq s_1 \leq \cdots \leq s_r$.
2. The high order quadrature coefficients $\beta_0, \beta_1, \cdots, \beta_r$.
3. Replacements $V(s_0)$, $V(s_1)$, $\cdots$, $V(s_r)$ of the slopes $u'(s_0)$, $u'(s_1)$, $\cdots$, $u'(s_r)$ satisfying

$$\sum_{a=1}^{r} \beta_a \big(u'(s_a) - V(s_a)\big) = O(k^q)$$

Standard RK2

Break points

$$s_0 = t_i, \quad s_1 = s_0 + \frac{k}{2}, \quad s_2 = s_0 + k.$$

Quadrature

We use the midpoint rule:

$$\beta_0 = 0, \quad \beta_1 = 1, \quad \beta_2 = 0.$$

Here, we make use of that for $\varphi \in C^\infty$,

$$\frac{1}{k}\int_{-\frac{k}{2}}^{\frac{k}{2}} \varphi(s)\, ds = \varphi(0) + O(k^2).$$

Slope replacement

Since $\beta_0 = \beta_2 = 0$, we only need the replacement $V(s_1)$ of $v(s_1)$. Let

$$\begin{aligned} U(s_1) &= u(s_0) + \frac{k}{2} f\big(s_0, u(s_0)\big) \\ V(s_1) &= f\big(s_1, U(s_1)\big) \end{aligned}$$

We notice that at the evaluation of the true solution $\hat{u}$

$$\begin{aligned} \hat{u}(s_1) - \hat{U}(s_1) &= \hat{u}(s_1) - \hat{u}(s_0) - \frac{k}{2} f\big(s_0, \hat{u}(s_0)\big) \\ &= \frac{k}{2}\Big(\hat{u}[s_0, s_1] - \hat{u}[s_0, s_0]\Big) \\ &= \frac{k^2}{4}\hat{u}[s_0, s_0, s_1], \end{aligned}$$

and

$$\begin{aligned} \Big|\sum_{a=0}^{2} \beta_a\big(\hat{u}'(s_a) - \hat{V}(s_a)\big)\Big| = \Big|\hat{v}(s_1) - \hat{V}(s_1)\Big| &= \Big|f\big(s_1, \hat{u}(s_1)\big) - f\big(s_1, \hat{U}(s_1)\big)\Big| \\ &\le \Lambda\Big|\hat{u}(s_1) - \hat{U}(s_1)\Big| \\ &\le \frac{\Lambda k^2}{4}\hat{u}[s_0, s_0, s_1]. \end{aligned}$$

RK2 is 2nd order consistent method

$$\begin{aligned} &\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=0}^{r} \beta_a \hat{V}(s_a) - \Big(\hat{u}'(t_i) + f(t_i, \hat{u}(t_i))\Big) \\ =&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=0}^{r} \beta_a \hat{V}(s_a) \\ =&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=0}^{r} \beta_a \hat{u}'(s_a) + O(k^2) \\ =&O(k^2). \end{aligned}$$

Standard RK3

Break points

$$s_0 = t_i, \quad s_1 = s_0 + \frac{k}{2}, \quad s_2 = s_0 + k, \quad s_3 = s_0 + k.$$

Quadrature

We use the simpson rule:

$$\beta_0 = \frac{1}{6}, \quad \beta_1 = \frac{4}{6}, \quad \beta_2 = \frac{1}{6}. \quad \text{We let } \beta_3 = 0.$$

Here, we make use of that for $\varphi \in C^\infty$,

$$\frac{1}{k}\int_0^k \varphi(s)\,ds = \frac{1}{6}\Big(\varphi(0) + 4\varphi\Big(\frac{k}{2}\Big) + \varphi(k)\Big) + O(k^3).$$

Slope replacement

Let

$$\begin{aligned}
&U(s_0) = u(s_0) && V(s_0) = f\big(s_0, U(s_0)\big)\\
&U(s_1) = u(s_0) + \frac{k}{2}V(s_0) && V(s_1) = f\big(s_1, U(s_1)\big)\\
&U(s_2) = u(s_0) + k\Big(-V(s_0) + 2V(s_1)\Big) && V(s_2) = f\big(s_2, U(s_2)\big).
\end{aligned}$$

We observe that

1. $$\hat{u}(s_0) - \hat{U}(s_0) = 0, \quad \hat{v}(s_0) - \hat{V}(s_0) = 0.$$

2. $$\hat{u}(s_1) - \hat{U}(s_1) = \hat{u}(s_1) - \hat{u}(s_0) - \frac{k}{2}\hat{u}'(s_0) = O(k^2)$$

   and this implies

   $$\hat{v}(s_1) - \hat{V}(s_1) = f\big(s_1, \hat{u}(s_1)\big) - f\big(s_1, \hat{U}(s_1)\big) = f\big(s_1, [\hat{u}(s_1), \hat{U}(s_1)]\big)\big(\hat{u}(s_1) - \hat{U}(s_1)\big).$$

3. $$\begin{aligned}
\hat{u}(s_2) - \hat{U}(s_2) &= \hat{u}(s_2) - \hat{u}(s_0) + k\hat{u}'(s_0) - 2k\hat{u}'(s_1) - 2k\big(\hat{V}(s_1) - \hat{u}'(s_1)\big)\\
&= \hat{u}(s_2) - \hat{u}(s_0) + k\hat{u}'(s_0) - 2k\hat{u}'(s_1) + O(k^3)\\
&= \hat{u}(s_2) - \hat{u}(s_0) - k\hat{u}'(s_1) + k\big(\hat{u}'(s_1) - \hat{u}'(s_0)\big) + O(k^3)\\
&= k\big(\hat{u}'(s_1) - \hat{u}'(s_0)\big) + O(k^3) = \frac{k^2}{2}\hat{u}'[s_0, s_1] + O(k^3).
\end{aligned}$$

and this implies

$$\begin{aligned}\hat{v}(s_2) - \hat{V}(s_2) &= f\big(s_2, \hat{u}(s_2)\big) - f\big(s_2, \hat{U}(s_2)\big) \\ &= f\big(s_2, [\hat{u}(s_2), \hat{U}(s_2)]\big)\big(\hat{u}(s_2) - \hat{U}(s_2)\big) \\ &= f\big(s_2, [\hat{u}(s_2), \hat{U}(s_2)]\big)\Big(\frac{k^2}{2}\hat{u}'[s_0, s_1]\Big) + O(k^3) \\ &= f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)\Big(\frac{k^2}{2}\hat{u}'[s_0, s_1]\Big) + O(k^3).\end{aligned}$$

4. We compute

$$\begin{aligned}&\sum_{a=0}^{3}\beta_a\big(\hat{u}'(s_a) - \hat{V}(s_a)\big) = \sum_{a=0}^{2}\beta_a\big(\hat{u}'(s_a) - \hat{V}(s_a)\big) = \sum_{a=1}^{2}\beta_a\big(\hat{u}'(s_a) - \hat{V}(s_a)\big) \\ =&\frac{f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)}{6}\Big(4\Big(\hat{u}(s_1) - \hat{u}(s_0) - \frac{k}{2}\hat{u}'(s_0)\Big) + \frac{k^2}{2}\hat{u}'[s_0, s_1]\Big) + O(k^3) \\ =&\frac{f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)}{6}\Big(2k\hat{u}[s_0, s_1] - 2k\hat{u}[s_0, s_0] + k\hat{u}[s_0, s_0] - k\hat{u}[s_1, s_1]\Big) + O(k^3) \\ =&\frac{f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)}{6}\Big(2k\hat{u}[s_0, s_1] - k\hat{u}[s_0, s_0] - k\hat{u}[s_1, s_1]\Big) + O(k^3) \\ =&\frac{f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)}{6}\Big(k\hat{u}[s_0, s_1] - k\hat{u}[s_0, s_0] + k\hat{u}[s_0, s_1] - k\hat{u}[s_1, s_1]\Big) + O(k^3) \\ =&\frac{f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)}{6}\Big(\frac{k^2}{2}\hat{u}[s_0, s_0, s_1] - \frac{k^2}{2}\hat{u}[s_0, s_1, s_1]\Big) + O(k^3) \\ =&\frac{f\big(s_1, [\hat{u}(s_1, \hat{U}(s_1)]\big)}{6}\Big(\frac{k^3}{4}\hat{u}[s_0, s_0, s_1, s_1]\Big) + O(k^3) \\ =&O(k^3)\end{aligned}$$

RK3 is 3rd order consistent method

$$\begin{aligned}&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=0}^{r}\beta_a\hat{V}(s_a) - \Big(\hat{u}'(t_i) + f(t_i, \hat{u}(t_i))\Big) \\ =&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=0}^{r}\beta_a\hat{V}(s_a) \\ =&\frac{\hat{u}(t_{i+1}) - \hat{u}(t_i)}{t_{i+1} - t_i} - \sum_{a=0}^{r}\beta_a\hat{u}'(s_a) + O(k^3) \\ =&O(k^3).\end{aligned}$$

The region of stability (non-growth) for $u' = \lambda u$.

As before, we let $k\lambda = z$ and characterize those $z \in \mathbb{C}$ so that

$$|u(t_{i+1})| \leq |u(t_i)|,$$

for the RK2 and RK3.

RK2

We have that

$$\begin{aligned} U(s_1) &= u(s_0) + \frac{k}{2} f\big(s_0, u(s_0)\big) = u(s_0) + \frac{k\lambda}{2} u(s_0) \\ V(s_1) &= f\big(s_1, U(s_1)\big) = \lambda\Big(u(s_0) + \frac{k\lambda}{2} u(s_0)\Big) \\ \frac{u(t_{i+1}) - u(t_i)}{k} &= V(s_1) = \lambda u(s_0) + \frac{k\lambda^2}{2} u(s_0) \\ \iff \quad u(t_{i+1}) &= u(t_i)\Big(1 + z + \frac{z^2}{2}\Big) \end{aligned}$$

Hence

$$K = \Big\{ z \in \mathcal{C} \mid \Big|1 + z + \frac{z^2}{2}\Big| \leq 1 \Big\}.$$

Let us consider the case z is real.

$$\begin{aligned} \Big|1 + z + \frac{z^2}{2}\Big| \leq 1 \quad &\iff \quad -1 \leq 1 + z + \frac{z^2}{2} \leq 1 \\ &\iff \quad 0 \leq 2 + z + \frac{z^2}{2} \quad \text{and} \quad z + \frac{z^2}{2} \leq 0 \end{aligned}$$

$$4 + 2z + z^2 = 3 + (z+1)^2 \geq 0 \quad \text{and} \quad z(2+z) \leq 0 \quad \iff \quad -2 \leq z \leq 0.$$

RK3

We have that

$$
\begin{aligned}
U(s_0) &= u(s_0), & V(s_0) &= \lambda u(s_0) \\
U(s_1) &= u(s_0) + \frac{k\lambda}{2} u(s_0) & V(s_1) &= \lambda u(s_0)\Big(1 + \frac{k\lambda}{2}\Big) \\
U(s_2) &= u(s_0) + k\big(- V(s_0) + 2V(s_1)\big) & & \\
&= u(s_0) + \big(- k\lambda u(s_0) + 2k\lambda u(s_0) + k^2\lambda^2 u(s_0)\big) & & \\
&= u(s_0)\Big(1 + k\lambda + k^2\lambda^2\Big) & V(s_2) &= u(s_0)\Big(1 + k\lambda + k^2\lambda^2\Big) \\
u(t_{i+1}) &= u(t_i) + \frac{k}{6}\Big(V(s_0) + 4V(s_1) + V(s_2)\Big) & & \\
&= u(t_i)\Big(1 + \frac{z}{6} + \frac{4z + 2z^2}{6} + \frac{z + z^2 + z^3}{6}\Big) & & \\
&= u(t_i)\Big(1 + z + \frac{z^2}{2} + \frac{z^3}{6}\Big) & &
\end{aligned}
$$

Hence

$$K = \Big\{ z \in \mathcal{C} \mid \Big|1 + z + \frac{z^2}{2} + \frac{z^3}{6}\Big| \leq 1 \Big\}.$$

Let us consider the case z is real and $z \leq 0$,

$$
\begin{aligned}
\Big|1 + z + \frac{z^2}{2} + \frac{z^3}{6}\Big| \leq 1 \quad &\iff \quad -1 \leq 1 + z + \frac{z^2}{2} + \frac{z^3}{6} \leq 1 \\
&\iff \quad 0 \leq 2 + z + \frac{z^2}{2} + \frac{z^3}{6} \quad \text{and} \quad z + \frac{z^2}{2} + \frac{z^3}{6} \leq 0 \\
p(z) = 12 + 6z + 3z^2 + z^3 \quad &\text{has one real root at } z_* \simeq -2.5127 \quad \text{and we require } z_* \leq z \leq 0 \\
&z(6 + 3z + z^2) \leq 0 \quad \text{if } z \text{ is real and } z \leq 0.
\end{aligned}
$$

Chapter 17

Butcher's table and Dormand & Prince 45

The technical document of scipy 1.15 shows that they provide ODE integrator implementations of

1. RK45 (default)
2. RK23
3. DOP853
4. Radau
5. BDF
6. LSODA

We have seen BDF2, an implicit multi-step method. LSODA is also a multi-step method. RK45, RK23, DOP853 are explicit RK, and Radau is implicit RK.

In this chapter, we try to understand the RK45, namely Dormand & Prince 45, explicit RK.

Butcher's table for explicit RK

In the standard RK3, for instance, we considered

$$
\begin{aligned}
& s_0 = t_i, \quad s_1 = t_i + \frac{k}{2}, \quad s_2 = t_i + k, \quad s_3 = t_i + k \\
& U(s_0) = u(s_0) && V(s_0) = f\big(s_0, U(s_0)\big) \\
& U(s_1) = u(s_0) + \frac{k}{2} V(s_0) && V(s_1) = f\big(s_1, U(s_1)\big) \\
& U(s_2) = u(s_0) + k\Big(- V(s_0) + 2V(s_1)\Big) && V(s_2) = f\big(s_2, U(s_2)\big).
\end{aligned}
$$

The data for the implementation is stored in the table

RK3

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
1	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$

General ℓ stages explicit RK can be put into the table

Explicit ℓ stages RK

c_1						
c_2	$\gamma_{2,1}$					
c_3	$\gamma_{3,1}$	$\gamma_{3,2}$				
c_4	$\gamma_{4,1}$	$\gamma_{4,2}$	$\gamma_{4,3}$			
$\vdots$	$\vdots$	$\vdots$	$\vdots$			
c_ℓ	$\gamma_{\ell,1}$	$\gamma_{\ell,2}$	$\gamma_{\ell,3}$	$\cdots$	$\gamma_{\ell,\ell-1}$	
$c_{\ell+1}$	b_1	b_2	b_3	$\cdots$	$b_{\ell-1}$	b_ℓ

1. We take time spacing *ratios* $0 = c_1 \le c_2 \le c_3 \le \cdots \le c_\ell \le c_{\ell+1} = 1$ so that

$$t_\alpha = t_i + c_\alpha k \quad \text{for each } \alpha = 1, \cdots, \ell + 1.$$

2. For each stage $\alpha = 2, \cdots, \ell + 1$,

$$
\begin{aligned}
& V(t_\beta) = f\big(t_\beta, U(t_\beta)\big), \quad \beta = 1, 2, \cdots, \alpha - 1. \\
& u(t_\alpha) = u(t_i) + (c_\alpha k)\Big[\sum_{\beta=1}^{\alpha-1} \tilde{\gamma}_{\alpha,\beta} V(t_\beta)\Big], \quad \text{where the weight adds to 1, } \sum_{\beta=1}^{\alpha-1} \tilde{\gamma}_{\alpha,\beta} = 1 \\
& \qquad = u(t_i) + k\Big[\sum_{\beta=1}^{\alpha-1} \gamma_{\alpha,\beta} V(t_\beta)\Big], \quad \text{where the weight adds to } c_\alpha, \sum_{\beta=1}^{\alpha-1} \gamma_{\alpha,\beta} = c_\alpha
\end{aligned}
$$

   We write $\tilde{\gamma}_{\ell+1,\beta} = b_\beta$

3. Since $c_{\ell+1}$ in the last row is always 1, it is usually omitted.

1. We find the coefficients set $\Gamma^{\ell,r}$ consisting of

$$\Big((c_\alpha),(\gamma_{\alpha,\beta}),(b_\alpha)\Big)$$

which results in ℓ-stages r-th order methods. A nonlinear system of equations on coefficients characterizes this set. We do not intend to introduce the system of equations.

2. Given that there is a way to collect $\Gamma^{\ell,r}$, here we try to understand the scipy default method Dormand & Prince 45

Dormand & Prince 45

It is a 6 stages 5th order explicit RK:

Dormand & Prince 45

0							
$\frac{2}{10}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{8}{10}$	$\frac{44}{45}$	$-\frac{168}{45}$	$\frac{160}{45}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
$u(t_i+k)$: 1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
$\tilde{u}(t_i+k)$: 1	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

1. We will explain what $u(t_i+k)$ and $\tilde{u}(t_i+k)$ are.

2. It has been proven by Butcher in '60s that 5-th order is possible only if $\ell \geq 6$.

Below, we give rough statements where logic is that are not perfectly logical. We explain in the context of a posteriari error analysis and the condition of the true solver M^f.

1. Considering an ODE system with a large size of unknowns, to take a big time step size k with high order method, potentially with sufficiently reasonable error bound, is a relevant objective.

2. At $t = t_i$ with initial $u(t_i)$, we try potentialy a reasonably big step size k.

3. At the time we computed the next step $u(t_i + k)$ using a certain method, we would like to sense whether the orbit was undergoing changes over the states (neighborhood of $u(t_i)$ and $u(t_i + k)$) where M^f over the neighborhood is well-conditioned or not.

4. This is of course, if we knew M^f is sufficiently well-conditioned, we would keep the big step size k, which could still guarantees a reasonably small solution error under this high order method, while if we knew M^f is not so much well-conditioned, we should reject the big step size and reduce k (by halves for example) and re-do the method.

5. To sense how much M^f is conditioned, one may do the following:

   (since to prove conditions of M^f theoretically on the fly is difficult)

   one prepares two approximate solutions $u(t_{i+1})$ and $\tilde{u}(t_{i+1})$ which gives (after interpolations) for instances,

   $$u'(t) - f(t, u(t)) = C_{method}\|u^{(r)}\|k^r + h.o.t.$$
   $$\tilde{u}'(t) - f(t, \tilde{u}(t)) = \tilde{C}_{method}\|u^{(\tilde{r})}\|k^{\tilde{r}} + h.o.t.$$

   Then, the testing side difference is of $k^{\min\{r,\tilde{r}\}}$. If dynamics of u is not so drastic such that derivatives are kept reasonably small and if M^f is well-conditioned, then the solution difference

   $$|u(t_i + k) - \tilde{u}(t_i + k)|$$

   would be reasonably small.

   If the difference is measured to be not so small, then those introduced hypotheses must have been broken somewhere, and we better reduce the time step and re-do the computation.

1. On top of above time step control mechanism, one feature that comes practically important is that among possible r-th order methods, we do want to choose the one where the coefficient C_{method} is small.

2. Among 6 stages 5-th order Butcher's tables, RK45 has been selected from the following considerations.

1. To employ two methods to sense the condition of the problem is certainly too costy, and one thus seek a certain RK table, where among the (c_α), 1 appears twice so that one low order and one high order approximations at $t_i + k$ are produced by the computations along the table.

2. Historically, the method appeared prior to Dormand & Prince 45 considered the following task:

   (a) Among possibly computable set of $\Gamma^{6,5}$ of 6 stages 5-th order explicit RK, one looks for those special cases where a 4-th order explicit RK is embeded in it, say using fewer stages.

   (b) It turns out that there are not many such choices.

3. Dormand & Prince did the opposite.

   (a) They first optimize the 5-th order method, and then low-order RK is sought for afterwards: Note the Butcher's table where the low order $\tilde{u}(t_{i+1})$ actually uses 7 stages !

   (b) As consequences, the coefficient C_{method} of 5-th order RK is relatively much smaller than other choices of 6 stages 5th order, and at the same time one can compare the two approximations to sense the condition of the problem.

Remark 17.1. RK23, DOP853 are designed with the same spirit. In the table of RK23, 2nd and 3rd order approximations are produced. In table of DOP853, 8th, 5th, and 3rd order approximations are produced.