Scientific computation

Sorin Mitran

Department of Mathematics
University of North Carolina at Chapel Hill

Abstract

The goal of scientific computation is the construction of quantitatively verifiable models of the natural world. Scientific computation intertwines mathematics, algorithm formulation, software and hardware engineering with domain-specific knowledge from the physical, life, or social sciences. It is the fusion of these disciplines that imparts a distinct identity to scientific computing, and mathematical concepts must be complemented by practical modeling considerations to achieve successful computational simulation.

Human interpretation of the complexity of the natural world has historically led to parallel developments in formulation of abstract concepts, construction of models, and use of computational aids. This is the case on the long time span from prehistoric formulation of the abstract concept of a number and use of tally sticks, to current efforts based on quantum mechanics and superconducting qubit electronics. A dichtomy arises between the formulation of mathematical concepts asserted purely by reason and practicable predictions of the natural world. The conflicting approaches are reconciled by modeling and approximation. Domain sciences such as physics, biology, or sociology construct the relevant models. Comparison of model predictions to observation can sometimes be carried out through devices as simple as Galileo's inclined plane. The much higher complexity of models of airplane flight, or cancer progression, or activity on social media requires different tools, paradigmatically represented by the modern digital computer. In this more complex setting, scientific computing is instrumental in extracting verifiable predictions from models.

Central to computational prediction is the idea of approximation: finding a simple mathematical representation of some complex model of reality. Archimedes approximated the area of a parabolic segment by that an increasing number of inscribed triangles. Nowadays meteorological observations are input to a digital computer to predict future rainfall. The weather model implemented on the computer contains numerous approximations. Remarkably, both models rely on the same technique of additive approximation: summation of successive corrections to obtain increased accuracy.

One of the goals of this textbook is to highlight how computational methods arise as expressions of just a few approximation approaches. Additive corrections underlie many methods ranging from interpolation to numerical integration, and benefit from a remarkably complete theoretical framework within linear algebra. Alternatively, successive function composition is a distinct approximation idea, with a theoretical framework that is not yet complete, and whose expression leads to the burgeoning field of "machine learning". The process by which a specific approximation approach leads to different types of algorithms is a unifying theme of the presentation, and hopefully not only illuminates well-known algorithms, but also serves as a guide to future research.

The presentation of topics and ideas is at the graduate level of study, but with a focus on unifying ideas rather than detailed techniques. Much of traditional numerical methods and some associated analysis is presented and seen to be related to real analysis and linear algebra. The same theoretical framework can be extended to probability spaces and account for random phenomena. The nonlinear approach to approximation that characterizes artificial neural networks requires a different conceptual framework which has yet to be crystallized. Though numerical methods are prevalent in scientific computation, this text also considers alternatives that should be part of a computational scientist's toolkit such as symbolic, topological, and geometric computation.

Scientific computing is not a theoretical exercise, and successful simulation relies on acquiring the skill set to implement mathematical concepts and approximation techniques into efficient code. This text intersperses method presentation and implementation, mostly in the Julia language. An associated electronic version of his textbook uses the TeXmacs scientific editing platform to enable live documents with embedded computational examples, allowing immediate experimentation with the presented algorithms.

Table of contents

Part I

Number Approximation

LECTURE 1: NUMBER APPROXIMATION

1. Numbers

1.1. Number sets

Most scientific disciplines introduce an idea of the amount of some entity or property of interest. Furthermore, the amount is usually combined with the concept of a *number*, an abstraction of the observation that the two sets $A = \{\text{Mary, Jane, Tom}\}$ and $B = \{\text{apple, plum, cherry}\}$ seem quite different, but we can match one distinct person to one distinct fruit as in $\{\text{Mary}\leftrightarrow\text{plum, Jane}\leftrightarrow\text{apple, Tom}\leftrightarrow\text{cherry}\}$. In contrast, we cannot do the same matching of distinct persons to a distinct color from the set $\{\text{red, green}\}$, and one of the colors must be shared between two persons. Formal definition of the concept of a number from the above observations is surprisingly difficult since it would be self-referential due to the apperance of the numbers "one" and "two". Leaving this aside, the key concept is that of *quantity* of some property of interest that is expressed through a number.

Several types of numbers have been introduced in mathematics to express different types of quantities, and the following will be used throughout this text:

- $\mathbb{N}$. The set of natural numbers, $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, infinite and countable, $\mathbb{N}_+ = \{1, 2, 3, \dots\}$;
- $\mathbb{Z}$. The set of integers, $\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \dots\}$, infinite and countable;
- $\mathbf{Q}$. The set of rational numbers $\mathbb{Q} = \{p/q, p \in \mathbb{Z}, q \in \mathbb{N}_+\}$, infinite and countable;
- $\mathbb{R}$. The set of real numbers, infinite, not countable, can be ordered;
- $\mathbb{C}$. The set of complex numbers, $\mathbb{C} = \{x + iy, x, y \in \mathbb{R}\}$, infinite, not countable, cannot be ordered.

These sets of numbers form a hierarchy, with $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$. The size of a set of numbers is an important aspect of its utility in describing natural phenomena. The set $S = \{\text{Mary, Jane, Tom}\}$ has three elements, and its size is defined by the *cardinal number*, $|S| = 3$. The sets $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ have an infinite number of elements, but the relation

$$z = \begin{cases} -n/2 & \text{for } n \text{ even} \\ (n+1)/2 & \text{for } n \text{ odd} \end{cases}$$

defines a one-to-one correspondence between $n \in \mathbb{N}$ and $z \in \mathbb{Z}$, so these sets are of the same size denoted by the *transfinite number* $\aleph_0$ (aleph-zero). The rationals can also be placed into a one-to-one correspondence with $\mathbb{N}$, hence

$$|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| = \aleph_0 .$$

In contrast there is no one-to-one mapping of the reals to the naturals, and the cardinality of the reals is $|\mathbb{R}| = \mathfrak{c}$ (Fraktur-script c). Georg Cantor established set theory and introduced a proof technique known as the diagonal argument to show that $\mathfrak{c} = 2^{\aleph_0}$. Intuitively, there are exponentially more reals than naturals.

1.2. Quantification

One of the foundations of the scientific method is *quantification*, ascribing numbers to phenomena of interest. To exemplify the utility of different types of number to describe natural phenomena, consider common salt (sodium chloride, Fig. 1) which has the chemical formula NaCl with the sodium ions (Na^+) and chloride ions (Cl^-) spatially organized in a cubic lattice, with an edge length $a = 5.6402$ Å (1 Å = 10^{-10} m) between atoms of the same type. Setting the origin of a Cartesian coordinate system $Oxyz$ at a sodium atom, the position of some atom within the lattice is

$$(x, y, z) = \left(i\frac{a}{2}, j\frac{a}{2}, k\frac{a}{2} \right).$$

Sodium atoms are found positions where $i + j + k$ is even, while chloride atoms are found at positions where $i + j + k$ is odd. The Cartesian coordinates (x, y, z) describe some arbitrary position in space, which is conceptualized as a continuum and placed into one-to-one correspondence with $\mathbb{R}^3$. A particular lattice position can be specified simply through a label consisting of three integers $(i, j, k) \in \mathbb{Z}^3$. The position can be recovered through a *scaling operation*

$$(x, y, z) = \frac{a}{2}(i, j, k),$$

and the number $a/2 \in \mathbb{R}$ that modifies the length scale from 1 to $a/2$, it is called a *scalar*.

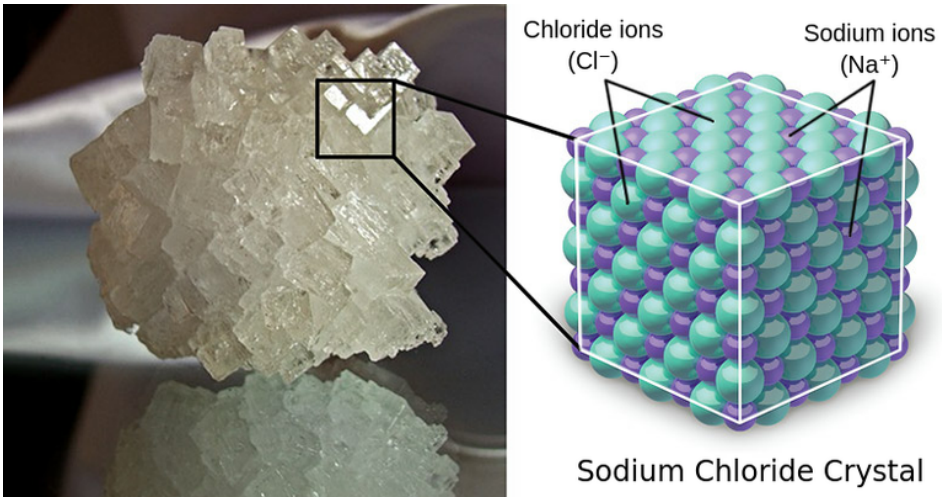


Figure 1. Left: Polycrystalline sodium chloride. Right: Cubic lattice structure of a single sodium chloride crystal

1.3. Computer number sets

A computer has a finite amount of memory, hence cannot represent all numbers, but rather subsets of the above number sets. Current digital computers internally use numbers represented through binary digits, or bits. Many computer number types are defined for specific purposes, and are often encountered in applications such as image representation or digital data acquisition. Here are the main types.

Subsets of $\mathbb{N}$. The number types `uint8`, `uint16`, `uint32`, `uint64` represent subsets of the natural numbers (unsigned integers) using 8, 16, 32, 64 bits respectively. An unsigned integer with b bits can store a natural number in the range from 0 to $2^b - 1$. Two arbitrary natural numbers, written as $\forall i, j \in \mathbb{N}$ can be added and will give another natural number, $k = i + j \in \mathbb{N}$. In contrast, addition of computer unsigned integers is only defined within the specific range 0 to $2^b - 1$. If $k > 2^b - 1$, the result might be displayed as the maximum possible value or as $k \bmod 2^b$.

```
∴ i=UInt8(15); j=UInt8(10); k=i+j
```

25

```
∴ i=UInt8(150); j=UInt8(200); k=i+j; [k i+j mod(350,256)]
```

$$[\ 94\ \ 94\ \ 94\ ] \tag{1}$$

```
∴ i=UInt8(150); j=UInt8(200); k=i-j; [k i-j mod(-50,256)]
```

$$[\ 206\ \ 206\ \ 206\ ] \tag{2}$$

```
∴ typeof(i-j)
```

UInt8

```
∴
```

Subsets of $\mathbb{Z}$. The number types `int8`, `int16`, `int32`, `int64` represent subsets of the integers. One bit is used to store the sign of the number, so the subset of $\mathbb{Z}$ that can be represented is from $1 - 2^{b-1}$ to $2^{b-1} - 1$.

```
∴ i=Int8(15); j=Int8(21); k=i+j
```

36

```
∴ i=Int8(100); j=Int8(101); k=i+j; [k i+j mod(201,128)-128]
```

$$\begin{bmatrix} -55 & -55 & -55 \end{bmatrix} \tag{3}$$

```
∴ typeof(k)
```

```
Int8
```

```
∴ [typemin(Int8) typemax(Int8)]
```

$$\begin{bmatrix} -128 & 127 \end{bmatrix} \tag{4}$$

```
∴
```

Subsets of $\mathbb{Q}, \mathbb{R}, \mathbb{C}$. Computers approximate the real numbers through the set $\mathbb{F}$ of *floating point numbers*. Floating point numbers that use $b = 32$ bits are known as *single precision*, while those that use $b = 64$ are *double precision*. A floating point number $x \in \mathbb{F}$ is stored internally as $x = \pm.B_1B_2\ldots B_m \times 2^{\pm b_1 b_2 \ldots b_e}$ where B_i, $i = 1, \ldots, m$ are bits within the *mantissa* of length m, and b_j, $j = 1, \ldots, e$ are bits within the *exponent*, along with signs $\pm$ for each. The default number type is usually double precision, more concisely referred to `Float64`. Common irrational constants such as e, π are predefined as irrationals and casting to `Float64` or `Float32` gives floating point approximation. Unicode notation is recognized. Specification of a decimal point indicates a floating point number; its absence indicates an integer.

```
∴ pi
```

π

```
∴ typeof(pi)
```

```
Irrational{:π}
```

```
∴ [Float32(pi) Float64(pi) Float64(π)]
```

$$\begin{bmatrix} 3.1415927410125732 & 3.141592653589793 & 3.141592653589793 \end{bmatrix} \tag{5}$$

```
∴ a=2.3; b=2; c=3.; [typeof(a) typeof(b) typeof(c)]
```

```
DataType[Float64 Int64 Float64]
```

```
∴
```

The approximation of the reals $\mathbb{R}$ by the floats $\mathbb{F}$ is characterized by: `floatmax()`, the largest float, `floatmin` the smallest positive float, and `eps()` known as *machine epsilon*. Machine epsilon highlights the differences between floating point and real numbers since it may be informally defined as the smallest number of form $\epsilon = 2^k \in \mathbb{F}$ that satisfies $1 + \epsilon \neq 1$. If $\varepsilon \in \mathbb{R}$ of course $1 + \varepsilon = 1$ implies $\varepsilon = 0$, but floating points exhibit "granularity", in the sense that over a unit interval there are small steps that are indistinguishable from zero due to the finite number of bits available for a float leading to $1 + \epsilon/2$ being indistiguishable from 1, and the apparently endless loop shown below actually terminates.

```
∴ eps=1.0;
```

```
∴ while (1.0+0.5*eps != 1.0)
    global eps;
    eps=0.5*eps;
  end
```

```
∴ eps
```

```
2.220446049250313e-16
```

The granularity of double precision expressed by machine epsilon is sufficient to represent natural phenomena, and floating point errors can usually be kept under control,

```
∴ [floatmin(Float32) floatmax(Float32) eps(Float32)]
```

$$\left[\begin{array}{ccc} 1.1754944\,e-38 & 3.4028235\,e\,38 & 1.1920929\,e-7 \end{array}\right] \tag{6}$$

```
∴ [floatmin(Float64) floatmax(Float64) eps(Float64)]
```

$$\left[\begin{array}{ccc} 2.2250738585072014\,e-308 & 1.7976931348623157\,e\,308 & 2.220446049250313\,e-16 \end{array}\right] \tag{7}$$

```
∴
```

Keep in mind that perfect accuracy is a mathematical abstraction, not encountered in nature. In fields such as sociology or psychology 3 digits of accuracy are excellent, in mechanical engineering this might increase to 6 digits, or in electronic engineering to 8 digits. The most precisely known physical constant is the Rydberg constant known to 12 digits, hence a mathematical statement such as

$$x = 2.630928345046124835031948631 9845$$

is unlikely to have any real significance, while

$$x = 2.631 \pm 0.0005$$

is much more informative.

Within the reals certain operations are undefined such as $1/0$. Special float constants are defined to handle such situations: `Inf` is a float meant to represent infinity, and `NaN` ("not a number") is meant to represent an undefinable result of an arithmetic operation.

```
∴ [1/0 -1.0/0.0 1/Inf -1/Inf Inf/Inf]
```

$$\left[\begin{array}{ccccc} \infty & -\infty & 0.0 & -0.0 & NaN \end{array}\right] \tag{8}$$

```
∴
```

Complex numbers $z \in \mathbb{C}$ are specified by two reals, in Cartesian form as $z = x + iy$, $x, y \in \mathbb{R}$ or in polar form as $z = \rho e^{i\theta}$, $\rho, \theta \in \mathbb{R}$, $\rho \geqslant 0$. The computer type complex is similarly defined from two floats and the additional constant I is defined to represent $\sqrt{-1} = i = e^{i\pi/2}$. Functions are available to obtain the real and imaginary parts within the Cartesian form, or the absolute value and argument of the polar form.

```
∴ z1=1+1im; z2=1-im; [z1+z2 z1/z2]
```

$$\left[\begin{array}{cc} 2.0 + 0.0i & -0.0 + 1.0i \end{array}\right] \tag{9}$$

```
∴ [real(z1) real(z2) real(z1+z2) real(z1/z2)]
```

$$\left[\begin{array}{cccc} 1.0 & 1.0 & 2.0 & -0.0 \end{array}\right] \tag{10}$$

```
∴ [imag(z1) imag(z2) imag(z1+z2) imag(z1/z2)]
```

$$\left[\begin{array}{cccc} 1.0 & -1.0 & 0.0 & 1.0 \end{array}\right] \tag{11}$$

```
∴ [abs(z1) abs(z2) abs(z1+z2) abs(z1/z2)]
```

$$\left[\begin{array}{cccc} 1.4142135623730951 & 1.4142135623730951 & 2.0 & 1.0 \end{array}\right] \tag{12}$$

```
∴ [angle(z1) angle(z2) angle(z1+z2) angle(z1/z2)]
```

$$\left[\begin{array}{cccc} 0.7853981633974483 & -0.7853981633974483 & 0.0 & 1.5707963267948966 \end{array}\right] \tag{13}$$

```
∴
```

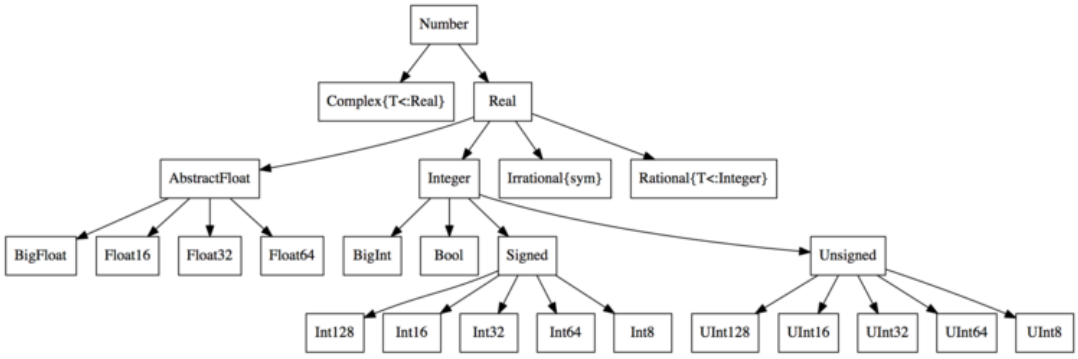


Figure 2. Hierarchy of number types in the `Julia` language.

2. Approximation

2.1. Axiom of floating point arithmetic

The reals $\mathbb{R}$ form an algebraic structure known as a field $(\mathbb{R}, +, \cdot)$. The set of floats together with floating point addition and multiplication are denoted as $(\mathbb{F}, \oplus, \odot)$. Operations with floats do not have the same properties as the reals, but are assumed to have a relative error bounded by machine epsilon

$$\forall x, y \in \mathbb{R}, \left| \frac{\mathrm{fl}(x) \circledast \mathrm{fl}(y) - x * y}{x * y} \right| \leq \epsilon, (\circledast, *) \in \{(\oplus, +), (\odot, \cdot)\},$$

where $\mathrm{fl}(x) \in \mathbb{F}$ is the floating point representation of $x \in \mathbb{R}$. The above is restated

$$\mathrm{fl}(x) \circledast \mathrm{fl}(y) = (x * y)\,(1 + \varepsilon), |\varepsilon| \leq \epsilon,$$

and accepted as an axiom for use in error analysis involving floating point arithmetic. Computer number sets are a first example of *approximation*: replacing some complicated object with a simpler one. It is one of the key mathematical ideas studied throughout this text.

2.2. Cummulative floating point operations

Care should be exercised about the cummulative effect of many floating point operations. An informative example is offered by Zeno's paradox of motion, that purports that fleet-footed Achilles could never overcome an initial head start of $D = 2$ given to the lethargic Tortoise since, as stated by Aristotle:

> In a race, the quickest runner can never over-take the slowest, since the pursuer must first reach the point whence the pursued started, so that the slower must always hold a lead.

The above is often formulated by considering that the first half of the initial head start must be overcome, then another half and so on. The distance traversed after N such steps is

$$D_N = 1 + \frac{1}{2} + \cdots + \frac{1}{2^N} = \frac{1 - (1/2)^{N+1}}{1 - (1/2)} = 2\left[1 - (1/2)^{N+1}\right] < 2.$$

Calculus resolves the paradox by rigorous definition of the limit $D = \lim_{N \to \infty} D_N = 2$ and definition of velocity as $v(t) = \lim_{\delta t \to 0} (D(t + \delta t) - D(t))/\delta t$, $\delta t = 1/N$, $D(t) = 2[1 - (1/2)^{t/\delta t}]$.

Undertake a numerical invesigation and consider two scenarios, with increasing or decreasing step sizes

$$D_N = 1 + \frac{1}{2} + \cdots + \frac{1}{2^N}, C_N = \frac{1}{2^N} + \frac{1}{2^{N-1}} + \cdots + 1.$$

In $(\mathbb{R}, +, \cdot)$ associativity ensures $D_N = C_N$.

```
∴ N=10; D=2.0 .^ (0:-1:-N); C=2.0 .^ (-N:1:0); sum(D)==sum(C)
```

true

```
∴ N=20; D=2.0 .^ (0:-1:-N); C=2.0 .^ (-N:1:0); sum(D)==sum(C)
```

true

```
∴
```

Irrespective of the value for N, $D_N = C_N$ in floating point arithmetic. Recall however that computers use binary representations internally, so division by powers of two might have unique features (indeed, it corresponds to a bit shift operation). Try subdividing the head start by a different number, perhaps π to get an "irrational" numerical investigation of Zeno's paradox of motion. Define now the distance S_N traversed by step sizes that are scaled by $1/\pi$ starting from one to T_N, traversed by step sizes scaled by π starting from π^{-N}

$$S_N = 1 + \frac{1}{\pi} + \frac{1}{\pi^2} + \cdots + \frac{1}{\pi^N}, T_N = \frac{1}{\pi^N} + \frac{1}{\pi^{N-1}} + \cdots + 1.$$

Again, in the reals the above two expressions are equal, $S_N = T_N$, but this is no longer verified computationally for all N, not even within a tolerance of machine epsilon.

```
∴ fpi=Float64(pi);
```

```
∴ N=10; S=fpi .^ (0:-1:-N); T=fpi .^ (-N:1:0); sum(S)==sum(T)
```

true

```
∴ N=20; S=fpi .^ (0:-1:-N); T=fpi .^ (-N:1:0); sum(S)==sum(T)
```

false

```
∴ sum(S)-sum(T)<eps(Float64)
```

false

```
∴
```

This example gives a first glimpse of the steps that need to be carried out in addition to mathematical analysis to fully characterize an algorithm. Since $S_N \neq T_N$, a natural question is whether one is more accurate than the other. For some arbitrary ratio a, the exact value is known

$$E_N = \frac{1 - (1/a)^{N+1}}{1 - (1/a)},$$

and can be used to evaluate the errors $|S_N - E_N|$, $|T_N - E_N|$.

```
∴ function E(N,a)
      (1-(1/a)^(N+1))/(1-(1/a))
  end;
```

```
∴ function εs(N,a)
      S=a .^ (0:-1:-N)
      abs(sum(S)-E(N,a))
  end;
```

```
∴ function εt(N,a)
      T=a .^ (-N:1:0)
      abs(sum(T)-E(N,a))
  end;
```

```
∴
```

Carrying out the computations leads to results in Fig. 3.

```
∴ n=30; errs=zeros(Float64,n); errt=zeros(Float64,n);
```

```
∴ for i=1:n
    errs[i]=εs(N,fpi); errt[i]=εt(N,fpi);
  end
```

```
∴ clf(); plot(1:n,errs,1:n,errt,marker="o"); title("Summation␣error"); grid("on"); xlabel("n");
  ylabel("εs,εt"); legend(["εs"; "εt"]);
```

```
∴
```

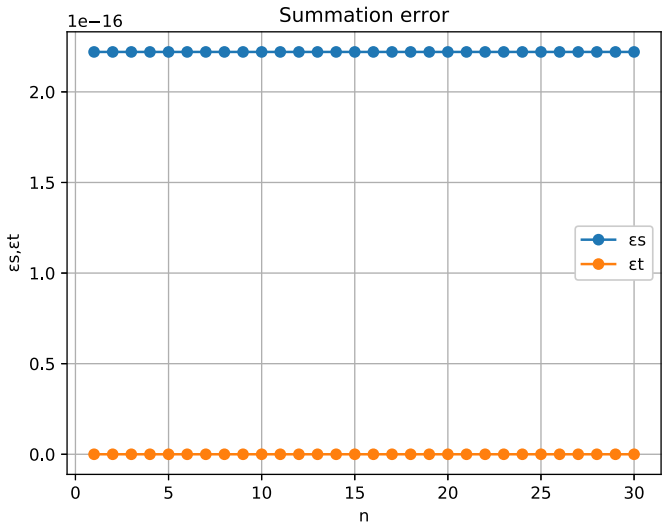


Figure 3. Summation order errors.

Note that errors are about the size of machine epsilon for S_N, but are zero for T_N, it seems that the summation ordering $T_N = a^{-N} + a^{-N+1} + \cdots + 1$ gives the exact value. A bit of reflection reinforces this interpretation: first adding small quantities allows for carry over digits to be accounted for.

This example is instructive beyond the immediate adage of "add small quantities first". It highlights the blend of empirical and analytical approaches that is prevalent in scientific computing.

3. Successive approximations

3.1. Sequences in $\mathbb{R}$

Single values given by some algorithm are of little value in the practice of scientific computing. The main goal is the construction of a sequence of approximations $\{x_n\}_{n\in\mathbb{N}}$ that enables assessment of the quality of an approximation. Recall from calculus that $\{x_n\}_{n\in\mathbb{N}}$ converges to x if $|x_n - x|$ can be made as small as desired for all n beyond some threshold. In precise mathematical language this is stated through:

DEFINITION. *$\{x_n\}_{n\in\mathbb{N}}$ converges to x if $\forall\varepsilon>0, \exists N(\varepsilon)$ such that $|x_n - x| < \varepsilon$ for $n > N(\varepsilon)$.*

Though it might seem natural to require a sequence of approximations to converge to an exact solution x

$$\lim_{n\to\infty} x_n = x,$$

such a condition is problematic on multiple counts:

1. the exact solution is rarely known;

2. the best approximation might be achieved for some finite range $n_1 \leqslant n \leqslant n_2$, rather than in the $n \longrightarrow \infty$ limit.

Both situations arise when approximating numbers and serve as useful reference points when considering approximation other types of mathematical objects such as functions. For example, the number π is readily defined in geometric terms as the ratio of circle circumference to diameter, but can only be approximately expressed by a rational number, e.g., $\pi \cong 22/7$. The exact value of π is only obtained as the limit of an infinite number of operations with rationals. There are many such infinite representations, one of which is the Leibniz series

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots .$$

No finite term

$$L_n = \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}$$

of the above Leibniz series equals $\pi/4$, i.e.,

$$\nexists n \in \mathbb{N} \text{ such that } L_n = \pi/4.$$

Rather, the Leibniz series should be understood as an algorithm, i.e., a sequence of elementary operations that leads to succesively more accurate approximations of $\pi/4$

$$\lim_{n\to\infty} L_n = \pi/4.$$

Complex analysis provides a convergence proof starting from properties of the arctan function

$$\frac{\mathrm{d}}{\mathrm{d}z}\arctan(z) = \frac{1}{1+z^2} \Longrightarrow \frac{\pi}{4} = \arctan(1) - \arctan(0) = \int_0^1 \frac{\mathrm{d}z}{1+z^2}.$$

For $|z| < 1$ the sequence $S_n = \sum_{k=0}^{n} (-z^2)^k$ of partial sums of a geometric series converges uniformly

$$\sum_{k=0}^{\infty} (-z^2)^k = \lim_{n\to\infty} \frac{1-(-z^2)^n}{1-(-z^2)} = \frac{1}{1+z^2},$$

and can be integrated term by term to give

$$\frac{\pi}{4} = \int_0^1 \sum_{k=0}^{\infty} (-z^2)^k \,\mathrm{d}z = \sum_{k=0}^{\infty} \int_0^1 (-z^2)^k \,\mathrm{d}z = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}.$$

This elegant result does not address however the points raised above: if π were not known, how could the convergence of the sequence $\{L_n\}_{n\in\mathbb{N}}$ be assessed? A simple numerical experiment indicates that the familiar value of π is only recovered for large n, with 10000 terms insufficient to ensure five significant digits.

```
∴ function L(n)
    L=1.0; s=-1.0
    for k=1:n
      L += s/(2*k+1); s = -s
    end
    return 4*L
  end
```

```
L
```

```
∴ [L(100) L(1000) L(10000) Float64(π)]
```

$$\begin{bmatrix} 3.1514934010709914 & 3.1425916543395442 & 3.1416926435905346 & 3.141592653589793 \end{bmatrix} \tag{14}$$

```
∴
```

3.2. Cauchy sequences

Instead of evaluating distance to an unknown limit, as in $|L_n-\pi|<\varepsilon$, one could evaluate if terms get closer to one another as in $|L_n-L_m|<\varepsilon$, a condition that can readily be checked in an algorithm.

DEFINITION. *$\{x_n\}_{n\in\mathbb{N}}$ is a Cauchy sequence if $\forall\varepsilon>0, \exists N(\varepsilon)$ such that $|x_n-x_m|<\varepsilon$ for all $m,n>N(\varepsilon)$.*

Note that the distance between *any* two terms after the threshold $N(\varepsilon)$ must be smaller than an arbitrary tolerance ε. For example the sequence $a_n=\sqrt{n}$ is not a Cauchy sequence even though the distance between successive terms can be made arbitrarily small

$$a_{n+1}-a_n=\sqrt{n+1}-\sqrt{n}=\frac{(\sqrt{n+1}-\sqrt{n})(\sqrt{n+1}+\sqrt{n})}{\sqrt{n+1}+\sqrt{n}}=\frac{1}{\sqrt{n+1}+\sqrt{n}}<\frac{1}{2\sqrt{n}}.$$

Verification of decreasing successive distance is therefore a necessary but not sufficient condition to assess whether a sequence is a Cauchy sequence. Furthermore, the distance between successive iterates is not necessarily an indication of the distance to the limit. Reprising the Leibniz example, successive terms can be further apart than the distance to the limit, though terms separated by 2 are closer than the distance to the limit (a consequence of the alternating Leibniz series signs)

```
∴ n=1000; [log10(abs(L(n)-L(n-1))) log10(abs(L(n)-π))]
```

$$\begin{bmatrix} -2.6991870973082537 & -3.000434185835426 \end{bmatrix} \tag{15}$$

```
∴  [log10(abs(L(n)-L(n-2))) log10(abs(L(n)-π))]
```

$$\begin{bmatrix} -5.698969895788488 & -3.000434185835426 \end{bmatrix} \tag{16}$$

```
∴
```

Another question is whether a Cauchy sequence is itself convergent. For sequences of reals this is true, but the Leibniz sequence furnishes a counterexample since it contains rationals and converges to an irrational. Such aspects that arise in number approximation sequences become even more important when considering approximation sequences composed of vectors or functions.

3.3. Sequences in $\mathbb{F}$

Consideration of floating point arithmetic indicates adaptation of the mathematical concept of convergence is required in scientific computation. Recall that machine epsilon ϵ is that largest number such that $1+\epsilon=1$ is true, and characterizes the granularity of the floating point system. A reasonable adaptation of the notion of convergence might be:

DEFINITION. $\{x_n\}_{n\in\mathbb{N}}$, $x_n\in\mathbb{F}$ *converges to* $x\in\mathbb{F}$ *if* $\forall\varepsilon>\epsilon, \exists N(\varepsilon)$ *such that* $|x_n-x|<\varepsilon$ *for* $n>N(\varepsilon)$.

What emerges is the need to consider a degree of uncertainty in an approximating sequence. If the uncertainty can be bounded to the intrinsic granularity of the number system, a good approximation is obtained.

Summary. The problem of approximating numbers uncovers generic aspects of scientific computing:

- different models of some phenomenon are possible and it is necessary to establish correspondence between models and of a model to theory;
- scientific computation seeks to establish viable approximation techniques for the mathematical objects that arise in models;
- correspondence of a model to theory is established through properties of approximation sequences, not single results of a particular approximation technique;
- physical limitations of computer memory require revisiting of mathematical concepts to characterize approximation sequence behavior, and impart a stochastic aspect to approximation techniques;
- computational experiments are a key aspect, giving an empirical aspect to scientific computing that is not found in deductive or analytical mathematics.

LECTURE 2: APPROXIMATION TECHNIQUES

1. Rate and order of convergence

The objective of scientific computation is to solve some problem $f(x)=0$ by constructing a sequence of approximations $\{x_n\}_{n\in\mathbb{N}}$. The condition suggested by mathematical analysis would be $x=\lim_{n\to\infty}x_n$, with $f(x)=0$. As already in the Leibniz series approximation of π, acceptable accuracy might only be obtained for large n. Since f could be an arbitrarily complex mathematical object, such slowly convering approximating sequences are of little practical interest. Scientific computing seeks approximations of the solution with rapidly decreasing error. This change of viewpoint with respect to analysis is embodied in the concepts of *rate* and *order of convergence*.

DEFINITION 1. $\{x_n\}_{n\in\mathbb{N}}$ *converges to* x *with rate* $r\in(0,1)$ *and order* p *if*

$$\lim_{n\to\infty}\frac{|x_{n+1}-x|}{|x_n-x|^p}=r. \tag{17}$$

As previously discussed, the above definition is of limited utility since:

a) The solution x is unknown;

b) The limit $n\longrightarrow\infty$ is impractical to attain.

Sequences converge faster for higher order p, but lower rate r. A more useful approach is to determine estimates of the rate and order of convergence over some range of iterations that are sufficiently accurate. Rewriting (1) as

$$\lim_{n\to\infty}\left(|x_{n+1}-x|-r\,|x_n-x|^p\right)=0,$$

suggests introducing the distance between successive iterates $d_n=|x_n-x_{n-1}|$, and considering the condition

$$|\,d_{n+1}-sd_n^q|\,\text{small for large}\,n.$$

DEFINITION 2. $\{x_n\}_{n\in\mathbb{N}}$ *approximates* x *with rate* s *and order* q *if there exist* $s,q\in\mathbb{R}$ *and* $n_1,n_2\in\mathbb{N}$ *such that*

$$|\,d_{n+1}-sd_n^q|<\epsilon,\text{for}\,n_1\leqslant n\leqslant n_2 \tag{18}$$

with $d_n=|x_n-x_{n-1}|$, $n\in\mathbb{N}$, ϵ *denotes machine epsilon.*

As an example, consider the derivative $g=f'$ of $f(x)=e^x-1$ at $x_0=0$, as given by the calculus definition

$$g(x_0)=f'(x_0)=\lim_{h\to0}\frac{f(x_0+h)-f(x_0)}{h},$$

and construct a sequence of approximations

$$g_n=\frac{f_n-f(0)}{h_n},\ f_n=f(h_n),\ h_n=2^{-n}.$$

Start with a numerical experiment, and compute the sequence $d_n=|g_n-g_{n-1}|$.

n	1	2	… N
$h_n=2^{-n}$	$1/2$	$1/4$	… $1/2^N$
f_n	f_1	f_2	… f_N
$g_n=(f_n-f(0))/h_n$	$g_1=(f_1-f(0))/2$	$g_2=(f_2-f(0))/4$	… $g_N=(f_N-f(0))/2^N$
$d_{n-1}=\lvert g_n-g_{n-1}\rvert$	–	$d_1=\lvert g_2-g_1\rvert$	… $d_{N-1}=\lvert g_N-g_{N-1}\rvert$

Table 1. Table presentation of calculations to construct approximation of derivative sequence for $f(x)=\tan x$, at $x_0=0$.

```
∴ N=24; n=1:N; h=2.0.^(-n); f(x) = exp(x)-1; x0=0; f0=f(x0);
∴ g = (f.(h).-f0) ./ h; d=abs.(g[2:N]-g[1:N-1]);
∴ n1=2; n2=8; [h[n1:n2] g[n1:n2] d[n1:n2]]
```

$$\begin{bmatrix} 0.25 & 1.1361016667509656 & 0.070914042216355 \\ 0.125 & 1.0651876245346106 & 0.033276281848861444 \\ 0.0625 & 1.0319113426857491 & 0.0161223027144608 \\ 0.03125 & 1.0157890399712883 & 0.007935690423401809 \\ 0.015625 & 1.0078533495478865 & 0.0039369071225365815 \\ 0.0078125 & 1.00391644242535 & 0.001960771808398931 \\ 0.00390625 & 1.001955670616951 & 0.0009784720235188615 \end{bmatrix} \tag{19}$$

```
∴
```

Investigation of the numerical results indicates increasing accuracy in the estimate of $g(x) = (e^x - 1)' = e^x$, $g(0) = 1$ with decreasing step size h. The distance between successive approximation sequence terms $d_n = |g_n - g_{n-1}|$ also decreases. It is more intuitive to analyze convergence behavior through a plot rather than a numerical table.

```
∴ clf(); plot(h[2:N],d,"-o"); xlabel("h"); ylabel("d");
∴ cd(homedir()*"//courses//MATH661//images"); savefig("L02Fig01a.eps");
∴
```

The intent of the rate and order of approximation definitions is to state that the distance between successive terms behaves as

$$d_{n+1} \cong s d_n^q,$$

in the hope that this is a Cauchy sequence, and successively closer terms actually indicate convergence. The convergence parameters (s, q) can be isolated by taking logarithms, $c_n = \log d_n$ leading to a linear dependence

$$c_{n+1} \cong q c_n + \log s.$$

Subtraction of successive terms gives $c_n - c_{n-1} \cong q(c_{n-1} - c_{n-2})$, leading to an average slope estimate

$$q \cong \frac{1}{N-3} \sum_{n=3}^{N-1} \frac{c_n - c_{n-1}}{c_{n-1} - c_{n-2}}$$

```
∴ c=log.(2,d); lh=log.(2,h[2:N]); clf(); plot(lh,c,"-o"); plot([-10,-20],[-10,-20],"k");
  plot([-10,-20],[-10,-30],"g");
∴ xlabel("log(h)"); ylabel("log(d)"); savefig("L02Fig01b.eps");
∴ num=c[3:N-1]-c[2:N-2]; den=c[2:N-2]-c[1:N-3];
∴ q = sum(num ./ den)/(N-3)
```

0.9920966582673338

```
∴
```

The above computations indicate $q \cong 1$, known as *linear convergence*. Figure 4b shows the common practice of depicting guide lines of slope 1 (black) and slope 2 (green) to visually ascertain the rate of convergence. Once the order of approximation q is determined, the rate of aproximation is estimated from

$$\log s \cong \frac{1}{N-2}\sum_{n=2}^{N-1}(c_n - q c_{n-1}).$$

```
∴ s=exp(sum(c[2:N-1]-q*c[1:N-2])/(N-2))
```

```
0.3252477724180383
```

```
∴
```

The above results suggest successive approximants become closer spaced according to

$$d_n \cong 0.124\, d_{n-1}$$

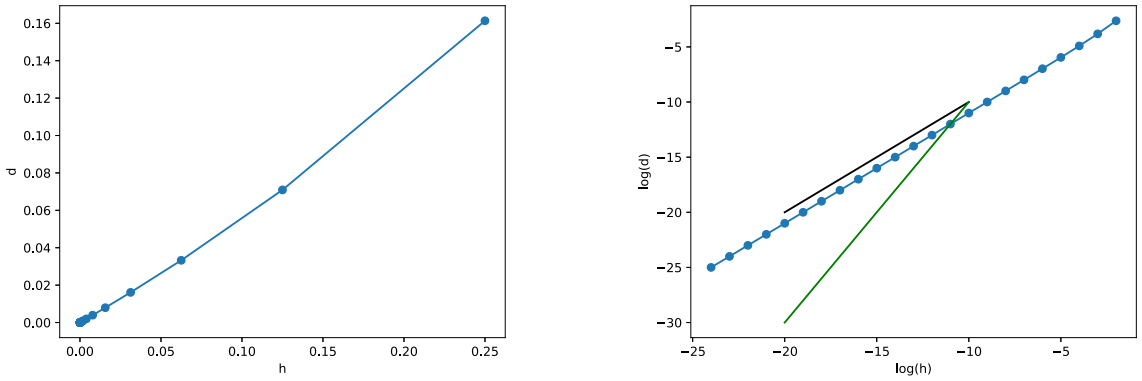


Figure 4. (a, left). Convergence plot; (b,right) Convergence plot in logarithmic coordinates.

Repeat the above experiment at $x_0 = \ln 2$, where $g(\ln 2) = 2$, and using a different approximation of the derivative

$$g_n = \frac{f(\ln 2 + h_n) - f(\ln 2 - h_n)}{2h_n}.$$

For this experiment, in addition to the rate and order of approximation (s, q), also determine the rate and order of convergence (r, p) using

$$b_n = |g_n - g(\pi/4)|, b_{n+1} \cong r b_n^p, a_n = \log b_n, a_{n+1} = p\, a_n + \log r.$$

```
∴ N=32; n=1:N; h=2.0.^(-n); f(x) = exp(x)-1; x0=log(2); f0=f(x0); g0=exp(x0);
```

```
∴ g = (f.(x0 .+ h).-f.(x0 .- h)) ./ (2*h); d=abs.(g[2:N]-g[1:N-1]);
```

```
∴ c=log.(2,d); lh=log.(2,h[2:N]); b=abs.(g[2:N].-g0); a=log.(2,b);
```

```
∴ plot(lh,c,"-o");            plot(lh,a,"-x");            plot([-10,-20],[-10,-20],"k");
  plot([-10,-20],[-10,-30],"g");
```

```
∴ xlabel("log(h)"); ylabel("c, a"); grid("on");
```

```
∴ savefig("L02Fig02.eps");
∴
```

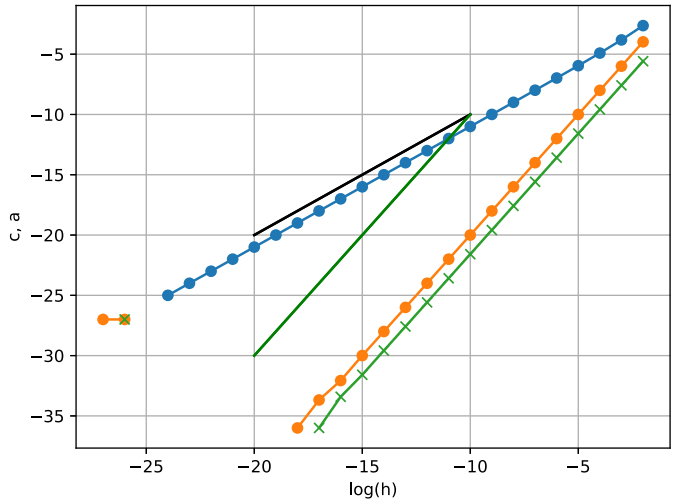


Figure 5. Typical convergence behavior for approximants of a derivative. Blue line shows first-order or linear convergence of approximation $f'(x_0) \cong (f(x_0+h) - f(x_0))/h$ for $f(x) = e^x - 1$ at $x_0 = 0$. The convergence curve is monotone, with decreasing error for all sample points due to fortuitous $f(x_0) = 0$. Green and orange lines indicate that the orders of convergence and approximation are quadratic for $f'(x_0) \cong (f(x_0+h) - f(x_0-h))/(2h)$ for $f(x) = e^x - 1$ at $x_0 = \log 2$. Now, $f(x_0) \neq 0$, and small differences in the numerator are no longer resolved by the floating point system leading to an *increase* in the error for $\log(h) < -20$. The numerical experiment indicates that order of approximation can be used interchangeably with order of convergence, i.e., closer spacing of successive approximations is often an indication of convergence.

2. Convergence acceleration

Given some approximation sequence $\{x_n\}_{n\in\mathbb{N}}$, $x_n \longrightarrow x$, with x solution of problem $f(x) = 0$, it is of interest to construct a more rapidly convergent sequence $\{y_n\}_{n\in\mathbb{N}}$, $y_n \longrightarrow x$. Knowledge of the order of convergence p can be used to achieve this purpose by writing

$$x_n - x \cong r(x_{n-1} - x)^p, x_{n-1} - x \cong r(x_{n-2} - x)^p, \tag{20}$$

and taking the ratio to obtain

$$\frac{x_n - x}{x_{n-1} - x} = \left(\frac{x_{n-1} - x}{x_{n-2} - x}\right)^p. \tag{21}$$

For $p \in \mathbb{N}$, the above is a polynomial equation of degree p that can be solved to obtain x. Since (20) is an approximation, solving (21) gives an approximation of the exact limit.

2.1. Aitken acceleration

One of the widely used acceleration techniques was published by Aitken (1926, but had been in use since Medieval times) for $p = 1$ in which case (21) gives

$$x_n x_{n-2} - (x_n + x_{n-2})x = x_{n-1}^2 - 2x_{n-1}x \Longrightarrow x = \frac{x_n x_{n-2} - x_{n-1}^2}{x_n - 2x_{n-1} + x_{n-2}}.$$

The above suggests that starting from $\{x_n\}_{n\in\mathbb{N}}$, the sequence $\{a_n\}_{n\in\mathbb{N}}$ with

$$a_n = \frac{x_n x_{n-2} - x_{n-1}^2}{x_n - 2x_{n-1} + x_{n-2}} = x_n - \frac{(x_n - x_{n-1})^2}{x_n - 2x_{n-1} + x_{n-2}},$$

might converge faster towards the limit. Investigate by revisiting the numerical experiment on approximation of the derivative $g = f'$ of $f(x) = e^x - 1$ at $x_0 = 0$, using

$$g_n = \frac{f_n - f(0)}{h_n}, f_n = f(h_n), h_n = 2^{-n}.$$

```
∴ N=24; n=1:N; h=2.0.^(-n); f(x) = exp(x)-1; x0=0; f0=f(x0);
∴ g = (f.(h).-f0) ./ h; a = copy(g);
∴ a[3:N] = g[3:N] - (g[3:N]-g[2:N-1]).^2 ./ (g[3:N]-2*g[2:N-1]+g[1:N-2]);
∴ lh=log.(2,h); d=log.(2,abs.(g.-1)); b=log.(2,abs.(a.-1));
∴ clf();      plot(lh,d,"-o");      plot(lh,b,"-x");      plot([-10,-20],[-10,-20],"k");
  plot([-10,-20],[-10,-30],"g");    xlabel("log(h)");    ylabel("g, a");    grid("on");
  savefig("L02Fig03.eps");
∴
```

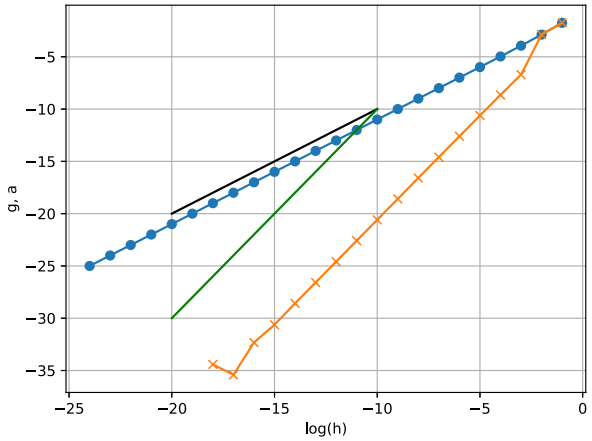


Figure 6. Aitken acceleration of linearly convergent sequence (blue dots) yields a close-to-quadratic convergent sequence (orange x).

3. Approximation correction types

Several approaches may be used in construction of an approximating sequence $\{x_n\}_{n\in\mathbb{N}}$. The approaches exemplified below for $x_n \in \mathbb{R}$, can be generalized when x_n is some other type of mathematical object.

3.1. Additive corrections

Returning to the Leibniz series

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots,$$

the sequence of approximations is $\{L_n\}_{n\in\mathbb{N}}$ with general term

$$L_n = \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}.$$

Note that successive terms are obtained by an additive correction

$$L_n = L_{n-1} + \frac{(-1)^n}{2n+1}, L_n \longrightarrow \frac{\pi}{4}.$$

Another example, again giving an approximation of π is the Srinivasa Ramanujan series

$$R_n = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{n} \frac{(4k)!\,(1103+26390k)}{(k!)^4\, 396^{4k}}, \lim_{n\to\infty} R_n = \frac{1}{\pi},$$

that can be used to obtain many digits of accuracy with just a few terms.

An example of the generalization of this approach is the Taylor series of a function. For example, the familiar sine power series

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots,$$

is analogous, but with rationals now replaced by monomials, and the limit is now a function $\sin\colon \mathbb{R} \longrightarrow [-1,1]$. The general term is

$$T_n(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1)!},$$

and the same type of additive correction appears, this time for functions,

$$T_n(x) = T_{n-1}(x) + \frac{(-1)^n x^{2n+1}}{(2n+1)!}, T_n(x) \longrightarrow \sin x.$$

3.2. Multiplicative corrections

Approximating sequences need not be constructed by adding a correction. Consider the approximation of $\pi/2$ given by Wallis's product (1656)

$$S_n = \left(\frac{2}{1}\cdot\frac{2}{3}\right)\cdot\left(\frac{4}{3}\cdot\frac{4}{5}\right)\cdot\left(\frac{6}{5}\cdot\frac{6}{7}\right)\ldots, S_n = \prod_{k=1}^{n} \frac{4k^2}{4k^2-1}, S_n \longrightarrow \frac{\pi}{2},$$

for which

$$S_n = S_{n-1}\cdot\left(\frac{4n^2}{4n^2-1}\right).$$

Another famous example is the Viète formula from 1593

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdot \ldots, \; V_n = \prod_{k=1}^{n} \frac{\mathop{\mathrm{N}}_{j=1}^{k} \sqrt[2]{2}}{2}$$

in which the correction is multiplicative with numerators given by nested radicals. Similar to the $\sum$ symbol for addition, and the $\prod$ symbol for multiplication, the N symbol is used to denote nested radicals

$$\mathop{\mathrm{N}}_{j=1}^{k} \sqrt[b]{a_j} = \sqrt[b_1]{a_1 + \sqrt[b_2]{a_2 + \sqrt[b_3]{a_3 + \cdots + \sqrt[b_k]{a_k}}}}.$$

In the case of the Viète formula, $a_j = 2$, $b_j = 2$ for all j.

3.3. Continued fractions

Yet another alternative is that of continued fractions, with one possible approximation of π given by

$$\pi + 3 = 6 + \cfrac{1^2}{6 + \cfrac{3^2}{6 + \cfrac{5^2}{6 + \ddots}}}. \tag{22}$$

A notation is introduced for continued fractions using the K symbol

$$F_n = b_0 + \mathop{\mathrm{K}}_{k=1}^{n} \frac{a_k}{b_k} = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \ddots}}}.$$

Using this notation, the sequences arising in the continued fraction representation of π are $\{a_n\}_{n\in\mathbb{N}}, \{b_n\}_{n\in\mathbb{N}}$ chosen as $a_k = (2k-1)^2$ for $k \in \mathbb{N}_+$, and $b_k = 6$ for $k \in \mathbb{N}$.

$$\pi = \lim_{n\to\infty} \left(6 + \mathop{\mathrm{K}}_{k=1}^{n} \frac{(2k-1)^2}{6} \right).$$

3.4. Composite corrections

The above correction techniques used arithmetic operations. The repeated radical coefficients in the Viète formula suggest consideration of repeated composition of arbitrary functions $t_0, t_1, \ldots, t_n$ to construct the approximant

$$T_n = t_0 \circ t_1 \circ \ldots \circ t_n = \bigodot_{k=0}^{n} t_k.$$

This is now a general framework, in which all of the preceeding correction approaches can be expressed. For example, the continued fraction formula (22) is recovered through the functions

$$t_0(z) = 6 + z, t_1(z) = \frac{1}{6+z}, \ldots, t_k(z) = \frac{(2k-1)^2}{6+z},$$

and evaluation of the composite function at $z = 0$

$$F_n = T_n(0).$$

This general framework is of significant current interest since such composition of nonlinear functions is the basis of deep neural network approximations.

Summary.

- The cornerstone of scientific computing is construction of approximating sequences.
- The problem of number approximation leads to definition of concepts and techniques that can be extended to more complex mathematical objects.
- A primary objective is the construction of efficient approximating sequences, with efficiency characterized through concepts such as order and speed of convergence.
- Though often enforced analytically, limiting behavior of the sequence is of secondary interest. As seen in the approximation of a derivative, the approximating sequence might diverge, yet give satisfactory answers for some range of indices.
- Though by far the most widely studied and used approach to approximation, additive corrections are not the only possibility.
- Alternative correction techniques include: multiplication, continued fractions, or repeated function composition.
- Repeated composition of functions is used in constructing deep neural network approximants.

LECTURE 3: PROBLEMS AND ALGORITHMS

1. Mathematical problems

1.1. Formalism for defining a mathematical problem

In general, mathematical problems can be thought of as mappings from some set of inputs X to some set of outputs Y. The mapping is often carried out through a function f, i.e., a procedure that associates a single $y \in Y$ to some input $x \in X$

$$f : X \longrightarrow Y, y = f(x), x \xrightarrow{f} y$$

Examples:

- Compute the square of a real:

$$X = \mathbb{R}, Y = \mathbb{R}, y = f(x) = x^2.$$

- Find x solution of $ax + b = c$ for given $a, b, c \in \mathbb{R}$, $a \neq 0$. The inputs to this problem are a, b, c and the output is the solution $(c - b)/a$

$$X = \mathbb{R}\backslash\{0\} \times \mathbb{R} \times \mathbb{R}, Y = \mathbb{R}, f(a, b, c) = (c - b)/a.$$

- Compute the innner product of two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$:

$$X = \mathbb{R}^n \times \mathbb{R}^n, Y = \mathbb{R}, y = f(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i=1}^{n} u_i v_i$$

  with u_i, v_i the components of $\boldsymbol{u}, \boldsymbol{v}$. Note that the input set is the Cartesian product of sets of vectors and the output set is the reals. Such functions defined from sets of vectors (more accurately vector spaces) to reals (more accurately scalars) are called *functionals*.

- Compute the definite integral

$$(u, v) = \int_a^b u(x)\, v(x)\, \mathrm{d}x,$$

  with f, g arbitrary continuous functions, denoted by $f, g \in C^{(0)}([a, b])$:

$$X = C^{(0)}([a, b]) \times C^{(0)}([a, b]), Y = \mathbb{R}.$$

  Again, this an example of a functional.

- Compute the derivative of a function $g \in C^{(1)}(\mathbb{R})$, with $C^{(k)}(\mathbb{R})$ the space of functions defined on $\mathbb{R}$ differentiable k times: $X = C^{(1)}(\mathbb{R})$, $Y = C^{(0)}(\mathbb{R})$, $f = \mathrm{d}/\mathrm{d}x$. Note that in this case X, Y are sets of functions, in which case f is referred to as an *operator*.

- Find the roots of a polynomial $p_n(x) = a_n x^n + \ldots + a_1 x + a_0$. The input is the polynomial specified by the vector of coefficients $\boldsymbol{a} \in \mathbb{R}^{n+1}$. The output is another vector $\boldsymbol{x} \in \mathbb{R}^n$ whose components are roots, $p_n(x_i) = 0$

$$X = \mathbb{R}^{n+1}, Y = \mathbb{R}^n.$$

  The function $f : X \longrightarrow Y$ cannot be written explicitly (corollary of Abel-Ruffini theorem), but there are approximations $\tilde{f}$ of the root-finding function that can be implemented such $\tilde{f} \cong f$.

Note that the specification of a mathematical problem requires definition of the triplet (X, Y, f).

Once a problem is specified, the natural question is to ascertain whether a solution is possible. Generally, simple affirmation of the existence of a solution is the objective of some field of mathematics (e.g., analysis, functional analysis). From the point of view of science, an essential question is not only existence but also:

1. how does the output $y = f(x)$ change if x changes?

2. what are the constructive methods to approximate y?

1.2. Vector space

The above general definition of a mathematical problem must be refined in order to assess magnitude of changes in inputs or outputs. A first step is to introduce some structure in the input and output sets X, Y. Using these sets, vector spaces $\mathcal{U} = (V, S, +, \cdot)$ are constructed, consisting of a set of vectors V, a set of scalars S, an addition operation $+$, and a scaling operation $\cdot$. The vector space is often referred to simply by its set of vectors V, when the set of scalars, addition operation, and scaling operation are self-evident in context.

Formally,a *vector space* $\mathcal{U}$ is defined by a set V whose elements satisfy certain scaling and addition properties, denoted all together by the 4-tuple $\mathcal{U} = (V, S, +, \cdot)$. The first element of the 4-tuple is a set whose elements are called *vectors*. The second element is a set of scalars, and the third is the vector addition operation. The last is the scaling operation, seen as multiplication of a vector by a scalar. The vector addition and scaling operations must satisfy rules suggested by positions or forces in three-dimensional space, which are listed in Table 1.1. In particular, a vector space requires definition of two distinguished elements: the zero vector $\mathbf{0} \in V$, and the identity scalar element $1 \in S$.

Addition rules for	$\forall \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in V$
$\boldsymbol{a} + \boldsymbol{b} \in V$	Closure
$\boldsymbol{a} + (\boldsymbol{b} + \boldsymbol{c}) = (\boldsymbol{a} + \boldsymbol{b}) + \boldsymbol{c}$	Associativity
$\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b} + \boldsymbol{a}$	Commutativity
$\mathbf{0} + \boldsymbol{a} = \boldsymbol{a}$	Zero vector
$\boldsymbol{a} + (-\boldsymbol{a}) = \mathbf{0}$	Additive inverse
Scaling rules for	$\forall \boldsymbol{a}, \boldsymbol{b} \in V, \forall x, y \in S$
$x\boldsymbol{a} \in V$	Closure
$x(\boldsymbol{a} + \boldsymbol{b}) = x\boldsymbol{a} + x\boldsymbol{b}$	Distributivity
$(x + y)\boldsymbol{a} = x\boldsymbol{a} + y\boldsymbol{a}$	Distributivity
$x(y\boldsymbol{a}) = (xy)\boldsymbol{a}$	Composition
$1\boldsymbol{a} = \boldsymbol{a}$	Scalar identity

Table 2. Vector space $\mathcal{U} = (V, S, +, \cdot)$ properties for arbitrary $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in V$

1.3. Norm

A first step is quantification of the changes in input or output, assumed to have the structure of a vector space, $\mathcal{X} = (X, \mathbb{R}, +, \cdot), \mathcal{Y} = (Y, \mathbb{R}, +, \cdot)$.

DEFINITION 3. *A norm on vector space $\mathcal{X}$ is a function $\|\,\|: X \longrightarrow \mathbb{R}_+$, that for any $x, y, z \in X$, $\alpha \in \mathbb{R}$ satisfies the properties:*

1. $\|x\| = 0$ *if and only if x=0.*
2. $\|ax\| = |a|\,\|x\|$
3. $\|x + y\| \leq \|x\| + \|y\|$

1.4. Condition number

The ratio of changes in output to changes in input is the absolute condition number of a problem.

DEFINITION 4. *The problem $f: X \longrightarrow Y$ has absolute condition number*

$$\hat{\kappa} = \lim_{\varepsilon \longrightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|f(x + \delta x) - f(x)\|}{\|\delta x\|}$$

To avoid influence of choice of reference unit, the relative condition number is also introduced.

DEFINITION 5. *The problem $f: X \longrightarrow Y$ has relative condition number*

$$\hat{\kappa} = \lim_{\varepsilon \longrightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|f(x+\delta x) - f(x)\|}{\|f(x)\|} \cdot \frac{\|x\|}{\|\delta x\|}.$$

2. Solution algorithm

2.1. Accuracy

In scientific computation, the mathematical problem $f: X \longrightarrow Y$ is approximated by an *algorithm* $\tilde{f}: \tilde{X} \longrightarrow \tilde{Y}$, in which is assumed to be computable, and $\tilde{X}, \tilde{Y}$ are vector spaces that approximate X, Y. As a first step in characterizing how well the algorithm $\tilde{f}$ approximates the problem f, consider that $\tilde{X} = X$ and $\tilde{Y} = Y$, i.e., there is no error in representation of the domain and codomain.

DEFINITION 6. *The absolute error of algorithm $\tilde{f}: X \longrightarrow Y$ that approximates the problem $f: X \longrightarrow Y$ is*

$$e = \|\tilde{f}(x) - f(x)\|.$$

DEFINITION 7. *The relative error of algorithm $\tilde{f}: X \longrightarrow Y$ that approximates the problem $f: X \longrightarrow Y$ is*

$$\varepsilon = \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}.$$

DEFINITION 8. *An algorithm $\tilde{f}: X \longrightarrow Y$ is accurate if there exists finite $M \in \mathbb{R}_+$ such that*

$$\varepsilon = \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq M \epsilon_{\text{mach}}$$

The above condition is also denoted as $\varepsilon = O(\epsilon_{\text{mach}})$

2.2. Stability

Algorithms should not catastrophically increase input errors. This is quantified in the concept of stability.

DEFINITION 9. *An algorithm $\tilde{f}: X \longrightarrow Y$ is forward stable if*

$$\|\tilde{x} - x\| / \|x\| = O(\epsilon_{\text{mach}}) \Longrightarrow \|\tilde{f}(x) - f(\tilde{x})\| / \|f(\tilde{x})\| = O(\epsilon_{\text{mach}})$$

The above states that the relative error in the output should be on the order of machine epsilon if the relative in the input is of order machine epsilon. Note that the constants in the order statements M, N are usually different from one another, $\|\tilde{x} - x\| / \|x\| \leq M \epsilon_{\text{mach}}$, $\|\tilde{f}(x) - f(\tilde{x})\| / \|f(\tilde{x})\| \leq N \epsilon_{\text{mach}}$.

DEFINITION 10. *An algorithm $\tilde{f}: X \longrightarrow Y$ is backward stable if from existence of some $\tilde{x}$ such that $\tilde{f}(x) = f(\tilde{x})$, it results that*

$$\|\tilde{x} - x\|/\|x\| = O(\epsilon_{\text{mach}}).$$

Backward stability asserts that the result of the algorithm on exact input data is the same as the solution to the mathematical problem for nearby data (with distance on order of machine epsilon).

Summary.

- Mathematical problems are stated as functions from a set of inputs X to a set of outputs Y, $f: X \longrightarrow Y$
- The difficulty of a mathematical problem is assessed by measuring the effect of changes in input
- To quantify changes in inputs and outputs, the framework of a normed vector space is introduced
- The ratio of norm of output change to norm of input change is the absolute condition number of a problem

$$\hat{\kappa} = \lim_{\varepsilon \longrightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \frac{\|f(x + \delta x) - f(x)\|}{\|\delta x\|}$$

- Algorithms are constructive approximations of mathemtical problems $\tilde{f}: X \longrightarrow Y$. The accuracy of an algorithm is assessed by comparison of the algorithm output to that of the mathematical problem through absolute error e and relative error ε

$$e = \|\tilde{f}(x) - f(x)\|, \varepsilon = \frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$$

- The tendency of an algorithm to amplify pertubations of input is assessed by the concept of stability
- Algorithms that do not amplify relative changes in input of the size of machine precision are forward stable.
- Algorithms that compute the exact result of a mathematical problem for changes in put of the size of machine precision are backward stable.

Part II

Linear Approximation

CHAPTER 1
LINEAR ALGEBRA

LECTURE 4: LINEAR COMBINATIONS

1. Finite-dimensional vector spaces

1.1. Overview

The definition from Table 1 of a vector space reflects everyday experience with vectors in Euclidean geometry, and it is common to refer to such vectors by descriptions in a Cartesian coordinate system. For example, a position vector $\boldsymbol{r}$ within the plane can be referred through the pair of coordinates (x, y). This intuitive understanding can be made precise through the definition of a vector space $\mathscr{R}_2 = (\mathbb{R}^2, \mathbb{R}, +, \cdot)$, called the real 2-space. Vectors within $\mathscr{R}_2$ are elements of $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R} = \{(x, y) \mid x, y \in \mathbb{R}\}$, meaning that a vector is specified through two real numbers, $\boldsymbol{r} \leftrightarrow (x, y)$. Addition of two vectors, $\boldsymbol{q} \leftrightarrow (s, t)$, $\boldsymbol{r} \leftrightarrow (x, y)$ is defined by addition of coordinates $\boldsymbol{q} + \boldsymbol{r} = (s + x, t + v)$. Scaling $\boldsymbol{r} \leftrightarrow (x, y)$ by scalar a is defined by $a\boldsymbol{r} \leftrightarrow (ax, ay)$. Similarly, consideration of position vectors in three-dimensional space leads to the definition of the $\mathscr{R}_3 = (\mathbb{R}^3, \mathbb{R}, +, \cdot)$, or more generally a real m-space $\mathscr{R}_m = (\mathbb{R}^m, \mathbb{R}, +, \cdot)$, $m \in \mathbb{N}$, $m > 0$.

Addition rules for	$\forall \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in V$
$\boldsymbol{a} + \boldsymbol{b} \in V$	Closure
$\boldsymbol{a} + (\boldsymbol{b} + \boldsymbol{c}) = (\boldsymbol{a} + \boldsymbol{b}) + \boldsymbol{c}$	Associativity
$\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b} + \boldsymbol{a}$	Commutativity
$\boldsymbol{0} + \boldsymbol{a} = \boldsymbol{a}$	Zero vector
$\boldsymbol{a} + (-\boldsymbol{a}) = \boldsymbol{0}$	Additive inverse
Scaling rules for	$\forall \boldsymbol{a}, \boldsymbol{b} \in V$, $\forall x, y \in S$
$x\boldsymbol{a} \in V$	Closure
$x(\boldsymbol{a} + \boldsymbol{b}) = x\boldsymbol{a} + x\boldsymbol{b}$	Distributivity
$(x + y)\boldsymbol{a} = x\boldsymbol{a} + y\boldsymbol{a}$	Distributivity
$x(y\boldsymbol{a}) = (xy)\boldsymbol{a}$	Composition
$1\boldsymbol{a} = \boldsymbol{a}$	Scalar identity

Table 1.1. Vector space $\mathcal{V} = (V, S, +, \cdot)$ properties for arbitrary $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in V$

Note however that there is no mention of coordinates in the definition of a vector space as can be seen from the list of properties in Table 1. The intent of such a definition is to highlight that besides position vectors, many other mathematical objects follow the same rules. As an example, consider the set of all continuous functions $C(\mathbb{R}) = \{f \mid f : \mathbb{R} \longrightarrow \mathbb{R}\}$, with function addition defined by the sum at each argument t, $(f + g)(t) = f(t) + g(t)$, and scaling by $a \in \mathbb{R}$ defined as $(af)(t) = af(t)$. Read this as: "given two continuous functions f and g, the function $f + g$ is defined by stating that its value for argument x is the sum of the two real numbers $f(t)$ and $g(t)$". Similarly: "given a continuous function f, the function af is defined by stating that its value for argument t is the product of the real numbers a and $f(t)$". Under such definitions $C^0 = (C(\mathbb{R}), \mathbb{R}, +, \cdot)$ is a vector space, but quite different from $\mathscr{R}_m$. Nonetheless, the fact that both C^0 and $\mathscr{R}_m$ are vector spaces can be used to obtain insight into the behavior of continuous functions from Euclidean vectors, and vice versa. This correspondence principle between discrete and continuous formulations is a recurring theme in scientific computation.

1.2. Real vector space $\mathscr{R}_m$

Column vectors. Since the real spaces $\mathscr{R}_m = (\mathbb{R}^m, \mathbb{R}, +, \cdot)$ play such an important role in themselves and as a guide to other vector spaces, familiarity with vector operations in $\mathscr{R}_m$ is necessary to fully appreciate the utility of linear algebra to a wide range of applications. Following the usage in geometry and physics, the m real numbers that specify a vector $\boldsymbol{u} \in \mathbb{R}^m$ are called the *components* of $\boldsymbol{u}$. The one-to-one correspondence between a vector and its components $\boldsymbol{u} \leftrightarrow (u_1, \ldots, u_m)$, is by convention taken to define an equality relationship,

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}, \tag{1.1}$$

with the components arranged vertically and enclosed in square brackets. Given two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$, and a scalar $a \in \mathbb{R}$, vector addition and scaling are defined in $\mathscr{R}_m$ by real number addition and multiplication of components

$$\boldsymbol{u} + \boldsymbol{v} = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ \vdots \\ u_m + v_m \end{bmatrix}, a\boldsymbol{u} = a \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} a u_1 \\ \vdots \\ a u_m \end{bmatrix}. \tag{1.2}$$

The vector space $\mathscr{R}_m$ is defined using the real numbers as the set of scalars, and constructing vectors by grouping together m scalars, but this approach can be extended to any set of scalars S, leading to the definition of the vector spaces $\mathscr{S}_n = (S^n, S, +, \cdot)$. These will often be referred to as *n-vector space of scalars*, signifying that the set of vectors is $V = S^n$.

To aid in visual recognition of vectors, the following notation conventions are introduced:

- vectors are denoted by lower-case bold Latin letters: $\boldsymbol{u}, \boldsymbol{v}$;
- scalars are denoted by normal face Latin or Greek letters: a, b, α, β;
- the components of a vector are denoted by the corresponding normal face with subscripts as in equation (1.1);
- related sets of vectors are denoted by indexed bold Latin letters: $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$.

Row vectors. Instead of the vertical placement or components into one *column*, the components of could have been placed horizontally in one *row* $[\ u_1\ \ldots\ u_m\ ]$, that contains the same data, differently organized. By convention vertical placement of vector components is the preferred organization, and $\boldsymbol{u}$ shall denote a *column vector* henceforth. A transpose operation denoted by a T superscript is introduced to relate the two representations

$$\boldsymbol{u}^T = [\ u_1\ \ldots\ u_m\ ],$$

and $\boldsymbol{u}^T$ is the notation used to denote a *row vector*.

◦ In Julia, horizontal placement of successive components in a row is denoted by a space.

Compatible vectors. Addition of real vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$ defines another vector $\boldsymbol{w} = \boldsymbol{u} + \boldsymbol{v} \in \mathbb{R}^m$. The components of $\boldsymbol{w}$ are the sums of the corresponding components of $\boldsymbol{u}$ and $\boldsymbol{v}$, $w_i = u_i + v_i$, for $i = 1, 2, \ldots, m$. Addition of vectors with different number of components is not defined, and attempting to add such vectors produces an error. Such vectors with different number of components are called *incompatible*, while vectors with the same number of components are said to be *compatible*. Scaling of $\boldsymbol{u}$ by a defines a vector $\boldsymbol{z} = a\boldsymbol{u}$, whose components are $z_i = a u_i$, for $i = 1, 2, \ldots, m$.

1.3. Working with vectors

Ranges. The vectors used in applications usually have a large number of components, $m \gg 1$, and it is important to become proficient in their manipulation. Previous examples defined vectors by explicit listing of their m components. This is impractical for large m, and support is provided for automated generation for often-encountered situations. First, observe that Table 1 mentions one distinguished vector, the zero element that is a member of any vector space $\mathbf{0} \in V$. The zero vector of a real vector space $\mathscr{R}_m$ is a column vector with m components, all of which are zero, and a mathematical convention for specifying this vector is $\mathbf{0}^T = [\ 0\ \ 0\ \ \ldots\ \ 0\ ] \in \mathbb{R}^m$. This notation specifies that transpose of the zero vector is the row vector with m zero components, also written through explicit indexing of each component as $\mathbf{0}_i = 0$, for $i = 1, \ldots, m$. Keep in mind that the zero vector $\mathbf{0}$ and the zero scalar 0 are different mathematical objects.

The ellipsis symbol in the mathematical notation is transcribed in Julia by the notion of a range, with `1:m` denoting all the integers starting from 1 to m, organized as a row vector. The notation is extended to allow for strides different from one, and the mathematical ellipsis $i = m, m-1, \ldots, 1$ is denoted as `m:-1:1`. In general `r:s:t` denotes the set of numbers $\{r, r+s, \ldots, r+ns\}$ with $r + ns \leqslant t$, and r, s, t real numbers and n a natural number, $r, s, t \in \mathbb{R}$, $n \in \mathbb{N}$. If there is no natural number n such that $r + ns \leqslant t$, an empty vector with no components is returned.

2. Linear combinations

2.1. Linear combination as a matrix-vector product

The expression $\boldsymbol{x} = x_1\,\boldsymbol{e}_1 + x_2\,\boldsymbol{e}_2 + \cdots + x_m\,\boldsymbol{e}_m$ expresses the idea of scaling vectors within a set and subsequent addition to form a new vector $\boldsymbol{x}$. The matrix $\boldsymbol{I} = [\ \boldsymbol{e}_1\ \ \boldsymbol{e}_2\ \ \ldots\ \ \boldsymbol{e}_m\ ]$ groups these vectors together in a single entity, and the scaling factors are the components of the vector $\boldsymbol{x}$. To bring all these concepts together it is natural to consider the notation

- $$\boldsymbol{x} = \boldsymbol{I}\boldsymbol{x},$$

as a generalization of the scalar expression $x = 1 \cdot x$. It is clear what the operation $\boldsymbol{I}\boldsymbol{x}$ should signify: it should capture the vector scaling and subsequent vector addition $x_1\,\boldsymbol{e}_1 + x_2\,\boldsymbol{e}_2 + \cdots + x_m\,\boldsymbol{e}_m$. A specific meaning is now ascribed to $\boldsymbol{I}\boldsymbol{x}$ by identifying two definitions to one another.

Linear combination. Repeateadly stating "vector scaling and subsequent vector addition" is unwieldy, so a special term is introduced for some given set of vectors $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$.

DEFINITION. (LINEAR COMBINATION) . *The linear combination of vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n \in V$ with scalars $x_1, x_2, \ldots, x_n \in S$ in vector space $(V, S, +, \cdot)$ is the vector $\boldsymbol{b} = x_1\,\boldsymbol{a}_1 + x_2\,\boldsymbol{a}_2 + \ldots x_n\,\boldsymbol{a}_n$.*

Matrix-vector product. Similar to the grouping of unit vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m$ into the identity matrix $\boldsymbol{I}$, a more concise way of referring to arbitrary vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ from the same vector space is the matrix $\boldsymbol{A} = [\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \ldots\ \ \boldsymbol{a}_n\ ]$. Combining these observations leads to the definition of a matrix-vector product.

DEFINITION. (MATRIX-VECTOR PRODUCT) . *In the vector space $(V, S, +, \cdot)$, the product of matrix $\boldsymbol{A} = [\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \ldots\ \ \boldsymbol{a}_n\ ]$ composed of columns $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n \in V$ with the vector $\boldsymbol{x} \in S_n$ whose components are scalars $x_1, x_2, \ldots, x_n \in S$ is the linear combination $\boldsymbol{b} = x_1\,\boldsymbol{a}_1 + x_2\,\boldsymbol{a}_2 + \ldots x_n\,\boldsymbol{a}_n = \boldsymbol{A}\boldsymbol{x} \in V$.*

2.2. Linear algebra problem examples

Linear combinations in E_2. Consider a simple example that leads to a common linear algebra problem: decomposition of forces in the plane along two directions. Suppose a force is given in terms of components along the Cartesian x, y-axes, $\boldsymbol{b} = b_x\,\boldsymbol{e}_x + b_y\,\boldsymbol{e}_y$, as expressed by the matrix-vector multiplication $\boldsymbol{b} = \boldsymbol{I}\boldsymbol{b}$. Note that the same force could be obtained by linear combination of other vectors, for instance the normal and tangential components of the force applied on an inclined plane with angle θ, $\boldsymbol{b} = x_t\,\boldsymbol{e}_t + x_n\,\boldsymbol{e}_n$, as in Figure 1.1. This defines an alternate reference system for the problem. The unit vectors along these directions are

- $$\boldsymbol{t} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \boldsymbol{n} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix},$$

```
∴ θ=π/6.; c=cos(θ); s=sin(θ); t=[c; s]; n=[-s; c];
∴
```

and can be combined into a matrix $\boldsymbol{A} = [\ \boldsymbol{t}\ \ \boldsymbol{n}\ ]$. The value of the components (x_t, x_n) are the scaling factors and can be combined into a vector $\boldsymbol{x} = [\ x_t\ \ x_n\ ]^T$. The same force must result irrespective of whether its components are given along the Cartesian axes or the inclined plane directions leading to the equality

- $$\boldsymbol{I}\boldsymbol{b} = \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}. \tag{1.9}$$

```
∴ b=[0.2; 0.4]; I*b
```

$$\begin{bmatrix} 0.2 \\ 0.4 \end{bmatrix} \tag{1.10}$$

```
∴
```

Interpret equation (1.9) to state that the vector $\boldsymbol{b}$ could be obtained either as a linear combination of $\boldsymbol{I}$, $\boldsymbol{b} = \boldsymbol{Ib}$, or as a linear combination of the columns of $\boldsymbol{A}$, $\boldsymbol{b} = \boldsymbol{Ax}$. Of course the simpler description seems to be $\boldsymbol{Ib}$ for which the components are already known. But this is only due to an arbitrary choice made by a human observer to define the force in terms of horizontal and vertical components. The problem itself suggests that the tangential and normal components are more relevant; for instance a friction force would be evaluated as a scaling of the normal force.

- The components of $\boldsymbol{b}$ in this more natural reference system are not known, but can be determined by solving the vector equality $\boldsymbol{Ax} = \boldsymbol{Ib} = \boldsymbol{b}$, known as a *linear system of equations*, implemented in many programming environments (Julia, Matlab, Octave) through the backslash operator `x=A\b`.

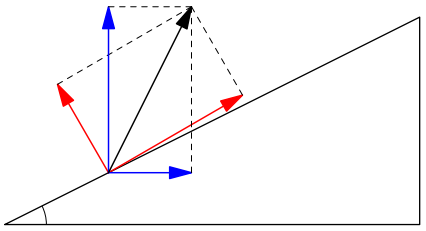


Figure 1.1. Alternative decompositions of force on inclined plane.

Linear combinations in $\mathbb{R}_m$ and $C^0[0, 2\pi)$. Linear combinations in a real space can suggest properties or approximations of more complex objects such as continuous functions. Let $C^0[0, 2\pi) = (C[0, 2\pi), \mathbb{R}, +, \cdot)$ denote the vector space of continuous functions that are periodic on the interval $[0, 2\pi)$, $C[0, \pi) = \{f \mid f : \mathbb{R} \longrightarrow \mathbb{R}, f(t) = f(t + 2\pi)\}$. Recall that vector addition is defined by $(f + g)(t) = f(t) + g(t)$, and scaling by $(af)(t) = af(t)$, for $f, g \in C[0, 2\pi)$, $a \in \mathbb{R}$. Familiar functions within this vector space are $\sin(kt)$, $\cos(kt)$ with $k \in \mathbb{N}$, and these can be recognized to intrinsically represent periodicity on $[0, 2\pi)$, a role analogous to the normal and tangential directions in the inclined plane example. Define now another periodic function $b(t + 2\pi) = b(t)$ by repeating the values $b(t) = t(\pi - t)(2\pi - t)$ from the interval $[0, 2\pi)$ on all intervals $[2p\pi, 2(p + 1)\pi]$, for $p \in \mathbb{Z}$. The function b is not given in terms of the "naturally" periodic functions $\sin(kt)$, $\cos(kt)$, but could it thus be expressed? This can be stated as seeking a linear combination $b(t) = \sum_{k=1}^{\infty} x_k \sin(kt)$, as studied in Fourier analysis. The coefficients x_k could be determined from an analytical formula involving calculus operations $x_k = \frac{1}{\pi}\int_0^{2\pi} b(t) \sin(kt)\, \mathrm{d}t$, but we'll seek an approximation using a linear combination of n terms

$$b(t) \cong \sum_{k=1}^{n} x_k \sin(kt), A(t) = [\ \sin(t)\ \ \sin(2t)\ \ \ldots\ \ \sin(nt)\ ], A : \mathbb{R} \longrightarrow \mathbb{R}^n.$$

Organize this as a matrix vector product $b(\boldsymbol{t}) \cong A(\boldsymbol{t})\, \boldsymbol{x}$, with

$$A(\boldsymbol{t}) = [\ \sin(\boldsymbol{t})\ \ \sin(2\boldsymbol{t})\ \ \ldots\ \ \sin(n\boldsymbol{t})\ ], \boldsymbol{x} = [\ x_1\ \ x_2\ \ \ldots\ \ x_n\ ]^T \in \mathbb{R}^n.$$

The idea is to sample the column vectors of $A(\boldsymbol{t})$ at the components of the vector $\boldsymbol{t} = [\ t_1\ \ t_2\ \ \ldots\ \ t_m\ ]^T \in \mathbb{R}^m$, $t_j = (j - 1)h$, $j = 1, 2, \ldots, m$, $h = \pi / m$. Let $\boldsymbol{b} = b(\boldsymbol{t})$, and $\boldsymbol{A} = A(\boldsymbol{t})$, denote the so-sampled b, A functions leading to the definition of a vector $\boldsymbol{b} \in \mathbb{R}^m$ and a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. There are n coefficients available to scale the column vectors of $\boldsymbol{A}$, and $\boldsymbol{b}$ has m components. For $m > n$ it is generally not possible to find $\boldsymbol{x}$ such that $\boldsymbol{Ax}$ would exactly equal $\boldsymbol{b}$, but as seen later the condition to be as close as possible to $\boldsymbol{b}$ leads to a well defined solution procedure. This is known as a least squares problem and is automatically applied in the `x=A\b` instruction when the matrix `A` is not square. As seen in the following numerical experiment and Figure 1.2, the approximation is excellent and the information conveyed by $m = 1000$ samples of $b(t)$ is now much more efficiently stored in the form chosen for the columns of $\boldsymbol{A}$ and the $n = 5$ scaling coefficients that are the components of $\boldsymbol{x}$.

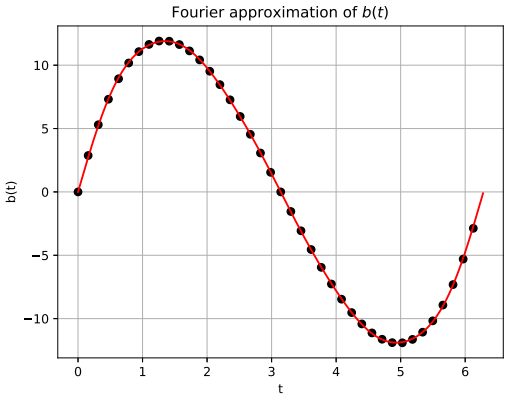


Figure 1.2. Comparison of least squares approximation (red line) with samples (black dots) of exact function $b(t) = t(\pi - t)(2\pi - t)$

Summary.

- A widely used framework for constructing additive approximations is the vector space algebraic space structure in which scaling and addition operations are defined
- In a vector space linear combinations are used to construct more complicated objects from simpler ones

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} = x_1 \boldsymbol{a}_1 + \cdots + x_n \boldsymbol{a}_n$$

LECTURE 5: LINEAR FUNCTIONALS AND MAPPINGS

1. Functions

1.1. Relations

A general procedure to relate input values from set X to output values from set Y is to first construct the set of all possible instances of $x \in X$ and $y \in Y$, which is the *Cartesian product* of X with Y, denoted as $X \times Y = \{(x, y) \mid x \in X, y \in Y\}$. Usually only some associations of inputs to outputs are of interest leading to the following definition.

DEFINITION. (RELATION) . *A relation R between two sets X, Y is a subset of the Cartesian product $X \times Y$, $R \subseteq X \times Y$.*

Associating an output to an input is also useful, leading to the definition of an *inverse relation* as $R^{-1} \subseteq Y \times X$, $R^{-1} = \{(y, x) \mid (x, y) \in R\}$. Note that an inverse exists for any relation, and the inverse of an inverse is the original relation, $(R^{-1})^{-1} = R$.

Homogeneous relations. Many types of relations are defined in mathematics and encountered in linear algebra. A commonly encountered type of relationship is from a set onto itself, known as a *homogeneous* relation. For homogeneous relations $H \subseteq A \times A$, it is common to replace the set membership notation $(a, b) \in H$ to state that $a \in A$ is in relationship H with $b \in A$, with a binary operator notation $a \overset{H}{\sim} b$. Familiar examples include the equality and less than relationships between reals, $E, L \subseteq \mathbb{R} \times \mathbb{R}$, in which $(a, b) \in E$ is replaced by $a = b$, and $(a, b) \in L$ is replaced by $a < b$. The equality relationship is its own inverse, and the inverse of the less than relationship is the greater than relation $G \subseteq \mathbb{R} \times \mathbb{R}$, $G = L^{-1}$, $a < b \implies b > a$. Homogeneous relations $H \subseteq A \times A$ are classified according to the following criteria.

Reflection. Relation H is reflexive if $(a, a) \in H$ for any $a \in A$. The equality relation $E \subseteq \mathbb{R} \times \mathbb{R}$ is reflexive, $\forall a \in A$, $a = a$, the less than relation $L \subseteq \mathbb{R} \times \mathbb{R}$ is not, $1 \in R, 1 \not< 1$.

Symmetry. Relation H is symmetric if $(a, b) \in H$ implies that $(b, a) \in H$, $(a, b) \in H \implies (b, a) \in H$. The equality relation $E \subseteq \mathbb{R} \times \mathbb{R}$ is symmetric, $a = b \implies b = a$, the less than relation $L \subseteq \mathbb{R} \times \mathbb{R}$ is not, $a < b \not\Rightarrow b < a$.

Anti-symmetry. Relation H is anti-symmetric if $(a,b)\in H$ for $a\neq b$, then $(b,a)\notin H$. The less than relation $L\subseteq\mathbb{R}\times\mathbb{R}$ is antisymmetric, $a<b\Longrightarrow b\neq a$.

Transitivity. Relation H is transitive if $(a,b)\in H$ and $(b,c)\in H$ implies $(a,c)\in H$. for any $a\in A$. The equality relation $E\subseteq\mathbb{R}\times\mathbb{R}$ is transitive, $a=b\wedge b=c\Longrightarrow a=c$, as is the less than relation $L\subseteq\mathbb{R}\times\mathbb{R}$, $a<b\wedge b<c\Longrightarrow a<c$.

Certain combinations of properties often arise. A homogeneous relation that is reflexive, symmetric, and transitive is said to be an *equivalence relation*. Equivalence relations include equality among the reals, or congruence among triangles. A homogeneous relation that is reflexive, anti-symmetric and transitive is a *partial order relation*, such as the less than or equal relation between reals. Finally, a homogeneous relation that is anti-symmetric and transitive is an *order relation*, such as the less than relation between reals.

1.2. Functions

Functions between sets X and Y are a specific type of relationship that often arise in science. For a given input $x\in X$, theories that predict a single possible output $y\in Y$ are of particular scientific interest.

DEFINITION. (FUNCTION) . *A function from set X to set Y is a relation $F\subseteq X\times Y$, that associates to $x\in X$ a single $y\in Y$.*

The above intuitive definition can be transcribed in precise mathematical terms as $F\subseteq X\times Y$ is a *function* if $(x,y)\in F$ and $(x,z)\in F$ implies $y=z$. Since it's a particular kind of relation, a function is a triplet of sets (X,Y,F), but with a special, common notation to denote the triplet by $f:X\longrightarrow Y$, with $F=\{(x,f(x))\,|\,x\in X, f(x)\in Y\}$ and the property that $(x,y)\in F\Longrightarrow y=f(x)$. The set X is the *domain* and the set Y is the *codomain* of the function f. The value from the domain $x\in X$ is the *argument* of the function associated with the *function value* $y=f(x)$. The function value y is said to be *returned* by evaluation $y=f(x)$.

As seen previously, a Euclidean space $E_m=(\mathbb{R}^m,\mathbb{R},+,\cdot)$ can be used to suggest properties of more complex spaces such as the vector space of continuous functions $C^0(\mathbb{R})$. A construct that will be often used is to interpret a vector within E_m as a function, since $\boldsymbol{v}\in\mathbb{R}^m$ with components $\boldsymbol{v}=\begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix}^T$ also defines a function $v:\{1,2,...,m\}\longrightarrow\mathbb{R}$, with values $v(i)=v_i$. As the number of components grows the function v can provide better approximations of some continuous function $f\in C^0(\mathbb{R})$ through the function values $v_i=v(i)=f(x_i)$ at distinct sample points $x_1,x_2,\dots,x_m$.

The above function examples are all defined on a domain of scalars or naturals and returned scalar values. Within linear algebra the particular interest is on functions defined on sets of vectors from some vector space $\mathcal{V}=(V,S,+,\cdot)$ that return either scalars $f:V\longrightarrow S$, or vectors from some other vector space $\mathcal{W}=(W,S,+,\cdot)$, $\boldsymbol{g}:V\longrightarrow W$. The codomain of a vector-valued function might be the same set of vectors as its domain, $\boldsymbol{h}:V\longrightarrow V$. The fundamental operation within linear algebra is the linear combination $a\boldsymbol{u}+b\boldsymbol{v}$ with $a,b\in S$, $\boldsymbol{u},\boldsymbol{v}\in V$. A key aspect is to characterize how a function behaves when given a linear combination as its argument, for instance $f(a\boldsymbol{u}+b\boldsymbol{v})$ or $\boldsymbol{g}(a\boldsymbol{u}+b\boldsymbol{v})$.

1.3. Linear functionals

Consider first the case of a function defined on a set of vectors that returns a scalar value. These can be interpreted as labels attached to a vector, and are very often encountered in applications from natural phenomena or data analysis.

DEFINITION. (FUNCTIONAL) . *A functional on vector space $\mathcal{V}=(V,S,+,\cdot)$ is a function from the set of vectors V to the set of scalars S of the vector space $\mathcal{V}$.*

DEFINITION. (LINEAR FUNCTIONAL) . *The functional $f:V\longrightarrow S$ on vector space $\mathcal{V}=(V,S,+,\cdot)$ is a linear functional if for any two vectors $\boldsymbol{u},\boldsymbol{v}\in V$ and any two scalars a,b*

$$f(a\boldsymbol{u}+b\boldsymbol{v})=af(\boldsymbol{u})+bf(\boldsymbol{v}). \tag{1.14}$$

Many different functionals may be defined on a vector space $\mathcal{V}=(V,S,+,\cdot)$, and an insightful alternative description is provided by considering the set of all linear functionals, that will be denoted as $V^*=\{f\,|\,f:V\longrightarrow S\}$. These can be organized into another vector space $\mathcal{V}^*=(V^*,S,+,\cdot)$ with vector addition of linear functionals $f,g\in V^*$ and scaling by $a\in S$ defined by

$$(f+g)(\boldsymbol{u})=f(\boldsymbol{u})+g(\boldsymbol{u}),(af)(\boldsymbol{u})=af(\boldsymbol{u}),\boldsymbol{u}\in V. \tag{1.15}$$

DEFINITION. (DUAL VECTOR SPACE) . *For some vector space $\mathcal{V}$, the vector space of linear functionals $\mathcal{V}^*$ is called the dual vector space.*

As is often the case, the above abstract definition can better be understood by reference to the familiar case of Euclidean space. Consider $\mathcal{R}_2 = (\mathbb{R}^2, \mathbb{R}, +, \cdot)$, the set of vectors in the plane with $\boldsymbol{x} \in \mathbb{R}^2$ the position vector from the origin $(0,0)$ to point X in the plane with coordinates (x_1, x_2). One functional from the dual space $\mathcal{R}_2^*$ is $f_2(\boldsymbol{x}) = x_2$, i.e., taking the second coordinate of the position vector. The linearity property is readily verified. For $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{R}_2$, $a, b \in \mathbb{R}$,

$$f_2(a\boldsymbol{x} + b\boldsymbol{y}) = ax_2 + by_2 = af_2(\boldsymbol{x}) + bf_2(\boldsymbol{y}).$$

Given some constant value $h \in \mathbb{R}$, the curves within the plane defined by $f_2(\boldsymbol{x}) = h$ are called the *contour lines* or *level sets* of f_2. Several contour lines and position vectors are shown in Figure 1.3. The utility of functionals and dual spaces can be shown by considering a simple example from physics. Assume that x_2 is the height above ground level and a vector $\boldsymbol{x}$ is the displacement of a body of mass m in a gravitational field. The mechanical work done to lift the body from ground level to height h is $W = mgh$ with g the gravitational acceleration. The mechanical work is the same for all displacements $\boldsymbol{x}$ that satisfy the equation $f_2(\boldsymbol{x}) = h$. The work expressed in units $mg\Delta h$ can be interpreted as the number of contour lines $f_2(\boldsymbol{x}) = n\Delta h$ intersected by the displacement vector $\boldsymbol{x}$. This concept of duality between vectors and scalar-valued functionals arises throughout mathematics, the physical and social sciences and in data science. The term "duality" itself comes from geometry. A point X in $\mathbb{R}^2$ with coordinates (x_1, x_2) can be defined either as the end-point of the position vector $\boldsymbol{x}$, or as the intersection of the contour lines of two functionals $f_1(\boldsymbol{x}) = x_1$ and $f_2(\boldsymbol{x}) = x_2$. Either geometric description works equally well in specifying the position of X, so it might seem redundant to have two such procedures. It turns out though that many quantities of interest in applications can be defined through use of both descriptions, as shown in the computation of mechanical work in a gravitational field.

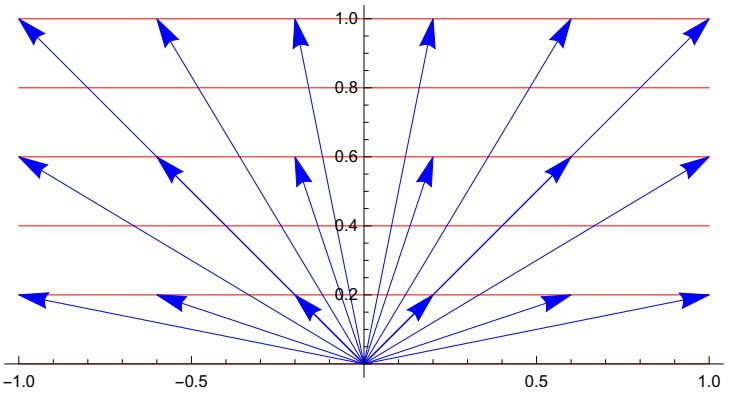


Figure 1.3. Vectors in E_2 and contour lines of the functional $f(\boldsymbol{x}) = x_2$

1.4. Linear mappings

Consider now functions $\boldsymbol{f}: V \longrightarrow W$ from vector space $\mathcal{V} = (V, S, +, \cdot)$ to another vector space $\mathcal{W} = (W, T, +, \cdot)$. As before, the action of such functions on linear combinations is of special interest.

DEFINITION. (LINEAR MAPPING) . *A function $\boldsymbol{f}: V \longrightarrow W$, from vector space $\mathcal{V} = (V, S, +, \cdot)$ to vector space $\mathcal{W} = (W, S, \oplus, \odot)$ is called a linear mapping if for any two vectors $\boldsymbol{u}, \boldsymbol{v} \in V$ and any two scalars $a, b \in S$*

$$\boldsymbol{f}(a\boldsymbol{u} + b\boldsymbol{v}) = a\boldsymbol{f}(\boldsymbol{u}) + b\boldsymbol{f}(\boldsymbol{v}). \tag{1.16}$$

The image of a linear combination $a\boldsymbol{u} + b\boldsymbol{v}$ through a linear mapping is another linear combination $a\boldsymbol{f}(\boldsymbol{u}) + b\boldsymbol{f}(\boldsymbol{v})$, and linear mappings are said to preserve the structure of a vector space, and called *homomorphisms* in mathematics. The codomain of a linear mapping might be the same as the domain in which case the mapping is said to be an *endomorphism*.

Matrix-vector multiplication has been introduced as a concise way to specify a linear combination

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} = x_1\,\boldsymbol{a}_1 + \cdots + x_n\,\boldsymbol{a}_n,$$

with $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ the columns of the matrix, $\boldsymbol{A} = [\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \ldots\ \ \boldsymbol{a}_n\ ]$. This is a linear mapping between the real spaces $\mathscr{R}_m$, $\mathscr{R}_n$, $\boldsymbol{f}: \mathbb{R}^m \longrightarrow \mathbb{R}^n$, and indeed any linear mapping between real spaces can be given as a matrix-vector product.

2. Measurements

Vectors within the real space $\mathscr{R}_m$ can be completely specified by m real numbers, even though m is large in many realistic applications. A vector within $C^0(\mathbb{R})$, i.e., a continuous function defined on the reals, cannot be so specified since it would require an infinite, non-countable listing of function values. In either case, the task of describing the elements of a vector space $\mathscr{V} = (V, S, +, \cdot)$ by simpler means arises. Within data science this leads to *classification problems* in accordance with some relevant criteria.

2.1. Equivalence classes

Many classification criteria are scalars, defined as a scalar-valued function $f: \mathscr{V} \longrightarrow S$ on a vector space, $\mathscr{V} = (V, S, +, \cdot)$. The most common criteria are inspired by experience with Euclidean space. In a Euclidean-Cartesian model $(\mathbb{R}^2, \mathbb{R}, +, \cdot)$ of the geometry of a plane Π, a point $O \in \Pi$ is arbitrarily chosen to correspond to the zero vector $\boldsymbol{0} = [\ 0\ \ 0\ ]^T$, along with two preferred vectors $\boldsymbol{e}_1, \boldsymbol{e}_2$ grouped together into the identity matrix $\boldsymbol{I}$. The position of a point $X \in \Pi$ with respect to O is given by the linear combination

$$\boldsymbol{x} = \boldsymbol{I}\boldsymbol{x} + \boldsymbol{0} = [\ \boldsymbol{e}_1\ \ \boldsymbol{e}_2\ ] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1\,\boldsymbol{e}_1 + x_2\,\boldsymbol{e}_2\,.$$

Several possible classifications of points in the plane are depicted in Figure 1.4: lines, squares, circles. Intuitively, each choice separates the plane into subsets, and a given point in the plane belongs to just one in the chosen family of subsets. A more precise characterization is given by the concept of a partition of a set.

DEFINITION. (PARTITION) . *A partition of a set is a grouping of its elements into non-empty subsets such that every element is included in exactly one subset.*

In precise mathematical terms, a partition of set S is $P = \{S_i \,|\, S_i \subset P, S_i \neq \emptyset, i \in I\}$ such that $\forall x \in S, \exists!\, j \in I$ for which $x \in S_j$. Since there is only one set ($\exists!$ signifies "exists and is unique") to which some given $x \in S$ belongs, the subsets S_i of the partition P are disjoint, $i \neq j \Longrightarrow S_i \cap S_j = \emptyset$. The subsets S_i are labeled by i within some index set I. The index set might be a subset of the naturals, $I \subset \mathbb{N}$ in which case the partition is countable, possibly finite. The partitions of the plane suggested by Figure 1.4 are however indexed by a real-valued label, $i \in \mathbb{R}$ with $I \subset \mathbb{R}$.

A technique which is often used to generate a partition of a vector space $\mathscr{V} = (V, S, +, \cdot)$ is to define an equivalence relation between vectors, $H \subseteq V \times V$. For some element $\boldsymbol{u} \in V$, the *equivalence class* of $\boldsymbol{u}$ is defined as all vectors $\boldsymbol{v}$ that are equivalent to $\boldsymbol{u}$, $\{\boldsymbol{v} \,|\, (\boldsymbol{u}, \boldsymbol{v}) \in H\}$. The set of equivalence classes of is called the *quotient set* and denoted as V/H, and the quotient set is a partition of V. Figure 1.4 depicts four different partitions of the plane. These can be interpreted geometrically, such as parallel lines or distance from the origin. With wider implications for linear algebra, the partitions can also be given in terms of classification criteria specified by functions.

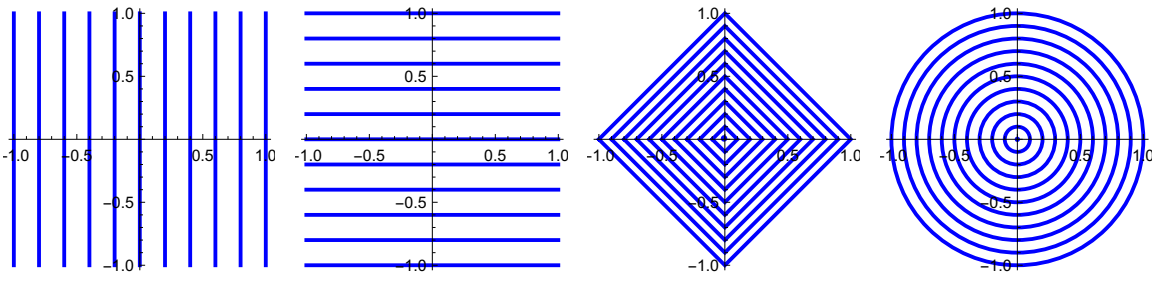


Figure 1.4. Equivalence classes within the plane

2.2. Norms

The partition of $\mathbb{R}^2$ by circles from Figure 1.4 is familiar; the equivalence classes are sets of points whose position vector has the same size, $\{\boldsymbol{x} = [\ x_1\ \ x_2\ ]^T \mid (\ x_1^2 + x_2^2)^{1/2} = r\}$, or is at the same distance from the origin. Note that familiarity with Euclidean geometry should not obscure the fact that some other concept of distance might be induced by the data. A simple example is statement of walking distance in terms of city blocks, in which the distance from a starting point to an address $x_1 = 3$ blocks east and $x_2 = 4$ blocks north is $x_1 + x_2 = 7$ city blocks, not the Euclidean distance $(\ x_1^2 + x_2^2)^{1/2} = 5$ since one cannot walk through the buildings occupying a city block.

The above observations lead to the mathematical concept of a *norm* as a tool to evaluate vector magnitude. Recall that a vector space is specified by two sets and two operations, $\mathcal{V} = (V, S, +, \cdot)$, and the behavior of a norm with respect to each of these components must be defined. The desired behavior includes the following properties and formal definition.

Unique value. The magnitude of a vector $\boldsymbol{v} \in V$ should be a unique scalar, requiring the definition of a function. The scalar could have irrational values and should allow ordering of vectors by size, so the function should be from V to $\mathbb{R}$, $f: V \longrightarrow \mathbb{R}$. On the real line the point at coordinate x is at distance $|x|$ from the origin, and to mimic this usage the norm of $\boldsymbol{v} \in V$ is denoted as $\|\boldsymbol{v}\|$, leading to the definition of a function $\|\ \|$: $V \longrightarrow \mathbb{R}_+$, $\mathbb{R}_+ = \{a \mid a \in \mathbb{R}, a \geqslant 0\}$.

Null vector case. Provision must be made for the only distinguished element of V, the null vector $\boldsymbol{0}$. It is natural to associate the null vector with the null scalar element, $\|\boldsymbol{0}\| = 0$. A crucial additional property is also imposed namely that the null vector is the *only* vector whose norm is zero, $\|\boldsymbol{v}\| = 0 \Longrightarrow \boldsymbol{v} = \boldsymbol{0}$. From knowledge of a single scalar value, an entire vector can be determined. This property arises at key junctures in linear algebra, notably in providing a link to another branch of mathematics known as analysis, and is needed to establish the fundamental theorem of linear algbera or the singular value decomposition encountered later.

Scaling. Transfer of the scaling operation $\boldsymbol{v} = a\boldsymbol{u}$ property leads to imposing $\|\boldsymbol{v}\| = |a|\,\|\boldsymbol{u}\|$. This property ensures commensurability of vectors, meaning that the magnitude of vector $\boldsymbol{v}$ can be expressed as a multiple of some standard vector magnitude $\|\boldsymbol{u}\|$.

Vector addition. Position vectors from the origin to coordinates $x, y > 0$ on the real line can be added and $|x + y| = |x| + |y|$. If however the position vectors point in different directions, $x > 0$, $y < 0$, then $|x + y| < |x| + |y|$. For a general vector space the analogous property is known as the *triangle inequality*, $\|\boldsymbol{u} + \boldsymbol{v}\| \leqslant \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ for $\boldsymbol{u}, \boldsymbol{v} \in V$.

DEFINITION. (NORM) . *A norm on the vector space $\mathcal{V} = (V, S, +, \cdot)$ is a function $\|\ \|: V \longrightarrow \mathbb{R}_+$ that for $\boldsymbol{u}, \boldsymbol{v} \in V$, $a \in S$ satisfies:*

1. $\|\boldsymbol{v}\| = 0 \Longrightarrow \boldsymbol{v} = \boldsymbol{0}$;
2. $\|\ a\boldsymbol{u}\| = |a|\,\|\boldsymbol{u}\|$;
3. $\|\boldsymbol{u} + \boldsymbol{v}\| \leqslant \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$.

Note that the norm is a functional, but the triangle inequality implies that it is not generally a linear functional. Returning to Figure 1.4, consider the functions $f_i: \mathbb{R}^2 \longrightarrow \mathbb{R}_+$ defined for $\boldsymbol{x} = [\ x_1\ \ x_2\ ]^T$ through values

$$f_1(\boldsymbol{x}) = |x_1|,\ f_2(\boldsymbol{x}) = |x_2|,\ f_3(\boldsymbol{x}) = |x_1| + |x_2|,\ f_4(\boldsymbol{x}) = (|x_1|^2 + |x_2|^2)^{1/2}.$$

Sets of constant value of the above functions are also equivalence classes induced by the equivalence relations E_i for $i = 1, 2, 3, 4$.

1. $f_1(\boldsymbol{x}) = c \Longrightarrow |x_1| = c$, $E_1 = \{(\boldsymbol{x}, \boldsymbol{y}) \mid f_1(\boldsymbol{x}) = f_1(\boldsymbol{y}) \Longleftrightarrow |x_1| = |y_1|\} \subseteq \mathbb{R}^2 \times \mathbb{R}^2$;
2. $f_2(\boldsymbol{x}) = c \Longrightarrow |x_2| = c$, $E_2 = \{(\boldsymbol{x}, \boldsymbol{y}) \mid f_2(\boldsymbol{x}) = f_2(\boldsymbol{y}) \Longleftrightarrow |x_2| = |y_2|\} \subseteq \mathbb{R}^2 \times \mathbb{R}^2$;
3. $f_3(\boldsymbol{x}) = c \Longrightarrow |x_1| + |x_2| = c$, $E_3 = \{(\boldsymbol{x}, \boldsymbol{y}) \mid f_3(\boldsymbol{x}) = f_3(\boldsymbol{y}) \Longleftrightarrow |x_1| + |x_2| = |y_1| + |y_2|\} \subseteq \mathbb{R}^2 \times \mathbb{R}^2$;
4. $f_4(\boldsymbol{x}) = c \Longrightarrow (|x_1|^2 + |x_2|^2)^{1/2} = c$, $E_4 = \{(\boldsymbol{x}, \boldsymbol{y}) \mid f_4(\boldsymbol{x}) = f_4(\boldsymbol{y}) \Longleftrightarrow (|x_1|^2 + |x_2|^2)^{1/2} = (|y_1|^2 + |y_2|^2)^{1/2}\} \subseteq \mathbb{R}^2 \times \mathbb{R}^2$.

These equivalence classes correspond to the vertical lines, horizontal lines, squares, and circles of Figure 1.4. Not all of the functions f_i are norms since $f_1(\boldsymbol{x})$ is zero for the non-null vector $\boldsymbol{x} = [\ 0\ \ 1\ ]^T$, and $f_2(\boldsymbol{x})$is zero for the non-null vector $\boldsymbol{x} = [\ 1\ \ 0\ ]^T$. The functions f_3 and f_4 are indeed norms, and specific cases of the following general norm.

DEFINITION. (p-NORM IN $\mathscr{R}_m$) . *The p-norm on the real vector space* $\mathscr{R}_m = (\mathbb{R}^m, \mathbb{R}, +, \cdot)$ *for* $p \geqslant 1$ *is the function* $\| \ \|_p$: $V \longrightarrow \mathbb{R}_+$ *with values* $\|\boldsymbol{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_m|^p)^{1/p}$, *or*

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{m} |x_i|^p \right)^{1/p} \text{ for } \boldsymbol{x} \in \mathbb{R}^m. \tag{1.17}$$

Denote by x_i the largest component in absolute value of $\boldsymbol{x} \in \mathbb{R}^m$. As p increases, $|x_i|^p$ becomes dominant with respect to all other terms in the sum suggesting the definition of an inf-norm by

$$\|\boldsymbol{x}\|_\infty = \max_{1 \leqslant i \leqslant m} |x_i| .$$

This also works for vectors with equal components, since the fact that the number of components is finite while $p \longrightarrow \infty$ can be used as exemplified for $\boldsymbol{x} = [\ a\ \ a\ \ \ldots\ \ a\ ]^T$, by $\|\boldsymbol{x}\|_p = (m|a|^p)^{1/p} = m^{1/p}|a|$, with $m^{1/p} \longrightarrow 1$.

Note that the Euclidean norm corresponds to $p = 2$, and is often called the 2-norm. The analogy between vectors and functions can be exploited to also define a p-norm for $C^0[a,b] = (C([a,b]), \mathbb{R}, +, \cdot)$, the vector space of continuous functions defined on $[a,b]$.

DEFINITION. (p-NORM IN $C^0[a,b]$) . *The p-norm on the vector space of continuous functions* $C^0[a,b]$ *for* $p \geqslant 1$ *is the function* $\| \ \|_p$: $V \longrightarrow \mathbb{R}_+$ *with values*

$$\|f\|_p = \left(\int_a^b |f(x)|^p \, \mathrm{d}x \right)^{1/p}, \text{for } f \in C[a,b]. \tag{1.18}$$

The integration operation $\int_a^b$ can be intuitively interpreted as the value of the sum $\sum_{i=1}^m$ from equation (1.17) for very large m and very closely spaced evaluation points of the function $f(x_i)$, for instance $|x_{i+1} - x_i| = (b-a)/m$. An inf-norm can also be define for continuous functions by

$$\|f\|_\infty = \sup_{x \in [a,b]} |f(x)|,$$

where sup, the supremum operation can be intuitively understood as the generalization of the max operation over the countable set $\{1, 2, \ldots, m\}$ to the uncountable set $[a,b]$.

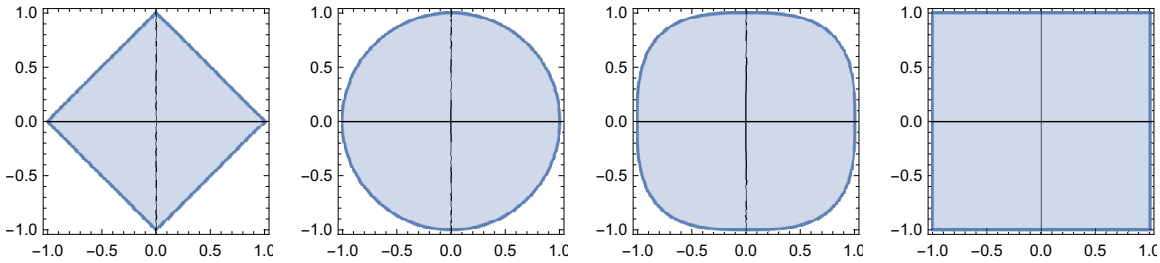


Figure 1.5. Regions within $\mathbb{R}^2$ for which $\|\boldsymbol{x}\|_p \leqslant 1$, for $p = 1, 2, 3, \infty$.

- Vector norms arise very often in applications since they can be used to classify data, and are implemented in most software systems as a `norm(x,p)` to evaluate the p-norm of a vector $\boldsymbol{x}$, with $p = 2$ as the default.

2.3. Inner product

Norms are functionals that define what is meant by the size of a vector, but are not linear. Even in the simplest case of the real line, the linearity relation $|x + y| = |x| + |y|$ is not verified for $x > 0$, $y < 0$. Nor do norms characterize the familiar geometric concept of orientation of a vector. A particularly important orientation from Euclidean geometry is orthogonality between two vectors. Another function is required, but before a formal definition some intuitive understanding is sought by considering vectors and functionals in the plane, as depicted in Figure 1.6. Consider a position vector $\boldsymbol{x} = [\ x_1\ \ x_2\ ]^T \in \mathbb{R}^2$ and the previously-encountered linear functionals

$$f_1, f_2 : \mathbb{R}^2 \longrightarrow \mathbb{R}, \ f_1(\boldsymbol{x}) = x_1, \ f_2(\boldsymbol{x}) = x_2.$$

The x_1 component of the vector $\boldsymbol{x}$ can be thought of as the number of level sets of f_1 times it crosses; similarly for the x_2 component. A convenient labeling of level sets is by their normal vectors. The level sets of f_1 have normal $\boldsymbol{e}_1^T = [\ 1\ \ 0\ ]$, and those of f_2 have normal vector $\boldsymbol{e}_2^T = [\ 0\ \ 1\ ]$. Both of these can be thought of as matrices with two columns, each containing a single component. The products of these matrices with the vector $\boldsymbol{x}$ gives the value of the functionals f_1, f_2

$$\boldsymbol{e}_1^T \boldsymbol{x} = [\ 1\ \ 0\ ] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 1 \cdot x_1 + 0 \cdot x_2 = x_1 = f_1(\boldsymbol{x}),$$

$$\boldsymbol{e}_2^T \boldsymbol{x} = [\ 0\ \ 1\ ] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \cdot x_1 + 1 \cdot x_2 = x_1 = f_2(\boldsymbol{x}).$$

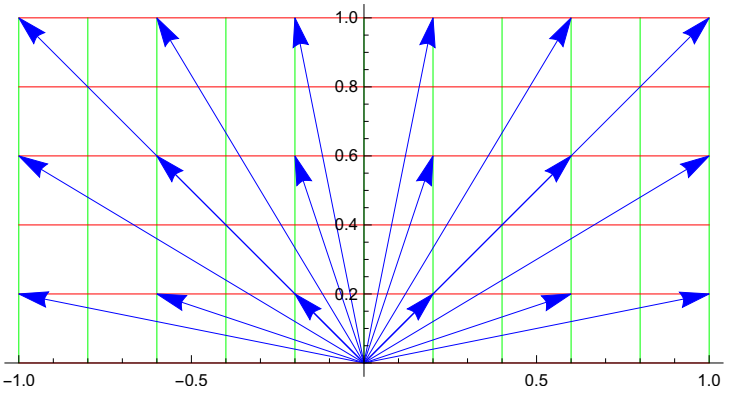


Figure 1.6. Euclidean space E_2 and its dual E_2^*.

In general, any linear functional f defined on the real space $\mathcal{R}_m$ can be labeled by a vector

$$\boldsymbol{a}^T = [\ a_1\ \ a_2\ \ \ldots\ \ a_m\ ],$$

and evaluated through the matrix-vector product $f(\boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{x}$. This suggests the definition of another function s: $\mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}$,

$$s(\boldsymbol{a}, \boldsymbol{x}) = \boldsymbol{a}^T \boldsymbol{x}.$$

The function s is called an *inner product*, has two vector arguments from which a matrix-vector product is formed and returns a scalar value, hence is also called a *scalar product*. The definition from an Euclidean space can be extended to general vector spaces. For now, consider the field of scalars to be the reals $S = \mathbb{R}$.

DEFINITION. (INNER PRODUCT) . *An inner product in the vector space $\mathcal{V} = (V, \mathbb{R}, +, \cdot)$ is a function s: $V \times V \longrightarrow \mathbb{R}$ with properties*

Symmetry. *For any $\boldsymbol{a}, \boldsymbol{x} \in V$, $s(\boldsymbol{a}, \boldsymbol{x}) = s(\boldsymbol{x}, \boldsymbol{a})$.*

Linearity in second argument. *For any $\boldsymbol{a}, \boldsymbol{x}, \boldsymbol{y} \in V$, $\alpha, \beta \in \mathbb{R}$, $s(\boldsymbol{a}, \alpha\boldsymbol{x} + \beta\boldsymbol{y}) = \alpha s(\boldsymbol{a}, \boldsymbol{x}) + \beta s(\boldsymbol{a}, \boldsymbol{y})$.*

Positive definiteness. *For any $\boldsymbol{x} \in V \backslash \{\boldsymbol{0}\}$, $s(\boldsymbol{x}, \boldsymbol{x}) > 0$.*

The inner product $s(\boldsymbol{a}, \boldsymbol{x})$ returns the number of level sets of the functional labeled by $\boldsymbol{a}$ crossed by the vector $\boldsymbol{x}$, and this interpretation underlies many applications in the sciences as in the gravitational field example above. Inner products also provide a procedure to evaluate geometrical quantities and relationships.

Vector norm. In $\mathcal{R}_m$ the number of level sets of the functional labeled by $\boldsymbol{x}$ crossed by $\boldsymbol{x}$ itself is identical to the square of the 2-norm

$$s(\boldsymbol{x}, \boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{x} = \|\boldsymbol{x}\|_2^2.$$

In general, the square root of $s(\boldsymbol{x}, \boldsymbol{x})$ satisfies the properties of a norm, and is called the norm induced by an inner product

$$\|\boldsymbol{x}\| = s(\boldsymbol{x}, \boldsymbol{x})^{1/2}.$$

A real space together with the scalar product $s(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$ and induced norm $\|\boldsymbol{x}\| = s(\boldsymbol{x}, \boldsymbol{x})^{1/2}$ defines an Euclidean vector space $\mathcal{E}_m$.

Orientation. In $\mathcal{E}_2$ the point specified by polar coordinates (r,θ) has the Cartesian coordinates $x_1 = r\cos\theta$, $x_2 = r\sin\theta$, and position vector $\boldsymbol{x} = [\ x_1\ \ x_2\ ]^T$. The inner product

$$\boldsymbol{e}_1^T\boldsymbol{x} = [\ 1\ \ 0\ ]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 1\cdot x_1 + 0\cdot x_2 = r\cos\theta,$$

is seen to contain information on the relative orientation of $\boldsymbol{x}$ with respect to $\boldsymbol{e}_1$. In general, the angle θ between two vectors $\boldsymbol{x}, \boldsymbol{y}$ with any vector space with a scalar product can be defined by

$$\cos\theta = \frac{s(\boldsymbol{x},\boldsymbol{y})}{[s(\boldsymbol{x},\boldsymbol{x})\,s(\boldsymbol{y},\boldsymbol{y})]^{1/2}} = \frac{s(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|},$$

which becomes

$$\cos\theta = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|},$$

in a Euclidean space, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$.

Orthogonality. In $\mathcal{E}_2$ two vectors are orthogonal if the angle between them is such that $\cos\theta = 0$, and this can be extended to an arbitrary vector space $\mathcal{U} = (V, \mathbb{R}, +, \cdot)$ with a scalar product by stating that $\boldsymbol{x}, \boldsymbol{y} \in V$ are orthogonal if $s(\boldsymbol{x},\boldsymbol{y}) = 0$. In $\mathcal{E}_m$ vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ are orthogonal if $\boldsymbol{x}^T\boldsymbol{y} = 0$.

3. Linear mapping composition

3.1. Matrix-matrix product

From two functions $f: A \longrightarrow B$ and $g: B \longrightarrow C$, a composite function, $h = g \circ f$, $h: A \longrightarrow C$ is defined by

$$h(x) = g(f(x)).$$

Consider linear mappings between Euclidean spaces $\boldsymbol{f}: \mathbb{R}^n \longrightarrow \mathbb{R}^m$, $\boldsymbol{g}: \mathbb{R}^m \longrightarrow \mathbb{R}^p$. Recall that linear mappings between Euclidean spaces are expressed as matrix vector multiplication

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{y}) = \boldsymbol{B}\boldsymbol{y}, \boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{B} \in \mathbb{R}^{p\times m}.$$

The composite function $\boldsymbol{h} = \boldsymbol{g} \circ \boldsymbol{f}$ is $\boldsymbol{h}: \mathbb{R}^n \longrightarrow \mathbb{R}^p$, defined by

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x})) = \boldsymbol{g}(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}.$$

Note that the intemediate vector $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x}$ is subsequently multiplied by the matrix $\boldsymbol{B}$. The composite function $\boldsymbol{h}$ is itself a linear mapping

$$\boldsymbol{h}(a\boldsymbol{x} + b\boldsymbol{y}) = \boldsymbol{B}\boldsymbol{A}(a\boldsymbol{x} + b\boldsymbol{y}) = \boldsymbol{B}\,(a\boldsymbol{A}\boldsymbol{x} + b\boldsymbol{A}\boldsymbol{y}) = \boldsymbol{B}(a\boldsymbol{u} + b\boldsymbol{v}) = a\boldsymbol{B}\boldsymbol{u} + b\boldsymbol{B}\boldsymbol{v} = a\boldsymbol{B}\boldsymbol{A}\boldsymbol{x} + b\boldsymbol{B}\boldsymbol{A}\boldsymbol{y} = a\boldsymbol{h}(\boldsymbol{x}) + b\boldsymbol{h}(\boldsymbol{y}),$$

so it also can be expressed a matrix-vector multiplication

$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}. \tag{1.23}$$

Using the above, $\boldsymbol{C}$ is defined as the product of matrix $\boldsymbol{B}$ with matrix $\boldsymbol{A}$

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{A}.$$

The columns of $\boldsymbol{C}$ can be determined from those of $\boldsymbol{A}$ by considering the action of $\boldsymbol{h}$ on the the column vectors of the identity matrix $\boldsymbol{I} = [\ \boldsymbol{e}_1\ \ \boldsymbol{e}_2\ \ \dots\ \ \boldsymbol{e}_n\ ] \in \mathbb{R}^{n\times n}$. First, note that

$$\boldsymbol{A}\boldsymbol{e}_j = [\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_n\ ]\begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} = \boldsymbol{a}_1, \dots, \boldsymbol{A}\boldsymbol{e}_j = [\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_n\ ]\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \boldsymbol{a}_j, \boldsymbol{A}\boldsymbol{e}_n = [\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_n\ ]\begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{a}_n. \tag{1.24}$$

The above can be repeated for the matrix $\boldsymbol{C} = [\ \boldsymbol{c}_1\ \ \boldsymbol{c}_2\ \ \dots\ \ \boldsymbol{c}_n\ ]$ giving

$$\boldsymbol{h}(\boldsymbol{e}_1) = \boldsymbol{C}\boldsymbol{e}_1 = \boldsymbol{c}_1, \dots, \boldsymbol{h}(\boldsymbol{e}_j) = \boldsymbol{C}\boldsymbol{e}_j = \boldsymbol{c}_j, \dots, \boldsymbol{h}(\boldsymbol{e}_n) = \boldsymbol{C}\boldsymbol{e}_n = \boldsymbol{c}_n. \tag{1.25}$$

Combining the above equations leads to $\boldsymbol{c}_j = \boldsymbol{B}\boldsymbol{a}_j$, or

$$\boldsymbol{C} = [\ \boldsymbol{c}_1\ \ \boldsymbol{c}_2\ \ \dots\ \ \boldsymbol{c}_n\ ] = \boldsymbol{B}\,[\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_n\ ].$$

◦ From the above the matrix-matrix product $\boldsymbol{C} = \boldsymbol{BA}$ is seen to simply be a grouping of all the products of $\boldsymbol{B}$ with the column vectors of $\boldsymbol{A}$,

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \dots & \boldsymbol{c}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}\,\boldsymbol{a}_1 & \boldsymbol{B}\boldsymbol{a}_2 & \dots & \boldsymbol{B}\boldsymbol{a}_n \end{bmatrix}.$$

Summary.

- Linear functionals $f : \mathcal{U} \longrightarrow S$ attach a scalar label to a vector, and preserve linear combinations

$$f(a\boldsymbol{u} + b\boldsymbol{v}) = af(\boldsymbol{u}) + bf(\boldsymbol{v})$$

- Linear functionals arise when establish vector magnitude and orientation
- Linear mappings $\boldsymbol{g} : \mathcal{U} \longrightarrow \mathcal{V}$ establish correspondences between vector spaces and preserve linear combinations

$$\boldsymbol{g}(a\boldsymbol{u} + b\boldsymbol{v}) = a\boldsymbol{g}(\boldsymbol{u}) + b\boldsymbol{g}(\boldsymbol{v})$$

- Composition of linear mappings is represented through matrix multiplication

LECTURE 6: FUNDAMENTAL MATRIX SPACES

1. Vector Subspaces

A central interest in scientific computation is to seek simple descriptions of complex objects. A typical situation is specifying an instance of some object of interest through an m-tuple $\boldsymbol{v} \in \mathbb{R}^m$ with large m. Assuming that addition and scaling of such objects can cogently be defined, a vector space is obtained, say over the field of reals with an Euclidean distance, E_m. Examples include for instance recordings of medical data (electroencephalograms, electrocardiograms), sound recordings, or images, for which m can easily reach into the millions. A natural question to ask is whether all the m real numbers are actually needed to describe the observed objects, or perhaps there is some intrinsic description that requires a much smaller number of descriptive parameters, that still preserves the useful idea of linear combination. The mathematical transcription of this idea is a vector subspace.

DEFINITION. (VECTOR SUBSPACE) . *$\mathcal{U} = (U, S, +, \cdot)$, $U \neq \emptyset$, is a vector subspace of vector space $\mathcal{V} = (V, S, +, \cdot)$ over the same field of scalars S, denoted by $\mathcal{U} \leq \mathcal{V}$, if $U \subseteq V$ and $\forall a, b \in S$, $\forall \boldsymbol{u}, \boldsymbol{v} \in U$, the linear combination $a\boldsymbol{u} + b\boldsymbol{v} \in U$.*

The above states a vector subspace must be closed under linear combination, and have the same vector addition and scaling operations as the enclosing vector space. The simplest vector subspace of a vector space is the null subspace that only contains the null element, $U = \{\boldsymbol{0}\}$. In fact any subspace must contain the null element $\boldsymbol{0}$, or otherwise closure would not be verified for the particular linear combination $\boldsymbol{u} + (-\boldsymbol{u}) = \boldsymbol{0}$. If $U \subset V$, then $\mathcal{U}$ is said to be a *proper subspace* of $\mathcal{V}$, denoted by $\mathcal{U} < \mathcal{V}$.

◦ Setting $n - m$ components equal to zero in the real space $\mathcal{R}_m$ defines a proper subspace whose elements can be placed into a one-to-one correspondence with the vectors within $\mathcal{R}_n$. For example, setting component m of $\boldsymbol{x} \in \mathbb{R}^m$ equal to zero gives $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_{m-1} & 0 \end{bmatrix}^T$ that while not a member of $\mathbb{R}^{m-1}$, it is in a one-to-one relation with $\boldsymbol{x}' = \begin{bmatrix} x_1 & x_2 & \dots & x_{m-1} \end{bmatrix}^T \in \mathbb{R}^{m-1}$. Dropping the last component of $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_{m-1} & y_m \end{bmatrix}^T$ gives vector $\boldsymbol{y}' = \begin{bmatrix} y_1 & y_2 & \dots & y_{m-1} \end{bmatrix} \in \mathbb{R}^{m-1}$, but this is no longer a one-to-one correspondence since for some given $\boldsymbol{y}'$, the last component y_m could take any value.

Vector subspaces arise in decomposition or partitioning of a vector space. The converse, composition of vector spaces $\mathcal{U} = (U, S, +, \cdot), \mathcal{V} = (V, S, +, \cdot)$ is defined in terms of linear combination. A vector $\boldsymbol{x} \in \mathbb{R}^3$ can be obtained as the linear combination

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x_2 \\ x_3 \end{bmatrix},$$

but also as

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 - a \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a \\ x_3 \end{bmatrix},$$

for some arbitrary $a \in \mathbb{R}$. In the first case, $\boldsymbol{x}$ is obtained as a unique linear combination of a vector from the set $U = \{\begin{bmatrix} x_1 & 0 & 0 \end{bmatrix}^T | \, x_1 \in \mathbb{R}\}$ with a vector from $V = \{\begin{bmatrix} 0 & x_2 & x_3 \end{bmatrix}^T | \, x_2, x_3 \in \mathbb{R}\}$. In the second case, there is an infinity of linear combinations of a vector from V with another from $W = \{\begin{bmatrix} x_1 & x_2 & 0 \end{bmatrix}^T | \, x_1, x_2 \in \mathbb{R}\}$ to the vector $\boldsymbol{x}$. This is captured by a pair of definitions to describe vector space composition.

DEFINITION. *Given two vector subspaces $\mathcal{U} = (U, S, +, \cdot), \mathcal{V} = (V, S, +, \cdot)$ of the space $\mathcal{W} = (W, S, +, \cdot)$, the sum is the vector space $\mathcal{U} + \mathcal{V} = (U + V, S, +, \cdot)$, where the sum of the two sets of vectors U, V is $U + V = \{\boldsymbol{u} + \boldsymbol{v} | \, \boldsymbol{u} \in U, \boldsymbol{v} \in V\}$.*

DEFINITION. *Given two vector subspaces $\mathcal{U} = (U, S, +, \cdot), \mathcal{V} = (V, S, +, \cdot)$ of the space $\mathcal{W} = (W, S, +, \cdot)$, the direct sum is the vector space $\mathcal{U} \oplus \mathcal{V} = (U \oplus V, S, +, \cdot)$, where the direct sum of the two sets of vectors U, V is $U \oplus V = \{\boldsymbol{u} + \boldsymbol{v} | \, \exists! \, \boldsymbol{u} \in U, \exists! \, \boldsymbol{v} \in V\}$. (unique decomposition)*

- Since the same scalar field, vector addition, and scaling is used , it is more convenient to refer to vector space sums simply by the sum of the vector sets $U + V$, or $U \oplus V$, instead of specifying the full tuplet for each space. This shall be adopted henceforth to simplify the notation.

In the previous example, the essential difference between the two ways to express $\boldsymbol{x} \in \mathbb{R}^3$ is that $U \cap V = \{\boldsymbol{0}\}$, but $V \cap W = \{\begin{bmatrix} 0 & a & 0 \end{bmatrix}^T | \, a \in \mathbb{R}\} \neq \{\boldsymbol{0}\}$, and in general if the zero vector is the only common element of two vector spaces then the sum of the vector spaces becomes a direct sum. In practice, the most important procedure to construct direct sums or check when an intersection of two vector subspaces reduces to the zero vector is through an inner product.

DEFINITION. *Two vector subspaces U, V of the real vector space $\mathbb{R}^m$ are orthogonal, denoted as $U \perp V$ if $\boldsymbol{u}^T \boldsymbol{v} = 0$ for any $\boldsymbol{u} \in U, \boldsymbol{v} \in V$.*

DEFINITION. *Two vector subspaces U, V of $U + V$ are orthogonal complements, denoted $U = V^\perp$, $V = U^\perp$ if they are orthogonal subspaces, $U \perp V$, and $U \cap V = \{\boldsymbol{0}\}$, i.e., the null vector is the only common element of both subspaces.*

The above concept of orthogonality can be extended to other vector subspaces, such as spaces of functions. It can also be extended to other choices of an inner product, in which case the term conjugate vector spaces is sometimes used. The concepts of sum and direct sum of vector spaces used linear combinations of the form $\boldsymbol{u} + \boldsymbol{v}$. This notion can be extended to arbitrary linear combinations.

DEFINITION. *In vector space $\mathcal{V} = (V, S, +, \cdot)$, the span of vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n \in V$, is the set of vectors reachable by linear combination*

$$\text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\} = \{\boldsymbol{b} \in V | \, \exists x_1, \ldots, x_n \in S \text{ such that } \boldsymbol{b} = x_1 \boldsymbol{a}_1 + \ldots + x_n \boldsymbol{a}_n\}.$$

Note that for real vector spaces a member of the span of the vectors $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\}$ is the vector $\boldsymbol{b}$ obtained from the matrix vector multiplication

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \ldots & \boldsymbol{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

From the above, the span is a subset of the co-domain of the linear mapping $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$.

2. Vector subspaces of a linear mapping

The wide-ranging utility of linear algebra results from a complete characterization of the behavior of a linear mapping between vector spaces $\boldsymbol{f}: U \longrightarrow V$, $\boldsymbol{f}(a\boldsymbol{u} + b\boldsymbol{v}) = a\boldsymbol{f}(\boldsymbol{u}) + b\boldsymbol{f}(\boldsymbol{v})$. For some given linear mapping the questions that arise are:

1. Can any vector within V be obtained by evaluation of $\boldsymbol{f}$?

2. Is there a single way that a vector within V can be obtained by evaluation of $\boldsymbol{f}$?

Linear mappings between real vector spaces $\boldsymbol{f}: \mathbb{R}^n \longrightarrow \mathbb{R}^m$, have been seen to be completely specified by a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. It is common to frame the above questions about the behavior of the linear mapping $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ through sets associated with the matrix $\boldsymbol{A}$. To frame an answer to the first question, a set of reachable vectors is first defined.

DEFINITION. *The column space (or range) of matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the set of vectors reachable by linear combination of the matrix column vectors*

$$C(\boldsymbol{A}) = \text{range}(\boldsymbol{A}) = \{\boldsymbol{b} \in \mathbb{R}^m | \exists \boldsymbol{x} \in \mathbb{R}^n \text{ such that } \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}\}.$$

By definition, the column space is included in the co-domain of the function $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, $C(\boldsymbol{A}) \subseteq \mathbb{R}^m$, and is readily seen to be a vector subspace of $\mathbb{R}^m$. The question that arises is whether the column space is the entire co-domain $C(\boldsymbol{A}) = \mathbb{R}^m$ that would signify that any vector can be reached by linear combination. If this is not the case then the column space would be a proper subset, $C(\boldsymbol{A}) \subset \mathbb{R}^m$, and the question is to determine what part of the co-domain cannot be reached by linear combination of columns of $\boldsymbol{A}$. Consider the orthogonal complement of $C(\boldsymbol{A})$ defined as the set vectors orthogonal to all of the column vectors of $\boldsymbol{A}$, expressed through inner products as

$$\boldsymbol{a}_1^T \boldsymbol{y} = 0, \boldsymbol{a}_2^T \boldsymbol{y} = 0, \ldots, \boldsymbol{a}_n^T \boldsymbol{y} = 0.$$

This can be expressed more concisely through the transpose operation

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \ldots & \boldsymbol{a}_n \end{bmatrix}, \boldsymbol{A}^T \boldsymbol{y} = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{y} \\ \boldsymbol{a}_2^T \boldsymbol{y} \\ \vdots \\ \boldsymbol{a}_n^T \boldsymbol{y} \end{bmatrix},$$

and leads to the definition of a set of vectors for which $\boldsymbol{A}^T \boldsymbol{y} = \boldsymbol{0}$

DEFINITION. *The left null space (or cokernel) of a matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *is the set*

$$N(\boldsymbol{A}^T) = \text{null}(\boldsymbol{A}^T) = \{\boldsymbol{y} \in \mathbb{R}^m | \boldsymbol{A}^T \boldsymbol{y} = \boldsymbol{0}\}.$$

Note that the left null space is also a vector subspace of the co-domain of $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, $N(\boldsymbol{A}^T) \subseteq \mathbb{R}^m$. The above definitions suggest that both the matrix and its transpose play a role in characterizing the behavior of the linear mapping $\boldsymbol{f} = \boldsymbol{A}\boldsymbol{x}$, so analagous sets are define for the transpose $\boldsymbol{A}^T$.

DEFINITION. *The row space (or corange) of a matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *is the set*

$$R(\boldsymbol{A}) = C(\boldsymbol{A}^T) = \text{range}(\boldsymbol{A}^T) = \{\boldsymbol{c} \in \mathbb{R}^n | \exists \boldsymbol{y} \in \mathbb{R}^m \, \boldsymbol{c} = \boldsymbol{A}^T \boldsymbol{y}\} \subseteq \mathbb{R}^n$$

DEFINITION. *The null space of a matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *is the set*

$$N(\boldsymbol{A}) = \text{null}(\boldsymbol{A}) = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\} \subseteq \mathbb{R}^n$$

Examples. Consider a linear mapping $\boldsymbol{f}: \mathbb{R}^n \longrightarrow \mathbb{R}^m$, defined by $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} = [\ y_1 \ \ \dots \ \ y_n\ ]^T$, with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

1. For $n = 1$, $m = 3$,

$$\boldsymbol{A} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{A}^T = [\ 1 \ \ 0 \ \ 0\ ],$$

the column space $C(\boldsymbol{A})$ is the y_1-axis, and the left null space $N(\boldsymbol{A}^T)$ is the y_2y_3-plane.

2. For $n = 2$, $m = 3$,

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = [\ \boldsymbol{a}_1 \ \ \boldsymbol{a}_2\ ], \boldsymbol{A}^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix},$$

the columns of $\boldsymbol{A}$ are colinear, $\boldsymbol{a}_2 = -\boldsymbol{a}_1$, and the column space $C(\boldsymbol{A})$ is the y_1-axis, and the left null space $N(\boldsymbol{A}^T)$ is the y_2y_3-plane, as before.

3. For $n = 2$, $m = 3$,

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{A}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

the column space $C(\boldsymbol{A})$ is the y_1y_2-plane, and the left null space $N(\boldsymbol{A}^T)$ is the y_3-axis.

4. For $n = 2$, $m = 3$,

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{A}^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \end{bmatrix},$$

the same $C(\boldsymbol{A})$, $N(\boldsymbol{A}^T)$ are obtained.

5. For $n=3$, $m=3$,

$$\boldsymbol{A}=\begin{bmatrix}1&1&3\\1&-1&-1\\1&1&3\end{bmatrix}=\begin{bmatrix}\boldsymbol{a}_1&\boldsymbol{a}_2&\boldsymbol{a}_3\end{bmatrix},$$

$$\boldsymbol{A}^T=\begin{bmatrix}1&1&1\\1&-1&1\\3&-1&3\end{bmatrix}=\begin{bmatrix}\boldsymbol{a}_1^T\\\boldsymbol{a}_2^T\\\boldsymbol{a}_3^T\end{bmatrix},\boldsymbol{A}^T\boldsymbol{y}=\begin{bmatrix}\boldsymbol{a}_1^T\boldsymbol{y}\\\boldsymbol{a}_2^T\boldsymbol{y}\\\boldsymbol{a}_3^T\boldsymbol{y}\end{bmatrix}$$

since $\boldsymbol{a}_3=\boldsymbol{a}_1+2\boldsymbol{a}_2$, the orthogonality condition $\boldsymbol{A}^T\boldsymbol{y}=\boldsymbol{0}$ is satisfied by vectors of form $\boldsymbol{y}=\begin{bmatrix}a&0&-a\end{bmatrix}$, $a\in\mathbb{R}$.

The above low dimensional examples are useful to gain initial insight into the significance of the spaces $C(\boldsymbol{A}), N(\boldsymbol{A}^T)$. Further appreciation can be gained by applying the same concepts to processing of images. A gray-scale image of size p_x by p_y pixels can be represented as a vector with $m=p_x p_y$ components, $\boldsymbol{b}\in[0,1]^m\subset\mathbb{R}^m$. Even for a small image with $p_x=p_y=128=2^7$ pixels along each direction, the vector $\boldsymbol{b}$ would have $m=2^{14}$ components. An image can be specified as a linear combination of the columns of the identity matrix

$$\boldsymbol{b}=\boldsymbol{I}\boldsymbol{b}=\begin{bmatrix}\boldsymbol{e}_1&\boldsymbol{e}_2&\dots&\boldsymbol{e}_m\end{bmatrix}\begin{bmatrix}b_1\\b_2\\\vdots\\b_m\end{bmatrix},$$

with b_i the gray-level intensity in pixel i. Similar to the inclined plane example from §1, an alternative description as a linear combination of another set of vectors $\boldsymbol{a}_1,\dots,\boldsymbol{a}_m$ might be more relevant. One choice of greater utility for image processing mimics the behavior of the set $\{1,\cos t,\cos 2t,\dots,\sin t,\sin 2t,\dots\}$ that extends the second example in §1, would be for $m=4$

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{a}_1&\boldsymbol{a}_2&\boldsymbol{a}_3&\boldsymbol{a}_4\end{bmatrix}=\begin{bmatrix}1&1&1&0\\1&1&0&1\\1&0&1&1\\1&0&0&0\end{bmatrix}.$$

3. Linear dependence

For the simple scalar mapping $f:\mathbb{R}\longrightarrow\mathbb{R}$, $f(x)=ax$, the condition $f(x)=0$ implies either that $a=0$ or $x=0$. Note that $a=0$ can be understood as defining a zero mapping $f(x)=0$. Linear mappings between vector spaces, $\boldsymbol{f}:U\longrightarrow V$, can exhibit different behavior, and the condtion $\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}$, might be satisfied for both $\boldsymbol{x}\neq\boldsymbol{0}$, and $\boldsymbol{A}\neq\boldsymbol{0}$. Analogous to the scalar case, $\boldsymbol{A}=\boldsymbol{0}$ can be understood as defining a zero mapping, $\boldsymbol{f}(\boldsymbol{x})=\boldsymbol{0}$.

In vector space $\mathcal{U}=(V,S,+,\cdot)$, vectors $\boldsymbol{u},\boldsymbol{v}\in V$ related by a scaling operation, $\boldsymbol{v}=a\boldsymbol{u}$, $a\in S$, are said to be colinear, and are considered to contain redundant data. This can be restated as $\boldsymbol{v}\in\text{span}\{\boldsymbol{u}\}$, from which it results that $\text{span}\{\boldsymbol{u}\}=\text{span}\{\boldsymbol{u},\boldsymbol{v}\}$. Colinearity can be expressed only in terms of vector scaling, but other types of redundancy arise when also considering vector addition as expressed by the span of a vector set. Assuming that $\boldsymbol{v}\notin\text{span}\{\boldsymbol{u}\}$, then the strict inclusion relation $\text{span}\{\boldsymbol{u}\}\subset\text{span}\{\boldsymbol{u},\boldsymbol{v}\}$ holds. This strict inclusion expressed in terms of set concepts can be transcribed into an algebraic condition.

DEFINITION. *The vectors $\boldsymbol{a}_1,\boldsymbol{a}_2,\dots,\boldsymbol{a}_n\in V$, are linearly dependent if there exist n scalars, $x_1,\dots,x_n\in S$, at least one of which is different from zero such that*

$$x_1\boldsymbol{a}_1+\dots+x_n\boldsymbol{a}_n=\boldsymbol{0}.$$

Introducing a matrix representation of the vectors

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_n \end{bmatrix}; \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

allows restating linear dependence as the existence of a non-zero vector, $\exists \boldsymbol{x} \neq \boldsymbol{0}$, such that $\boldsymbol{Ax} = \boldsymbol{0}$. Linear dependence can also be written as $\boldsymbol{Ax} = \boldsymbol{0} \nRightarrow \boldsymbol{x} = \boldsymbol{0}$, or that one cannot deduce from the fact that the linear mapping $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{Ax}$ attains a zero value that the argument itself is zero. The converse of this statement would be that the only way to ensure $\boldsymbol{Ax} = \boldsymbol{0}$ is for $\boldsymbol{x} = \boldsymbol{0}$, or $\boldsymbol{Ax} = \boldsymbol{0} \Longrightarrow \boldsymbol{x} = \boldsymbol{0}$, leading to the concept of linear independence.

DEFINITION. *The vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n \in V$, are linearly independent if the only n scalars, $x_1, \dots, x_n \in S$, that satisfy*

$$x_1\boldsymbol{a}_1 + \dots + x_n\boldsymbol{a}_n = \boldsymbol{0}, \tag{1.33}$$

are $x_1 = 0$, $x_2 = 0$,...,$x_n = 0$.

4. Basis and dimension

Vector spaces are closed under linear combination, and the span of a vector set $\mathcal{B} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots\}$ defines a vector subspace. If the entire set of vectors can be obtained by a spanning set, $V = \operatorname{span} \mathcal{B}$, extending $\mathcal{B}$ by an additional element $\mathbb{C} = \mathcal{B} \cup \{\boldsymbol{b}\}$ would be redundant since $\operatorname{span} \mathcal{B} = \operatorname{span} \mathbb{C}$. This is recognized by the concept of a basis, and also allows leads to a characterization of the size of a vector space by the cardinality of a basis set.

DEFINITION. *A set of vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_n \in V$ is a basis for vector space $\mathbb{U} = (V, S, +, \cdot)$ if*

1. *$\boldsymbol{u}_1, \dots, \boldsymbol{u}_n$ are linearly independent;*
2. *$\operatorname{span}\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_n\} = V$.*

DEFINITION. *The number of vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_n \in V$ within a basis is the dimension of the vector space $\mathbb{U} = (V, S, +, \cdot)$.*

5. Dimension of matrix spaces

The domain and co-domain of the linear mapping $\boldsymbol{f}: U \longrightarrow V$, $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{Ax}$, are decomposed by the spaces associated with the matrix $\boldsymbol{A}$. When $U = \mathbb{R}^n$, $V = \mathbb{R}^m$, the following vector subspaces associated with the matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ have been defined:

- $C(\boldsymbol{A})$ the column space of $\boldsymbol{A}$
- $C(\boldsymbol{A}^T)$ the row space of $\boldsymbol{A}$
- $N(\boldsymbol{A})$ the null space of $\boldsymbol{A}$
- $N(\boldsymbol{A}^T)$ the left null space of $\boldsymbol{A}$, or null space of $\boldsymbol{A}^T$

DEFINITION. *The rank of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the dimension of its column space and is equal to the dimension of its row space.*

DEFINITION. *The nullity of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the dimension of its null space.*

FUNDAMENTAL THEOREM OF LINEAR ALGEBRA

1. Partition of linear mapping domain and codomain

A partition of a set S has been introduced as a collection of subsets $P=\{S_i|\, S_i\subset P, S_i\neq\emptyset\}$ such that any given element $x\in S$ belongs to only one set in the partition. This is modified when applied to subspaces of a vector space, and a partition of a set of vectors is understood as a collection of subsets such that any vector except $\mathbf{0}$ belongs to only one member of the partition.

Linear mappings between vector spaces $\boldsymbol{f}: U\longrightarrow V$ can be represented by matrices $\boldsymbol{A}$ with columns that are images of the columns of a basis $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\dots\}$ of U

$$\boldsymbol{A}=[\ \boldsymbol{f}(\boldsymbol{u}_1)\ \ \boldsymbol{f}(\boldsymbol{u}_2)\ \ \dots\ ].$$

Consider the case of real finite-dimensional domain and co-domain, $\boldsymbol{f}:\mathbb{R}^n\longrightarrow\mathbb{R}^m$, in which case $\boldsymbol{A}\in\mathbb{R}^{m\times n}$,

$$\boldsymbol{A}=[\ \boldsymbol{f}(\boldsymbol{e}_1)\ \ \boldsymbol{f}(\boldsymbol{e}_2)\ \ \dots\ \ \boldsymbol{f}(\boldsymbol{e}_n)\ ]=[\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_n\ ].$$

◦ **Example 1.1.** Rotation by θ in $\mathbb{R}^2$ is obtained from

$$\boldsymbol{f}(\boldsymbol{e}_1)=\begin{bmatrix}\cos\theta\\ \sin\theta\end{bmatrix}, f(\boldsymbol{e}_2)=\begin{bmatrix}-\sin\theta\\ \cos\theta\end{bmatrix}$$

leading to

$$\boldsymbol{A}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}.$$

The column space of $\boldsymbol{A}$ is a vector subspace of the codomain, $C(\boldsymbol{A})\leq\mathbb{R}^m$, but according to the definition of dimension if $n<m$ there remain non-zero vectors within the codomain that are outside the range of $\boldsymbol{A}$,

$$n<m\Longrightarrow\exists\boldsymbol{v}\in\mathbb{R}^m, \boldsymbol{v}\neq\mathbf{0}, \boldsymbol{v}\notin C(\boldsymbol{A}).$$

All of the non-zero vectors in $N(\boldsymbol{A}^T)$, namely the set of vectors orthogonal to all columns in $\boldsymbol{A}$ fall into this category. The above considerations can be stated as

$$C(\boldsymbol{A})\leq\mathbb{R}^m,\ \ N(\boldsymbol{A}^T)\leq\mathbb{R}^m,\ \ C(\boldsymbol{A})\perp N(\boldsymbol{A}^T)\ \ C(\boldsymbol{A})+N(\boldsymbol{A}^T)\leq\mathbb{R}^m\,.$$

The question that arises is whether there remain any non-zero vectors in the codomain that are not part of $C(\boldsymbol{A})$ or $N(\boldsymbol{A}^T)$. The fundamental theorem of linear algebra states that there no such vectors, that $C(\boldsymbol{A})$ is the orthogonal complement of $N(\boldsymbol{A}^T)$, and their direct sum covers the entire codomain $C(\boldsymbol{A})\oplus N(\boldsymbol{A}^T)=\mathbb{R}^m$.

LEMMA 1.2. *Let $\mathcal{U},\mathcal{V}$, be subspaces of vector space $\mathcal{W}$. Then $\mathcal{W}=\mathcal{U}\oplus\mathcal{V}$ if and only if*

i. $\mathcal{W}=\mathcal{U}+\mathcal{V}$, and

ii. $\mathcal{U}\cap\mathcal{V}=\{\mathbf{0}\}$.

Proof. *$\mathcal{W}=\mathcal{U}\oplus\mathcal{V}\Longrightarrow\mathcal{W}=\mathcal{U}+\mathcal{V}$ by definition of direct sum, sum of vector subspaces. To prove that $\mathcal{W}=\mathcal{U}\oplus\mathcal{V}\Longrightarrow\mathcal{U}\cap\mathcal{V}=\{\mathbf{0}\}$, consider $\boldsymbol{w}\in\mathcal{U}\cap\mathcal{V}$. Since $\boldsymbol{w}\in\mathcal{U}$ and $\boldsymbol{w}\in\mathcal{V}$ write*

$$\boldsymbol{w}=\boldsymbol{w}+\mathbf{0}\ \ (\boldsymbol{w}\in\mathcal{U},\mathbf{0}\in\mathcal{V}),\ \ \boldsymbol{w}=\mathbf{0}+\boldsymbol{w}\ \ (\mathbf{0}\in\mathcal{U},\boldsymbol{w}\in\mathcal{V}),$$

and since expression $\boldsymbol{w}=\boldsymbol{u}+\boldsymbol{v}$ is unique, it results that $\boldsymbol{w}=\mathbf{0}$. Now assume (i),(ii) and establish an unique decomposition. Assume there might be two decompositions of $\boldsymbol{w}\in\mathcal{W}$, $\boldsymbol{w}=\boldsymbol{u}_1+\boldsymbol{v}_1$, $\boldsymbol{w}=\boldsymbol{u}_2+\boldsymbol{v}_2$, with $\boldsymbol{u}_1,\boldsymbol{u}_2\in\mathcal{U}$, $\boldsymbol{v}_1,\boldsymbol{v}_2\in\mathcal{V}$. Obtain $\boldsymbol{u}_1+\boldsymbol{v}_1=\boldsymbol{u}_2+\boldsymbol{v}_2$, or $\boldsymbol{x}=\boldsymbol{u}_1-\boldsymbol{u}_2=\boldsymbol{v}_2-\boldsymbol{v}_1$. Since $\boldsymbol{x}\in\mathcal{U}$ and $\boldsymbol{x}\in\mathcal{V}$ it results that $\boldsymbol{x}=\mathbf{0}$, and $\boldsymbol{u}_1=\boldsymbol{u}_2$, $\boldsymbol{v}_1=\boldsymbol{v}_2$, i.e., the decomposition is unique. □

In the vector space $U+V$ the subspaces U,V are said to be orthogonal complements if $U\perp V$, and $U\cap V=\{\mathbf{0}\}$. When $U\leq\mathbb{R}^m$, the orthogonal complement of U is denoted as $U^\perp$, $U\oplus U^\perp=\mathbb{R}^m$.

THEOREM. *Given the linear mapping associated with matrix $\boldsymbol{A}\in\mathbb{R}^{m\times n}$ we have:*

1. *$C(\boldsymbol{A})\oplus N(\boldsymbol{A}^T)=\mathbb{R}^m$, the direct sum of the column space and left null space is the codomain of the mapping*
2. *$C(\boldsymbol{A}^T)\oplus N(\boldsymbol{A})=\mathbb{R}^n$, the direct sum of the row space and null space is the domain of the mapping*

3. $C(\boldsymbol{A})\perp N(\boldsymbol{A}^T)$ *and* $C(\boldsymbol{A})\cap N(\boldsymbol{A}^T)=\{\boldsymbol{0}\}$*, the column space is orthogonal to the left null space, and they are orthogonal complements of one another,*

$$C(\boldsymbol{A})=N(\boldsymbol{A}^T)^\perp,\ \ N(\boldsymbol{A}^T)=C(\boldsymbol{A})^\perp\,.$$

4. $C(\boldsymbol{A}^T)\perp N(\boldsymbol{A})$ *and* $C(\boldsymbol{A}^T)\cap N(\boldsymbol{A})=\{\boldsymbol{0}\}$*, the row space is orthogonal to the null space, and they are orthogonal complements of one another,*

$$C(\boldsymbol{A}^T)=N(\boldsymbol{A})^\perp,\ \ N(\boldsymbol{A})=C(\boldsymbol{A}^T)^\perp\,.$$

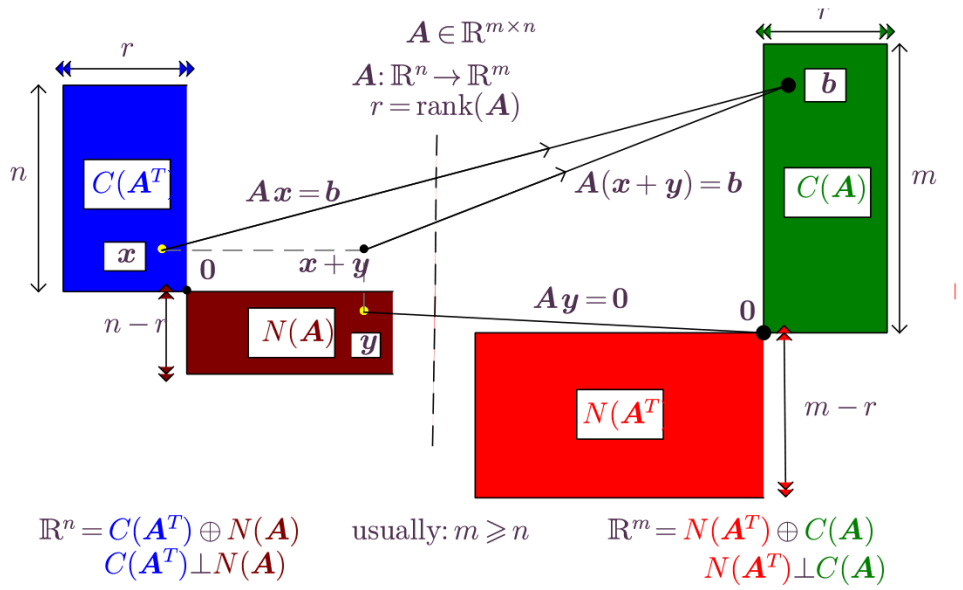


Figure 1.7. Graphical represenation of the Fundamental Theorem of Linear Algebra, Gil Strang, *Amer. Math. Monthly* **100**, 848-855, 1993.

Consideration of equality between sets arises in proving the above theorem. A standard technique to show set equality $A=B$, is by double inclusion, $A\subseteq B\wedge B\subseteq A\Longrightarrow A=B$. This is shown for the statements giving the decomposition of the codomain $\mathbb{R}^m$. A similar approach can be used to decomposition of $\mathbb{R}^n$.

i. $C(\boldsymbol{A})\perp N(\boldsymbol{A}^T)$ (column space is orthogonal to left null space).

Proof. Consider arbitrary $\boldsymbol{u}\in C(\boldsymbol{A})$, $\boldsymbol{v}\in N(\boldsymbol{A}^T)$. By definition of $C(\boldsymbol{A})$, $\exists\boldsymbol{x}\in\mathbb{R}^n$ such that $\boldsymbol{u}=\boldsymbol{A}\boldsymbol{x}$, and by definition of $N(\boldsymbol{A}^T)$, $\boldsymbol{A}^T\boldsymbol{v}=\boldsymbol{0}$. Compute $\boldsymbol{u}^T\boldsymbol{v}=(\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{v}=\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{v}=\boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{v})=\boldsymbol{x}^T\boldsymbol{0}=0$, hence $\boldsymbol{u}\perp\boldsymbol{v}$ for arbitrary $\boldsymbol{u}$, $\boldsymbol{v}$, and $C(\boldsymbol{A})\perp N(\boldsymbol{A}^T)$. □

ii. $C(\boldsymbol{A})\cap N(\boldsymbol{A}^T)=\{\boldsymbol{0}\}$ ($\boldsymbol{0}$ is the only vector both in $C(\boldsymbol{A})$ and $N(\boldsymbol{A}^T)$).

Proof. (By contradiction, *reductio ad absurdum*). Assume there might be $\boldsymbol{b}\in C(\boldsymbol{A})$ and $b\in N(\boldsymbol{A}^T)$ and $\boldsymbol{b}\neq\boldsymbol{0}$. Since $\boldsymbol{b}\in C(\boldsymbol{A})$, $\exists\boldsymbol{x}\in\mathbb{R}^n$ such that $\boldsymbol{b}=\boldsymbol{A}\boldsymbol{x}$. Since $\boldsymbol{b}\in N(\boldsymbol{A}^T)$, $\boldsymbol{A}^T\boldsymbol{b}=\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x})=\boldsymbol{0}$. Note that $\boldsymbol{x}\neq 0$ since $\boldsymbol{x}=0\Longrightarrow\boldsymbol{b}=0$, contradicting assumptions. Multiply equality $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}$ on left by $\boldsymbol{x}^T$,

$$\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}\Longrightarrow(\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x})=\boldsymbol{b}^T\boldsymbol{b}=\|\boldsymbol{b}\|^2=0,$$

thereby obtaining $\boldsymbol{b}=0$, using norm property 3. Contradiction.

□

iii. $C(\boldsymbol{A}) \oplus N(\boldsymbol{A}^T) = \mathbb{R}^m$

Proof. (iii) and (iv) have established that $C(\boldsymbol{A}), N(\boldsymbol{A}^T)$ are orthogonal complements

$$C(\boldsymbol{A}) = N(\boldsymbol{A}^T)^\perp, N(\boldsymbol{A}^T) = C(\boldsymbol{A})^\perp.$$

By Lemma 2 it results that $C(\boldsymbol{A}) \oplus N(\boldsymbol{A}^T) = \mathbb{R}^m$. □

The remainder of the FTLA is established by considering $\boldsymbol{B} = \boldsymbol{A}^T$, e.g., since it has been established in (v) that $C(\boldsymbol{B}) \oplus N(\boldsymbol{A}^T) = \mathbb{R}^n$, replacing $\boldsymbol{B} = \boldsymbol{A}^T$ yields $C(\boldsymbol{A}^T) \oplus N(\boldsymbol{A}) = \mathbb{R}^m$, etc.

Summary.

- Vector subspaces are subsets of a vector space closed under linear combination
- The simplest vector subspace is $\{\mathbf{0}\}$
- Linear mappings are represented by matrices
- Associated with matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ that represents mapping $f: \mathbb{R}^n \longrightarrow \mathbb{R}^m$ are four fundamental subspaces:
    1. $C(\boldsymbol{A}) \leq \mathbb{R}^m$ the column space of $\boldsymbol{A}$ containing vectors $\boldsymbol{b}$ reachable by $\boldsymbol{Ax}$, $\boldsymbol{b} = \boldsymbol{Ax}$
    2. $N(\boldsymbol{A}^T) \leq \mathbb{R}^m$ the left null space of $\boldsymbol{A}$ containing vectors $\boldsymbol{y}$ orthogonal to columns $\boldsymbol{A}$, $\boldsymbol{A}^T\boldsymbol{y} = 0$
    3. $C(\boldsymbol{A}^T) \leq \mathbb{R}^n$ the row space of $\boldsymbol{A}$
    4. $N(\boldsymbol{A}) \leq \mathbb{R}^n$ the null space of $\boldsymbol{A}$

LECTURE 7: THE SINGULAR VALUE DECOMPOSITION

1. Mappings as data

1.1. Vector spaces of mappings and matrix representations

A vector space $\mathcal{L}$ can be formed from all linear mappings from the vector space $\mathcal{U} = (U, S, +, \cdot)$ to another vector space $\mathcal{V} = (V, S, +, \cdot)$

$$\mathcal{L} = \{L, S, +, \cdot\}, \quad L = \{\boldsymbol{f} \mid \boldsymbol{f}: U \longrightarrow V, \boldsymbol{f}(a\boldsymbol{u} + b\boldsymbol{v}) = af(\boldsymbol{u}) + bf(\boldsymbol{v})\},$$

with addition and scaling of linear mappings defined by $(\boldsymbol{f}+\boldsymbol{g})(\boldsymbol{u})=\boldsymbol{f}(\boldsymbol{u})+\boldsymbol{g}(\boldsymbol{u})$ and $(a\boldsymbol{f})(\boldsymbol{u})=a\boldsymbol{f}(\boldsymbol{u})$. Let $B=\{\boldsymbol{u}_1,\boldsymbol{u}_2,\dots\}$ denote a basis for the domain U of linear mappings within $\mathscr{L}$, such that the linear mapping $\boldsymbol{f}\in\mathscr{L}$ is represented by the matrix

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{f}(\boldsymbol{u}_1) & \boldsymbol{f}(\boldsymbol{u}_2) & \dots\end{bmatrix}.$$

When the domain and codomain are the real vector spaces $U=\mathbb{R}^n$, $V=\mathbb{R}^m$, the above is a standard matrix of real numbers, $\boldsymbol{A}\in\mathbb{R}^{m\times n}$. For linear mappings between infinite dimensional vector spaces, the matrix is understood in a generalized sense to contain an infinite number of columns that are elements of the codomain V. For example, the indefinite integral is a linear mapping between the vector space of functions that allow differentiation to any order,

$$\int:C^\infty\longrightarrow C^\infty \quad v(t)=\int u(t)\,\mathrm{d}t$$

and for the monomial basis $B=\{1,t,t^2,\dots\}$, is represented by the generalized matrix

$$\boldsymbol{A}=\begin{bmatrix}t & \frac{1}{2}t^2 & \frac{1}{3}t^3 & \dots\end{bmatrix}.$$

Truncation of the MacLaurin series $u(t)=\sum_{j=1}^{\infty}u_j t^j$, with $u_j=u^{(j)}(0)/j!\in\mathbb{R}$ to n terms, and sampling of $u\in C^\infty$ at points $t_1,\dots,t_m$, forms a standard matrix of real numbers

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{t} & \frac{1}{2}\boldsymbol{t}^2 & \frac{1}{3}\boldsymbol{t}^3 & \dots\end{bmatrix}\in\mathbb{R}^{m\times n},\quad \boldsymbol{t}^j=\begin{bmatrix}t_1^j\\ \vdots\\ t_m^j\end{bmatrix}.$$

Values of function $u\in C^\infty$ at $t_1,\dots,t_m$ are approximated by

$$\boldsymbol{u}=\boldsymbol{B}\boldsymbol{x}=\begin{bmatrix}u(t_1) & \dots & u(t_m)\end{bmatrix}^T,$$

with $\boldsymbol{x}$ denoting the coordinates of $\boldsymbol{u}$ in basis $\boldsymbol{B}$. The above argument states that the coordinates $\boldsymbol{y}$ of $\boldsymbol{v}$, the primitive of $\boldsymbol{u}$ are given by

$$\boldsymbol{y}=\boldsymbol{A}\boldsymbol{x},$$

as can be indeed verified through term-by-term integration of the MacLaurin series.

As to be expected, matrices can also be organized as vector space $\mathscr{M}$, which is essentially the representation of the associated vector space of linear mappings,

$$\mathscr{M}=(M,S,+,\cdot)\quad M=\{\boldsymbol{A}\,|\,\boldsymbol{A}=\begin{bmatrix}\boldsymbol{f}(\boldsymbol{u}_1) & \boldsymbol{f}(\boldsymbol{u}_2) & \dots\end{bmatrix}\}\,.$$

The addition $\boldsymbol{C}=\boldsymbol{A}+\boldsymbol{B}$ and scaling $\boldsymbol{S}=a\boldsymbol{R}$ of matrices is given in terms of the matrix components by

$$c_{ij}=a_{ij}+b_{ij},\, s_{ij}=ar_{\mathrm{ij}}\,.$$

1.2. Measurement of mappings

From the above it is apparent that linear mappings and matrices can also be considered as data, and a first step in analysis of such data is definition of functionals that would attach a single scalar label to each linear mapping of matrix. Of particular interest is the definition of a norm functional that characterizes in an appropriate sense the size of a linear mapping.

Consider first the case of finite matrices with real components $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ that represent linear mappings between real vector spaces $\boldsymbol{f}: \mathbb{R}^m \longrightarrow \mathbb{R}^n$. The columns $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ could be placed into a single column vector $\boldsymbol{c}$ with mn components

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_n \end{bmatrix}.$$

Subsequently the norm of the matrix $\boldsymbol{A}$ could be defined as the norm of the vector $\boldsymbol{c}$. An example of this approach is the Frobenius norm

$$\|\boldsymbol{A}\|_F = \|\boldsymbol{c}\|_2 = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}.$$

A drawback of the above approach is that the structure of the matrix and its close relationship to a linear mapping is lost. A more useful characterization of the size of a mapping is to consider the amplification behavior of linear mapping. The motivation is readily understood starting from linear mappings between the reals $f: \mathbb{R} \longrightarrow \mathbb{R}$, that are of the form $f(x) = ax$. When given an argument of unit magnitude $|x| = 1$, the mapping returns a real number with magnitude $|a|$. For mappings $\boldsymbol{f}: \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ within the plane, arguments that satisfy $\|\boldsymbol{x}\|_2 = 1$ are on the unit circle with components $\boldsymbol{x} = [\, \cos\theta \;\; \sin\theta \,]$ have images through $\boldsymbol{f}$ given analytically by

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{bmatrix} \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \cos\theta \boldsymbol{a}_1 + \sin\theta \boldsymbol{a}_2,$$

and correspond to ellipses.

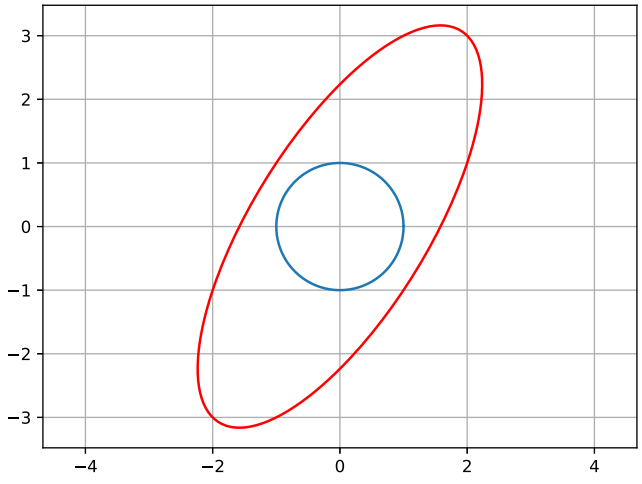


Figure 1.8. Mapping of unit circle by $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, $\boldsymbol{A} = \begin{bmatrix} 2 & -1 \\ 3 & 1 \end{bmatrix}$.

From the above the mapping associated $\boldsymbol{A}$ amplifies some directions more than others. This suggests a definition of the size of a matrix or a mapping by the maximal amplification unit norm vectors within the domain.

DEFINITION. *For vector spaces U, V with norms $\|\,\|_U : U \longrightarrow \mathbb{R}_+$, $\|\,\|_V : V \longrightarrow \mathbb{R}_+$, the induced norm of $\boldsymbol{f} : U \longrightarrow V$ is*

$$\|\boldsymbol{f}\| = \sup_{\|\boldsymbol{x}\|_U = 1} \|\boldsymbol{f}(\boldsymbol{x})\|_V .$$

DEFINITION. *For vector spaces $\mathbb{R}^n, \mathbb{R}^m$ with norms $\|\,\|^{(n)} : U \longrightarrow \mathbb{R}_+$, $\|\,\|^{(m)} : V \longrightarrow \mathbb{R}_+$, the induced norm of matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is*

$$\|\boldsymbol{A}\| = \sup_{\|\boldsymbol{x}\|^{(n)} = 1} \|\boldsymbol{A}\boldsymbol{x}\|^{(m)} .$$

In the above, any vector norm can be used within the domain and codomain.

2. The Singular Value Decomposition (SVD)

The fundamental theorem of linear algebra partitions the domain and codomain of a linear mapping $\boldsymbol{f} : U \longrightarrow V$. For real vectors spaces $U = \mathbb{R}^n$, $V = \mathbb{R}^m$ the partition properties are stated in terms of spaces of the associated matrix $\boldsymbol{A}$ as

$$C(\boldsymbol{A}) \oplus N(\boldsymbol{A}^T) = \mathbb{R}^m \quad C(\boldsymbol{A}) \perp N(\boldsymbol{A}^T) \quad C(\boldsymbol{A}^T) \oplus N(\boldsymbol{A}) = \mathbb{R}^n \quad C(\boldsymbol{A}^T) \perp N(\boldsymbol{A}) \ .$$

The dimension of the column and row spaces $r = \dim C(\boldsymbol{A}) = \dim C(\boldsymbol{A}^T)$ is the rank of the matrix, $n - r$ is the nullity of $\boldsymbol{A}$, and $m - r$ is the nullity of A^T. A infinite number of bases could be defined for the domain and codomain. It is of great theoretical and practical interest to define bases with properties that facilitate insight or computation.

2.1. Orthogonal matrices

The above partitions of the domain and codomain are orthogonal, and suggest searching for orthogonal bases within these subspaces. Introduce a matrix representation for the bases

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m \end{bmatrix} \in \mathbb{R}^{m \times m}, \boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_n \end{bmatrix} \in \mathbb{R}^{n \times n},$$

with $C(\boldsymbol{U}) = \mathbb{R}^m$ and $C(\boldsymbol{V}) = \mathbb{R}^n$. Orthogonality between columns $\boldsymbol{u}_i$, $\boldsymbol{u}_j$ for $i \neq j$ is expressed as $\boldsymbol{u}_i^T \boldsymbol{u}_j = 0$. For $i = j$, the inner product is positive $\boldsymbol{u}_i^T \boldsymbol{u}_i > 0$, and since scaling of the columns of $\boldsymbol{U}$ preserves the spanning property $C(\boldsymbol{U}) = \mathbb{R}^m$, it is convenient to impose $\boldsymbol{u}_i^T \boldsymbol{u}_i = 1$. Such behavior is concisely expressed as a matrix product

$$\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_m,$$

with $\boldsymbol{I}_m$ the identity matrix in $\mathbb{R}^m$. Expanded in terms of the column vectors of $\boldsymbol{U}$ the first equality is

$$\begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m \end{bmatrix}^T \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1^T \\ \boldsymbol{u}_2^T \\ \vdots \\ \boldsymbol{u}_m^T \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1^T \boldsymbol{u}_1 & \boldsymbol{u}_1^T \boldsymbol{u}_2 & \dots & \boldsymbol{u}_1^T \boldsymbol{u}_m \\ \boldsymbol{u}_2^T \boldsymbol{u}_1 & \boldsymbol{u}_2^T \boldsymbol{u}_2 & \dots & \boldsymbol{u}_2^T \boldsymbol{u}_m \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{u}_m^T \boldsymbol{u}_1 & \boldsymbol{u}_m^T \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m^T \boldsymbol{u}_m \end{bmatrix} = \boldsymbol{I}_m .$$

It is useful to determine if a matrix $\boldsymbol{X}$ exists such that $\boldsymbol{UX} = \boldsymbol{I}_m$, or

$$\boldsymbol{UX} = \boldsymbol{U}\left[\begin{array}{cccc}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_m\end{array}\right] = \left[\begin{array}{cccc}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_m\end{array}\right].$$

The columns of $\boldsymbol{X}$ are the coordinates of the column vectors of $\boldsymbol{I}_m$ in the basis $\boldsymbol{U}$, and can readily be determined

$$\boldsymbol{U}\boldsymbol{x}_j = \boldsymbol{e}_j \Longrightarrow \boldsymbol{U}^T\boldsymbol{U}\boldsymbol{x}_j = \boldsymbol{U}^T\boldsymbol{e}_j \Longrightarrow \boldsymbol{I}_m\boldsymbol{x}_j = \begin{bmatrix}\boldsymbol{u}_1^T \\ \boldsymbol{u}_2^T \\ \vdots \\ \boldsymbol{u}_m^T\end{bmatrix}\boldsymbol{e}_j \Longrightarrow \boldsymbol{x}_j = (\boldsymbol{U}^T)_j,$$

where $(\boldsymbol{U}^T)_j$ is the j^{th} column of $\boldsymbol{U}^T$, hence $\boldsymbol{X} = \boldsymbol{U}^T$, leading to

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} = \boldsymbol{U}\boldsymbol{U}^T.$$

Note that the second equality

$$\left[\begin{array}{cccc}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m\end{array}\right]\left[\begin{array}{cccc}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m\end{array}\right]^T = \left[\begin{array}{cccc}\boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m\end{array}\right]\begin{bmatrix}\boldsymbol{u}_1^T \\ \boldsymbol{u}_2^T \\ \vdots \\ \boldsymbol{u}_m^T\end{bmatrix} = \boldsymbol{u}_1\boldsymbol{u}_1^T + \boldsymbol{u}_2\boldsymbol{u}_2^T + \cdots + \boldsymbol{u}_m\boldsymbol{u}_m^T = \boldsymbol{I}$$

acts as normalization condition on the matrices $\boldsymbol{U}_j = \boldsymbol{u}_j\boldsymbol{u}_j^T$.

DEFINITION. *A square matrix $\boldsymbol{U}$ is said to be orthogonal if $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$.*

2.2. Intrinsic basis of a linear mapping

Given a linear mapping $\boldsymbol{f}: U \longrightarrow V$, expressed as $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, the simplest description of the action of $\boldsymbol{A}$ would be a simple scaling, as exemplified by $\boldsymbol{g}(\boldsymbol{x}) = a\boldsymbol{x}$ that has as its associated matrix $a\boldsymbol{I}$. Recall that specification of a vector is typically done in terms of the identity matrix $\boldsymbol{b} = \boldsymbol{I}\boldsymbol{b}$, but may be more insightfully given in some other basis $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{I}\boldsymbol{b}$. This suggests that especially useful bases for the domain and codomain would reduce the action of a linear mapping to scaling along orthogonal directions, and evaluate $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ by first re-expressing $\boldsymbol{y}$ in another basis $\boldsymbol{U}$, $\boldsymbol{U}\boldsymbol{s} = \boldsymbol{I}\boldsymbol{y}$ and re-expressing $\boldsymbol{x}$ in another basis $\boldsymbol{V}$, $\boldsymbol{V}\boldsymbol{r} = \boldsymbol{I}\boldsymbol{x}$. The condition that the linear operator reduces to simple scaling in these new bases is expressed as $s_i = \sigma_i r_i$ for $i = 1, \dots, \min(m, n)$, with σ_i the scaling coefficients along each direction which can be expressed as a matrix vector product $\boldsymbol{s} = \boldsymbol{\Sigma}\boldsymbol{r}$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is of the same dimensions as $\boldsymbol{A}$ and given by

$$\boldsymbol{\Sigma} = \begin{bmatrix}\sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0\end{bmatrix}.$$

Imposing the condition that $\boldsymbol{U}, \boldsymbol{V}$ are orthogonal leads to

$$\boldsymbol{U}\boldsymbol{s} = \boldsymbol{y} \Longrightarrow \boldsymbol{s} = \boldsymbol{U}^T\boldsymbol{y}, \boldsymbol{V}\boldsymbol{r} = \boldsymbol{x} \Longrightarrow \boldsymbol{r} = \boldsymbol{V}^T\boldsymbol{x},$$

which can be replaced into $\boldsymbol{s} = \boldsymbol{\Sigma}\boldsymbol{r}$ to obtain

$$\boldsymbol{U}^T\boldsymbol{y} = \boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} \Longrightarrow \boldsymbol{y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}.$$

From the above the orthogonal bases $\boldsymbol{U}, \boldsymbol{V}$ and scaling coefficients $\boldsymbol{\Sigma}$ that are sought must satisfy $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$.

THEOREM. *Every matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ has a singular value decomposition (SVD)*

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T,$$

with properties:

1. $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ *is an orthogonal matrix,* $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_m$;
2. $\boldsymbol{V} \in \mathbb{R}^{m\times m}$ *is an orthogonal matrix,* $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_n$;
3. $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ *is diagonal,* $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_p)$, $p = \min(m, n)$, *and* $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_p \geqslant 0$.

Proof. *The proof of the SVD makes use of properties of the norm, concepts from analysis and complete induction. Adopting the 2-norm set* $\sigma_1 = \|A\|_2$,

$$\sigma_1 = \sup_{\|\boldsymbol{x}\|_2 = 1} \|\boldsymbol{A}\boldsymbol{x}\|_2.$$

The domain $\|\boldsymbol{x}\|_2 = 1$ *is compact (closed and bounded), and the extreme value theorem implies that* $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ *attains its maxima and minima, hence there must exist some vectors* $\boldsymbol{u}_1, \boldsymbol{v}_1$ *of unit norm such that* $\sigma_1\boldsymbol{u}_1 = \boldsymbol{A}\boldsymbol{v}_1 \Longrightarrow \sigma_1 = \boldsymbol{u}_1^T\boldsymbol{A}\boldsymbol{v}_1$. *Introduce orthogonal bases* $\boldsymbol{U}_1$, $\boldsymbol{V}_1$ *for* $\mathbb{R}^m, \mathbb{R}^n$ *whose first column vectors are* $\boldsymbol{u}_1, \boldsymbol{v}_1$, *and compute*

$$\boldsymbol{U}_1{}^T\boldsymbol{A}\boldsymbol{V}_1 = \begin{bmatrix} \boldsymbol{u}_1^T \\ \vdots \\ \boldsymbol{u}_m^T \end{bmatrix} \begin{bmatrix} \boldsymbol{A}\boldsymbol{v}_1 & \ldots & \boldsymbol{A}\boldsymbol{v}_n \end{bmatrix} = \begin{bmatrix} \sigma_1 & \boldsymbol{w}^T \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix} = \boldsymbol{C}.$$

In the above $\boldsymbol{w}^T$ *is a row vector with* $n-1$ *components* $\boldsymbol{u}_1^T\boldsymbol{A}\boldsymbol{v}_j$, $j = 2, \ldots, n$, *and* $\boldsymbol{u}_i^T\boldsymbol{A}\boldsymbol{v}_1$ *must be zero for* $\boldsymbol{u}_1$ *to be the direction along which the maximum norm* $\|\boldsymbol{A}\boldsymbol{v}_1\|$ *is obtained. Introduce vectors*

$$\boldsymbol{y} = \begin{bmatrix} \sigma_1 \\ \boldsymbol{w} \end{bmatrix}, \boldsymbol{z} = \boldsymbol{C}\boldsymbol{y} = \begin{bmatrix} \sigma_1^2 + \boldsymbol{w}^T\boldsymbol{w} \\ \boldsymbol{B}\boldsymbol{w} \end{bmatrix},$$

and $\|\boldsymbol{C}\boldsymbol{y}\|_2 = \|\boldsymbol{z}\|_2 \geqslant \sigma_1^2 + \boldsymbol{w}^T\boldsymbol{w} + \|\boldsymbol{B}\boldsymbol{w}\|_1 \geqslant \sigma_1^2 + \boldsymbol{w}^T\boldsymbol{w} = \|\boldsymbol{y}\|_2^2 = \sqrt{\sigma_1^2 + \boldsymbol{w}^T\boldsymbol{w}}\ \|\boldsymbol{y}\|_2$. *From* $\|\boldsymbol{U}_1{}^T\boldsymbol{A}\boldsymbol{V}_1\| = \|\boldsymbol{A}\| = \sigma_1 = \|\boldsymbol{C}\| \geqslant \sigma_1^2 + \boldsymbol{w}^T\boldsymbol{w}$ *it results that* $\boldsymbol{w} = \boldsymbol{0}$. *By induction, assume that* $\boldsymbol{B}$ *has a singular value decomposition,* $\boldsymbol{B} = \boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^T$, *such that*

$$\boldsymbol{U}_1{}^T\boldsymbol{A}\boldsymbol{V}_1 = \begin{bmatrix} \sigma_1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^T \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{\Sigma}_2 \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{V}_2^T \end{bmatrix},$$

and the orthogonal matrices arising in the singular value decomposition of $\boldsymbol{A}$ *are*

$$\boldsymbol{U} = \boldsymbol{U}_1 \begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{U}_2 \end{bmatrix}, \boldsymbol{V}^T = \begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{V}_2^T \end{bmatrix} \boldsymbol{V}_1^T.$$

□

The scaling coefficients σ_j are called the *singular values* of $\boldsymbol{A}$. The columns of $\boldsymbol{U}$ are called the *left singular vectors*, and those of $\boldsymbol{V}$ are called the *right singular vectors*.

The fact that the scaling coefficients are norms of $\boldsymbol{A}$ and submatrices of $\boldsymbol{A}$, $\sigma_1 = \|\boldsymbol{A}\|$, is crucial importance in applications. Carrying out computation of the matrix products

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_r & \boldsymbol{u}_{r+1} & \dots & \boldsymbol{u}_m \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_r^T \\ \boldsymbol{v}_{r+1}^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_r & \boldsymbol{u}_{r+1} & \dots & \boldsymbol{u}_m \end{bmatrix} \begin{bmatrix} \sigma_1 \boldsymbol{v}_1^T \\ \sigma_2 \boldsymbol{v}_2^T \\ \vdots \\ \sigma_r \boldsymbol{v}_r^T \\ \vdots \\ 0 \end{bmatrix}$$

leads to a representation of $\boldsymbol{A}$ as a sum

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T, r \leqslant \min(m, n).$$

$$\boldsymbol{A} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T$$

Each product $\boldsymbol{u}_i \boldsymbol{v}_i^T$ is a matrix of rank one, and is called a rank-one update. Truncation of the above sum to p terms leads to an approximation of $\boldsymbol{A}$

$$\boldsymbol{A} \cong \boldsymbol{A}_p = \sum_{i=1}^{p} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T.$$

In very many cases the singular values exhibit rapid, exponential decay, $\sigma_1 \gg \sigma_2 \gg \cdots$, such that the approximation above is an accurate representation of the matrix $\boldsymbol{A}$.

Figure 1.9. Successive SVD approximations of Andy Warhol's painting, *Marilyn Diptych* (~1960), with $k = 10, 20, 40$ rank-one updates.

2.3. SVD solution of linear algebra problems

The SVD can be used to solve common problems within linear algebra.

Change of coordinates. To change from vector coordinates $\boldsymbol{b}$ in the canonical basis $\boldsymbol{I} \in \mathbb{R}^{m\times m}$ to coordinates $\boldsymbol{x}$ in some other basis $\boldsymbol{A} \in \mathbb{R}^{m\times m}$, a solution to the equation $\boldsymbol{Ib} = \boldsymbol{Ax}$ can be found by the following steps.

1. Compute the SVD, $\boldsymbol{U\Sigma V}^T = \boldsymbol{A}$;
2. Find the coordinates of $\boldsymbol{b}$ in the orthogonal basis $\boldsymbol{U}$, $\boldsymbol{c} = \boldsymbol{U}^T\boldsymbol{b}$;
3. Scale the coordinates of $\boldsymbol{c}$ by the inverse of the singular values $y_i = c_i/\sigma_i$, $i = 1,\ldots,m$, such that $\Sigma\, \boldsymbol{y} = \boldsymbol{c}$ is satisfied;
4. Find the coordinates of $\boldsymbol{y}$ in basis $\boldsymbol{V}^T$, $\boldsymbol{x} = \boldsymbol{Vy}$.

Best 2-norm approximation. In the above $\boldsymbol{A}$ was assumed to be a basis, hence $r = \text{rank}(\boldsymbol{A}) = m$. If columns of $\boldsymbol{A}$ do not form a basis, $r < m$, then $\boldsymbol{b} \in \mathbb{R}^m$ might not be reachable by linear combinations within $C(\boldsymbol{A})$. The closest vector to $\boldsymbol{b}$ in the norm is however found by the same steps, with the simple modification that in Step 3, the scaling is carried out only for non-zero singular values, $y_i = c_i/\sigma_i$, $i = 1,\ldots,r$.

The pseudo-inverse. From the above, finding either the solution of $\boldsymbol{Ax} = \boldsymbol{Ib}$ or the best approximation possible if $\boldsymbol{A}$ is not of full rank, can be written as a sequence of matrix multiplications using the SVD

$$(\boldsymbol{U\Sigma V}^T)\,\boldsymbol{x} = \boldsymbol{b} \Longrightarrow \boldsymbol{U}\,(\boldsymbol{\Sigma V}^T\boldsymbol{x}) = \boldsymbol{b} \Longrightarrow (\boldsymbol{\Sigma V}^T\boldsymbol{x}) = \boldsymbol{U}^T\boldsymbol{b} \Longrightarrow \boldsymbol{V}^T\boldsymbol{x} = \boldsymbol{\Sigma}^+\boldsymbol{U}^T\boldsymbol{b} \Longrightarrow \boldsymbol{x} = \boldsymbol{V\Sigma}^+\boldsymbol{U}^T\boldsymbol{b},$$

where the matrix $\boldsymbol{\Sigma}^+ \in \mathbb{R}^{n\times m}$ (notice the inversion of dimensions) is defined as a matrix with elements σ_i^{-1} on the diagonal, and is called the pseudo-inverse of $\boldsymbol{\Sigma}$. Similarly the matrix

$$\boldsymbol{A}^+ = \boldsymbol{V\Sigma}^+\boldsymbol{U}^T$$

that allows stating the solution of $\boldsymbol{Ax} = \boldsymbol{b}$ simply as $\boldsymbol{x} = \boldsymbol{A}^+\boldsymbol{b}$ is called the *pseudo-inverse* of $\boldsymbol{A}$. Note that in practice $\boldsymbol{A}^+$ is not explicitly formed. Rather the notation $\boldsymbol{A}^+$ is simply a concise reference to carrying out steps 1-4 above.

LECTURE 8: LEAST SQUARES PROBLEMS

A typical scenario in many sciences is acquisition of m numbers to describe some object that is understood to actually require only $n \ll m$ parameters. For example, m voltage measurements u_i of an alternating current could readily be reduced to three parameters, the amplitude, phase and frequency $u(t) = a\sin(\omega t + \varphi)$. Very often a simple first-degree polynomial approximation $y = ax + b$ is sought for a large data set $D = \{(x_i, y_i), i = 1,\ldots,m\}$. All of these are instances of data compression, a problem that can be solved in a linear algebra framework.

1. Projection

Consider a partition of a vector space U into orthogonal subspaces $U = V \oplus W$, $V = W^\perp$, $W = V^\perp$. Within the typical scenario described above $U = \mathbb{R}^m$, $V \subset \mathbb{R}^m$, $W \subset \mathbb{R}^m$, $\dim V = n$, $\dim W = m - n$. If $\boldsymbol{V} = [\ \boldsymbol{v}_1\ \ldots\ \boldsymbol{v}_n\ ] \in \mathbb{R}^{m\times n}$ is a basis for V and $\boldsymbol{W} = [\ \boldsymbol{w}_1\ \ldots\ \boldsymbol{w}_{m-n}\ ] \in \mathbb{R}^{m\times(m-n)}$ is a basis for W, then $\boldsymbol{U} = [\ \boldsymbol{v}_1\ \ldots\ \boldsymbol{v}_n\ \boldsymbol{w}_1\ \ldots\ \boldsymbol{w}_{m-n}\ ]$ is a basis for U. Even though the matrices $\boldsymbol{V}, \boldsymbol{W}$ are not necessarily square, they are said to be orthonormal, in the sense that all columns are of unit norm and orthogonal to one another. Computation of the matrix product $\boldsymbol{V}^T\boldsymbol{V}$ leads to the formation of the identity matrix within $\mathbb{R}^n$

$$\boldsymbol{V}^T\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix} [\ \boldsymbol{v}_1\ \ \boldsymbol{v}_2\ \ \ldots\ \ \boldsymbol{v}_n\ ] = \begin{bmatrix} \boldsymbol{v}_1^T\boldsymbol{v}_1 & \boldsymbol{v}_1^T\boldsymbol{v}_2 & \ldots & \boldsymbol{v}_1^T\boldsymbol{v}_n \\ \boldsymbol{v}_2^T\boldsymbol{v}_1 & \boldsymbol{v}_2^T\boldsymbol{v}_2 & \ldots & \boldsymbol{v}_2^T\boldsymbol{v}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{v}_n^T\boldsymbol{v}_1 & \boldsymbol{v}_n^T\boldsymbol{v}_2 & \ldots & \boldsymbol{v}_n^T\boldsymbol{v}_n \end{bmatrix} = \boldsymbol{I}_n.$$

Similarly, $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_{m-n}$. Whereas for the square orthogonal matrix $\boldsymbol{U}$ multiplication both on the left and the right by its transpose leads to the formation of the identity matrix

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}_m,$$

the same operations applied to rectangular orthonormal matrices lead to different results

$$\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_n, \boldsymbol{V}\boldsymbol{V}^T = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \vdots \\ \boldsymbol{v}_n^T \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{v}_i \boldsymbol{v}_i^T, \text{rank}(\boldsymbol{v}_i \boldsymbol{v}_i^T) = 1$$

A simple example is provided by taking $\boldsymbol{V} = \boldsymbol{I}_{m,n}$, the first n columns of the identity matrix in which case

$$\boldsymbol{V}\boldsymbol{V}^T = \sum_{i=1}^{n} \boldsymbol{e}_i \boldsymbol{e}_i^T = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{m \times m}.$$

- Applying $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^T$ to some vector $\boldsymbol{b} \in \mathbb{R}^m$ leads to a vector $\boldsymbol{r} = \boldsymbol{P}\boldsymbol{b}$ whose first n components are those of $\boldsymbol{b}$, and the remaining $m-n$ are zero. The subtraction $\boldsymbol{b} - \boldsymbol{r}$ leads to a new vector $\boldsymbol{s} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{b}$ that has the first components equal to zero, and the remaining $m-n$ the same as those of $\boldsymbol{b}$. Such operations are referred to as *projections*, and for $\boldsymbol{V} = \boldsymbol{I}_{m,n}$ correspond to projection onto the $\text{span}\{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n\}$.

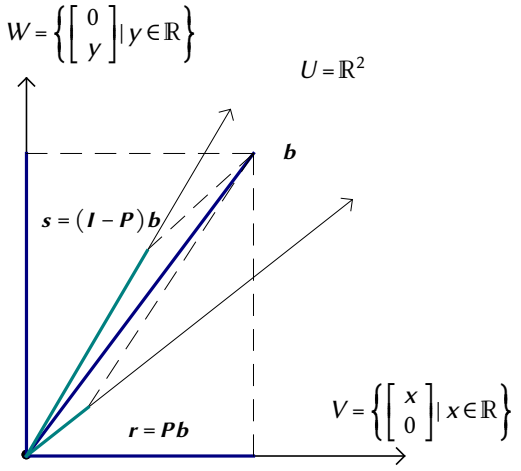


Figure 1.10. Projection in $\mathbb{R}^2$. The vectors $\boldsymbol{r}, \boldsymbol{s} \in \mathbb{R}^2$ have two components, but could be expressed through scaling of $\boldsymbol{e}_1, \boldsymbol{e}_2$.

Returning to the general case, the orthogonal matrices $\boldsymbol{U} \in \mathbb{R}^{m \times m}$, $\boldsymbol{V} \in \mathbb{R}^{m \times n}$, $\boldsymbol{W} \in \mathbb{R}^{m \times (m-n)}$ are associated with linear mappings $\boldsymbol{b} = \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{U}\boldsymbol{x}$, $\boldsymbol{r} = \boldsymbol{g}(\boldsymbol{b}) = \boldsymbol{P}\boldsymbol{b}$, $\boldsymbol{s} = \boldsymbol{h}(\boldsymbol{b}) = (\boldsymbol{I} - \boldsymbol{P})\,\boldsymbol{b}$. The mapping $\boldsymbol{f}$ gives the components in the $\boldsymbol{I}$ basis of a vector whose components in the $\boldsymbol{U}$ basis are $\boldsymbol{x}$. The mappings $\boldsymbol{g}, \boldsymbol{h}$ project a vector onto $\text{span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_n\}$, $\text{span}\{\boldsymbol{w}_1, \dots, \boldsymbol{w}_{m-n}\}$, respectively. When $\boldsymbol{V}, \boldsymbol{W}$ are orthogonal matrices the projections are also orthogonal $\boldsymbol{r} \perp \boldsymbol{s}$. Projection can also be carried out onto nonorthogonal spanning sets, but the process is fraught with possible error, especially when the angle between basis vectors is small, and will be avoided henceforth.

Notice that projection of a vector already in the spanning set simply returns the same vector, which leads to a general definition.

DEFINITION. *The mapping is called a projection if* $f \circ f = f$*, or if for any* $u \in U$*,* $f(f(u)) = f(u)$*. With* P *the matrix associated* f*, a projection matrix satisfies* $P^2 = P$*.*

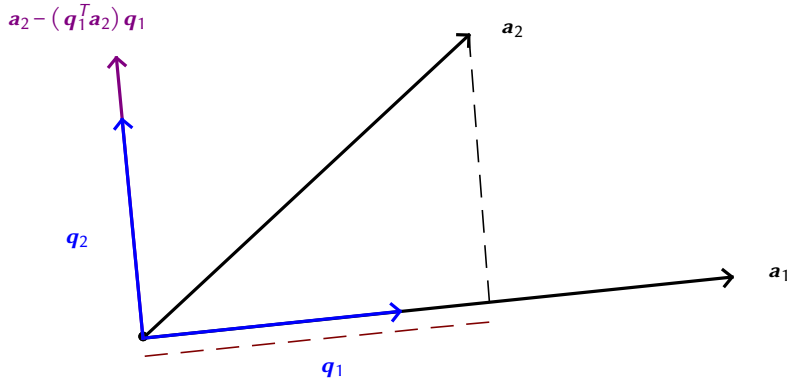
$$P = VV^T$$

$$P^2 = PP = VV^TVV^T = V\left(V^TV\right)V^T = VIV^T = VV^T = P$$

2. Gram-Schmidt

Orthonormal vector sets $\{q_1, \ldots, q_n\}$ are of the greatest practical utility, leading to the question of whether some such a set can be obtained from an arbitrary set of vectors $\{a_1, \ldots, a_n\}$. This is possible for independent vectors, through what is known as the Gram-Schmidt algorithm

1. Start with an arbitrary direction a_1
2. Divide by its norm to obtain a unit-norm vector $q_1 = a_1/\|a_1\|$
3. Choose another direction a_2
4. Subtract off its component along previous direction(s) $a_2 - (q_1^T a_2)\, q_1$
5. Divide by norm $q_2 = (a_2 - (q_1^T a_2)\, q_1)/\|a_2 - (q_1^T a_2)\, q_1\|$
6. Repeat the above



$$P_1 a_2 = (q_1 q_1^T) a_2 = q_1 (q_1^T a_2) = (q_1^T a_2)\, q_1$$

The above geometrical description can be expressed in terms of matrix operations as

$$A = \begin{pmatrix} a_1 & a_2 & \ldots & a_n \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & \ldots & q_n \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & \ldots & r_{1n} \\ 0 & r_{22} & r_{23} & \ldots & r_{2n} \\ 0 & 0 & r_{33} & \ldots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \ldots & r_{mn} \end{pmatrix} = QR,$$

equivalent to the system

$$\begin{cases} \boldsymbol{a}_1 = r_{11}\boldsymbol{q}_1 \\ \boldsymbol{a}_2 = r_{12}\boldsymbol{q}_1 + r_{22}\boldsymbol{q}_2 \\ \vdots \\ \boldsymbol{a}_n = r_{1n}\boldsymbol{q}_1 + r_{2n}\boldsymbol{q}_2 + \ldots + r_{nn}\boldsymbol{q}_n \end{cases}.$$

The system is easily solved by *forward substitution* resulting in what is known as the (modified) *Gram-Schmidt algorithm*, transcribed below both in pseudo-code and in Julia.

Algorithm (Gram-Schmidt)

Given n vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$
Initialize $\boldsymbol{q}_1 = \boldsymbol{a}_1, .., \boldsymbol{q}_n = \boldsymbol{a}_n$, $\boldsymbol{R} = \boldsymbol{I}_n$
for $i = 1$ to n
  $r_{ii} = (\boldsymbol{q}_i^T \boldsymbol{q}_i)^{1/2}$
  if $r_{ii} < \epsilon$ break;
  $\boldsymbol{q}_i = \boldsymbol{q}_i / r_{ii}$
  for $j = i+1$ to n
    $r_{ij} = \boldsymbol{q}_i^T \boldsymbol{a}_j$; $\boldsymbol{q}_j = \boldsymbol{q}_j - r_{ij}\boldsymbol{q}_i$
  end
end
return $\boldsymbol{Q}, \boldsymbol{R}$

```
∴ function mgs(A)
    m,n=size(A); Q=copy(A); R=zeros(n,n)
    for i=1:n
      R[i,i]=sqrt(Q[:,i]'*Q[:,i])
      if (R[i,i]<eps())
        break
      end
      Q[:,i]=Q[:,i]/R[i,i]
      for j=i+1:n
        R[i,j]=Q[:,i]'*A[:,j]
        Q[:,j]=Q[:,j]-R[i,j]*Q[:,i]
      end
    end
    return Q,R
  end;
∴
```

- Note that the normalization condition $\|\boldsymbol{q}_{ii}\| = 1$ is satisifed by two values $\pm r_{ii}$, so results from the above implementation might give orthogonal vectors $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n$ of different orientations than those returned by the Octave `qr` function. The implementation provided by computational packages such as Octave contain many refinements of the basic algorithm and it's usually preferable to use these in application

By analogy to arithmetic and polynomial algebra, the Gram-Schmidt algorithm furnishes a *factorization*

$$\boldsymbol{QR} = \boldsymbol{A}$$

with $\boldsymbol{Q} \in \mathbb{R}^{m \times n}$ with orthonormal columns and $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ an upper triangular matrix, known as the QR-factorization. Since the column vectors within $\boldsymbol{Q}$ were obtained through linear combinations of the column vectors of $\boldsymbol{A}$ we have

$$C(\boldsymbol{A}) = C(\boldsymbol{Q}) \neq C(\boldsymbol{R})$$

3. QR solution of linear algebra problems

The QR-factorization can be used to solve basic problems within linear algebra.

3.1. Transformation of coordinates

Recall that when given a vector $\boldsymbol{b} \in \mathbb{R}^m$, an implicit basis is assumed, the canonical basis given by the column vectors of the identity matrix $\boldsymbol{I} \in \mathbb{R}^{m \times m}$. The coordinates $\boldsymbol{x}$ in another basis $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ can be found by solving the equation

$$\boldsymbol{Ib} = \boldsymbol{b} = \boldsymbol{Ax},$$

by an intermediate change of coordinates to the orthogonal basis $\boldsymbol{Q}$. Since the basis $\boldsymbol{Q}$ is orthogonal the relation $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$ holds, and changes of coordinates from $\boldsymbol{I}$ to $\boldsymbol{Q}$, $\boldsymbol{Qc} = \boldsymbol{b}$, are easily computed $\boldsymbol{c} = \boldsymbol{Q}^T\boldsymbol{b}$. Since matrix multiplication is associative

$$\boldsymbol{b} = \boldsymbol{Ax} = (\boldsymbol{QR})\boldsymbol{x} = \boldsymbol{Q}\,(\boldsymbol{Rx}),$$

the relations $\boldsymbol{Rx} = \boldsymbol{Q}^T\boldsymbol{b} = \boldsymbol{c}$ are obtained, stating that $\boldsymbol{x}$ also contains the coordinates of $\boldsymbol{c}$ in the basis $\boldsymbol{R}$. The three steps are:

1. Compute the QR-factorization, $\boldsymbol{QR} = \boldsymbol{A}$;
2. Find the coordinates of $\boldsymbol{b}$ in the orthogonal basis $\boldsymbol{Q}$, $\boldsymbol{c} = \boldsymbol{Q}^T\boldsymbol{b}$;
3. Find the coordinates of $\boldsymbol{x}$ in basis $\boldsymbol{R}$, $\boldsymbol{Rx} = \boldsymbol{c}$.

Since $\boldsymbol{R}$ is upper-triangular,

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1m} \\ 0 & r_{22} & r_{23} & \dots & r_{2m} \\ 0 & 0 & r_{33} & \dots & r_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dots & r_{mm} \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m-1} \\ x_m \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m-1} \\ c_m \end{bmatrix}$$

the coordinates of $\boldsymbol{c}$ in the $\boldsymbol{R}$ basis are easily found by *back substitution*.

Algorithm (Back substitution)

Given R upper-triangular, vectors $\boldsymbol{c}$
for $i = m$ down to 1
  if $r_{ii} < \epsilon$ break;
  $x_i = c_i / r_{ii}$
  for $j = i$-1 down to 1
    $c_j = c_j - r_{ji}\,x_i$
  end
end
return $\boldsymbol{x}$

```
∴ function bcks(R,c)
    m,n=size(R); x=zeros(m,1)
    for i=m:-1:1
      x[i]=c[i]/R[i,i]
      for j=i-1:-1:1
        c[j]=c[j]-R[j,i]*x[i]
      end
    end
    return x
  end;
∴
```

- The above operations are carried out in the background by the backslash operation `A\b` to solve `A*x=b`, inspired by the scalar mnemonic $ax = b \Longrightarrow x = (1/a)\,b$.

3.2. General orthogonal bases

The above approch for the real vector space $\mathcal{R}_m$ can be used to determine orthogonal bases for any other vector space by appropriate modification of the scalar product. For example, within the space of smooth functions $C^\infty[-1, 1]$ that can differentiated an arbitrary number of times, the Taylor series

$$f(x) = f(0)\cdot 1 + f'(0)\cdot x + \tfrac{1}{2}f''(0)\cdot x^2 + \cdots + \tfrac{1}{n!}f^{(n)}(0)\cdot x^n + \cdots +$$

is seen to be a linear combination of the monomial basis $\boldsymbol{M} = [\ 1\ \ x\ \ x^2\ \ \dots\ ]$ with scaling coefficients $\left\{f(0), f'(0), \tfrac{1}{2}f''(0), \dots\right\}$. The scalar product

$$(f, g) = \int_{-1}^{1} f(x)\,g(x)\,\mathrm{d}x$$

can be seen as the extension to the $[-1,1]$ continuum of a the vector dot product. Orthogonalization of the monomial basis with the above scalar product leads to the definition of another family of polynomials, known as the Legendre polynomials

$$Q_0(x) = \left(\tfrac{1}{2}\right)^{1/2} \cdot 1,\ Q_1(x) = \left(\tfrac{3}{2}\right)^{1/2} \cdot x,\ Q_2(x) = \left(\tfrac{5}{8}\right)^{1/2} \cdot (3x^2 - 1),\ Q_4(x) = \left(\tfrac{7}{8}\right)^{1/2} \cdot (5x^3 - 3x), \ldots .$$

- The Legendre polynomials are usually given with a different scaling such that $P_k(1) = 1$, rather than the unit norm condition $\|Q_k\| = (Q_k, Q_k)^{1/2} = 1$. The above results can be recovered by sampling of the interval $[-1,1]$ at points $x_i = (i-1)\,h - 1$, $h = 2/(m-1)$, $i = 1, \ldots, m$, by approximation of the integral by a Riemann sum

$$\int_{-1}^{1} f(x)\, L_j(x)\, \mathrm{d}x \cong h \sum_{i=1}^{m} f(x_i) L_j(x_i) = h \boldsymbol{f}^T \boldsymbol{L}_j.$$

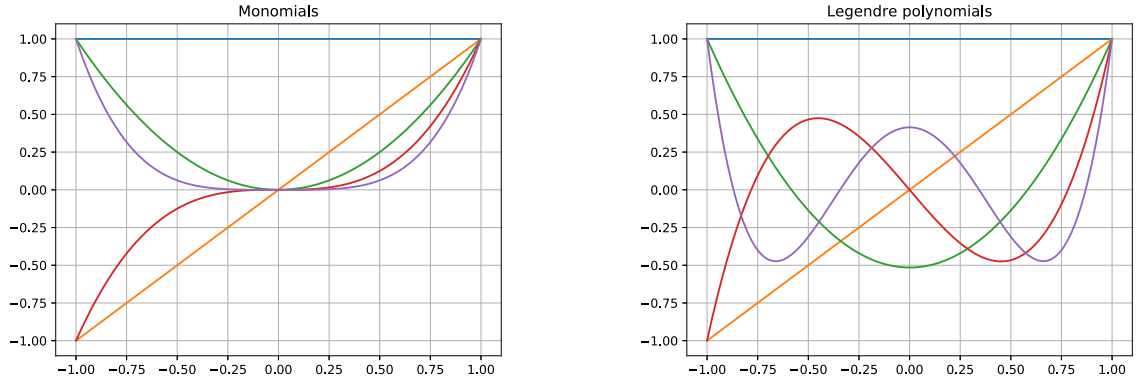


Figure 1.11. Comparison of monomial basis (left) to Legendre polynomial basis (right). The "resolution" of $P_3(x)$ can be interpreted as the number of crossings of the $y = 0$ ordinate axis, and is greater than that of the corresponding monomial x^3.

3.3. Least squares

The approach to compressing data $D = \{(x_i, y_i) | \, i = 1, \ldots, m\}$ suggested by calculus concepts is to form the sum of squared differences between $y(x_i)$ and y_i, for example for $y(x) = a_0 + a_1 x$ when carrying out linear regression,

$$S(a_0, a_1) = \sum_{i=1}^{m} (y(x_i) - y_i)^2 = \sum_{i=1}^{m} (a_0 + a_1 x_i - y_i)^2$$

and seek (a_0, a_1) that minimize $S(a_0, a_1)$. The function $S(a_0, a_1) \geqslant 0$ can be thought of as the height of a surface above the $a_0 a_1$ plane, and the gradient ∇S is defined as a vector in the direction of steepest slope. When at some point on the surface if the gradient is different from the zero vector $\nabla S \neq \mathbf{0}$, travel in the direction of the gradient would increase the height, and travel in the opposite direction would decrease the height. The minimal value of S would be attained when no local travel could decrease the function value, which is known as stationarity condition, stated as $\nabla S = 0$. Applying this to determining the coefficients (a_0, a_1) of a linear regression leads to the equations

$$\frac{\partial S}{\partial a_0} = 0 \Longrightarrow 2 \sum_{i=1}^{m} (a_0 + a_1 x_i - y_i) = 0 \Longleftrightarrow m a_0 + \left(\sum_{i=1}^{m} x_i \right) a_1 = \sum_{i=1}^{m} y_i,$$

$$\frac{\partial S}{\partial a_1} = 0 \Longrightarrow 2 \sum_{i=1}^{m} (a_0 + a_1 x_i - y_i) x_i = 0 \Longleftrightarrow \left(\sum_{i=1}^{m} x_i \right) a_0 + \left(\sum_{i=1}^{m} x_i^2 \right) a_1 = \sum_{i=1}^{m} x_i y_i.$$

The above calculations can become tedious, and do not illuminate the geometrical essence of the calculation, which can be brought out by reformulation in terms of a matrix-vector product that highlights the particular linear combination that is sought in a liner regression. Form a vector of errors with components $e_i = y(x_i) - y_i$, which for linear regression is $y(x) = a_0 + a_1 x$. Recognize that $y(x_i)$ is a linear combination of 1 and x_i with coefficients a_0, a_1, or in vector form

$$\boldsymbol{e} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} - \boldsymbol{y} = \begin{pmatrix} \mathbf{1} & \boldsymbol{x} \end{pmatrix} \boldsymbol{a} - \boldsymbol{y} = \boldsymbol{A}\boldsymbol{a} - \boldsymbol{y}$$

The norm of the error vector $\|\boldsymbol{e}\|$ is smallest when $\boldsymbol{A}\boldsymbol{a}$ is as close as possible to $\boldsymbol{y}$. Since $\boldsymbol{A}\boldsymbol{a}$ is within the column space of $C(\boldsymbol{A})$, $\boldsymbol{A}\boldsymbol{a} \in C(\boldsymbol{A})$, the required condition is for $\boldsymbol{e}$ to be orthogonal to the column space

$$\boldsymbol{e} \perp C(\boldsymbol{A}) \Longrightarrow \boldsymbol{A}^T\boldsymbol{e} = \begin{pmatrix} \mathbf{1}^T \\ \boldsymbol{x}^T \end{pmatrix} \boldsymbol{e} = \begin{pmatrix} \mathbf{1}^T\boldsymbol{e} \\ \boldsymbol{x}^T\boldsymbol{e} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \mathbf{0}$$

$$\boldsymbol{A}^T\boldsymbol{e} = \mathbf{0} \Longleftrightarrow \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{a} - \boldsymbol{y}) = 0 \Longleftrightarrow (\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{a} = \boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{b}.$$

The above is known as the normal system, with $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ is the normal matrix. The system $\boldsymbol{N}\boldsymbol{a} = \boldsymbol{b}$ can be interpreted as seeking the coordinates in the $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ basis of the vector $\boldsymbol{b} = \boldsymbol{A}^T\boldsymbol{y}$. An example can be constructed by randomly perturbing a known function $y(x) = a_0 + a_1 x$ to simulate measurement noise and compare to the approximate $\tilde{\boldsymbol{a}}$ obtained by solving the normal system.

1. Generate some data on a line and perturb it by some random quantities

```
∴ m=100; x=LinRange(0,1,m); a=[2; 3];
∴ a0=a[1]; a1=a[2]; yex=a0 .+ a1*x; y=(yex+rand(m,1) .- 0.5);
∴
```

2. Form the matrices $\boldsymbol{A}$, $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$, vector $\boldsymbol{b} = \boldsymbol{A}^T\boldsymbol{y}$

```
∴ A=ones(m,2); A[:,2]=x; N=A'*A; b=A'*y;
∴
```

3. Solve the system $\boldsymbol{N}\boldsymbol{a} = \boldsymbol{b}$, and form the linear combination $\tilde{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{a}$ closest to $\boldsymbol{y}$

```
∴ atilde=N\b; [a atilde]
```

$$\begin{bmatrix} 2.0 & 1.9215699010834906 \\ 3.0 & 3.03714411616737 \end{bmatrix} \tag{1.40}$$

```
∴
```

The normal matrix basis $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ can however be an ill-advised choice. Consider $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ given by

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{bmatrix} = \begin{bmatrix} 1 & \cos\theta \\ 0 & \sin\theta \end{bmatrix},$$

where the first column vector is taken from the identity matrix $\boldsymbol{a}_1 = \boldsymbol{e}_1$, and second is the one obtained by rotating it with angle θ. If $\theta = \pi/2$, the normal matrix is orthogonal, $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$, but for small θ, $\boldsymbol{A}$ and $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ are approximated as

$$\boldsymbol{A} \cong \begin{bmatrix} 1 & 1 \\ 0 & \theta \end{bmatrix}, \boldsymbol{N} = \begin{bmatrix} \boldsymbol{n}_1 & \boldsymbol{n}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & \theta^2 \end{bmatrix}.$$

When θ is small $\boldsymbol{a}_1, \boldsymbol{a}_2$ are almost colinear, and $\boldsymbol{n}_1, \boldsymbol{n}_2$ even more so. This can lead to amplification of small errors, but can be avoided by recognizing that the best approximation in the 2-norm is identical to the Euclidean concept of orthogonal projection. The orthogonal projector onto $C(\boldsymbol{A})$ is readily found by QR-factorization, and the steps to solve least squares become

1. Compute $\boldsymbol{QR} = \boldsymbol{A}$
2. The projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{A}$ is $\boldsymbol{z} = \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{y}$, and has coordinates $\boldsymbol{c} = \boldsymbol{Q}^T\boldsymbol{y}$ in the orthogonal basis $\boldsymbol{Q}$.
3. The same $\boldsymbol{z}$ can also obtained by linear combination of the columns of $\boldsymbol{A}$, $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{a} = \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{y}$, and replacing $\boldsymbol{A}$ with its QR-factorization gives $\boldsymbol{QRa} = \boldsymbol{Qc}$, that leads to the system $\boldsymbol{Ra} = \boldsymbol{c}$, solved by back-substitution.

```
∴ Q,R=mgs(A); c=Q'*y;
```

```
∴ aQR=R\c; [a atilde aQR]
```

$$\begin{bmatrix} 2.0 & 1.9215699010834906 & 1.9065620791027633 \\ 3.0 & 3.03714411616737 & 3.0503545613518166 \end{bmatrix} \tag{1.41}$$

```
∴
```

The above procedure carried over to approximation by higher degree polynomials.

```
∴ m=100; n=6; x=LinRange(0,1,m); a=rand(-10:10,n,1); A=ones(m,1);
```

```
∴ for j=1:n-1
    global A
    A = [A x.^j];
  end
```

```
∴ yex=A*a; y=yex .+ 0.001*(rand(m,1) .- 0.5); N=A'*A;
```

```
∴ b=A'*y;
```

```
∴ atilde=N\b; Q,R=mgs(A); c=Q'*y;
```

```
∴ aQR=R\c; [a atilde aQR]
```

$$\begin{bmatrix} 10.0 & 10.000017230385858 & 10.000017230388655 \\ -1.0 & -0.9992383469668031 & -0.9992383470406295 \\ 5.0 & 4.989621421635248 & 4.989621422094993 \\ -1.0 & -0.9668390169144284 & -0.9668390180268995 \\ -3.0 & -3.0392026170942916 & -3.0392026159405616 \\ -7.0 & -6.984207346901643 & -6.984207347332273 \end{bmatrix} \tag{1.42}$$

```
∴
```

Given data $\boldsymbol{b}$, form $\boldsymbol{A}$, find $\boldsymbol{x}$, such that $\|\boldsymbol{e}\| = \|\boldsymbol{Ax} - \boldsymbol{b}\|$ is minimized

$$\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{Ax}$$

4. Projection of mappings

4.1. Reduced matrices

The least-squares problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{y} - \boldsymbol{Ax}\| \tag{1.43}$$

focuses on a simpler representation of a data vector $\boldsymbol{y}\in\mathbb{R}^m$ as a linear combination of column vectors of $\boldsymbol{A}\in\mathbb{R}^{m\times n}$. Consider some phenomenon modeled as a function between vector spaces $\boldsymbol{f}: X \longrightarrow Y$, such that for input parameters $\boldsymbol{x}\in X$, the state of the system is $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$. For most models $\boldsymbol{f}$ is differentiable, a transcription of the condition that the system should not exhibit jumps in behavior when changing the input parameters. Then by appropriate choice of units and origin, a linearized model

$$\boldsymbol{y} = \boldsymbol{Ax}, \boldsymbol{A}\in\mathbb{R}^{m\times n},$$

is obtained if $\boldsymbol{y}\in C(\boldsymbol{A})$, expressed as (1) if $\boldsymbol{y}\notin C(\boldsymbol{A})$.

A simpler description is often sought, typically based on recognition that the inputs and outputs of the model can themselves be obtained as linear combinations $\boldsymbol{x} = \boldsymbol{Bu}$, $\boldsymbol{y} = \boldsymbol{Cv}$, involving a smaller set of parameters $\boldsymbol{u}\in\mathbb{R}^q$, $\boldsymbol{v}\in\mathbb{R}^p$, $p < m$, $q < n$. The column spaces of the matrices $\boldsymbol{B}\in\mathbb{R}^{n\times q}$, $\boldsymbol{C}\in\mathbb{R}^{m\times p}$ are vector subspaces of the original set of inputs and outputs, $C(\boldsymbol{B}) \leq \mathbb{R}^n$, $C(\boldsymbol{C}) \leq \mathbb{R}^m$. The sets of column vectors of $\boldsymbol{B}, \boldsymbol{C}$ each form a *reduced basis* for the system inputs and outputs if they are chosed to be of full rank. The reduced bases are assumed to have been orthonormalized through the Gram-Schmidt procedure such that $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{I}_q$, and $\boldsymbol{C}^T\boldsymbol{C} = \boldsymbol{I}_p$. Expressing the model inputs and outputs in terms of the reduced basis leads to

$$\boldsymbol{Cv} = \boldsymbol{ABu} \Longrightarrow \boldsymbol{v} = \boldsymbol{C}^T\boldsymbol{ABu} \Longrightarrow \boldsymbol{v} = \boldsymbol{Ru}.$$

The matrix $\boldsymbol{R} = \boldsymbol{C}^T\boldsymbol{AB}\in\mathbb{R}^{p\times q}$ is called the *reduced system matrix* and is associated with a mapping $\boldsymbol{g}: U \longrightarrow V$, that is a restriction to the U, V vector subspaces of the mapping $\boldsymbol{f}$. When $\boldsymbol{f}$ is an endomorphism, $\boldsymbol{f}: X \longrightarrow X$, $m = n$, the same reduced basis is used for both inputs and outputs, $\boldsymbol{x} = \boldsymbol{Bu}$, $\boldsymbol{y} = \boldsymbol{Bv}$, and the reduced system is

$$\boldsymbol{v} = \boldsymbol{Ru}, \boldsymbol{R} = \boldsymbol{B}^T\boldsymbol{AB}.$$

Since $\boldsymbol{B}$ is assumed to be orthogonal, the projector onto $C(\boldsymbol{B})$ is $\boldsymbol{P}_{\boldsymbol{B}} = \boldsymbol{BB}^T$. Applying the projector on the inital model

$$\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{y} = \boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{Ax}$$

leads to $\boldsymbol{BB}^T\boldsymbol{y} = \boldsymbol{BB}^T\boldsymbol{Ax}$, and since $\boldsymbol{v} = \boldsymbol{B}^T\boldsymbol{y}$ the relation $\boldsymbol{Bv} = \boldsymbol{BB}^T\boldsymbol{ABu}$ is obtained, and conveniently grouped as

$$\boldsymbol{Bv} = \boldsymbol{B}\left(\boldsymbol{B}^T\boldsymbol{AB}\right)\boldsymbol{u} \Longrightarrow \boldsymbol{Bv} = \boldsymbol{B}(\boldsymbol{Ru}),$$

again leading to the reduced model $\boldsymbol{v} = \boldsymbol{B}\boldsymbol{u}$. The above calculation highlights that the reduced model is a projection of the full model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ on $C(\boldsymbol{B})$.

4.2. Dynamical system model reduction

An often encountered situation is the reduction of large-dimensional dynamical system

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{D}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f}, \boldsymbol{M}, \boldsymbol{D}, \boldsymbol{K} \in \mathbb{R}^{m\times m}, \boldsymbol{x}, \boldsymbol{f} : \mathbb{R}_+ \longrightarrow \mathbb{R}^m, \tag{1.44}$$

$$\dot{\boldsymbol{x}} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}, \ddot{\boldsymbol{x}} = \frac{\mathrm{d}\dot{\boldsymbol{x}}}{\mathrm{d}t},$$

a generalization to multiple degrees of freedom of the damped oscillator equation

$$m\ddot{x} + d\dot{x} + kx = f.$$

In (1.44), $\boldsymbol{x}(t)$ are the time-depenent coordinates of the system, $\boldsymbol{f}(t)$ the forces acting on the system, and $\boldsymbol{M}, \boldsymbol{D}, \boldsymbol{K}$ are the mass, drag, stiffness matrices, respectively.

When $m \gg 1$, a reduced description is sought by linear combination of $n \ll m$ basis vectors

$$\boldsymbol{x} \cong \tilde{\boldsymbol{x}} = \boldsymbol{B}\boldsymbol{y} \Longrightarrow \boldsymbol{M}\boldsymbol{B}\ddot{\boldsymbol{y}} + \boldsymbol{D}\boldsymbol{B}\dot{\boldsymbol{y}} + \boldsymbol{K}\boldsymbol{B}\boldsymbol{y} = \boldsymbol{f}$$

Choose $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ to have orthonormal columns, and project (1.44) onto $C(\boldsymbol{B})$ by multiplication with the projector $\boldsymbol{P} = \boldsymbol{B}\boldsymbol{B}^T$

$$\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{M}\boldsymbol{B}\ddot{\boldsymbol{y}} + \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}\dot{\boldsymbol{y}} + \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{K}\boldsymbol{B}\boldsymbol{y} = \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{f} \Longrightarrow$$

$$\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{M}\boldsymbol{B}\ddot{\boldsymbol{y}} + \boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}\dot{\boldsymbol{y}} + \boldsymbol{B}^T\boldsymbol{K}\boldsymbol{B}\boldsymbol{y} - \boldsymbol{B}^T\boldsymbol{f}) = \boldsymbol{0} \Longleftrightarrow \boldsymbol{B}\boldsymbol{z} = \boldsymbol{0}.$$

Since $N(\boldsymbol{B}) = \{\boldsymbol{0}\}$, deduce $\boldsymbol{z} = \boldsymbol{0}$, hence

$$\boldsymbol{B}^T\boldsymbol{M}\boldsymbol{B}\ddot{\boldsymbol{y}} + \boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}\dot{\boldsymbol{y}} + \boldsymbol{B}^T\boldsymbol{K}\boldsymbol{B}\boldsymbol{y} = \boldsymbol{B}^T\boldsymbol{f}.$$

Introduce notations

$$\tilde{\boldsymbol{M}} = \boldsymbol{B}^T\boldsymbol{M}\boldsymbol{B}, \tilde{\boldsymbol{D}} = \boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}, \tilde{\boldsymbol{K}} = \boldsymbol{B}^T\boldsymbol{K}\boldsymbol{B}$$

for the reduced mass, drag, stiffness matrices, with $\tilde{\boldsymbol{M}}, \tilde{\boldsymbol{D}}, \tilde{\boldsymbol{K}} \in \mathbb{R}^{n\times n}$ of smaller size. The reduced coordinates and forces are

$$\tilde{\boldsymbol{f}} = \boldsymbol{B}^T\boldsymbol{f}, \ \boldsymbol{y}, \tilde{\boldsymbol{f}} \in \mathbb{R}^n.$$

The resulting reduced dynamical system is

$$\tilde{\boldsymbol{M}}\ddot{\boldsymbol{y}} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{y}} + \tilde{\boldsymbol{K}}\boldsymbol{y} = \tilde{\boldsymbol{f}}.$$

5. Reduced bases

One elemenet is missing from the description of model reduction above: how is $\boldsymbol{B}$ determined? Domain-specific knowledge can often dictate an appropriate basis (e.g., Fourier basis fo periodic phenomena). An alternative approach is to extract an appropriate basis from observations of a phenomenon, known as *data-driven modeling*.

5.1. Correlation matrices

Correlation coefficient. Consider two functions $x_1, x_2 : \mathbb{R} \longrightarrow \mathbb{R}$, that represent data streams in time of inputs $x_1(t)$ and outputs $x_2(t)$ of some system. A basic question arising in modeling and data science is whether the inputs and outputs are themselves in a functional relationship. This usually is a consequence of incomplete knowledge of the system, such that while x_1, x_2 might be assumed to be the most relevant input, output quantities, this is not yet fully established. A typical approach is to then carry out repeated measurements leading to a data set $D = \{(x_1(t_i), x_2(t_i)) \mid i = 1, \ldots, N\}$, thus defining a relation. Let $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^N$ denote vectors containing the input and output values. The *mean values* μ_1, μ_2 of the input and output are estimated by the statistics

$$\mu_1 \cong \bar{x}_1 = \frac{1}{N}\sum_{i=1}^{N} x_1(t_i) = E[x_1], \mu_2 \cong \bar{x}_2 = \frac{1}{N}\sum_{i=1}^{N} x_2(t_i) = E[x_2],$$

where E is the expectation seen to be a linear mapping, $E : \mathbb{R}^N \longrightarrow \mathbb{R}$ whose associated matrix is

$$\boldsymbol{E} = \frac{1}{N}[\ 1\ \ 1\ \ \ldots\ \ 1\ ],$$

and the means are also obtained by matrix vector multiplication (linear combination),

$$\bar{x}_1 = \boldsymbol{E}\boldsymbol{x}_1,\ \bar{x}_2 = \boldsymbol{E}\boldsymbol{x}_2.$$

Deviation from the mean is measured by the *standard deviation* defined for x_1, x_2 by

$$\sigma_1 = \sqrt{E[(x_1 - \mu_1)^2]},\ \sigma_2 = \sqrt{E[(x_2 - \mu_2)^2]}.$$

Note that the standard deviations are no longer linear mappings of the data.

Assume that the origin is chosen such that $\bar{x}_1 = \bar{x}_2 = 0$. One tool to estalish whether the relation D is also a function is to compute the *correlation coefficient*

$$\rho(x_1, x_2) = \frac{E[x_1 x_2]}{\sigma_1 \sigma_2} = \frac{E[x_1 x_2]}{\sqrt{E[x_1^2]\, E[x_2^2]}},$$

that can be expressed in terms of a scalar product and 2-norm as

$$\rho(x_1, x_2) = \frac{\boldsymbol{x}_1^T \boldsymbol{x}_2}{\|\boldsymbol{x}_1\|\, \|\boldsymbol{x}_2\|}.$$

Squaring each side of the norm property $\|\boldsymbol{x}_1 + \boldsymbol{x}_2\| \leq \|\boldsymbol{x}_1\| + \|\boldsymbol{x}_2\|$, leads to

$$(\boldsymbol{x}_1 + \boldsymbol{x}_2)^T (\boldsymbol{x}_1 + \boldsymbol{x}_2) \leq \boldsymbol{x}_1^T \boldsymbol{x}_1 + \boldsymbol{x}_2^T \boldsymbol{x}_2 + 2\,\|\boldsymbol{x}_1\|\, \|\boldsymbol{x}_2\| \Longrightarrow \boldsymbol{x}_1^T \boldsymbol{x}_2 \leq \|\boldsymbol{x}_1\|\, \|\boldsymbol{x}_2\|,$$

known as the Cauchy-Schwarz inequality, which implies $-1 \leqslant \rho(x_1, x_2) \leqslant 1$. Depending on the value of ρ, the variables $x_1(t), x_2(t)$ are said to be:

1. *uncorrelated*, if $\rho = 0$;
2. *correlated*, if $\rho = 1$;
3. *anti-correlated*, if $\rho = -1$.

The numerator of the correlation coefficient is known as the covariance of x_1, x_2

$$\operatorname{cov}(x_1, x_2) = E[x_1 x_2].$$

The correlation coefficient can be interpreted as a normalization of the covariance, and the relation

$$\operatorname{cov}(x_1, x_2) = \boldsymbol{x}_1^T \boldsymbol{x}_2 = \rho(x_1, x_2)\, \|\boldsymbol{x}_1\|\, \|\boldsymbol{x}_2\|,$$

is the two-variable version of a more general relationship encountered when the system inputs and outputs become vectors.

Patterns in data. Consider now a related problem, whether the input and output parameters $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ thought to characterize a system are actually well chosen, or whether they are redundant in the sense that a more insightful description is furnished by $\boldsymbol{u} \in \mathbb{R}^q$, $\boldsymbol{v} \in \mathbb{R}^p$ with fewer components $p < m, q < n$. Applying the same ideas as in the correlation coefficient, a sequence of N measurements is made leading to data sets

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n \end{bmatrix} \in \mathbb{R}^{N \times n}, \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \dots & \boldsymbol{y}_n \end{bmatrix} \in \mathbb{R}^{N \times m}.$$

Again, by appropriate choice of the origin the means of the above measurements is assumed to be zero

$$E[\boldsymbol{x}] = \boldsymbol{0}, E[\boldsymbol{y}] = \boldsymbol{0}.$$

Covariance matrices can be constructed by

$$\boldsymbol{C}_{\boldsymbol{X}} = \boldsymbol{X}^T \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^T \boldsymbol{x}_1 & \boldsymbol{x}_1^T \boldsymbol{x}_2 & \dots & \boldsymbol{x}_1^T \boldsymbol{x}_n \\ \boldsymbol{x}_2^T \boldsymbol{x}_1 & \boldsymbol{x}_2^T \boldsymbol{x}_2 & \dots & \boldsymbol{x}_2^T \boldsymbol{x}_n \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_n^T \boldsymbol{x}_1 & \boldsymbol{x}_n^T \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n^T \boldsymbol{x}_n \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Consider now the SVDs of $C_{\boldsymbol{X}} = \boldsymbol{N} \boldsymbol{\Lambda} \boldsymbol{N}^T$, $\boldsymbol{X} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{S}^T$, and from

$$\boldsymbol{C}_{\boldsymbol{X}} = \boldsymbol{X}^T \boldsymbol{X} = (\boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{S}^T)^T \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{S}^T = \boldsymbol{S} \boldsymbol{\Sigma}^T \boldsymbol{U}^T \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{S}^T = \boldsymbol{S} \boldsymbol{\Sigma}^T \boldsymbol{\Sigma} \boldsymbol{S}^T = \boldsymbol{N} \boldsymbol{\Lambda} \boldsymbol{N}^T,$$

identify $\boldsymbol{N} = \boldsymbol{S}$, and $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^T \boldsymbol{\Sigma}$.

Recall that the SVD returns an order set of singular values $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant$, and associated singular vectors. In many applications the singular values decrease quickly, often exponentially fast. Taking the first q singular modes then gives a basis set suitable for mode reduction

$$\boldsymbol{x} = \boldsymbol{S}_q \boldsymbol{u} = \begin{bmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \dots & \boldsymbol{s}_q \end{bmatrix} \boldsymbol{u}.$$

6. Stochastic systems - Karhunen-Loève theorem

The data reduction inherent in SVD representations is a generic feature of natural phenomena. A paradigm for physical systems is the evolution of correlated behavior against a backdrop of thermal enery, typically represented as a form of noise.

One mathematical technique to model such systems is the definition of a stochastic process $\{X_t\}_{a\leqslant t\leqslant b}$, where for each fixed t, X_t is a random variable, i.e., a measurable function $X\colon \Omega \longrightarrow E$ from a set of possible outcomes Ω to a measurable space E. The set Ω is the sample space of a probability triple $(\Omega, \mathcal{F}, P)$, where for $\forall S \subseteq E$

$$P(X \in S) = P(\{\omega \in \Omega\} | X(\omega) \in S|).$$

A measurable space is a set coupled with procedure to determine measurable subsets, known as a σ-algebra.

THEOREM. *Let X_t be a zero-mean ($\mathbb{E}[X_t] = 0$), square-integrable stochastic process defined over probability space $(\Omega, \mathcal{F}, P)$ indexed by $t \in \mathbb{R}$, $a \leqslant t \leqslant b$. Then X_t admits a representation*

$$X_t = \sum_{k=1}^{\infty} Z_k e_k(t),$$

with

$$Z_k = \int_a^b X_t e_k(t)\, \mathrm{d}t, \mathbb{E}[Z_k] = 0, \mathbb{E}[Z_i, Z_j] = \delta_{ij}\sigma_j.$$

LECTURE 9: LINEAR SYSTEMS

1. Gaussian elimination and row echelon reduction

Suppose now that $A\,x = b$ admits a unique solution. How to find it? We are especially interested in constructing a general procedure, that will work no matter what the size of A might be. This means we seek an *algorithm* that precisely specifies the steps that lead to the solution, and that we can program a computing device to carry out automatically. One such algorithm is *Gaussian elimination.*

Consider the system

$$\begin{cases} x_1 + 2x_2 - x_3 = 2 \\ 2x_1 - x_2 + x_3 = 2 \\ 3x_1 - x_2 - x_3 = 1 \end{cases}$$

The idea is to combine equations such that we have one fewer unknown in each equation. Ask: with what number should the first equation be multiplied in order to eliminate x_1 from sum of equation 1 and equation 2? This number is called a Gaussian multiplier, and is in this case -2. Repeat the question for eliminating x_1 from third equation, with multiplier -3.

$$\begin{cases} x_1 + 2x_2 - x_3 = 2 \\ 2x_1 - x_2 + x_3 = 2 \\ 3x_1 - x_2 - x_3 = 1 \end{cases} \Longrightarrow \begin{cases} x_1 + 2x_2 - x_3 = 2 \\ -5x_2 + 3x_3 = -2 \\ -7x_2 + 2x_3 = -5 \end{cases}$$

Now, ask: with what number should the second equation be multiplied to eliminate x_2 from sum of second and third equations. The multiplier is in this case $-7/5$.

$$\begin{cases} x_1 + 2x_2 - x_3 = 2 \\ -5x_2 + 3x_3 = -2 \\ -7x_2 + 2x_3 = -5 \end{cases} \Longrightarrow \begin{cases} x_1 + 2x_2 - x_3 = 2 \\ -5x_2 + 3x_3 = -2 \\ -\frac{11}{5}x_3 = -\frac{11}{5} \end{cases}$$

Starting from the last equation we can now find $x_3 = 1$, replace in the second to obtain $-5x_2 = -5$, hence $x_2 = 1$, and finally replace in the first equation to obtain $x_1 = 1$.

The above operations only involve coefficients. A more compact notation is therefore to work with what is known as the "bordered matrix"

$$\begin{pmatrix} 1 & 2 & -1 & 2 \\ 2 & -1 & 1 & 2 \\ 3 & -1 & -1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -1 & 2 \\ 0 & -5 & 3 & -2 \\ 0 & -7 & 2 & -5 \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -1 & 2 \\ 0 & -5 & 3 & -2 \\ 0 & 0 & -\frac{11}{5} & -\frac{11}{5} \end{pmatrix}$$

Once the above *triangular* form has been obtain, the solution is found by back substitution, in which we seek to form the identity matrix in the first 3 columns, and the solution is obtained in the last column.

$$\begin{pmatrix} 1 & 2 & -1 & 2 \\ 0 & -5 & 3 & -2 \\ 0 & 0 & -\frac{11}{5} & -\frac{11}{5} \end{pmatrix} \sim \begin{pmatrix} 1 & 2 & -1 & 2 \\ 0 & -5 & 3 & -2 \\ 0 & 0 & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

2. LU-factorization

- We have introduced Gaussian elimination as a procedure to solve the linear system $Ax = b$ ("find coordinates of vector b in terms of column vectors of matrix A"), $x, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times m}$
- We now reinterpret Gaussian elimination as a sequence of matrix multiplications applied to A to obtain a simpler, upper triangular form.

2.1. Example for $m = 3$

Consider the system $A\,x = b$

$$\begin{cases} x_1 + 2\,x_2 - x_3 &= 2 \\ 2\,x_1 - x_2 + x_3 &= 2 \\ 3\,x_1 - x_2 - x_3 &= 1 \end{cases}$$

with

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & -1 & -1 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

We ask if there is a matrix L_1 that could be multiplied with A to produce a result $L_1 A$ with zeros under the main diagonal in the first column. First, gain insight by considering multiplication by the identity matrix, which leaves A unchanged

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & -1 & -1 \end{pmatrix}$$

In the first stage of Gaussian multiplication, the first line remains unchanged, so we deduce that L_1 should have the same first line as the identity matrix

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ ? & ? & ? \\ ? & ? & ? \end{pmatrix} \begin{pmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \\ 0 & -7 & 2 \end{pmatrix}$$

Next, recall the way Gaussian multipliers were determined: find number to multiply first line so that added to second, third lines a zero is obtained. This leads to the form

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ ? & 1 & 0 \\ ? & 0 & 1 \end{pmatrix}$$

Finally, identify the missing entries with the Gaussian multipliers to determine

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}$$

Verify by carrying out the matrix multiplication

$$L_1 A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & -1 \\ 2 & -1 & 1 \\ 3 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \\ 0 & -7 & 2 \end{pmatrix}$$

Repeat the above reasoning to come up with a second matrix L_2 that forms a zero under the main diagonal in the second column

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -7/5 & 1 \end{pmatrix}$$

$$L_2 L_1 A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -7/5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \\ 0 & -7 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -1 \\ 0 & -5 & 3 \\ 0 & 0 & -11/5 \end{pmatrix} = U$$

We have obtained a matrix with zero entries under the main diagonal (an upper triangular matrix) by a sequence of matrix multiplications.

2.2. General m case

From the above, we assume that we can form a sequence of multiplier matrices such that the result is an upper triangular matrix U

$$L_{m-1} \ldots L_2 L_1 A = U$$

- Recall the basic operation in row echelon reduction: constructing a linear combination of rows to form zeros beneath the main diagonal, e.g.

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ a_{31} & a_{32} & \ldots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mm} \end{pmatrix} \sim \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ 0 & a_{22} - \frac{a_{21}}{a_{11}} a_{12} & \ldots & a_{2m} - \frac{a_{21}}{a_{11}} a_{1m} \\ 0 & a_{32} - \frac{a_{31}}{a_{11}} a_{12} & \ldots & a_{3m} - \frac{a_{31}}{a_{11}} a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m2} - \frac{a_{m1}}{a_{11}} a_{12} & \ldots & a_{mm} - \frac{a_{m1}}{a_{11}} a_{1m} \end{pmatrix}$$

- This can be stated as a matrix multiplication operation, with $l_{i1} = a_{i1}/a_{11}$

$$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ -l_{21} & 1 & 0 & \ldots & 0 \\ -l_{31} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -l_{m1} & 0 & 0 & \ldots & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ a_{31} & a_{32} & \ldots & a_{3m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mm} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ 0 & a_{22} - l_{21} a_{12} & \ldots & a_{2m} - l_{21} a_{1m} \\ 0 & a_{32} - l_{31} a_{12} & \ldots & a_{3m} - l_{31} a_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m2} - l_{m1} a_{12} & \ldots & a_{mm} - l_{m1} a_{1m} \end{pmatrix}$$

DEFINITION. *The matrix*

$$\boldsymbol{L}_k = \begin{pmatrix} 1 & \ldots & 0 & \ldots & 1 \\ 0 & \ddots & 0 & \ldots & 0 \\ 0 & \ldots & 1 & \ldots & 0 \\ 0 & \ldots & -l_{k+1,k} & \ldots & 0 \\ 0 & \ldots & -l_{k+2,k} & \ldots & 0 \\ \vdots & \ldots & \vdots & \ddots & \vdots \\ 0 & \ldots & -l_{m,k} & \ldots & 1 \end{pmatrix}$$

with $l_{i,k} = a_{i,k}^{(k)} / a_{k,k}^{(k)}$, and $\boldsymbol{A}^{(k)} = \left(a_{i,j}^{(k)}\right)$ the matrix obtained after step k of row echelon reduction (or, equivalently, Gaussian elimination) is called a Gaussian multiplier matrix.

- For $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ nonsingular, the successive steps in row echelon reduction (or Gaussian elimination) correspond to successive multiplications on the left by Gaussian multiplier matrices

$$\boldsymbol{L}_{m-1}\boldsymbol{L}_{m-2}\ldots\boldsymbol{L}_2\boldsymbol{L}_1\boldsymbol{A} = \boldsymbol{U}$$

- The inverse of a Gaussian multiplier is

$$\boldsymbol{L}_k^{-1} = \begin{pmatrix} 1 & \ldots & 0 & \ldots & 1 \\ 0 & \ddots & 0 & \ldots & 0 \\ 0 & \ldots & 1 & \ldots & 0 \\ 0 & \ldots & l_{k+1,k} & \ldots & 0 \\ 0 & \ldots & l_{k+2,k} & \ldots & 0 \\ \vdots & \ldots & \vdots & \ddots & \vdots \\ 0 & \ldots & l_{m,k} & \ldots & 1 \end{pmatrix} = \boldsymbol{I} - (\boldsymbol{L}_k - \boldsymbol{I})$$

- From $(\boldsymbol{L}_{m-1}\boldsymbol{L}_{m-2}\ldots\boldsymbol{L}_2\boldsymbol{L}_1)\boldsymbol{A} = \boldsymbol{U}$ obtain

$$\boldsymbol{A} = (\boldsymbol{L}_{m-1}\boldsymbol{L}_{m-2}\ldots\boldsymbol{L}_2\boldsymbol{L}_1)^{-1}\boldsymbol{U} = \boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdot\ldots\cdot\boldsymbol{L}_{m-1}^{-1}\boldsymbol{U} = \boldsymbol{L}\,\boldsymbol{U}$$

- Due to the simple form of $\boldsymbol{L}_k^{-1}$ the matrix $\boldsymbol{L}$ is easily obtained as

$$\boldsymbol{L} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ l_{2,1} & 1 & 0 & \ldots & 0 & 0 \\ l_{3,1} & l_{3,2} & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{m-1,1} & l_{m-1,2} & l_{m-1,3} & \ldots & 1 & 0 \\ l_{m,1} & l_{m,2} & l_{m,3} & \ldots & l_{m,m-1} & 1 \end{pmatrix}$$

We will show that this indeed possible if $A\,x = b$ admits a unique solution. Furthermore, the product of lower triangular matrices is lower triangular, and the inverse of a lower triangular matrix is lower triangular (same applies for upper triangular matrices). Introduce the notation

$$L^{-1} = L_{m-1}\ldots L_2\,L_1$$

and obtain

$$L^{-1}\,A = U$$

or

$$A = L\,U$$

The above result permits a basic insight into Gaussian elimination: the procedure depends on "factoring" the matrix A into two "simpler" matrices L, U. The idea of representing a matrix as a product of simpler matrices is fundamental to linear algebra, and we will come across it repeatedly.

For now, the factorization allows us to devise the following general approach to solving $A\,x = b$

1. Find the factorization $L\,U = A$

2. Insert the factorization into $A\,x = b$ to obtain $(L\,U)\,x = L\,(U\,x) = L\,y = b$, where the notation $y = U\,x$ has been introduced. The system

$$L\,y = b$$

   is easy to solve by forward substitution to find y for given b

3. Finally find x by backward substitution solution of

$$U\,x = y$$

Algorithm Gauss elimination without pivoting

for $s = 1$ to $m - 1$
  for $i = s + 1$ to m
    $t = -a_{is}/a_{ss}$
    for $j = s + 1$ to m
      $a_{ij} = a_{ij} + t \cdot a_{sj}$
    $b_i = b_i + t \cdot b_s$

for $s = m$ downto 1
  $x_s = b_s/a_{ss}$
  for $i = 1$ to $s - 1$
    $b_i = b_i - a_{is} \cdot x_s$

return x

Algorithm Gauss elimination with partial pivoting

$p = 1{:}\,m$ (initialize row permutation vector)
for $s = 1$ to $m - 1$
  piv = abs($a_{p(s),s}$)
  for $i = s + 1$ to m
    mag = abs($a_{p(i),s}$)
    if mag > piv then
      piv = mag; $k = p(s); p(s) = p(i); p(i) = k$
    if piv $< \epsilon$ then break("Singular matrix")
    $t = -a_{p(i)\,s}/a_{p(s)\,s}$
    for $j = s + 1$ to m
      $a_{p(i)\,j} = a_{p(i)\,j} + t \cdot a_{p(s)\,j}$
    $b_{p(i)} = b_{p(i)} + t \cdot b_{p(s)}$

for $s = m$ downto 1
  $x_s = b_{p(s)}/a_{p(s)\,s}$

for $i = 1$ to $s-1$
$b_{p(i)} = b_{p(i)} - a_{p(i)s} \cdot x_s$

return x

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

Singular value decomposition	Gram-Schmidt	Lower-upper
Transformation of coordinates	$\boldsymbol{Ax} = \boldsymbol{b}$	
$\boldsymbol{U\Sigma V^T} = \boldsymbol{A}$	$\boldsymbol{QR} = \boldsymbol{A}$	$\boldsymbol{LU} = \boldsymbol{A}$
$(\boldsymbol{U\Sigma V^T})\boldsymbol{x} = \boldsymbol{b} \Longrightarrow \boldsymbol{Uy} = \boldsymbol{b} \Longrightarrow \boldsymbol{y} = \boldsymbol{U^T b}$	$(\boldsymbol{QR})\boldsymbol{x} = \boldsymbol{b} \Longrightarrow \boldsymbol{Qy} = \boldsymbol{b},\ \boldsymbol{y} = \boldsymbol{Q^T b}$	$(\boldsymbol{LU})\boldsymbol{x} = \boldsymbol{b} \Longrightarrow \boldsymbol{Ly} = \boldsymbol{b}$ (forward sub to find)$\mathbf{y}$
$\boldsymbol{\Sigma z} = \boldsymbol{y} \Longrightarrow \boldsymbol{z} = \boldsymbol{\Sigma^+ y}$	$\boldsymbol{Rx} = \boldsymbol{y}$ (back sub to find $\boldsymbol{x}$)	$\boldsymbol{Ux} = \boldsymbol{y}$ (back sub to find $\boldsymbol{x}$)
$\boldsymbol{V^T x} = \boldsymbol{z} \Longrightarrow \boldsymbol{x} = \boldsymbol{Vz}$		

3. Inverse matrix

By analogy to the simple scalar equation $a\,x = b$ with solution $x = a^{-1}\,b$ when $a \neq 0$, we are interested in writing the solution to a linear system $A\,x = b$ as $x = A^{-1}\,b$ for $A \in \mathbb{R}^{m \times m}$, $x \in \mathbb{R}^m$. Recall that solving $A\,x = b = I\,b$ corresponds to expressing the vector b as a linear combination of the columns of A. This can only be done if the columns of A form a basis for $\mathbb{R}^m$, in which case we say that A is *non-singular*.

DEFINITION 1.3. *For matrix $A \in \mathbb{R}^{m \times m}$ non-singular the inverse matrix is denoted by A^{-1} and satisfies the properties*

$$A\,A^{-1} = A^{-1}\,A = I$$

3.1. Gauss-Jordan algorithm

Computation of the inverse A^{-1} can be carried out by repeated use of Gauss elimination. Denote the inverse by $B = A^{-1}$ for a moment and consider the inverse matrix property $A\,B = I$. Introducing the column notation for B, I leads to

$$A(\ B_1\ \ ...\ \ B_m\ ) = (\ e_1\ \ ...\ \ e_m\ )$$

and identification of each column in the equality states

$$A\,B_k = e_k, k = 1, 2, .., m$$

with e_k the column unit vector with zero components everywhere except for a 1 in row k. To find the inverse we need to simultaneously solve the m linear systems given above.

Gauss-Jordan algorithm example. Consider

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 3 & 1 \\ 2 & -1 & -2 \end{pmatrix}$$

To find the inverse we solve the systems $A\,B_1 = e_1, A\,B_2 = e_2, A\,B_3 = e_3$. This can be done simultaneously by working with the matrix A bordered by I

$$(A|I) = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 4 & -2 & 0 & 0 & 1 \end{pmatrix}$$

Carry out now operations involving linear row combinations and permutations to bring the left side to I

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 4 & -2 & 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 2 & -2 & -2 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & -3 & -3 & -1 & 1 \end{pmatrix} \sim$$

$$\sim \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & \frac{1}{3} & -\frac{1}{3} \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 1 & 1 & \frac{1}{3} & -\frac{1}{3} \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \frac{1}{3} & \frac{1}{6} \\ 0 & 0 & 1 & 1 & \frac{1}{3} & -\frac{1}{3} \end{pmatrix} \sim$$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & -\frac{1}{3} & -\frac{1}{6} \\ 0 & 1 & 0 & 0 & \frac{1}{3} & \frac{1}{6} \\ 0 & 0 & 1 & 1 & \frac{1}{3} & -\frac{1}{3} \end{pmatrix}$$

to obtain

$$A^{-1} = \begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{6} \\ 0 & \frac{1}{3} & \frac{1}{6} \\ 1 & \frac{1}{3} & -\frac{1}{3} \end{pmatrix}$$

LU FACTORIZATION OF STRUCTURED MATRICES

The special structure of a matrix can be exploited to obtain more efficient factorizations. Evaluation of the linear combination $\boldsymbol{Ax} = x_1\boldsymbol{a}_1 + \cdots + x_n\boldsymbol{a}_n$ requires mn floating point operations (flops) for $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. Evaluation of p linear combinations $\boldsymbol{AX}$, $\boldsymbol{X} \in \mathbb{C}^{n\times p}$ requires mnp flops. If it is possible to evaluate $\boldsymbol{Ax}$ with fewer operations, the matrix is said to be structured. Examples include:

- Banded matrices $\boldsymbol{A} = [a_{ij}]$, $a_{ij} = 0$ if $i - j > l$ or $j - i > u$, with l, u denoting the lower and upper bandwidths. If $l = u = 0$ the matrix is diagonal. If $l = u = b$ the matrix is said to have bandwidth $B = 2b + 1$, i.e., for $b = 1$, the matrix is tridiagonal, and for $b = 2$ the matrix is pentadiagonal. Lower triangular matrices have $u = 0$, while upper triangular matrices have $l = 0$. The $\boldsymbol{Ax}$ product requires $(l + u + 1)m$ flops.

- Sparse matrices have r non-zero elements per row or c non-zero elements per column. The $\boldsymbol{Ax}$ product requires rm or cn flops

- Circulant matrices $\boldsymbol{A} = [a_{ij}]$ are sqaure and have $a_{ij} = f(i - j)$, a property that can be exploited to compute $\boldsymbol{Ax}$ using $O(m \log m)$ operations

- For square, rank-deficient matrices $\boldsymbol{A} \in \mathbb{C}^{m\times m}$, $\text{rank}\,(\boldsymbol{A}) = r$, $\boldsymbol{Ax}$ can be evaluated in $O(km)$ flops
- When $\boldsymbol{A}, \boldsymbol{X}$ are symmetric (hence square), $\boldsymbol{AX}$ requires $O(m^3/2)$ flops instead of m^3.

1. Cholesky factorization of positive definite hermitian matrices

1.1. Symmetric matrices, hermitian matrices

Special structure of a matrix is typically associated with underlying symmetries of a particular phenomenon. For example, the law of action and reaction in dynamics (Newton's third law) leads to real symmetric matrices, $\boldsymbol{A} \in \mathbb{R}^{m\times m}$, $\boldsymbol{A}^T = \boldsymbol{A}$. Consider a system of m point masses with nearest-neighbor interactions on the real line where the interaction force depends on relative position. Assume that the force exerted by particle $i+1$ on particle i is linear

$$f_{i+1,i} = f(u_{i+1} - u_i) = k(u_{i+1} - u_i),$$

with u_i denoting displacement from an equilibrium position. The law of action and reaction then states that

$$f_{i,i+1} = -f_{i+1,i} = k(u_i - u_{i+1}).$$

If the same force law holds at all positions, then

$$f_{i-1,i} = k(u_{i-1} - u_i).$$

The force on particle i is given by the sum of forces from neighboring particles $i-1, i+1$

$$f_i = f_{i-1,i} + f_{i+1,i} = k(u_{i-1} - u_i) + k(u_{i+1} - u_i) = k(u_{i+1} - 2u_i + u_{i-1}).$$

Introducing $\boldsymbol{f}, \boldsymbol{u} \in \mathbb{R}^m$, and assuming $u_0 = u_{m+1} = 0$, the above is stated as

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{u},$$

with $\boldsymbol{K} = k\,\text{diag}([\ 1\ \ -2\ \ 1\ ])$ is a symmetric matrix, $\boldsymbol{K} = \boldsymbol{K}^T$, a direct consequence of the law of action and reaction. The matrix $\boldsymbol{K}$ is in this case tridiagonal as a consequence of the assumption of nearest-neighbor interactions. Recall that matrices represent linear mappings, hence

$$\boldsymbol{K} = [\ \boldsymbol{f}(\boldsymbol{e}_1)\ \ \boldsymbol{f}(\boldsymbol{e}_2)\ \ \ldots\ \ \boldsymbol{f}(\boldsymbol{e}_m)\ ],$$

with $\boldsymbol{f}(\boldsymbol{u})$ the force-displacement linear mapping, Fig. 1.12, obtaining the same symmetric, tri-diagonal matrix.

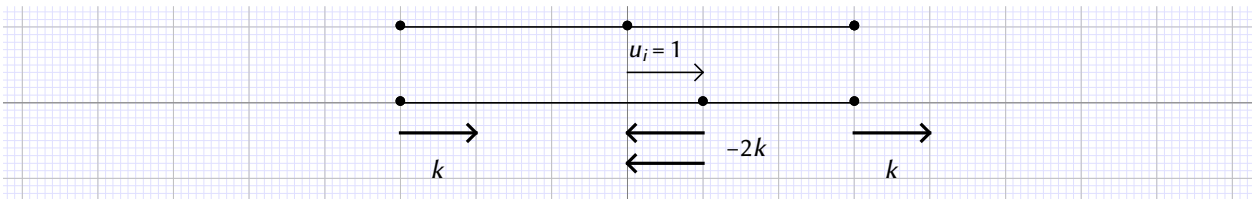


Figure 1.12. Image of $\boldsymbol{e}_i$ through mapping representing a linear force is $\boldsymbol{f}(\boldsymbol{e}_i) = k[\ \ldots\ 1\ -2\ 1\ \ldots\ ]^T$.

This concept can be extended to complex matrices $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ through $\boldsymbol{A}^* = \boldsymbol{A}$, in which case $\boldsymbol{A}$ is said to be self-adjoint or hermitian. Again, this property is often associated with desired physical properties, such as the requirement of real observable quantitites in quantum mechanics. Diagonal elements of a hermitian matrix must be real, and for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^m$, the computation

$$\overline{\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{y}} = (\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{y})^* = \boldsymbol{y}^* \boldsymbol{A}^* \boldsymbol{x} = \boldsymbol{y}^* \boldsymbol{A} \boldsymbol{x},$$

implies for $\boldsymbol{x} = \boldsymbol{y}$ that

$$\overline{\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x}} = \boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x},$$

hence $\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x}$ is real.

1.2. Positive-definite matrices

The work (i.e., energy stored in the system) done by all the forces in the above point mass system is

$$\mathcal{W} = \frac{1}{2} \boldsymbol{u}^T \boldsymbol{K} \boldsymbol{u},$$

and physical considerations state that $\mathcal{W} \geqslant 0$. This leads the following definitions.

DEFINITION. *A hermitian matrix $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ is positive definite if for any non-zero $\boldsymbol{x} \in \mathbb{C}^m$, $\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x} > 0$.*

DEFINITION. *A hermitian matrix $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ is positive semi-definite if for any non-zero $\boldsymbol{x} \in \mathbb{C}^m$, $\boldsymbol{x}^* \boldsymbol{A} \boldsymbol{x} \geqslant 0$.*

If $\boldsymbol{A}$ is hermitian positive definite, then so is $\boldsymbol{X}^* \boldsymbol{A} \boldsymbol{X}$ for any $\boldsymbol{X} \in \mathbb{C}^{m\times n}$. Choosing

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_n \end{bmatrix} \in \mathbb{C}^{m\times n}$$

gives $\boldsymbol{A}_n = \boldsymbol{X}^* \boldsymbol{A} \boldsymbol{X}$, the n^{th} principal submatrix of $\boldsymbol{A}$, itself a hermitian positive definite matrix. Choosing $\boldsymbol{X} = \boldsymbol{e}_j$shows that the j^{th} diagonal element of $\boldsymbol{A}$ is positive, $a_{jj} = \boldsymbol{e}_j^T \boldsymbol{A} \boldsymbol{e}_j > 0$

1.3. Symmetric factorization of positive-definite hermitian matrices

The structure of a hermitian positive definite matrix $\boldsymbol{A} \in \mathbb{C}^{m\times m}$, can be preserved by modification of *LU*-factorization. The resulting algorithm is known as Cholesky factorization, and its first stage is stated as

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \boldsymbol{w}^* \\ \boldsymbol{w} & \boldsymbol{B} \end{bmatrix} = \begin{bmatrix} \alpha & \boldsymbol{0} \\ \boldsymbol{w}/\alpha & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} 1 & \boldsymbol{0}^* \\ \boldsymbol{0} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} \alpha & \boldsymbol{w}^*/\alpha \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \alpha & \boldsymbol{0} \\ \boldsymbol{w}/\alpha & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \alpha & \boldsymbol{w}^*/\alpha \\ \boldsymbol{0} & \boldsymbol{C} \end{bmatrix} = \begin{bmatrix} a_{11} & \boldsymbol{w}^* \\ \boldsymbol{w} & \boldsymbol{C} + \boldsymbol{w}\boldsymbol{w}^*/a_{11} \end{bmatrix},$$

whence $\boldsymbol{C} = \boldsymbol{B} - \boldsymbol{w}\boldsymbol{w}^*/a_{11}$. Repeating the stage-1 step

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{A}_1 \boldsymbol{L}_1^*,$$

leads to

$$\boldsymbol{A} = \boldsymbol{L}_1 \boldsymbol{L}_2 \boldsymbol{A}_2 \boldsymbol{L}_2^* \boldsymbol{L}_1^* = \cdots = \boldsymbol{L}\boldsymbol{L}^*, \boldsymbol{L} = \boldsymbol{L}_1 \boldsymbol{L}_2 \ldots \boldsymbol{L}_m.$$

The resulting Cholesky algorithm is half as expensive as standard LU-factorization.

Algorithm (Cholesky factorization, $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^*$)

$\boldsymbol{L} = \boldsymbol{A}$
for $i = 1{:}\, m$
  for $j = i + 1{:}\, m$
    $L[j{:}\, m, j] = L[j{:}\, m, j] - L[j{:}\, m, i]\, \bar{L}[j, i] / L[i, i]$
  $L[i{:}\, m, i] = L[i{:}\, m, i] / \sqrt{L[i, i]}$

2. iLU-factorization of sparse matrices

The two-dimensional extension of the nearest-neighbor interacting point mass system

3. Determinants

- $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ a square matrix, $\det(\boldsymbol{A}) \in \mathbb{R}$ is the oriented volume enclosed by the column vectors of $\boldsymbol{A}$ (a parallelipiped)
- Geometric interpretation of determinants
- Determinant calculation rules
- Algebraic definition of a determinant

DEFINITION. *The determinant of a square matrix* $\boldsymbol{A} = (\, \boldsymbol{a}_1 \ \ldots \ \boldsymbol{a}_m \,) \in \mathbb{R}^{m \times m}$ *is a real number*

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mm} \end{vmatrix} \in \mathbb{R}$$

giving the (oriented) volume of the parallelepiped spanned by matrix column vectors.

- $m = 2$

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

- $m = 3$

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

- Computation of a determinant with $m = 2$

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

- Computation of a determinant with $m = 3$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{31}a_{12}a_{23} - a_{13}a_{22}a_{31} - a_{23}a_{32}a_{11} - a_{33}a_{12}a_{21}$$

- Where do these determinant computation rules come from? Two viewpoints
  - *Geometric viewpoint*: determinants express parallelepiped volumes
  - *Algebraic viewpoint*: determinants are computed from all possible products that can be formed from choosing a factor from each row and each column

- $m = 2$

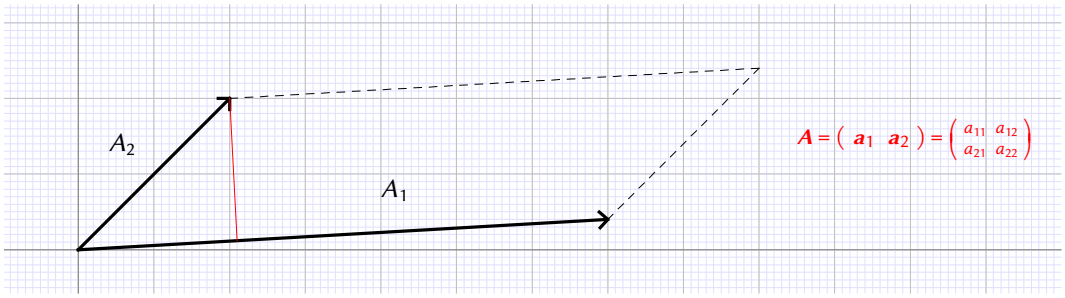


Figure 1.13.

- In two dimensions a ``parallelepiped'' becomes a parallelogram with area given as

$$(\text{Area}) = (\text{Length of Base}) \times (\text{Length of Height})$$

- Take $\boldsymbol{a}_1$ as the base, with length $b = \|\boldsymbol{a}_1\|$. Vector $\boldsymbol{a}_1$ is at angle φ_1 to x_1-axis, $\boldsymbol{a}_2$ is at angle φ_2 to x_2-axis, and the angle between $\boldsymbol{a}_1$, $\boldsymbol{a}_2$ is $\theta = \varphi_2 - \varphi_1$. The height has length

$$h = \|\boldsymbol{a}_2\| \sin\theta = \|\boldsymbol{a}_2\| \sin(\varphi_2 - \varphi_1) = \|\boldsymbol{a}_2\| (\sin\varphi_2 \cos\varphi_1 - \sin\varphi_1 \cos\varphi_2)$$

- Use $\cos\varphi_1 = a_{11}/\|\boldsymbol{a}_1\|$, $\sin\varphi_1 = a_{12}/\|\boldsymbol{a}_1\|$, $\cos\varphi_2 = a_{21}/\|\boldsymbol{a}_2\|$, $\sin\varphi_2 = a_{22}/\|\boldsymbol{a}_2\|$

$$(\text{Area}) = \|\boldsymbol{a}_1\|\,\|\boldsymbol{a}_2\|(\sin\varphi_2\cos\varphi_1 - \sin\varphi_1\cos\varphi_2) = a_{11}a_{22} - a_{12}a_{21}$$

- The geometric interpretation of a determinant as an oriented volume is useful in establishing rules for calculation with determinants:
  - Determinant of matrix with repeated columns is zero (since two edges of the parallelepiped are identical). Example for $m = 3$

$$\Delta = \begin{vmatrix} a & a & u \\ b & b & v \\ c & c & w \end{vmatrix} = abw + bcu + cav - ubc - vca - wab = 0$$

    This is more easily seen using the column notation

$$\Delta = \det(\ \boldsymbol{a}_1\ \ \boldsymbol{a}_1\ \ \boldsymbol{a}_3\ \ \dots\ ) = 0$$

  - Determinant of matrix with linearly dependent columns is zero (since one edge lies in the 'hyperplane' formed by all the others)

- Separating sums in a column (similar for rows)

$$\det(\ \boldsymbol{a}_1 + \boldsymbol{b}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ ) = \det(\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ ) + \det(\ \boldsymbol{b}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ )$$

  with $\boldsymbol{a}_i, \boldsymbol{b}_1 \in \mathbb{R}^m$

- Scalar product in a column (similar for rows)

$$\det(\ \alpha\boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ ) = \alpha\det(\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ )$$

  with $\alpha \in \mathbb{R}$

- Linear combinations of columns (similar for rows)

$$\det(\ \boldsymbol{a}_1\ \ \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ ) = \det(\ \boldsymbol{a}_1\ \ \alpha\boldsymbol{a}_1 + \boldsymbol{a}_2\ \ \dots\ \ \boldsymbol{a}_m\ )$$

  with $\alpha \in \mathbb{R}$.

- A determinant of size m can be expressed as a sum of determinants of size $m-1$ by expansion along a row or column

$$\begin{aligned}
\begin{vmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mm} \end{vmatrix} = {} & a_{11}\begin{vmatrix} a_{22} & a_{23} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m2} & a_{m3} & \dots & a_{mm} \end{vmatrix} - \\
& a_{12}\begin{vmatrix} a_{21} & a_{23} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m3} & \dots & a_{mm} \end{vmatrix} + \\
& a_{13}\begin{vmatrix} a_{21} & a_{22} & a_{24} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m4} & \dots & a_{mm} \end{vmatrix} - \\
& \dots \\
& +(-1)^{m+1}a_{1m}\begin{vmatrix} a_{21} & a_{23} & \dots & a_{2,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m3} & \dots & a_{m,m-1} \end{vmatrix}
\end{aligned}$$

- The formal definition of a determinant

$$\det A = \sum_{\sigma \in \Sigma} v(\sigma) a_{1i_1} a_{2i_2} \ldots a_{mi_m}$$

requires $mm!$ operations, a number that rapidly increases with m

- A more economical determinant is to use row and column combinations to create zeros and then reduce the size of the determinant, an algorithm reminiscent of Gauss elimination for systems

Example:

$$\begin{vmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ -2 & -1 & 4 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 0 & 2 & 4 \\ 0 & 3 & 10 \end{vmatrix} = \begin{vmatrix} 2 & 4 \\ 3 & 10 \end{vmatrix} = 20 - 12 = 8$$

The first equality comes from linear combinations of rows, i.e. row 1 is added to row 2, and row 1 multiplied by 2 is added to row 3. These linear combinations maintain the value of the determinant. The second equality comes from expansion along the first column

3.1. Cross product

- Consider $u, v \in \mathbb{R}^3$. We've introduced the idea of a scalar product

$$u \cdot v = u^T v = u_1 v_1 + u_2 v_2 + u_3 v_3$$

in which from two vectors one obtains a scalar

- We've also introduced the idea of an exterior product

$$u v^T = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix} = \begin{pmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \end{pmatrix}$$

in which a matrix is obtained from two vectors

- Another product of two vectors is also useful, the cross product, most conveniently expressed in determinant-like form

$$u \times v = \begin{vmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix} = (u_2 v_3 - v_2 u_3) \boldsymbol{e}_1 + (u_3 v_1 - v_3 u_1) \boldsymbol{e}_2 + (u_1 v_2 - v_1 u_2) \boldsymbol{e}_3$$

LECTURE 10: STABILIZED ORTHOGONAL FACTORIZATIONS

1. Conditioning of linear algebra problems

Recall that the relative condition number of a mathematical problem $f: X \longrightarrow Y$ characterizes the amplification by f of perturbations in its argument

$$\kappa = \lim_{\varepsilon \longrightarrow 0} \sup_{\|\delta x\| \leq \varepsilon} \left(\frac{\|f(x+\delta x) - f(x)\|}{\|f(x)\|} / \frac{\|\delta x\|}{\|x\|} \right).$$

Linear combination. The basic operation of linear combination $\boldsymbol{Ax}$, $\boldsymbol{A} \in \mathbb{C}^{m \times n}$, seen as the problem $\mathbb{C}^n \xrightarrow{f} \mathbb{C}^m$ has the condition number

$$\kappa = \sup_{\delta x} \left(\frac{\|\boldsymbol{A}\delta \boldsymbol{x}\|}{\|\boldsymbol{Ax}\|} / \frac{\|\delta \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \right) = \sup_{\delta x} \left(\frac{\|\boldsymbol{A}\delta \boldsymbol{x}\|}{\|\delta \boldsymbol{x}\|} \right) \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{Ax}\|} = \|\boldsymbol{A}\| \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{Ax}\|}.$$

The matrix norm property $\|\boldsymbol{Ay}\| \leqslant \|\boldsymbol{A}\| \|\boldsymbol{y}\|$ can be used to obtain

$$\|\boldsymbol{x}\| = \|\boldsymbol{I}_n \boldsymbol{x}\| = \|\boldsymbol{A}^+ \boldsymbol{Ax}\| \leqslant \|\boldsymbol{A}^+\| \|\boldsymbol{Ax}\| \Longrightarrow \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{Ax}\|} \leqslant \|\boldsymbol{A}^+\|$$

leading to

$$\kappa \leqslant \|\boldsymbol{A}\| \|\boldsymbol{A}^+\| = \kappa(\boldsymbol{A}),$$

where $\kappa(\boldsymbol{A})$ is the condition number of the matrix $\boldsymbol{A}$. If $\boldsymbol{A}$ is of full rank with $m > n$, the 2-norm condition number is given by the ratio of largest to smallest singular values.

$$\|\boldsymbol{A}\| = \sigma_1, \|\boldsymbol{A}^+\| = 1/\sigma_n \Longrightarrow \kappa(\boldsymbol{A}) = \sigma_1/\sigma_n \geqslant 1.$$

By convention, if $\boldsymbol{A}$ is singular, the condition number $\kappa(\boldsymbol{A}) = \infty$.

Coordinate transformation. For $\boldsymbol{A} \in \mathbb{C}^{m \times m}$ of full rank, the coordinates of vector $\boldsymbol{b} \in \mathbb{C}^m$ expressed in the $\boldsymbol{I}$ basis can be transformed its coordinates $\boldsymbol{x} \in \mathbb{C}^m$ in the $\boldsymbol{A}$ basis by solving the linear system $\boldsymbol{Ax} = \boldsymbol{Ib}$, with the solution $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ (so written formally, even though the inverse is almost never explicitly computed). This is simply another linear combination of the columns of $\boldsymbol{A}^{-1}$, hence the problem $\boldsymbol{f}: \mathbb{C}^m \longrightarrow \mathbb{C}^m$, $\boldsymbol{f}(\boldsymbol{b}) = \boldsymbol{A}^{-1}\boldsymbol{b}$ again has a condition number bounded by the condition number of the matrix $\boldsymbol{A}$.

$$\kappa \leqslant \|\boldsymbol{A}^{-1}\| \|\boldsymbol{A}\| = \kappa(\boldsymbol{A}) = \kappa(\boldsymbol{A}^{-1}).$$

Operator perturbation. Instead of changing the input data as above, the linear mapping represented by the matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ might itself be perturbed. Two mathematical problems may now be formulated:

1. For fixed $\boldsymbol{b} \in \mathbb{C}^m$, $\boldsymbol{f}: \mathbb{C}^{m \times n} \longrightarrow \mathbb{C}^n$, $\boldsymbol{f}(\boldsymbol{A}, \boldsymbol{b}) = \boldsymbol{A}^+ \boldsymbol{b} = \boldsymbol{x}$. Perturbation of the input $\boldsymbol{A}$ induces perturbation of $\boldsymbol{x}$ in order for $\boldsymbol{b}$ to be kept fixed

   $$(\boldsymbol{A} + \delta \boldsymbol{A})(\boldsymbol{x} + \delta \boldsymbol{x}) = \boldsymbol{b}.$$

   Using $\boldsymbol{Ax} = \boldsymbol{b}$, and assuming that $\delta \boldsymbol{A} \delta \boldsymbol{x}$ is negligible gives

   $$\boldsymbol{A}\delta \boldsymbol{x} + \delta \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Longrightarrow \delta \boldsymbol{x} = -\boldsymbol{A}^+ \delta \boldsymbol{A}\boldsymbol{x},$$

hence the relative condition number is

$$\kappa = \frac{\|\boldsymbol{A}^+ \, \delta \boldsymbol{A} \, \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta \boldsymbol{A}\|} \leqslant \frac{\|\boldsymbol{A}^+\| \, \|\, \delta \boldsymbol{A} \, \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta \boldsymbol{A}\|} \leqslant \frac{\|\boldsymbol{A}^+\| \, \|\, \delta \boldsymbol{A}\| \, \|\, \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \cdot \frac{\|\boldsymbol{A}\|}{\|\delta \boldsymbol{A}\|} = \kappa(\boldsymbol{A}).$$

For all above linear algebra problems the condition number is bounded by the associated matrix condition number. Unitary matrices $\boldsymbol{Q} \in \mathbb{C}^{m \times m}$ have $\kappa(\boldsymbol{Q}) = 1$, and define an orthonormal basis for $\mathbb{C}^m$. A rank-deficient matrix $\boldsymbol{Z} \in \mathbb{C}^{m \times m}$ has $\kappa(\boldsymbol{Z}) = \infty$, and corresponds to a linearly dependent vector set $\{z_1, \ldots, z_m\}$. The behavior of many numerical approximation procedures based upon linear combinations is determined by condition number of the basis set.

- *Monomial basis with uniform sampling*. Sampling the monomial basis on interval $[a, b]$ at $t_i = ih + a$, $i = 0, m$, $h = (b - a)/(m - 1)$ leads to the Vandermonde matrix

$$\boldsymbol{V} = \begin{bmatrix} \mathbf{1} & \boldsymbol{t} & \ldots & \boldsymbol{t}^m \end{bmatrix},$$

an extremely ill-conditioned matrix (Fig.). This can readily be surmised from the example $a = 0$, $b = 1$, in which case for large m the last columns of $\boldsymbol{V}$ become ever more colinear to the same $\boldsymbol{e}_m$ vector. Series expansions based on the monomials such as the Taylor series

$$f(t) = f(0) + f'(0)t + \cdots + \frac{f^{(n)}(0)}{n!} t^n + \cdots$$

are highly sensitive to pertubations, small changes in $f(t)$ lead to large changes in the coordinates $\{f(0), f'(0), \ldots\}$.

```
∴ function Vandermonde(a,b,m)
    t=LinRange(a,b,m); v=ones(m,1); V=copy(v)
    for j=2:m
      v = v .* t; V=[V v]
    end
    return V
  end;
```

```
∴
```

- *Monomial basis with Chebyshev sampling*. Changing the sampling so that points are clustered towards the interval endpoints reduces the condition number at fixed number of sampling points m, but the same limiting behavior for large m is obtained.

```
∴ function VandermondeC(m)
    δ=π/(2*m); θ=LinRange(δ,π-δ,m)
    t=cos.(θ)
    v=ones(m,1); V=copy(v)
    for j=2:m
      v = v .* t; V=[V v]
    end
    return V
  end;
```

```
∴
```

- *Triangular basis with uniform sampling*. LU-factorization of the monomial basis leads to a different family of polynomials, known as a triangular basis

$$\{1, t - x_1, (t - x_1) \cdot (t - x_2), \ldots, (t - x_1) \cdot \ldots \cdot (t - x_{m-1})\},$$

where $\{x_1, \ldots, x_m\}$ are known as the nodes of the system. These can be chosen to uniformly sample an interval. As to be expected, applying a sequence of non-unitary linear transformations onto an ill-conditioned basis yields even worse conditioning.

```
∴ function Triangular(a,b,m)
    x=LinRange(a,b,m); T=ones(m,1); Tj=copy(T); t=copy(x)
    for j=2:m
      Tj = Tj .* (t .- x[j-1]); T=[T Tj]
    end
    return T
  end;
```

```
∴
```

- *Triangular basis with Chebyshev sampling*. Adopting Chebyshev sampling ameliorates the conditioning, but only marginally.

```
∴ function TriangularC(m)
    δ=π/(2*m); θ=LinRange(δ,π-δ,m)
    x=cos.(θ); T=ones(m,1); Tj=copy(T); t=copy(x)
    for j=2:m
      Tj = Tj .* (t .- x[j-1]); T=[T Tj]
    end
    return T
  end;
```

```
∴
```

- 

Figure 1.14. Monomial basis with: (o) uniform sampling, (x) Chebyshev sampling. Triangular basis with: (+) uniform sampling, (*) Chebyshev sampling.

```
∴ mr=5:5:100; κVDMU=log10.(cond.(Vandermonde.(-1,1,mr)));
```

```
∴ κVDMC=log10.(cond.(VandermondeC.(mr)));
∴ κTU=log10.(cond.(Triangular.(-1,1,mr)));
∴ κTC=log10.(cond.(TriangularC.(mr)));
∴
```

```
∴ x=collect(mr); clf();
∴ plot(x,κVDMU,"o-",x,κVDMC,"x-",κTU,"+-",κTC,"*-");
∴ grid("on"); title("Condition␣number␣κ␣of␣polynomial␣bases");
∴ xlabel("Number␣of␣sample␣points"); ylabel("lg(κ)");
∴ pre=homedir()*"/courses/MATH661/images/";
∴ savefig(pre*"PolyBasesCondNr.eps");
∴
```

2. Orthogonal factorization through Householder reflectors

The Gram-Schmidt procedure constructs an orthogonal factorization by linear combinations of the column vectors of $\boldsymbol{A}\in\mathbb{C}^{m\times n}$, $m\geqslant n$, $\operatorname{rank}(\boldsymbol{A})=n$

$$\boldsymbol{A}\boldsymbol{R}_1\boldsymbol{R}_2\ldots\boldsymbol{R}_n=\boldsymbol{Q}\Longrightarrow\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R},\boldsymbol{R}=\boldsymbol{R}_n^{-1}\ldots\boldsymbol{R}_1^{-1}.$$

In exact arithmetic $C(\boldsymbol{Q})=C(\boldsymbol{A})$ by construction, and $\kappa(\boldsymbol{Q})=1$, but the sequence of multiplications with $\boldsymbol{R}_1,\ldots,\boldsymbol{R}_n$ might amplify perturbations in $\boldsymbol{A}$ (due for example to floating point representation errors or inherent uncertainty in knowledge of $\boldsymbol{A}$). The problem $\boldsymbol{f}:\mathbb{C}^{m\times n}\longrightarrow C^{m\times n}\times\mathbb{C}^{n\times n}$, $\boldsymbol{A}\overset{\boldsymbol{f}}{\longrightarrow}\boldsymbol{Q},\boldsymbol{R}$ has condition number

$$\kappa=\frac{\|\delta\boldsymbol{Q}\|}{\|\boldsymbol{Q}\|}\cdot\frac{\|\boldsymbol{A}\|}{\|\delta\boldsymbol{A}\|}+\frac{\|\delta\boldsymbol{R}\|}{\|\boldsymbol{R}\|}\cdot\frac{\|\boldsymbol{A}\|}{\|\delta\boldsymbol{A}\|},$$

and numerical experimentation (Fig. 1.15) readily exhibits large condition numbers.

An alternative approach is to obtain an orthogonal factorization through unitary transformations

$$\boldsymbol{Q}_n\ldots\boldsymbol{Q}_1\boldsymbol{A}=\boldsymbol{R}\Longrightarrow\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{R},\boldsymbol{Q}=\boldsymbol{Q}_1^*\ldots\boldsymbol{Q}_n^*.$$

Unitary transformations do not modify vector 2-norms or relative orientations

$$\|\boldsymbol{Q}\boldsymbol{x}\|^2=\boldsymbol{x}^*\boldsymbol{Q}^*\boldsymbol{Q}\boldsymbol{x}=\|\boldsymbol{x}\|^2,(\boldsymbol{Q}\boldsymbol{y})^*(\boldsymbol{Q}\boldsymbol{x})=\boldsymbol{y}^*\boldsymbol{x},$$

and are hence said to be isometric. In Euclidean space reflections and rotations are isometric.

Figure 1.15. *QR*-conditioning: (o) modified Gram-Schmidt, (x) Householder.

Construction of an isometric reflection transformation suitable for a QR factorization is represented in Fig. 1.16. Let vector $\boldsymbol{x} \in \mathbb{C}^{m+1-k}$ represent the portion of the k^{th} column from the diagonal downwards in stage k of reduction of $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ to upper triangular form

$$\boldsymbol{Q}_{k-1} \ldots \boldsymbol{Q}_1 \boldsymbol{A} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{C} \\ \boldsymbol{0} & \boldsymbol{B} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{b}_2 & \ldots & \boldsymbol{b}_{n-k} \end{bmatrix}.$$

The next stage of in reduction to upper triangular form is the isometric transformation of $\boldsymbol{x}$ into $\pm\|\boldsymbol{x}\|\,\boldsymbol{e}_1$, with $\boldsymbol{e}_1 \in \mathbb{C}^{m+1-k}$ the unit vector along the first direction. With $\boldsymbol{v} = \pm\|\boldsymbol{x}\boldsymbol{e}_1\| - \boldsymbol{x}$, $\boldsymbol{q} = \boldsymbol{v}/\|\boldsymbol{v}\|$, the projection of $\boldsymbol{x}$ onto the span of $\boldsymbol{v}$, $C(\boldsymbol{v})$ is

$$\boldsymbol{y} = \boldsymbol{P}_{\boldsymbol{v}} \boldsymbol{x} = \boldsymbol{q}\boldsymbol{q}^* \boldsymbol{x},$$

and its complementary projector onto $N(\boldsymbol{v}^*)$ is

$$\boldsymbol{z} = \boldsymbol{P}_{\perp \boldsymbol{v}} = (\boldsymbol{I} - \boldsymbol{q}\boldsymbol{q}^*)\boldsymbol{x}.$$

The reflector transforming $\boldsymbol{x}$ into $\pm\|\boldsymbol{x}\|\,\boldsymbol{e}_1$ is obtained by doubling the above displacements, and is known as a Householder reflector

$$\boldsymbol{H} = \boldsymbol{I} - 2\,\boldsymbol{q}\boldsymbol{q}^*.$$

Of the two possibilities $\pm\|\boldsymbol{x}\|\,\boldsymbol{e}_1$, the choice

$$\boldsymbol{v} = -\text{sign}(x_1)\|\boldsymbol{x}\|\,\boldsymbol{e}_1 - \boldsymbol{x},$$

avoids loss of floating accuracy $\boldsymbol{x} \cong \|\boldsymbol{x}\|\,\boldsymbol{e}_1$. For $\boldsymbol{x} \in \mathbb{C}^{m+1-k}$, $\text{sign}(x_1) = \exp(\arg(x_1))$.

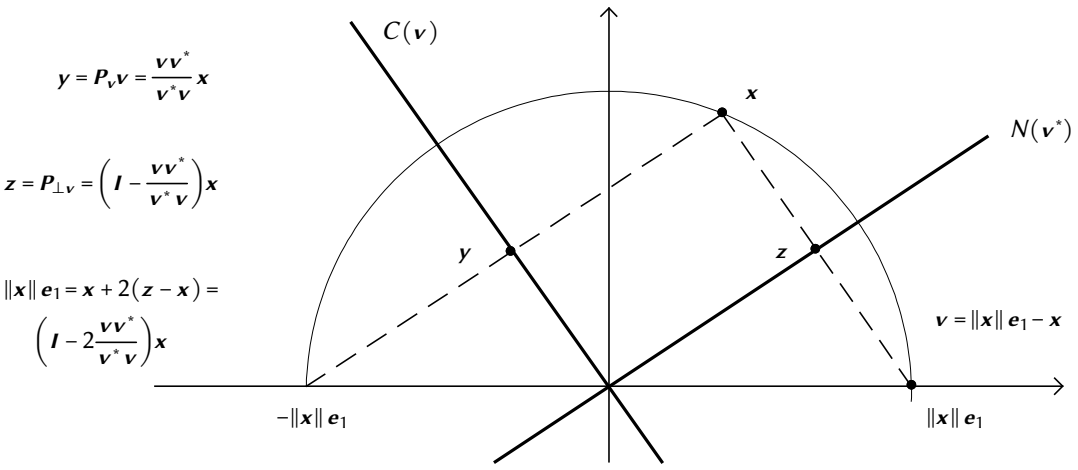


Figure 1.16. Geometry of Householder reflector

The resulting Householder QR-factorization is given

Input: $A \in \mathbb{C}^{m \times n}$
$Q = \mathbf{0}_{m,n}$
for $k = 1{:}n$
  $x = A[k{:}m, k]$
  $v = \text{sign}(x_1)\,\|x\| + x$
  $q = v/\|v\|$; $Q[k{:}m, k] = q$
  for $j = k{:}n$
    $A[k{:}m, j] = A[k{:}m, j] - 2\,q\,(q^* A[k{:}m, j])$

```
∴ function HouseholderQR(A)
    m,n=size(A)
    Q=zeros(m,n); R=copy(A)
    for k=1:n
      x=R[k:m,k]
      e1=zeros(size(x)); e1[1]=1
      v=sign(x[1])*norm(x)*e1+x
      q=v/norm(v); Q[k:m,k]=q
      for j=k:n
        aj=R[k:m,j]; c=2*q'*aj
        R[k:m,j]=aj.-c*q
      end
    end
    return Q,R
  end;
∴
```

Note that the above implementation does not return the Q matrix, but rather the $Q_1, \ldots, Q_n$ reflectors from which Q can be reconstructed if needed. Usually though, the Q matrix itself is not required, but rather the product Qu which can readily be evaluated as $Q_n \ldots Q_1 u$. The Householder reflector algorithm is typically the default procedure in QR-factorizations implemented in software systems, and as seen in (Fig. 1.15), leads to much better conditioning.

3. Orthogonal factorization through Given rotators

An alternative approach to orthogonal factorization utilizes isometric rotation transformations of the form

$$R(i,k,\theta) = I + (\cos\theta - 1)(e_i e_i^* + e_k e_k^*) - \sin\theta\,(e_i e_k^* - e_k e_i^*),$$

with the rotation angle θ chosen to nullify the subdiagonal element (i,k), $i > k$

$$(R(i,k,\theta)\,A)_{ik} = a_{kk}\sin\theta + a_{ik}\cos\theta = 0 \Longrightarrow \theta_{ik} = \arctan\left(-\frac{a_{ik}}{a_{kk}}\right).$$

Composition of repeated rotations $Q_{ik} = R(i,k,\theta_{ik})$ can be organized to lead to an upper triangular matrix

$$Q_{mn} \ldots Q_{32}\, Q_{m1} \ldots Q_{31}\, Q_{21}\, A = R.$$

Whereas Householder reflectors work on entire vectors, Givens rotators nullify individual subdiagonal elements. For full matrices Householder reflectors typically require fewer floating point operations, but the special structure of a sparse matrix is better exploited by use of Givens rotators.

Input: $\boldsymbol{A} \in \mathbb{C}^{m\times n}$
$\boldsymbol{Q} = \mathbf{0}_{m,n}$
for $k = 1{:}n$
 for $i = k+1{:}m$
 $\theta = \arctan(-a_{ik}/a_{kk})$
 $c = \cos(\theta)$; $s = \sin(\theta)$
 for $j = k{:}n$
 $u = a_{kj}$; $v = a_{ij}$
 $a_{kj} = c\,u - s\,v$
 $a_{ij} = s\,u + c\,v$

```
∴ function GivensQR(A)
    m,n=size(A)
    Q=zeros(m,n); R=copy(A)
    for k=1:n
      for i=k+1:m
        θ = atan(-R[i,k],R[k,k]); Q[i,k]=
        c = cos(θ); s = sin(θ)
        for j=k:n
          u = R[k,j]; v = R[i,j]
          R[k,j]=c*u-s*v
          R[i,j]=s*u+c*v
        end
      end
    end
    return Q,R
  end;
```

∴

As in the Householder implementation the above implementation returns data to reconstruct $\boldsymbol{Q}$ if needed.

LECTURE 11: THE EIGENVALUE PROBLEM

1. Definitions

Linear endomorphisms $\boldsymbol{f}: \mathbb{C}^m \longrightarrow \mathbb{C}^m$, represented by $\boldsymbol{A} \in \mathbb{C}^{m\times m}$, can exhibit invariant directions $\boldsymbol{x} \neq \mathbf{0}$ for which

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x},$$

known as eigenvectors, with associated eigenvalue $\lambda \in \mathbb{C}$. Eigenvectors are non-zero elements of the null space of $\boldsymbol{A} - \lambda \boldsymbol{I}$,

$$(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{x} = \mathbf{0},$$

and the null-space is referred to as the eigenspace of $\boldsymbol{A}$ for eigenvalue λ, $\mathcal{E}_{\boldsymbol{A}}(\lambda) = N(\boldsymbol{A} - \lambda \boldsymbol{I})$.

Non-zero solutions are obtained if $\boldsymbol{A} - \lambda \boldsymbol{I}$ is rank-deficient (singular), or has linearly dependent columns in which case

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0 \Longrightarrow \det(\lambda \boldsymbol{I} - \boldsymbol{A}) = \begin{vmatrix} \lambda - a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & \lambda - a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & \lambda - a_{mm} \end{vmatrix} = 0.$$

From the determinant definition as "sum of all products choosing an element from row/column", it results that

$$\det(\lambda \boldsymbol{I} - \boldsymbol{A}) = \lambda^m + c_1\lambda^{m-1} + \ldots + c_{m-1}\lambda + c_m = p_A(\lambda),$$

known as the characteristic polynomial associated with the matrix $\boldsymbol{A}$, and of degree m. The characteristic polynomial is *monic*, meaning that the coefficient of the highest power λ^m is equal to one. The fundamental theorem of algebra states that $p_A(\lambda)$ of degree m has m roots, hence $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ has m eigenvalues (not necessarily distinct), and m associated eigenvectors. This can be stated in matrix form as

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\Lambda},$$

with

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_m \end{bmatrix}, \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_m),$$

the eigenvector matrix and eigenvalue matrix, respectively. By definition, the matrix $\boldsymbol{A}$ is *diagonalizable* if $\boldsymbol{X}$ is of full rank, in which case the *eigendecomposition* of $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}.$$

1.1. Coordinate transformations

The statement $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, that eigenvector $\boldsymbol{x}$ is an invariant direction of the operator $\boldsymbol{A}$ along which the effect of operator is scaling by λ, suggests that similar behavior would be obtained under a coordinate transformation $\boldsymbol{T}\boldsymbol{y} = \boldsymbol{I}\boldsymbol{x} = \boldsymbol{x}$. Assuming $\boldsymbol{T}$ is of full rank and introducing $\boldsymbol{B} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}$, this leads to

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{T}\boldsymbol{y} = \lambda\boldsymbol{x} = \lambda\boldsymbol{T}\boldsymbol{y} \Longrightarrow \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\boldsymbol{y} = \lambda\boldsymbol{y}.$$

Upon coordinate transformation, the eigenvalues (scaling factors along the invariant directions) stay the same. Metric-preserving coordinate transformations are of particular interest, in which case the transformation matrix is unitary $\boldsymbol{B} = \boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{Q}$.

DEFINITION. *Matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{m\times m}$ are said to be similar, $\boldsymbol{B} \sim \boldsymbol{A}$, if there exists some full rank matrix $\boldsymbol{T} \in \mathbb{C}^{m\times m}$ such that $\boldsymbol{B} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}$.*

PROPOSITION. *Similar matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{m\times m}$, $\boldsymbol{B} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}$, have the same eigenvalues, and eigenvectors $\boldsymbol{x}$ of $\boldsymbol{A}$, $\boldsymbol{y}$ of $\boldsymbol{B}$ are related through $\boldsymbol{x} = \boldsymbol{T}\boldsymbol{y}$.*

Since the eigenvalues of $\boldsymbol{B} \sim \boldsymbol{A}$ are the same, and a polynomial is completely specified by its roots and coefficient of highest power, the characteristic polynomials of $\boldsymbol{A}, \boldsymbol{B}$ must be the same

$$p_{\boldsymbol{A}}(\lambda) = \prod_{k=1}^{m} (\lambda - \lambda_k) = p_{\boldsymbol{B}}(\lambda).$$

This can also be verified through the determinant definition

$$p_{\boldsymbol{B}}(t) = \det(\lambda\boldsymbol{I} - \boldsymbol{B}) = \det(\lambda\boldsymbol{T}^{-1}\boldsymbol{T} - \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}) = \det(\boldsymbol{T}^{-1}(\lambda\boldsymbol{I} - \boldsymbol{A})\boldsymbol{T}) = \det(\boldsymbol{T}^{-1})\det(\lambda\boldsymbol{I} - \boldsymbol{A})\det(\boldsymbol{T}) = p_{\boldsymbol{A}}(\lambda),$$

since $\det(\boldsymbol{T}^{-1}) = 1/\det(\boldsymbol{T})$.

1.2. Paradigmatic eigenvalue problem solutions

◦ **Reflection matrix**. The matrix

$$\boldsymbol{H} = \boldsymbol{I} - 2\,\boldsymbol{q}\boldsymbol{q}^T \in \mathbb{R}^{2\times 2}, \|\boldsymbol{q}\| = 1,$$

is the two-dimensional Householder reflector across $N(\boldsymbol{q}^T)$. Vectors colinear with $\boldsymbol{q}$ change direction along the same orientation upon reflection, while vectors orthogonal to $\boldsymbol{q}$ (i.e., in the null space $\boldsymbol{N}(\boldsymbol{q}^T)$) are unchanged. It is therefore to be expected that $\lambda_1 = -1$, $\boldsymbol{x}_1 = \boldsymbol{q}$, and $\lambda_2 = 1$, $\boldsymbol{q}^T\boldsymbol{x}_2 = 0$. This is readily verified

$$\boldsymbol{H}\boldsymbol{q} = (\boldsymbol{I} - 2\,\boldsymbol{q}\boldsymbol{q}^T)\,\boldsymbol{q} = \boldsymbol{q} - 2\,\boldsymbol{q} = -\boldsymbol{q},$$

$$\boldsymbol{H}\boldsymbol{x}_2 = (\boldsymbol{I} - 2\,\boldsymbol{q}\boldsymbol{q}^T)\,\boldsymbol{x}_2 = \boldsymbol{x}_2.$$

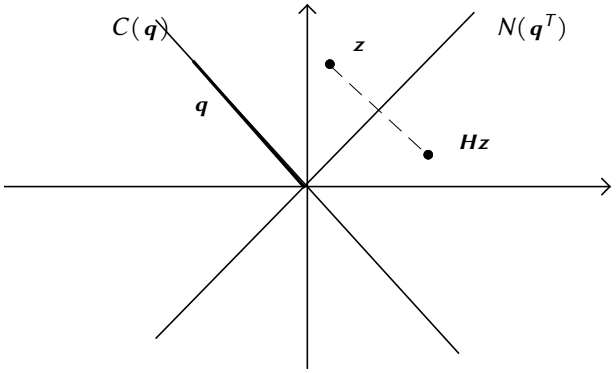


Figure 1.17. Reflector in two dimensions

◦ **Rotation matrix**. The matrix

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

represents the isometric rotation of two-dimensional vectors. If $\theta = 0$, $\boldsymbol{R} = \boldsymbol{I}$ with eigenvalues $\lambda_1 = \lambda_2 = 1$, and eigenvector matrix $\boldsymbol{X} = \boldsymbol{I}$. For $\theta = \pi$, the eigenvalues are $\lambda_1 = \lambda_2 = -1$, again with eigenvector matrix $\boldsymbol{X} = \boldsymbol{I}$. If $\sin\theta \neq 0$, the orientation of any non-zero $\boldsymbol{x} \in \mathbb{R}^2$ changes upon rotation by θ. The characteristic polynomial has complex roots

$$p(\lambda) = (\lambda - \cos\theta)^2 + \sin^2\theta \Longrightarrow \lambda_{1,2} = \cos\theta \pm i\sin\theta = e^{\pm i\theta}$$

and the directions of invariant orientation have complex components (are outside the real plane $\mathbb{R}^2$)

$$\boldsymbol{X} = \begin{bmatrix} 1 & -1 \\ i & i \end{bmatrix}, \boldsymbol{R}\boldsymbol{X} = \begin{bmatrix} e^{-i\theta} & -e^{i\theta} \\ ie^{-i\theta} & ie^{i\theta} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ i & i \end{bmatrix}\begin{bmatrix} e^{-i\theta} & 0 \\ 0 & e^{i\theta} \end{bmatrix}.$$

◦ **Second-order differentiation matrix**. Eigenvalues of matrices arising from discretization of continuum operators can be obtained from the operator eigenproblem. The second-order differentiation operator ∂_x^2 has eigenvalues $-\xi^2$ associated with eigenfunctions $\sin(\xi x)$

$$\partial_x^2 \sin(\xi x) = -\xi^2 \sin(\xi x).$$

Sampling of $\sin(\xi x)$ at $x_k = kh$, $k = 1,\ldots,m$, $h = \pi/(m+1)$ leads to the vector $\boldsymbol{u} \in \mathbb{R}^m$ with components $u_k = \sin(\xi kh)$. The boundary conditions at the sampling interval end-points affect the eigenvalues. Imposing $\sin(\xi x) = 0$, at $x = 0$ and $x = \pi$ leads to $\xi \in \mathbb{Z}$. The derivative can be approximated at the sample points through

$$u_k'' \cong \frac{\sin[\xi(x_k+h)] - 2\sin[\xi x_k] + \sin[\xi(x_k-h)]}{h^2} = \frac{2}{h^2}(\cos(\xi h) - 1)\sin(\xi kh) = -\frac{4}{h^2}\sin^2\left(\frac{\xi h}{2}\right)\sin(\xi kh).$$

The derivative approximation vector $\boldsymbol{u}'' = [\,u_k''\,]_{k=1,\ldots m}$ results from a linear mapping $\boldsymbol{u}'' = \boldsymbol{D}\boldsymbol{u}$, and the matrix

$$\boldsymbol{D} = \frac{1}{h^2}\,\mathrm{diag}([\;1\;\;-2\;\;1\;]),$$

has eigenvectors $\boldsymbol{u}$ and eigenvalues $-(4/h^2)\sin^2(\xi h/2)$, $\xi = 1, 2, \ldots, m$. In the limit of an infinite number of sampling points the continuum eigenvalues are obtained, exemplifying again the correspondence principle between discrete and continuum representations

$$\lim_{h\to 0} -\frac{4}{h^2}\sin^2\left(\frac{\xi h}{2}\right) = -\xi^2.$$

1.3. Matrix eigendecomposition

A solution $\boldsymbol{X}, \boldsymbol{\Lambda}$ to the eigenvalue problem $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\Lambda}$ always exists, but the eigenvectors of $\boldsymbol{A}$ do not always form a basis set, i.e., $\boldsymbol{X}$ is not always of full rank. The factorized form of the characteristic polynomial of $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ is

$$p_{\boldsymbol{A}}(\lambda) = \det(\lambda \boldsymbol{I} - \boldsymbol{A}) = \prod_{k=1}^{K} (\lambda - \lambda_k)^{m_k},$$

with $K \leqslant m$ denoting the number of distinct roots of $p_{\boldsymbol{A}}(\lambda)$, and m_k is the *algebraic multiplicity* of eigenvalue λ_k, defined as the number of times the root λ_k is repeated. Let $\mathcal{E}_k$ denote the associated eigenspace $\mathcal{E}_k = \mathcal{E}_{\boldsymbol{A}}(\lambda_k) = N(\boldsymbol{A} - \lambda_k \boldsymbol{I})$. The dimension of $\mathcal{E}_k$ denoted by n_k is the *geometric multiplicity* of eigenvalue λ_k. The eigenvector matrix is of full rank when the vector sum of the eigenspaces covers $\mathbb{C}^m$, as established by the following results.

PROPOSITION. *The geometric multiplicity is at least 1,* $n_k \geqslant 1$.

Proof. By contradiction if $n_k = \dim \mathcal{E}_k$, then $\mathcal{E}_k = \{\boldsymbol{0}\}$, but eigenvectors cannot be null. □

PROPOSITION. *If* $\lambda_i \neq \lambda_j$ *then* $\mathcal{E}_i \cap \mathcal{E}_j = \{\boldsymbol{0}\}$ *(the eigenspaces of distinct eigenvalues are disjoint)*

Proof. Let $\boldsymbol{x} \in \mathcal{E}_i$, hence $\boldsymbol{A}\boldsymbol{x} = \lambda_i \boldsymbol{x}$ and $\boldsymbol{x} \in \mathcal{E}_j$, hence $\boldsymbol{A}\boldsymbol{x} = \lambda_j \boldsymbol{x}$. Subtraction gives

$$\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} = (\lambda_i - \lambda_j)\boldsymbol{x}.$$

Since $\lambda_i \neq \lambda_j$ it results that $\boldsymbol{x} = \mathbf{0}$. □

PROPOSITION. *The geometric multiplicity of an eigenvalue is less or equal to its algebraic multiplicity,*

$$0 < n_k = \dim(N(\boldsymbol{A} - \lambda_k \boldsymbol{I})) \leqslant m_k.$$

Proof. Let $\hat{\boldsymbol{V}} \in \mathbb{C}^{m \times n_k}$ be an orthonormal basis for $N(\boldsymbol{A} - \lambda_k \boldsymbol{I})$. By definition of a null space, $\boldsymbol{y} \in N(\boldsymbol{A} - \lambda_k \boldsymbol{I})$

$$(\boldsymbol{A} - \lambda_k \boldsymbol{I})\,\boldsymbol{y} = \mathbf{0} \Longrightarrow \boldsymbol{A}\boldsymbol{y} = \lambda_k \boldsymbol{y},$$

i.e., every vector of the eigenspace is an eigenvector with eigenvalue λ_k. Then

$$\boldsymbol{A}\hat{\boldsymbol{V}} = \boldsymbol{A}\begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_{n_k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{v}_1 & \boldsymbol{A}\boldsymbol{v}_2 & \dots & \boldsymbol{A}\boldsymbol{v}_{n_k} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_{n_k} \end{bmatrix}.$$

Form the unitary matrix $\boldsymbol{V} = \begin{bmatrix} \hat{\boldsymbol{V}} & \boldsymbol{Z} \end{bmatrix} \in \mathbb{C}^{m \times m}$, and compute

$$\boldsymbol{V}^* \boldsymbol{A} \boldsymbol{V} = \begin{bmatrix} \hat{\boldsymbol{V}}^* \\ \boldsymbol{Z}^* \end{bmatrix} \boldsymbol{A} \begin{bmatrix} \hat{\boldsymbol{V}} & \boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{V}}^* \\ \boldsymbol{Z}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{A}\hat{\boldsymbol{V}} & \boldsymbol{A}\boldsymbol{Z} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{V}}^*\boldsymbol{A}\hat{\boldsymbol{V}} & \hat{\boldsymbol{V}}^*\boldsymbol{A}\boldsymbol{Z} \\ \boldsymbol{Z}^*\boldsymbol{A}\hat{\boldsymbol{V}} & \boldsymbol{Z}^*\boldsymbol{A}\boldsymbol{Z} \end{bmatrix}.$$

Since $\boldsymbol{V}$ is unitary, obtain

$$\hat{\boldsymbol{V}}^*\boldsymbol{A}\hat{\boldsymbol{V}} = \lambda \begin{bmatrix} \boldsymbol{v}_1^* \\ \boldsymbol{v}_2^* \\ \vdots \\ \boldsymbol{v}_{n_k}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_{n_k} \end{bmatrix} = \lambda \boldsymbol{I}_{n_k}, \boldsymbol{Z}^*\boldsymbol{A}\hat{\boldsymbol{V}} = \lambda \begin{bmatrix} \boldsymbol{z}_1^* \\ \boldsymbol{z}_2^* \\ \vdots \\ \boldsymbol{z}_{m-n_k}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_{n_k} \end{bmatrix} = \mathbf{0},$$

where $\boldsymbol{I}_{n_k}$ is the $n_k \times n_k$ identity matrix, and in the above $\mathbf{0}$ denotes a $(m - n_k) \times n_k$ matrix of zeros. The matrix

$$\boldsymbol{B} = \boldsymbol{V}^*\boldsymbol{A}\boldsymbol{V} = \begin{bmatrix} \lambda \boldsymbol{I} & \boldsymbol{C} \\ \mathbf{0} & \boldsymbol{D} \end{bmatrix}$$

is similar to $\boldsymbol{A}$ and has the same eigenvalues. Since $\det(z\boldsymbol{I} - \boldsymbol{B}) = \det((z - \lambda)\boldsymbol{I})\det(\boldsymbol{D})$, the algebraic multiplicity of λ must be at least n_k, i.e., $n_k \leqslant m_k$. □

DEFINITION 1.4. *An eigenvalue for which the geometric multiplicity is less than the algebraic multiplicity is said to be defective.*

◦ **Example.** Non-defective matrices exist, for example

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \boldsymbol{X} = \boldsymbol{I}, \boldsymbol{\Lambda} = \operatorname{diag}(\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}).$$

◦ **Example.** Non-defective matrices with repeated eigenvalues exist, for example

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{X} = \boldsymbol{I}, \boldsymbol{\Lambda} = \mathrm{diag}([\ 1\ \ 1\ \ 1\ ]).$$

◦ **Example.** Defective matrices exist, for example

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix},$$

has eigenvalue $\lambda = 3$ with algebraic multiplicity $m_1 = 3$. Reduction to row-echelon form of $\boldsymbol{A} - \lambda \boldsymbol{I}$ leads to

$$\boldsymbol{A} - \lambda \boldsymbol{I} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

and $N(\boldsymbol{A} - \lambda \boldsymbol{I}) = \langle \boldsymbol{e}_1 \rangle$, i.e., the geometric multiplicity is equal to 1. The above is known as a Jordan block.

PROPOSITION 1.5. *A matrix is diagonalizable if the geometric multiplicity of each eigenvalue is equal to the algebraic multiplicity of that eigenvalue.*

Proof. Recall that $\boldsymbol{A}$ is diagonalizable if the eigenvector matrix $\boldsymbol{X}$ is of full rank. Since the eigenspaces $\mathcal{E}_j$ of the K distinct eigenvalues are disjoint, the column space of $\boldsymbol{X}$ is the direct vector sum of the eigenspaces

$$C(\boldsymbol{X}) = \mathcal{E}_1 \oplus \ldots \oplus \mathcal{E}_K.$$

The dimension of $C(\boldsymbol{X})$ is therefore given by the sum of the eigenspace dimensions

$$\dim C(\boldsymbol{X}) = \sum_{k=1}^{K} n_k \leqslant \sum_{k=1}^{K} m_k = m.$$

Since $n_k \leqslant m_k$, the only possibility for $\boldsymbol{X}$ to be of full rank, $\dim C(\boldsymbol{X}) = m$, is for $n_k = m_k$. □

1.4. Matrix properties from eigenvalues

Eigenvalues as roots of the characteristic polynomial

$$p_A(\lambda) = \det(\lambda \boldsymbol{I} - \boldsymbol{A}) = \lambda^m + c_1 \lambda^{m-1} + \ldots + c_{m-1}\lambda + c_m = \prod_{k=1}^{m} (\lambda - \lambda_k)$$

reveal properties of a matrix $\boldsymbol{A} \in \mathbb{C}^{m \times m}$. The evaluation of $p_A(0)$ leads to

$$\det(-\boldsymbol{A}) = (-1)^m \det(\boldsymbol{A}) = (-1)^m \prod_{k=1}^{m} \lambda_k,$$

hence the determinant of a matrix is given by the product of its eigenvalues

$$\det(\boldsymbol{A}) = \prod_{k=1}^{m} \lambda_k.$$

The trace of a matrix is the sum of its diagonal elements is equal to the sum of its eigenvalues

$$\operatorname{tr}(\boldsymbol{A}) = \sum_{k=1}^{m} a_{kk} = \sum_{k=1}^{m} \lambda_k,$$

a relationship established by the Vieta formulas.

1.5. Matrix eigendecomposition applications

Whereas the SVD, QR, LU decompositions can be applied to general matrices $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ with m not necessarily equal to n, the eigendecomposition requires $\boldsymbol{A} \in \mathbb{C}^{m\times m}$, and hence is especially relevant in the characterization of endomorphisms. A generic time evolution problem is stated as

$$\partial_t \boldsymbol{u} = \boldsymbol{u}_t = \boldsymbol{f}(\boldsymbol{u}), \boldsymbol{u}(0) = \boldsymbol{u}_0, \boldsymbol{u}: \mathbb{R}_+ \longrightarrow \mathbb{C}^m,$$

stating that the rate of change in the state variables $\boldsymbol{u}$ characterizing some system is a function of the current state through the function $\boldsymbol{f}: \mathbb{C}^m \longrightarrow \mathbb{C}^m$, an endomorphism. An approximation of $\boldsymbol{f}$ is furnished by the MacLaurin series

$$\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{v} + \boldsymbol{A}\boldsymbol{u} + O(\|\boldsymbol{u}\|^2), \boldsymbol{v} = \boldsymbol{f}(\boldsymbol{0}), \boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\boldsymbol{0}).$$

Truncation at first order gives a linear ODE system $\boldsymbol{u}_t = \boldsymbol{v} + \boldsymbol{A}\boldsymbol{u}$, that can be formally integrated to give

$$\boldsymbol{u}(t) = \boldsymbol{v}t + e^{t\boldsymbol{A}}\boldsymbol{u}_0.$$

The matrix exponential $e^{t\boldsymbol{A}}$ is defined as

$$e^{t\boldsymbol{A}} = \boldsymbol{I} + \frac{1}{1!}t\boldsymbol{A} + \frac{1}{2!}(t\boldsymbol{A})^2 + \frac{1}{3!}(t\boldsymbol{A})^3 + \ldots.$$

Evaluation of $\boldsymbol{A}^n$ requires $n-1$ matrix multiplications or $(n-1)m^3$ floating point operations. However, if the eigendecomposition of $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$ is available the matrix exponential can be evaluate in only $2m^3$ operations since

$$\boldsymbol{A}^k = (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1})(\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1})\ldots(\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}) = \boldsymbol{X}\boldsymbol{\Lambda}^k\boldsymbol{X}^{-1},$$

leads to

$$e^{t\boldsymbol{A}} = \boldsymbol{X}e^{t\boldsymbol{\Lambda}}\boldsymbol{X}^{-1}.$$

2. Computation of the SVD

The existence of the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ was establish by a constructive procedure by complete induction. However the proof depends on determining the singular values, e.g., $\sigma_1 = \|\boldsymbol{A}\|$. The existence of the singular values was established by an argument from analysis, that the norm function on a compact domain must attain its extrema. This however leaves open the problem of effectively determining the singular values. In practive the singular values and vectors are determined by solving the eigenvalue problem for $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$

$$\boldsymbol{A}^*\boldsymbol{A} = (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*)^*(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*) = \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^*\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* = \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^* \Longrightarrow (\boldsymbol{A}^*\boldsymbol{A})\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma},$$

$$\boldsymbol{A}\boldsymbol{A}^* = (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*)(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*)^* = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*\boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^* = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\boldsymbol{U}^* \Longrightarrow (\boldsymbol{A}\boldsymbol{A}^*)\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T.$$

From the above the left singular vectors $\boldsymbol{U}$ are eigenvectors of $\boldsymbol{A}\boldsymbol{A}^*$, and the right singular vectors are eigenvectors of $\boldsymbol{A}^*\boldsymbol{A}$. Both $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$ have the same eigenvalues that are the squared singular values.

LECTURE 12: POWER ITERATIONS

1. Reduction to triangular form

The relevance of eigendecompositions $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$ to repeated application of the linear operator $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ as in

$$e^{t\boldsymbol{A}} = \boldsymbol{I} + \frac{1}{1!}t\boldsymbol{A} + \frac{1}{2!}t^2\boldsymbol{A}^2 + \cdots = \boldsymbol{X}e^{t\boldsymbol{\Lambda}}\boldsymbol{X}^{-1},$$

suggests that algorithms that construct powers of $\boldsymbol{A}$ might reveal eigenvalues. This is indeed the case and leads to a class of algorithms of wide applicability in scientific computation. First, observe that taking condition numbers gives

$$\mu(\boldsymbol{A}) = \mu(\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}) \leqslant \mu^2(\boldsymbol{X})\,\mu(\boldsymbol{\Lambda}) = (|\lambda|_{\max}/|\lambda|_{\min}),$$

where $|\lambda|_{\max}$, $|\lambda|_{\min}$ are the eigenvalues of maximum and minimum absolute values. While these express an intrinsic property of the operator $\boldsymbol{A}$, the factor $\mu^2(\boldsymbol{X})$ is associated with the conditioning of a change of coordinates, and a natural question is whether it is possible to avoid any ill-conditioning associated with a basis set $\boldsymbol{X}$ that is close to linear dependence. The answer to this line of inquiry is given by the following result.

SCHUR THEOREM. *For any $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ there exists $\boldsymbol{Q}$ unitary and $\boldsymbol{T}$ upper triangular such that $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{T}\boldsymbol{Q}^*$.*

Proof. *Proceed by induction, starting from an arbitrary eigenvalue λ and eigenvector $\boldsymbol{x}$. Let $\boldsymbol{u}_1 = \boldsymbol{x}/\|\boldsymbol{x}\|$, the first column vector of a unitary matrix $\boldsymbol{U} = [\ \boldsymbol{u}_1\ \ \boldsymbol{V}\ ]$. Then*

$$\boldsymbol{U}^*\boldsymbol{A}\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}_1^* \\ \boldsymbol{V}^* \end{bmatrix} \boldsymbol{A} [\ \boldsymbol{u}_1\ \ \boldsymbol{V}\ ] = \begin{bmatrix} \boldsymbol{u}_1^* \\ \boldsymbol{V}^* \end{bmatrix} [\ \boldsymbol{A}\boldsymbol{u}_1\ \ \boldsymbol{A}\boldsymbol{V}\ ] = \begin{bmatrix} \boldsymbol{u}_1^* \\ \boldsymbol{V}^* \end{bmatrix} [\ \lambda\boldsymbol{u}_1\ \ \boldsymbol{A}\boldsymbol{V}\ ] = \begin{bmatrix} \lambda_1 & \boldsymbol{b}^* \\ \boldsymbol{0} & \boldsymbol{C} \end{bmatrix},$$

with $C \in \mathbb{C}^{(m-1)\times(m-1)}$ that by the inductive hypothesis can be written as $C = WSW^*$, with W unitary, S upper triangular. The matrix

$$Q = U \begin{bmatrix} 1 & 0 \\ 0 & W \end{bmatrix}$$

is a product of unitary matrices, hence itself unitary. The computation

$$Q^* A Q = \left(U \begin{bmatrix} 1 & 0 \\ 0 & W \end{bmatrix} \right)^* A U \begin{bmatrix} 1 & 0 \\ 0 & W \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & W^* \end{bmatrix} U^* A U \begin{bmatrix} 1 & 0 \\ 0 & W \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & W^* \end{bmatrix} \begin{bmatrix} \lambda_1 & b^* \\ 0 & C \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & W \end{bmatrix} = \begin{bmatrix} \lambda_1 & b^* \\ 0 & S \end{bmatrix} = T,$$

then shows that T is indeed triangular. □

The eigenvalues of an upper triangular matrix are simply its diagonal elements, so the Schur factorization is an eigenvalue-revealing factorization.

2. Power iteration for real symmetric matrices

When the operator A expresses some physical phenomenon, the principle of action and reaction implies that $A \in \mathbb{R}^{m\times m}$ is symmetric, $A = A^T$ and has real eigenvalues. Componentwise, symmetry of $A = [a_{ij}]$ implies $a_{ij} = a_{ji}$. Consider $Ax = \lambda x$, and take the adjoint to obtain $x^T A^T = \bar{\lambda} x^T$, or $x^T A = \bar{\lambda} x^T$ since A is symmetric. Form scalar products $x^T A x = \lambda x^T x$, $x^T A^T x = \bar{\lambda} x^T x$, and subtract to obtain

$$0 = (\lambda - \bar{\lambda}) x^T x \Longrightarrow \lambda = \bar{\lambda} \Longrightarrow \lambda \in \mathbb{R},$$

since $x \neq \mathbf{0}$, an eigenvector.

Example. Consider a linear array of identical mass-springs. The i^{th} point mass obeys the dynamics

$$m\ddot{x}_i = k(x_{i+1} - x_i) - k(x_i - x_{i-1}) = k(x_{i+1} - 2x_i + x_{i-1}),$$

expressed in matrix form as $\ddot{x} = Ax$, with A symmetric.

For a real symmetric matrix the Schur theorem states that

$$A = A^T \Longrightarrow (QTQ^T) = QT^T Q^T \Longrightarrow T = T^T,$$

and since a symmetric triangular matrix is diagonal, the Schur factorization is also an eigendecomposition, and the eigenvector matrix Q is a basis, $C(Q) = \mathbb{R}^m$.

2.1. The power iteration idea

Assume initially that the eigenvalues are distinct and ordered $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|$. Repeated application of A on an arbitrary vector $v = Qc \in \mathbb{R}^m = C(Q)$ is expressed as

$$A^n v = (Q\Lambda Q^T)^n Qc = (Q\Lambda Q^T)(Q\Lambda Q^T)\ldots(Q\Lambda Q^T)\, Qc = Q\Lambda^n c,$$

a linear combination of the columns of $\boldsymbol{Q}$ (eigenvectors of $\boldsymbol{A}$) with coefficients $\boldsymbol{\Lambda}^n \boldsymbol{c} = [\ \lambda_1^n c_1\ \ \lambda_2^n c_2\ \ \dots\ \ \lambda_m^n c_m\ ]^T$.

- For large enough n, $|\lambda_1| > |\lambda_k|$, $k = 2, \dots, n$, leads to a dominant contribution along the direcion of eigenvector $\boldsymbol{q}_1$

$$\boldsymbol{A}^n \boldsymbol{v} = \boldsymbol{Q}\boldsymbol{\Lambda}^n \boldsymbol{c} = \lambda_1^n c_1\, \boldsymbol{q}_1 + \cdots + \lambda_m^n c_m \boldsymbol{q}_m \cong \lambda_1^n c_1\, \boldsymbol{q}_1.$$

  This gives a procedure for finding one eigenvector of a matrix, and the Schur theorem proof suggests a recursive algorithm to find all eigenvalues can be defined.

The sequence of normalized eigenvector approximants $\boldsymbol{v}_n = \boldsymbol{A}^n \boldsymbol{v} / \|\boldsymbol{A}^n \boldsymbol{v}\|$ is linearly convergent at rate $r = |\lambda_2 / \lambda_1|$.

2.2. Rayleigh quotient

To estimate the eigenvalue revealed by power iteration, formulate the least squares problem

$$\min_c \|\boldsymbol{A}\boldsymbol{v} - \boldsymbol{v}c\|,$$

that seeks the best approximation of one power iteration $\boldsymbol{A}\boldsymbol{v}$ as a linear combination of the initial vector $\boldsymbol{v}$. Of course, if $\boldsymbol{v} = \boldsymbol{q}$ is an eigenvector, then the solution would be $c = \lambda$, the associated eigenvalue. The projector onto $C(\boldsymbol{v})$ is

$$\boldsymbol{P} = \frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}},$$

that when applied to $\boldsymbol{A}\boldsymbol{v}$ gives the equation

$$\boldsymbol{P}\boldsymbol{A}\boldsymbol{v} = \frac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{A}\boldsymbol{v} = \frac{\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v} = c\boldsymbol{v} \Longrightarrow c = \frac{\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}.$$

The function $r: \mathbb{R}^m \longrightarrow \mathbb{R}$,

$$r(\boldsymbol{v}) = \frac{\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}},$$

is known as the Rayleigh quotient which, evaluated for an eigenvector, gives $r(\boldsymbol{q}) = \lambda$. To determine how well the eigenvalue is approximated, carry out a Taylor series in the vicinity of an eigenvector $\boldsymbol{q}$

$$r(\boldsymbol{v}) = r(\boldsymbol{q}) + \frac{1}{1!}[\nabla_{\boldsymbol{v}} r(\boldsymbol{q})]^T (\boldsymbol{v} - \boldsymbol{q}) + O(\|\boldsymbol{v} - \boldsymbol{q}\|^2),$$

where $\nabla_{\boldsymbol{v}} r$ is the gradient of $r(\boldsymbol{v})$

$$\nabla_{\boldsymbol{v}} r = \begin{bmatrix} \dfrac{\partial r}{\partial v_1} \\ \vdots \\ \dfrac{\partial r}{\partial v_m} \end{bmatrix}.$$

Compute the gradient through differentiation of the Rayleigh quotient

$$\nabla_{\boldsymbol{v}} r(\boldsymbol{v}) = \frac{\nabla_{\boldsymbol{v}}(\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v})}{\boldsymbol{v}^T\boldsymbol{v}} - \frac{(\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v})}{(\boldsymbol{v}^T\boldsymbol{v})^2}\nabla_{\boldsymbol{v}}(\boldsymbol{v}^T\boldsymbol{v}).$$

Noting that $\nabla_{\boldsymbol{v}} v_i = \boldsymbol{e}_i$, the i^{th} column of $\boldsymbol{I}$, the gradient of $\boldsymbol{v}^T\boldsymbol{v}$ is

$$\nabla_{\boldsymbol{v}} (\boldsymbol{v}^T\boldsymbol{v}) = \nabla_{\boldsymbol{v}} \sum_{i=1}^{m} v_i^2 = \sum_{i=1}^{m} \nabla_{\boldsymbol{v}} v_i^2 = \sum_{i=1}^{m} 2\, v_i \nabla_{\boldsymbol{v}} v_i = 2 \sum_{i=1}^{m} v_i \boldsymbol{e}_i = 2\boldsymbol{v}.$$

To compute $\nabla_{\boldsymbol{v}}(\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v})$, let $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}$, and since $\boldsymbol{A}$ is symmetric $\boldsymbol{u}^T = \boldsymbol{v}^T\boldsymbol{A}^T = \boldsymbol{v}^T\boldsymbol{A}$, leading to

$$\nabla_{\boldsymbol{v}}(\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}) = \nabla_{\boldsymbol{v}}(\boldsymbol{u}^T\boldsymbol{v}) = \sum_{i=1}^{m} \nabla_{\boldsymbol{v}} (u_i v_i) = \sum_{i=1}^{m} u_i \nabla_{\boldsymbol{v}} v_i + \sum_{i=1}^{m} v_i \nabla_{\boldsymbol{v}} u_i.$$

Use $u_i = \sum_{j=1}^{m} a_{ij} v_j$ also expressed as $u_j = \sum_{i=1}^{m} a_{ji} v_i$ by swapping indices to obtain

$$\nabla_{\boldsymbol{v}} u_i = \sum_{j=1}^{m} a_{ij} \nabla_{\boldsymbol{v}} v_j = \sum_{j=1}^{m} a_{ij} \boldsymbol{e}_j$$

and therefore

$$\sum_{i=1}^{m} v_i \nabla_{\boldsymbol{v}} u_i = \sum_{i=1}^{m} v_i \sum_{j=1}^{m} a_{ij} \boldsymbol{e}_j = \sum_{j=1}^{m} \sum_{i=1}^{m} a_{ij} v_i \boldsymbol{e}_j = \sum_{j=1}^{m} \sum_{i=1}^{m} a_{ij} v_i \boldsymbol{e}_j.$$

Use symmetry of $\boldsymbol{A}$ to write

$$\sum_{i=1}^{m} a_{ij} v_i = \sum_{i=1}^{m} a_{ji} v_i = u_j,$$

and substitute above to obtain

$$\sum_{i=1}^{m} v_i \nabla_{\boldsymbol{v}} u_i = \sum_{j=1}^{m} u_j \boldsymbol{e}_j = \boldsymbol{u} = \boldsymbol{A}\boldsymbol{v}.$$

Gathering the above results

$$\nabla_{\boldsymbol{v}} (\boldsymbol{v}^T\boldsymbol{v}) = 2\boldsymbol{v}, \nabla_{\boldsymbol{v}}(\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}) = 2\boldsymbol{A}\boldsymbol{v},$$

gives the following gradient of the Rayleigh quotient

$$\nabla_{\boldsymbol{v}} r(\boldsymbol{v}) = \frac{2}{\boldsymbol{v}^T\boldsymbol{v}} (\boldsymbol{A}\boldsymbol{v} - r(\boldsymbol{v})\, \boldsymbol{v}).$$

When evaluated at $\boldsymbol{v} = \boldsymbol{q}$, obtain $\nabla_{\boldsymbol{v}} r(\boldsymbol{q}) = \boldsymbol{0}$, implying that near an eigenvector the Rayleigh quotient approximation of an eigenvalue is of quadratic accuracy,

$$r(\boldsymbol{v}) - \lambda = O(\|\boldsymbol{v} - \boldsymbol{q}\|^2).$$

2.3. Refining the power iteration idea

Power iteration furnishes the largest eigenvalue. Further eigenvalues can be found by use of the following properties:

- $(\lambda, \boldsymbol{q})$ eigenpair of $\boldsymbol{A} \Longrightarrow (\lambda - \mu, \boldsymbol{q})$ eigenpair of $\boldsymbol{A} - \mu\boldsymbol{I}$;
- $(\lambda, \boldsymbol{q})$ eigenpair of $\boldsymbol{A} \Longrightarrow (1/\lambda, \boldsymbol{q})$ eigenpair of $\boldsymbol{A}^{-1}$.

If μ is a known initial approximation of the eigenvalue then the inverse power iteration $\boldsymbol{v}_n = (\boldsymbol{A} - \mu \boldsymbol{I})^{-1} \boldsymbol{v}_{n-1}$, actually implemented as successive solution of linear systems

$$(\boldsymbol{A} - \mu \boldsymbol{I})\, \boldsymbol{v}_n = \boldsymbol{v}_{n-1},$$

leads to a sequence of Rayleigh quotients $\boldsymbol{r}(\boldsymbol{v}_n)$ that converges quadratically to an eigenvalue close to μ. An important refinement of the idea is to change the shift at each iteration which leads to cubic order of convergence

Algorithm (Rayleigh quotient iteration)

Given $\boldsymbol{v}, \boldsymbol{A}$
$\mu = \boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} / \boldsymbol{v}^T \boldsymbol{v}$
for $i = 1$ to $n_{\max}$
  $\boldsymbol{w} = (\boldsymbol{A} - \mu \boldsymbol{I}) \backslash \boldsymbol{v}$ (solve linear system)
  $\boldsymbol{v} = \boldsymbol{w} / \|\boldsymbol{w}\|$
  $\lambda = \boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v}$
  if $|\lambda - \mu| < \varepsilon$ exit
  $\mu = \lambda$
end
return $\lambda, \boldsymbol{v}$

Power iteration can be applied simultaneously to multiple directions at once

Algorithm (Simultaneous iteration)

Given $\boldsymbol{A}$
$\boldsymbol{Q} = \boldsymbol{I}$; $\boldsymbol{\mu} = \mathrm{diag}(\boldsymbol{A})$
for $i = 1$ to $n_{\max}$
  $\boldsymbol{V} = \boldsymbol{A}\boldsymbol{Q}$ (power iteration applied to multiple directions)
  $\boldsymbol{Q}\boldsymbol{R} = \boldsymbol{V}$ (orthogonalize new directions)
  $\lambda = \mathrm{diag}(\boldsymbol{Q}^T \boldsymbol{A} \boldsymbol{Q})$
  if $\|\lambda - \boldsymbol{\mu}\| < \varepsilon$ exit
end
return $\lambda, \boldsymbol{Q}$

LECTURE 13: EIGENVALUE-REVEALING FACTORIZATIONS

CHAPTER 2
SCALAR FUNCTION APPROXIMATION

LECTURE 14: INTERPOLATION

The linear algebra concepts arising from study of linear mappings between vector spaces $\boldsymbol{f}: U \longrightarrow V$, $\boldsymbol{f}(\alpha \boldsymbol{u} + \beta \boldsymbol{v}) = \alpha f(\boldsymbol{u}) + \beta \boldsymbol{f}(\boldsymbol{v})$, are widely applicable to nonlinear functions also. The study of nonlinear approximation starts with the simplest case of approximation of a function with scalar values and arguments, $f: \mathbb{R} \longrightarrow \mathbb{R}$ through additive corrections.

1. Function spaces

An immediate application of the linear algebra framework is to construct vector spaces of real functions $\mathscr{F} = (F, +, \cdot)$, with $F = \{f | f: \mathbb{R} \longrightarrow \mathbb{R}\}$, and the addition and scaling operations induced from $\mathbb{R}$,

$$(\alpha f + \beta g)(t) = \alpha f(t) + g(t), f, g \in F, \alpha, \beta \in \mathbb{R}.$$

Comparing with the real vector space $(\mathbb{R}^m, +, \cdot)$ in which the analogous operation is $\alpha \boldsymbol{u} + \beta \boldsymbol{v}, \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m, \alpha, \beta \in \mathbb{R}$, or componentwise

$$(\alpha \boldsymbol{u} + \beta \boldsymbol{v})_i = \alpha u_i + \beta v_i, i = 1, 2, \ldots, m,$$

the key difference that arises is the dimension of the set of vectors. Finite-dimensional vectors within $\mathbb{R}^m$ can be regarded as functions defined on a finite set $\boldsymbol{u} \Longleftrightarrow u: \{1, 2, \ldots, m\} \longrightarrow \mathbb{R}$, with $u(i) = u_i$. The elements of F are however functions defined on $\mathbb{R}$, a set with cardinality $\mathfrak{c} = 2^{\aleph_0}$, with $\aleph_0$ the cardinality of the naturals $\mathbb{N}$. This leads to a review of the concept of a basis for this infinite-dimensional case.

1.1. Infinite dimensional basis set

In the finite dimensional case $\boldsymbol{B} \in \mathbb{R}^{m \times m}$ constituted a basis if any vector $\boldsymbol{y} \in \mathbb{R}^m$ could be expressed uniquely as a linear combination of the column vectors of

$$\forall \boldsymbol{y} \in \mathbb{R}^m, \exists! \boldsymbol{c} \in \mathbb{R}^m \text{ such that } \boldsymbol{y} = \boldsymbol{B}\boldsymbol{c} = c_1 \boldsymbol{b}_1 + \cdots + c_m \boldsymbol{b}_m.$$

While the above finite sum is well defined, there is no consistent definition of an infinite sum of vectors. As a simple example, in the vector space of real numbers $\mathscr{R}_1 = (\mathbb{R}, +, \cdot)$, any finite sum of reals is well defined, for instance

$$S_n = \sum_{k=0}^{n} (-1)^k = \begin{cases} 1 & \text{if } n \text{ even} \\ 0 & \text{if } n \text{ odd} \end{cases}$$

but the limit $S_{n \to \infty}$ cannot be determined. This leads to the necessity of seeking *finite-dimensional* linear combinations to span a vector space $\mathscr{V} = (V, S, +, \cdot)$. First, define linear independence of an infinite (possibly uncountable) set of vectors $\mathscr{B} = \{v_\gamma | \gamma \in \Gamma, v_\gamma \in V\}$, where Γ is some indexing set.

DEFINITION. *The vector set $\mathscr{B} = \{v_\gamma | \gamma \in \Gamma, v_\gamma \in V\}$, is linearly independent if the only $n \in \mathbb{N}$ scalars, $x_1, \ldots, x_n \in S$, that satisfy*

$$x_1 v_{\gamma_1} + \ldots + x_n v_{\gamma_n} = 0, \gamma_i \in \Gamma \tag{2.1}$$

are $x_1 = 0$, $x_2 = 0$,...,$x_n = 0$.

The important aspect of the above definition is that all finite vector subsets are linearly independent. The same approach is applied in the definition of a spanning set.

DEFINITION. *Vectors within the set* $\mathscr{B} = \{v_\gamma \,|\, \gamma \in \Gamma, v_\gamma \in V\}$, *span V, stated as* $V = \text{span}(\mathscr{B})$, *if for any* $u \in V$ *there exist* $n \in \mathbb{N}$ *scalars,* $x_1, \ldots, x_n \in S$, *such that*

$$x_1 v_{\gamma_1} + \ldots + x_n v_{\gamma_n} = u, \gamma_i \in \Gamma. \tag{2.2}$$

This now allows a generally-applicable definition of basis and dimension.

DEFINITION. *The vector set* $\mathscr{B} = \{v_\gamma \,|\, \gamma \in \Gamma, v_\gamma \in V\}$ *is a basis for vector space* $\mathscr{V} = (V, S, +, \cdot)$ *if*

1. $\mathscr{B}$ *is linearly independent;*
2. $\text{span}(\mathscr{B}) = V$.

DEFINITION. *The dimension of a vector space* $\mathscr{V} = (V, S, +, \cdot)$ *is the cardinality of a basis set* $\mathscr{B}$, $\dim(\mathscr{V}) = |\mathscr{B}|$.

The use of finite sums to define linear independence and bases is not overly restrictive since it can be proven that every *vector space has a basis*. The proof of this theorem is based on Zorn's lemma from set theory, and asserts exsitence of a basis, but no constructive procedure. The difficulty in practical construction of bases for infinite dimensional vector spaces is ascertained through basic examples.

Example. $\mathscr{R}_\infty$. As a generalization of $\mathscr{R}_m = (\mathbb{R}^m, \mathbb{R}, +, \cdot)$, consider the vector space of real sequences $\{x_n\}_{n \in \mathbb{N}}$ represented as a vectors with a countably infinite number of components $\boldsymbol{x} = [\ x_1\ \ x_2\ \ x_3\ \ \ldots\ ]^T$. Linear combinations are defined by

$$\alpha \boldsymbol{x} + \beta \boldsymbol{y} = [\ \alpha x_1 + \beta y_1\ \ \alpha x_2 + \beta y_2\ \ \alpha x_3 + \beta y_3\ \ \ldots\ ]^T.$$

Let $\boldsymbol{e}_i$ denote the vector of all zeros except the i^{th} position. In $\mathbb{R}^m$, the identity matrix $\boldsymbol{I} = [\ \boldsymbol{e}_1\ \ \ldots\ \ \boldsymbol{e}_m\ ]$ was a basis, but this does not generalize to $\mathbb{R}^\infty$; for example the vector $\boldsymbol{v} = [\ 1\ \ 1\ \ 1\ \ \ldots\ ]^T$ cannot be obtained by finite linear combination of the $\boldsymbol{e}_i$ vectors. In fact, there is no countable set of vectors that spans $\mathbb{R}^\infty$.

Example. $P(\mathbb{R})$. The vector space of polynomials $P(\mathbb{R}) = \{p \,|\, p(t) = c_n t^n + c_{n-1} t^{n-1} + \cdots + c_0, n \in \mathbb{N}, c_i \in \mathbb{R}, i = 0, 1, \ldots, N\}$ on the real line has an easily constructed basis, namely the set of the monomials

$$\mathscr{B}(t) = \{t^n \,|\, n \in \mathbb{N}\},$$

an infinite set with the cardinality as the naturals $|\mathscr{B}| = |\mathbb{N}| = \aleph_0$.

1.2. Alternatives to the concept of a basis

The difficulty in ascribing significance to an infinite sum of vectors $\sum_{i=1}^{\infty} \boldsymbol{v}_i$ can be resolved by endowing the vector space with additional structure, in particular a way to define convergence of the partial sums

$$\boldsymbol{s}_n = \sum_{i=1}^{n} \boldsymbol{v}_i$$

to a limit $\lim_{n\to\infty} \boldsymbol{s}_n = \boldsymbol{v}$.

Fourier series. One approach is the introduction of an inner product $(\boldsymbol{u}, \boldsymbol{v})$ and the associated norm $\|\boldsymbol{u}\| = (\boldsymbol{u}, \boldsymbol{v})^{1/2}$. A considerable advantage of this approach is that it not only allows infinite linear combinations, but also definition of orthonormal spanning sets. An example is the vector space of continuous functions defined on $[-\pi, \pi]$ with the inner product

$$(f, g) = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\, g(t)\, \mathrm{d}t,$$

and norm $\|f\| = (f, f)^{1/2}$. An orthonormal spanning set for $C[-\pi, \pi]$ is given by

$$\left\{\frac{1}{2}\right\} \bigcup \{\cos(nx) | n \in \mathbb{N}^+\} \bigcup \{\sin(nx) | n \in \mathbb{N}^+\}.$$

Vector spaces with an inner product are known as *Hilbert spaces*.

Taylor series. Convergence of infinite sums can be determined through a norm, without the need of an inner product. An example is the space of real-analytic functions with the inf-norm

$$\|f\|_\infty = \sup_x |f(t)|,$$

for which a spanning set is given by the monomials $\{1, t, t^2, \dots\}$, and the infinite exapnsion

$$f(t) = \sum_{k=0}^{\infty} a_k (t - c)^k$$

is convergent, with coefficients given by the Taylor series

$$f(t) = f(c) + \frac{f'(c)}{1!}(t - c) + \cdots, a_k = \frac{f^{(k)}(c)}{k!}.$$

Note that orthogonality of the spanning set cannot be established, absent an inner product.

1.3. Common function spaces

Several function spaces find widespread application in scientific computation. An overview is provided in Table 2.1.

$B(\mathbb{R})$	bounded functions		
$C(\mathbb{R})$	continuous functions	$C^r(\mathbb{R})$	with continuous derivatives up to r
$C_c(\mathbb{R})$	with compact support	$C_c^r(\mathbb{R})$	and compact support
$C_0(\mathbb{R})$	that vanish at infinity	$C^\infty(\mathbb{R})$	smooth functions
$L^p(\mathbb{R})$	with finite p-norm $\|f\|_p = (\int_{-\infty}^{\infty} \|f(t)\|^2 \mathrm{d}t)^{1/p}$	$W^{k,p}(\mathbb{R})$	Sobolev space, with norm $\|f\|_{k,p} = \left(\sum_{i=0}^{k} \|f^{(i)}\|_p^p\right)^{1/p}$

Table 2.1. Common vector spaces of functions

2. Interpolation

The interpolation problem seeks the representation of a function f known only through a sample data set $\mathcal{D} = \{(x_i, y_i = f(x_i)) | i = 0,\ldots,m\} \subset \mathbb{R} \times \mathbb{R}$, by an approximant $p(t)$, obtained through combination of elements from some family of computable functions, $\mathcal{B} = \{b_0, \ldots, b_n | b_k : \mathbb{R} \longrightarrow \mathbb{R}\}$. The approximant $p(t)$ is an interpolant of $\mathcal{D}$ if

$$p(x_i) = f(x_i) = y_i, i = 0, \ldots, m,$$

or $p(t)$ passes through the known poles (x_i, y_i) of the function f. The objective is to use $p(t)$ thus determined to approximate the function f at other points. Assuming $x_0 < x_1 < \cdots < x_m$, evaluation of $p(t)$ at $t \in (x_0, x_m)$ is an *interpolation*, while evaluation at $t < x_0$ or $t > x_m$, is an *extrapolation*. The basic problems arising in interpolation are:

- choice of the family from which to build the approximant $p(t)$;
- choice of the combination technique;
- estimation of the error of the approximation given some knowledge of f.

◦ Algorithms for interpolation of real functions can readily be extended to more complicated objects, e.g., interpolation of matrix representations of operators. Implementation is aided by programming language polymorphism as in Julia.

2.1. Additive corrections

As to be expected, a widely used combination technique is linear combination,

$$p(t) = c_0 b_0(t) + c_1 b_1(t) + \cdots + c_n b_n(t).$$

The idea is to capture the nonlinearity of $f(t)$ through the functions $b_0(t),\ldots,b_n(t)$, while maintaining the framework of linear combinations. Sampling of $b_j(t)$ at the poles x_i of a data set $\mathcal{D}$, constructs the vectors

$$\boldsymbol{b}_j = b_j(\boldsymbol{x}) = [\ b_j(x_0)\ \ \ldots\ \ b_j(x_m)\ ]^T \in \mathbb{R}^{m+1},$$

which gathered together into a matrix leads to the formulation of the interpolation problem as

$$\boldsymbol{B}\boldsymbol{c} = \boldsymbol{y} = \boldsymbol{I}\boldsymbol{y}, \boldsymbol{B} \in \mathbb{R}^{(m+1)\times(n+1)}. \tag{2.3}$$

Before choosing some specific function set $\mathcal{B}$, some general observations are useful.

1. The function values $y_i = f(x_i)$, $i = 0, \ldots, m$, are directly incorporated into the interpolation problem (2.3). Any estimate of the error at other points requires additional information on f. Such information can be furnished by bounds on the function values, or knowledge of its derivatives for example.

2. A solution to (2.3) exists if $\boldsymbol{y} \in C(\boldsymbol{B})$. Economical interpolations would use $n < m$ functions in the set $\mathcal{B}$, hopefully $n \ll m$.

2.2. Polynomial interpolation

Monomial form of interpolating polynomial. As noted above, the vector space of polynomials $P(\mathbb{R})$ has an easily constructed basis, that of the monomials $m_j(t) = t^j$ which shall be organized as a row vector of functions

$$\mathcal{M}(t) = [\ 1\ \ t\ \ t^2\ \ \ldots\ ].$$

With $\mathscr{M}_{n+1}(t)$ denoting the first $n+1$ monomials

$$\mathscr{M}_{n+1}(t) = \begin{bmatrix} 1 & t & \dots & t^n \end{bmatrix},$$

a polynomial of degree n is the linear combination

$$p(t) = \mathscr{M}_{n+1}(t)\,\boldsymbol{a} = \begin{bmatrix} 1 & t & \dots & t^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = a_0 + a_1 t + \cdots + a_n t^n.$$

Let $\boldsymbol{M} \in \mathbb{R}^{(m+1)\times(n+1)}$ denote the matrix obtained from evaluation of the first $n+1$ monomials at the sample points $\boldsymbol{x} = \begin{bmatrix} x_0 & x_1 & \dots & x_m \end{bmatrix}^T$,

$$\boldsymbol{M} = \mathscr{M}_{n+1}(\boldsymbol{x}).$$

The above notation conveys that a finite-dimensional matrix $\boldsymbol{M} \in \mathbb{R}^{(m+1)\times(n+1)}$ is obtained from evaluation of the row vector of the monomial basis function $\mathscr{M}(x): \mathbb{R} \longrightarrow \mathbb{R}^{n+1}$, at the column vector of sample points $\boldsymbol{x} \in \mathbb{R}^{m+1}$. The interpolation condition $p(\boldsymbol{x}) = \boldsymbol{y}$ leads to the linear system

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{y}. \tag{2.4}$$

For a solution to exist for arbitrary $\boldsymbol{y}$, $\boldsymbol{M}$ must be of full rank, hence $m = n$, in which case $\boldsymbol{M}$ becomes the Vandermonde matrix

$$\boldsymbol{M} = \begin{bmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^n \end{bmatrix},$$

known to be ill-conditioned. Since $\boldsymbol{M}$ is square and of full rank, (2.4) has a unique solution.

Finding the polynomial coefficients by solving the above linear system requires $O(n^3/3)$ operations. Evaluation of the monomial form is economically accomplished in $O(n)$ operations through Horner's scheme

$$p(t) = a_0 + (a_1 + \cdots + (a_{n-2} + (a_{n-1} + a_n t) \cdot t) \cdot t) \cdot t. \tag{2.5}$$

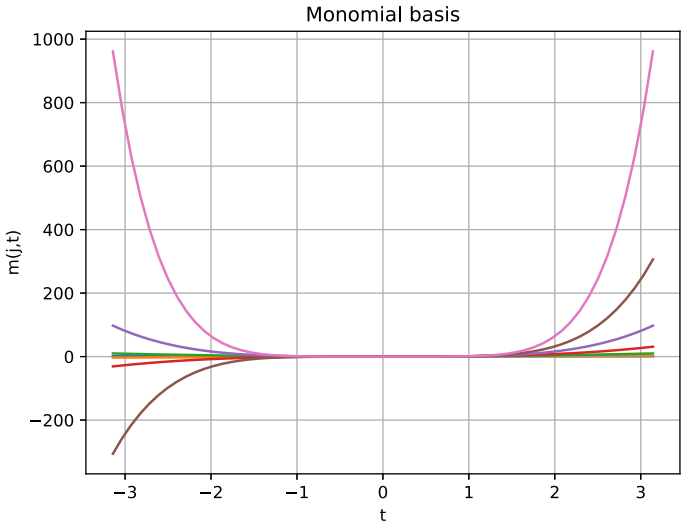


Figure 2.1. Monomial basis over interval $[-\pi, \pi]$

◦ **Algorithm (Horner's scheme)**

Input: $t \in \mathbb{R}, \boldsymbol{a} \in \mathbb{R}^{n+1}$
Output: $p(t) = a_0 + a_1 t + \cdots + a_n t^n$
$p = a_n$
for $k = n-1$ downto 0
  $p = a_k + p \cdot t$
end
return p

Lagrange form of interpolating polynomial. It is possible to reduce the operation count to find the interpolating polynomial by carrying out an LU decomposition of the monomial matrix $\boldsymbol{M}$,

$$\mathscr{M}_{n+1}(\boldsymbol{x}) = \boldsymbol{M} = \boldsymbol{L}\,\boldsymbol{U}.$$

Let $\mathscr{L}_{n+1}(t) = [\ \ell_0(t)\ \ \ell_1(t)\ \ \ldots\ \ \ell_n(t)\ ]$ denote another set of basis functions that evaluates to the identity matrix at the sample points $\boldsymbol{x}$, such that $\mathscr{L}_{n+1}(\boldsymbol{x}) = \boldsymbol{I}$,

$$\mathscr{M}_{n+1}(\boldsymbol{x}) = \boldsymbol{M} = \boldsymbol{L}\,\boldsymbol{U} = \boldsymbol{I}\boldsymbol{L}\,\boldsymbol{U} = \mathscr{L}_{n+1}(\boldsymbol{x})\,\boldsymbol{L}\,\boldsymbol{U}.$$

For arbitrary t, the relationship

$$\mathscr{M}_{n+1}(t) = \mathscr{L}_{n+1}(t)\,\boldsymbol{L}\,\boldsymbol{U},$$

describes a linear mapping between the monomials $\mathscr{M}_{n+1}(t)$ and the $\mathscr{L}_{n+1}(t)$ functions, a mapping which is invertible since $\boldsymbol{M} = \boldsymbol{L}\,\boldsymbol{U}$ is of full rank

$$\mathscr{L}_{n+1}(t) = \mathscr{M}_{n+1}(t)\,\boldsymbol{U}^{-1}\,\boldsymbol{L}^{-1}.$$

Note that organization of bases as row vectors of functions leads to linear mappings expressed through right factors.

◦ The LU factorization of the Vandermonde matrix can be determined analytically, as exemplified for $n = 3$ by

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & \dfrac{x_0 - x_2}{x_0 - x_1} & 1 & 0 \\ 1 & \dfrac{x_0 - x_3}{x_0 - x_1} & \dfrac{(x_0 - x_3)\,(x_3 - x_1)}{(x_0 - x_2)\,(x_2 - x_1)} & 1 \end{pmatrix} \begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 & x_1^3 - x_0^3 \\ 0 & 0 & (x_0 - x_2)\,(x_1 - x_2) & (x_0 - x_2)\,(x_1 - x_2)\,(x_0 + x_1 + x_2) \\ 0 & 0 & 0 & -((x_0 - x_3)\,(x_3 - x_1)\,(x_3 - x_2)) \end{pmatrix}$$

◦ Both factors can be inverted analytically, e.g., for $n = 3$,

$$\boldsymbol{L}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \dfrac{x_1 - x_2}{x_0 - x_1} & \dfrac{x_2 - x_0}{x_0 - x_1} & 1 & 0 \\ \dfrac{(x_1 - x_3)\,(x_3 - x_2)}{(x_0 - x_1)\,(x_0 - x_2)} & \dfrac{(x_0 - x_3)\,(x_2 - x_3)}{(x_0 - x_1)\,(x_1 - x_2)} & \dfrac{(x_0 - x_3)\,(x_1 - x_3)}{(x_0 - x_2)\,(x_2 - x_1)} & 1 \end{pmatrix},$$

$$\boldsymbol{U}^{-1} = \begin{pmatrix} 1 & \dfrac{x_0}{x_0 - x_1} & -\dfrac{x_0\,x_1}{(x_0 - x_2)\,(x_2 - x_1)} & \dfrac{x_0\,x_1\,x_2}{(x_0 - x_3)\,(x_3 - x_1)\,(x_3 - x_2)} \\ 0 & \dfrac{1}{x_1 - x_0} & \dfrac{x_0 + x_1}{(x_0 - x_2)\,(x_2 - x_1)} & -\dfrac{x_1\,x_2 + x_0\,(x_1 + x_2)}{(x_0 - x_3)\,(x_3 - x_1)\,(x_3 - x_2)} \\ 0 & 0 & \dfrac{1}{(x_0 - x_2)\,(x_1 - x_2)} & \dfrac{x_0 + x_1 + x_2}{(x_0 - x_3)\,(x_3 - x_1)\,(x_3 - x_2)} \\ 0 & 0 & 0 & -\dfrac{1}{(x_0 - x_3)\,(x_3 - x_1)\,(x_3 - x_2)} \end{pmatrix}.$$

- The functions that result for $n = 3$

$$\left\{ \frac{(t-x_1)(t-x_2)(t-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}, \frac{(t-x_0)(t-x_2)(t-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}, \frac{(t-x_0)(t-x_1)(t-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}, \frac{(t-x_0)(t-x_1)(t-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \right\},$$

  can be generalized as

$$\ell_i(t) = \prod_{j=0}^{n\,'} \frac{t-x_j}{x_i-x_j},$$

  known as the *Lagrange basis set*, where the prime on the product symbol skips the index $j = i$. Note that each member of the basis is a polynomial of degree n.

By construction, through the condition $\mathcal{L}_{n+1}(\boldsymbol{x}) = \boldsymbol{I}$, a Lagrange basis function evaluated at a sample point is

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

A polynomial of degree n is expressed as a linear combinations of the Lagrange basis functions by

$$p(t) = \mathcal{L}_{n+1}(t)\boldsymbol{c} = \begin{bmatrix} \ell_0(t) & \ell_1(t) & \dots & \ell_n(t) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = c_0\ell_0(t) + c_1\ell_1(t) + \dots c_n\ell_n(t).$$

The interpolant of data $\{(x_i, y_i = f(x_i)), i = 0, 1, \dots, n\}$ is determined through the conditions

$$p(\boldsymbol{x}) = \boldsymbol{y} = \mathcal{L}_{n+1}(\boldsymbol{x})\boldsymbol{c} = \boldsymbol{I}\boldsymbol{c} = \boldsymbol{c} \Longrightarrow \boldsymbol{c} = \boldsymbol{y},$$

i.e., the linear combination coefficients are simply the sampled function values $c_i = y_i = f(x_i)$.

$$p(t) = \sum_{i=0}^{n} y_i \ell_i(t) = \sum_{i=0}^{n} y_i \prod_{j=0}^{n\,'} \frac{t-x_j}{x_i-x_j}. \tag{2.7}$$

Determining the linear combination coefficients may be without cost, but evaluation of the Lagrange form (2.7) of the interpolating polynomial requires $O(n^2)$ operations, significantly more costly than the $O(n)$ operations required by Horner's scheme (2.5)

- **Algorithm (Lagrange evaluation)**

  Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
  Output: $p(t) = \sum_{i=0}^{n} y_i \prod_{j=0}^{n\,'} (t-x_j)/(x_i-x_j)$
  $p = 0$
  for $i = 0$ to n
    $w = 1$
    for $j = 0$ to n
      if $j \neq i$ then $w = w\,(t-x_j)/(x_i-x_j)$
    end
    $p = p + w \cdot y_i$
  end

return p

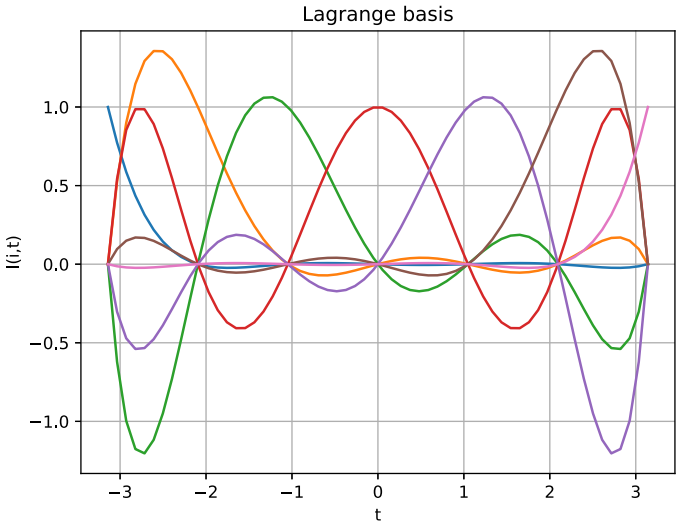


Figure 2.2. Lagrange basis for $n=6$ for $\sin(x)$ over interval $[-\pi,\pi]$

A reformulation of the Lagrange basis can however reduce the operation count. Let $\ell(t)=\prod_{k=0}^{n}(t-x_k)$, and rewrite $\ell_i(t)$ as

$$\ell_i(t)=\prod_{j=0}^{n\ '}\frac{t-x_j}{x_i-x_j}=\ell(t)\frac{w_i}{t-x_i},$$

with the weights

$$w_i=\prod_{j=0}^{n\ '}\frac{1}{x_i-x_j},$$

depending only on the function sample arguments x_i, but not on the function values y_i. The interpolating polynomial is now

$$p(t)=\sum_{i=0}^{n}y_i\ell_i(t)=\ell(t)\sum_{i=0}^{n}y_i\frac{w_i}{t-x_i}.$$

Interpolation of the function $g(t)=1$ would give

$$1=\ell(t)\sum_{i=0}^{n}\frac{w_i}{t-x_i},$$

and taking the ratio yields

$$p(t)=\frac{\sum_{i=0}^{n}y_i\dfrac{w_i}{t-x_i}}{\sum_{i=0}^{n}\dfrac{w_i}{t-x_i}},\tag{2.9}$$

known as the barycentric Lagrange formula (by analogy to computation of a center of mass). Evaluation of the weights w_i costs $O(n^2)$ operations, but can be done once for any set of x_i. The evaluation of $p(t)$ now becomes an $O(2n)$ process, comparable in cost to Horner's scheme.

◦ **Algorithm (Barycentric Lagrange evaluation)**

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
Output: $p(t) = \left(\sum_{i=0}^{n} y_i \, \frac{w_i}{t - x_i} \right) / \left(\sum_{i=0}^{n} \frac{w_i}{t - x_i} \right)$
for $i = 0$ to n
  $w_i = 1$
 for $j = 0$ to n
  if $j \neq i$ $w_i = w_i / (x_i - x_j)$
 end
end
$q = r = 0$
for $i = 0$ to n
  $s = w_i / (t - x_i)$; $q = q + y_i s$; $r = r + s$
end
$p = q / r$
return p

Newton form of interpolating polynomial. Inverting only one factor of the $\mathcal{M}_{n+1}(t) = \mathcal{L}_{n+1}(t) \, \boldsymbol{L} \boldsymbol{U}$ mapping yields yet another basis set $\mathcal{S}(t) = [\ N_0(t) \ \ N_1(t) \ \ N_2(t) \ \ \ldots\ ]$

$$\mathcal{M}_{n+1}(t) \, \boldsymbol{U}^{-1} = \mathcal{L}_{n+1}(t) \, \boldsymbol{L} = \mathcal{S}_{n+1}(t) \, .$$

◦ The first four basis polynomials are

$$\left\{ 1, \frac{t - x_0}{x_1 - x_0}, \frac{(t - x_0)(t - x_1)}{(x_2 - x_0)(x_2 - x_1)}, \frac{(t - x_0)(t - x_1)(t - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \right\},$$

with $N_0(t) = 1$, and in general

$$N_k(t) = \prod_{j=0}^{k-1} \frac{t - x_j}{x_k - x_j},$$

for $k > 0$.

Computation of the scaling factors $w_k = 1 / \prod_{j=0}^{k-1} (x_k - x_j)$ would require $O(n^2/2)$ operations, but can be avoided by redefining the basis set as $\mathcal{N}(t) = [\ n_0(t) \ \ n_1(t) \ \ n_2(t) \ \ \ldots\ ]$, with $n_0(t) = 1$, and

$$n_k(t) = \prod_{j=0}^{k-1} (t - x_j),$$

known as the *Newton basis*. As usual, the coefficients $\boldsymbol{d} \in \mathbb{R}^{n+1}$ of the linear combination of Newton polynomials

$$p(t) = \mathcal{N}_{n+1}(t) \boldsymbol{c} = [\ n_0(t) \ \ n_1(t) \ \ \ldots \ \ n_n(t)\ ] \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_n \end{bmatrix} = d_0 \, n_0(t) + d_1 \, n_1(t) + \ldots + d_n \, n_n(t),$$

are determined from the interpolation conditions $p(\boldsymbol{x}) = \boldsymbol{y}$. The resulting linear system is of triangular form,

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & x_1 - x_0 & 0 & \cdots & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & \cdots & 0 \\ 1 & x_3 - x_0 & (x_3 - x_0)(x_3 - x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & \cdots & \prod_{j=0}^{n-1} (x_n - x_j) \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

and readily solved by forward substitution.

◦ The first few coefficients are

$$d_0 = y_0,\ d_1 = \frac{y_1 - d_0}{x_1 - x_0} = \frac{y_1 - y_0}{x_1 - x_0},$$

$$d_2 = \frac{y_2 - (x_2 - x_0)d_1 - d_0}{(x_2 - x_0)(x_2 - x_1)} = \frac{y_2 - (x_2 - x_0)\dfrac{y_1 - y_0}{x_1 - x_0} - y_0}{(x_2 - x_0)(x_2 - x_1)} = \frac{\dfrac{y_2 - y_1}{x_2 - x_1} - \dfrac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}.$$

The forward substitution is efficiently expressed through the definition of divided differences

$$[y_i] = y_i, [y_{i+1}, y_i] = \frac{[y_{i+1}] - [y_i]}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, [y_{i+2}, y_{i+1}, y_i] = \frac{[y_{i+2}, y_{i+1}] - [y_{i+1}, y_i]}{x_{i+2} - x_i},$$

or in general, the k^{th} divided difference

$$[y_{i+k}, y_{i+k-1}, \ldots, y_i] = \frac{[y_{i+k}, y_{i+k-1}, \ldots, y_{i+1}] - [y_{i+k-1}, y_{i+k-1}, \ldots, y_i]}{x_{i+k} - x_i},$$

given in terms of the $(k-1)$ divided differences. The forward substitution computations are conveniently organized in a table, useful both for hand computation and also for code implementation.

i	x_i	$[y_i]$	$[y_i, y_{i-1}]$	$[y_i, y_{i-1}, y_{i-2}]$	
0	x_0	y_0	–	–	
1	x_1	y_1	$\dfrac{y_1 - y_0}{x_1 - x_0}$	–	
2	x_2	y_2	$\dfrac{y_2 - y_1}{x_2 - x_1}$	$\dfrac{[y_2, y_1] - [y_1, y_0]}{x_2 - x_0}$	$\ddots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$		
n	x_n	y_n	$\dfrac{y_n - y_{n-1}}{x_n - x_{n-1}}$	$\dfrac{[y_n, y_{n-1}] - [y_{n-1}, y_{n-2}]}{x_n - x_{n-2}}$	$\cdots$ $\dfrac{[y_n, \ldots, y_1] - [y_{n-1}, \ldots, y_0]}{x_n - x_0}$

Table 2.2. Table of divided differences. The Newton basis coefficients $\boldsymbol{d}$ are the diagonal terms.

◦ **Algorithm (Forward substitution, Newton coefficients)**

Input: $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}$
Output: $\boldsymbol{d} \in \mathbb{R}^{n+1}$

$\boldsymbol{d} = \boldsymbol{y}$
for $i = 1$ to n
  for $j = n$ downto i
    $d_j = (d_j - d_{j-1})/(x_j - x_{j-i})$
  end
end

The above algorithm requires only $O(n)$ operations, and the Newton form of the interpolating polynomial

$$p(t) = [y_0] \cdot 1 + [y_1, y_0] \cdot (t - x_0) + [y_1, y_0] \cdot (t - x_0)(t - x_1) + \cdots + + [y_n, y_{n-1}, \ldots, y_0] \cdot (t - x_0) \cdot (t - x_1) \cdot \ldots \cdot (t - x_{n-1}),$$

can also be evaluated in $O(n)$ operations

- **Algorithm (Newton polynomial evaluation)**

  Input: $\boldsymbol{x}, \boldsymbol{d} \in \mathbb{R}^{n+1}, t \in \mathbb{R}$
  Output: $p(t) = d_0 + d_1 t + \cdots + d_n t^n$
  $p = d_0$; $r = 1$
  for $k = 1$ to n
    $r = r \cdot (t - x_{k-1})$
    $p = p + d_k \cdot r$
  end
  return p

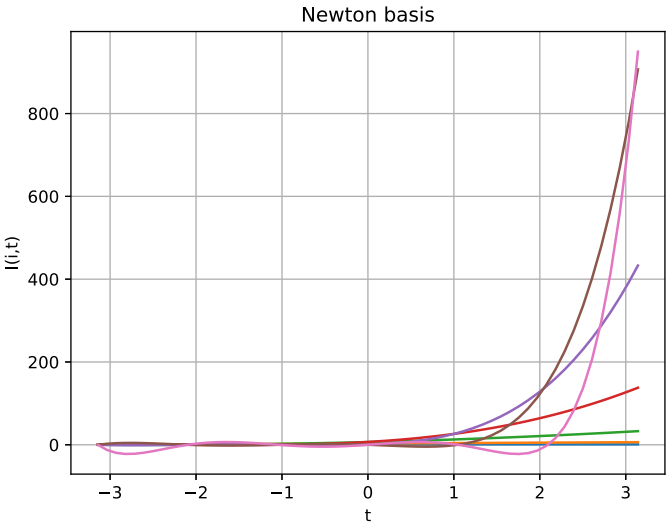


Figure 2.3. Newton basis for $n = 6$ for $\sin(x)$ over interval $[-\pi, \pi]$

3. Interpolation error

As mentioned, a polynomial interpolant of $f : \mathbb{R} \longrightarrow \mathbb{R}$ already incorporates the function values $y_i = f(x_i), i = 0, \ldots, n$, so additional information on f is required to estimate the error

$$e(t) = f(t) - p(t),$$

when t is not one of the sample points. One approach is to assume that f is smooth, $f \in C^\infty(\mathbb{R})$, in which case the error is given by

$$f(t) - p(t) = \frac{f^{(n+1)}(\xi_t)}{(n+1)!} \prod_{i=0}^{n} (t - x_i) = \frac{f^{(n+1)}(\xi_t)}{(n+1)!} w(t), \tag{2.13}$$

for some $\xi_t \in [x_0, x_n]$, assuming $x_0 < x_1 < \cdots < x_n$. The above error estimate is obtained by repeated application of Rolle's theorem to the function

$$\Phi(u) = f(u) - p(u) - \frac{f(t) - p(t)}{w(t)} w(u),$$

that has $n+1$ roots at $t, x_0, x_1, ..., x_n$, hence its $(n+1)$-order derivative must have a root in the interval (x_0, x_n), denoted by ξ_t

$$\Phi^{(n+1)}(\xi_t) = \frac{\mathrm{d}^{n+1}\Phi}{\mathrm{d}u^{n+1}}(\xi_t) = 0 = f^{(n+1)}(\xi_t) - \frac{f(t) - p(t)}{w(t)} (n+1)!,$$

establishing (2.13). Though the error estimate seems promising due to the $(n+1)!$ in the denominator, the derivative $f^{(n+1)}$ can be arbitrarily large even for a smooth function. This is the behavior that arises in the Runge function $f(t) = 1/[1 + (5t)^2]$ (Fig. 2.4), for which a typical higher-order derivative appears as

◦ $$f^{(10)} = \frac{35437500000000\,(107421875\,t^{10} - 64453125\,t^8 + 7218750\,t^6 - 206250\,t^4 + 1375\,t^2 - 1)}{(25\,t^2 + 1)^{11}}, \|f^{(10)}\|_\infty \cong 3.5 \times 10^{13}.$$

The behavior might be attributable to the presence of poles of f in the complex plane at $t_{1,2} = \pm i/5$, but the interpolant of the holomorphic function $g(t) = \exp(-(5t)^2)$, with a similar power series to f,

◦ $$f(t) \cong 1 - 25\,t^2 + 625\,t^4 - 15625\,t^6 + O(t^7),$$
$$g(t) \cong 1 - 25\,t^2 + \frac{625\,t^4}{2} - \frac{15625\,t^6}{6} + O(t^7),$$

also exhibits large errors (Fig. 2.4), and also has a high-order derivative of large norm $\|g\|_\infty \cong 3 \times 10^{11}$.

◦ $$g^{(10)}(t) = 1562500000\,e^{-25t^2}\,(62500000\,t^{10} - 56250000\,t^8 + 15750000\,t^6 - 1575000\,t^4 + 47250\,t^2 - 189),$$

◦

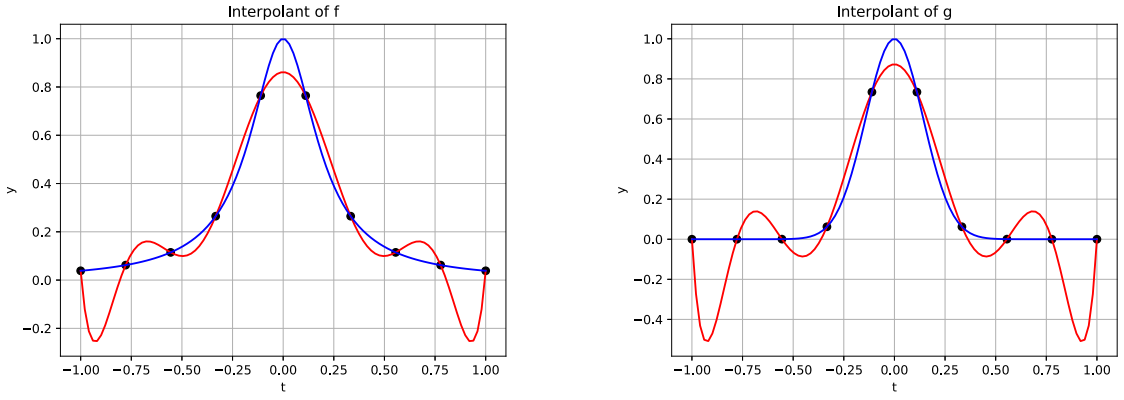


Figure 2.4. Interpolants of $f(t) = 1/[1 + (5t)^2]$, $g(t) = = \exp(-(5t)^2)$, based on equidistant sample points.

3.1. Error minimization - Chebyshev polynomials

Since $\|f^{(n+1)}\|_\infty$ is problem-specific, the remaining avenue to error control suggested by formula (2.13) is a favorable choice of the sample points x_i, $i = 0, \ldots, n$. The $w(t)$ polynomial

$$w(t) = \prod_{i=0}^{n} (t - x_i)$$

is monic (coefficient of highest power is unity), and any interval $[a, b] \subset \mathbb{R}$ can be mapped to the $[-1, 1]$ interval by $t = 2(s-a)/(b-a) - 1$, leading to consideration of the problem

$$\min_{\boldsymbol{x}} \|w\|_\infty = \min_{\boldsymbol{x}} \max_{-1 \leq t \leq 1} |w(t)|,$$

i.e., finding the sample points or roots of $w(t)$ that lead to the smallest possible inf-norm of $w(t)$. Plots of the Lagrange basis (L18, Fig. 2), or Legendre basis, suggest study of basis functions that have distinct roots in the interval $[-1, 1]$ and attain the same maximum. The trigonometric functions satisfy these criteria, and can be used to construct the Chebyshev family of polynomials through

$$T_n(x) = \cos[n \cos^{-1} x] = \cos(n\theta), \cos\theta = x, \theta = \cos^{-1} x.$$

The trigonometric identity

$$\cos[(n+1)\theta] + \cos[(n-1)\theta] = 2\cos\theta\cos(n\theta)$$

leads to a recurrence relation for the Chebyshev polynomials

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x),\ T_0(x) = 1,\ T_1(x) = x.$$

The definition in terms of the cosine function easily leads to the roots, $T_n(x_i) = 0$,

$$\cos[n\theta] = 0 \Longrightarrow n\theta_i = (2i-1)\frac{\pi}{2} \Longrightarrow \theta_i = \frac{2i-1}{2n}\pi \Longrightarrow x_i = \cos\left[\frac{2i-1}{2n}\pi\right],\ i = 1, \ldots, n,$$

and extrema x_j, $T_n(x_j) = (-1)^j$

$$\cos[n\theta] = \pm 1 \Longrightarrow n\theta_j = j\pi \Longrightarrow x_j = \cos\left[\frac{j\pi}{n}\right],\ j = 0, 1, \ldots, n.$$

The Chebyshev polynomials are not themselves monic, but can readily be rescaled through

$$P_n(x) = 2^{1-n} T_n(x),\ n > 0,\ P_0(x) = 1.$$

Since $|T_n(x)| = |\cos(n\theta)|$, the above suggests that the monic polynomials P_n have $\|P_n\|_\infty = 2^{1-n}$, small for large n, and are indeed among all possible monic polynomials defined on $[-1, 1]$ the ones with the smallest inf-norm.

THEOREM. *The monic polynomial* $p\colon [-1,1] \longrightarrow \mathbb{R}$ *has a inf-norm lower bound*

$$\|p\|_\infty = \max_{-1\leqslant t\leqslant 1} |p(t)| \geqslant 2^{1-n}.$$

Proof. *By contradiction, assume the monic polynomial* $p\colon [-1,1] \longrightarrow \mathbb{R}$ *has* $\|q\|_\infty < 2^{1-n}$. *Construct a comparison with the Chebyshev polynomials by evaluating p at the extrema* $x_j = \cos(j\pi/n)$,

$$(-1)^j p(x_j) \leqslant |p(x_j)| < 2^{1-n} = (-1)^j P_n(x_j) = (-1)^j 2^{1-n} T_n(x_j).$$

Since the above states $(-1)^j p(x_j) < (-1)^j P_n(x_j)$ *deduce*

$$(-1)^j [p(x_j) - P_n(x_j)] < 0, \text{for } j = 0, 1, \ldots, n \tag{2.14}$$

However, p, P_n *both monic implies that* $p(x_j) - P_n(x_j)$ *is a polynomial of degree* $n-1$ *that would change signs* $n+1$ *times to satisfy (2.14), and thus have n roots contradicting the fundamental theorem of algebra.* □

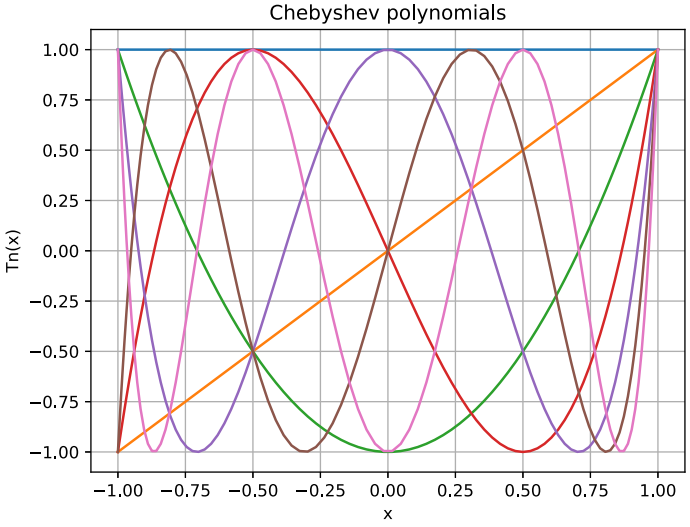


Figure 2.5. First $n = 6$ Chebyshev polynomials

3.2. Best polynomial approximant

Based on the above, the optimal choice of $n+1$ sample points is given by the roots $x_j = \cos(\theta_j)$ of the Chebyshev polynomial of $(n+1)^{\text{th}}$ degree $T_{n+1}(x)$, for which $\cos[(n+1)\theta] = 0$,

$$x_j = \cos\left[\frac{\pi}{n+1}\left(\frac{1}{2} + j\right)\right], j = 0, \ldots, n,$$

For this choice of sample points the interpolation error has the bound

$$|f(t)-p_n(t)| = \left| \frac{f^{(n+1)}(\xi_t)}{(n+1)!} \prod_{i=0}^{n} (t-x_i) \right| \leqslant \frac{|f^{(n+1)}(\xi_t)|}{(n+1)!} \|P_{n+1}\|_\infty \leqslant \frac{\|f^{(n+1)}\|_\infty}{(n+1)!\,2^n}.$$

LECTURE 15: DERIVATIVE INTERPOLATION

1. Interpolation in function and derivative values - Hermite interpolation

In addition to sampling of the function $f:\mathbb{R}\longrightarrow\mathbb{R}$, information on the derivatives of f might also be available, as in the data set

$$\mathcal{D}' = \{(x_i, y_i = f(x_i), y_i' = f'(x_i)), i = 0, 1, \ldots, n\}. \tag{2.15}$$

The extended data set can again be interpolated by a polynomial, this time of degree $2n+1$ given in the monomial, Lagrange or Newton form.

Monomial form of interpolating polynomial. Using the monomial basis

$$\mathscr{M}_{2n+1}(t) = [\ 1 \ \ t \ \ t^2 \ \ t^3 \ \ \ldots \ \ t^{2n+1}\ ],$$

the interpolating polynomial is

$$p(t) = \mathscr{M}_{2n+1}(t)\,\boldsymbol{a} = [\ 1 \ \ t \ \ \ldots \ \ t^{2n+1}\ ] \begin{bmatrix} 0 \\ a_1 \\ \vdots \\ a_{2n+1} \end{bmatrix} = a_0 + a_1 t + \cdots + a_{2n+1} t^{2n+1},$$

with derivative

$$p'(t) = a_1 + 2a_2 t + \cdots + (2n+1) a_{2n+1} t^{2n}.$$

The above suggests constructing a basis set of monomials and their derivatives

$$\mathscr{M}'_{2n+1}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 & \ldots & t^{2n+1} \\ 0 & 1 & 2t & 3t^2 & \ldots & (2n+1)\,t^{2n} \end{bmatrix},$$

to allow setting the function $p(x_i) = y_i$, and derivative conditions $p(x_i) = y_i'$. The columns of $\mathscr{M}'_{2n+1}(t)$ are linearly independent since

$$\alpha \begin{bmatrix} t^j \\ jt^{j-1} \end{bmatrix} + \beta \begin{bmatrix} t^k \\ kt^{k-1} \end{bmatrix} = 0,$$

can only be satisfied for all t by $\alpha = \beta = 0$.

- Sampling at $\boldsymbol{x} \in \mathbb{R}^{(n+1)}$ gives $\boldsymbol{M} = \mathscr{M}'_{2n+1}(\boldsymbol{x}) \in \mathbb{R}^{(2n+2)\times(2n+2)}$, e.g., for $n = 2$, the matrix is

$$\boldsymbol{M} = \begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & x_0^4 & x_0^5 \\ 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ 0 & 1 & 2x_0 & 3x_0^2 & 4x_0^3 & 5x_0^4 \\ 0 & 1 & 2x_1 & 3x_1^2 & 4x_1^3 & 5x_1^4 \\ 0 & 1 & 2x_2 & 3x_2^2 & 4x_2^3 & 5x_2^4 \end{pmatrix},$$

  is obtained.

For general n, $\boldsymbol{M}$ is of full rank for distinct sample points with a determinant reminiscent of that of the Vandermonde matrix

$$\det(\boldsymbol{M}) = \prod_{0 \le i < j \le n} (x_i - x_j)^4.$$

The interpolation conditions lead to the linear system

$$\boldsymbol{M}\boldsymbol{a} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \end{bmatrix},$$

whose solution requires $O([2(n+1)]^3/3)$ operations. An error formula is again obtained by repeated application of Rolle's theorem, i.e., for p interpolant of data set $\mathscr{D}'$, $\exists \xi_t \in (x_0, x_n)$ such that

$$f(t) - p(t) = \frac{f^{(2n+2)}(\xi_t)}{(2n+2)!} \prod_{j=0}^{n} (t - x_j)^2.$$

The above approach generalizes to higher-order derivatives, e.g., for

$$\mathscr{D}'' = \{(x_i, y_i = f(x_i), y_i' = f'(x_i), y_i'' = f''(x_i)), i = 0, 1, \ldots, n\}, \tag{2.16}$$

the basis set is

$$\mathscr{M}''_{3n+2}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 & \ldots & t^{3n+2} \\ 0 & 1 & 2t & 3t^2 & \ldots & (3n+2)\,t^{3n+1} \\ 0 & 0 & 2 & 6t & \ldots & (3n+2)(3n+1)t^{3n} \end{bmatrix},$$

with interpolant

$$p(t) = \mathscr{M}''_{3n+2}(t)\boldsymbol{a},$$

with $\boldsymbol{a} \in \mathbb{R}^{3(n+1)}$ determined by solving

$$\boldsymbol{M}\boldsymbol{a} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \\ \boldsymbol{y}'' \end{bmatrix}$$

with $\boldsymbol{M} = \mathscr{M}''_{3n+2}(\boldsymbol{x}) \in \mathbb{R}^{(3n+3)\times(3n+3)}$, and error formula

$$f(t) - p(t) = \frac{f^{(3n+3)}(\xi_t)}{(3n+3)!} \prod_{j=0}^{n} (t - x_j)^3.$$

Lagrange form of interpolating polynomial. As in the function value interpolation case, a basis set that evaluates to an identity matrix when sampled at $\boldsymbol{x} \in \mathbb{R}^{n+1}$ is obtained by LU-factorization of the sampled monomial matrix

$$\mathscr{M}'_{2n+1}(\boldsymbol{x}) = \boldsymbol{M} = \boldsymbol{L}\boldsymbol{U} = \boldsymbol{I}\boldsymbol{L}\boldsymbol{U} = \mathscr{L}'_{2n+1}(\boldsymbol{x})\,\boldsymbol{L}\boldsymbol{U},$$

that for arbitrary t leads to the basis set

$$\mathscr{L}'_{2n+1}(t) = \mathscr{M}'_{2n+1}(t)\,\boldsymbol{U}^{-1}\boldsymbol{L}^{-1} = \begin{bmatrix} a_0(t) & a_1(t) & \dots & a_n(t) & b_0(t) & b_1(t) & \dots & b_n(t) \\ a'_0(t) & a'_1(t) & \dots & a'_n(t) & b'_0(t) & b'_1(t) & \dots & b'_n(t) \end{bmatrix}.$$

The interpolating polynomial of data set $\mathscr{D}' = \{(x_i, y_i = f(x_i), y'_i = f'(x_i)), i = 0, 1, \dots, n\}$ is

$$p(t) = \sum_{i=0}^{n} y_i\, a_i(t) + \sum_{i=0}^{n} y'_i\, b_i(t),$$

where the basis functions can be expressed in terms of the Lagrange polynomials

$$\ell_i(t) = \prod_{j=0}^{n\,\prime} \frac{t - x_j}{x_i - x_j},$$

as

$$a_i(t) = [1 - 2(t - x_i)\ell'_i(x_i)]\ell_i^2(t),\; b_i(t) = (t - x_i)\ell_i^2(t),$$

and have the properties

$$a_i(x_j) = \delta_{ij},\; a'_i(x_j) = 0,\; b_i(x_j) = 0,\; b'_i(x_j) = \delta_{ij}.$$

◦ As, an example, consider the LU-factorization of matrix $\boldsymbol{M} = \mathscr{M}'_{2n+1}(\boldsymbol{x})$ for $n = 1$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & -\dfrac{1}{x_0 - x_1} & 1 & 0 \\ 0 & -\dfrac{1}{x_0 - x_1} & -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & \dfrac{1}{x_1 - x_0} & 1 & 0 \\ 0 & \dfrac{1}{x_1 - x_0} & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 0 & x_1 - x_0 & x_1^2 - x_0^2 & x_1^3 - x_0^3 \\ 0 & 0 & x_0 - x_1 & 2x_0^2 - x_1 x_0 - x_1^2 \\ 0 & 0 & 0 & (x_0 - x_1)^2 \end{pmatrix}.$$

◦ The factor inverses are

$$\boldsymbol{L}^{-1}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \dfrac{1}{x_1-x_0} & \dfrac{1}{x_0-x_1} & 1 & 0 \\ \dfrac{2}{x_1-x_0} & \dfrac{2}{x_0-x_1} & 1 & 1 \end{pmatrix}, \boldsymbol{U}^{-1}=\begin{pmatrix} 1 & \dfrac{x_0}{x_0-x_1} & \dfrac{x_0\,x_1}{x_0-x_1} & -\dfrac{x_0^2\,x_1}{(x_0-x_1)^2} \\ 0 & \dfrac{1}{x_1-x_0} & -\dfrac{x_0+x_1}{x_0-x_1} & \dfrac{x_0\,(x_0+2\,x_1)}{(x_0-x_1)^2} \\ 0 & 0 & \dfrac{1}{x_0-x_1} & -\dfrac{2\,x_0+x_1}{(x_0-x_1)^2} \\ 0 & 0 & 0 & \dfrac{1}{(x_0-x_1)^2} \end{pmatrix}$$

◦ The functions that result

$$\left\{\left[1-2(t-x_0)\frac{1}{x_0-x_1}\right]\left(\frac{t-x_1}{x_0-x_1}\right)^2,\left[1-2(t-x_1)\frac{1}{x_1-x_0}\right]\left(\frac{t-x_0}{x_1-x_0}\right)^2,(t-x_0)\left(\frac{t-x_1}{x_0-x_1}\right)^2,(t-x_1)\left(\frac{t-x_0}{x_1-x_0}\right)^2\right\},$$

are indeed expressed in terms of $\ell_i(t)$ as

$$\{[1-2(t-x_0)\ell_0'(x_0)]\ell_0^2(t),[1-2(t-x_1)\ell_1'(x_1)]\ell_0^2(t),(t-x_0)\ell_0^2(t),(t-x_1)\ell_1^2(t)\}.$$

The procedure can be extended to derivatives of arbitrary order, e.g., the data set $\mathcal{D}''$ is interpolated by

$$p(t)=\sum_{i=0}^{n} y_i\,a_i(t)+\sum_{i=0}^{n} y_i'\,b_i(t)+\sum_{i=0}^{n} y_i''\,c_i(t),$$

where the Lagrange basis polynomials are given as

$$\mathcal{L}''_{3n+2}(t)=\mathcal{M}''_{3n+2}(t)\,\boldsymbol{U}^{-1}\boldsymbol{L}^{-1}=\begin{bmatrix} a_0(t) & \dots & a_n(t) & b_0(t) & \dots & b_n(t) & c_0(t) & \dots & c_n(t) \\ a_0'(t) & \dots & a_n'(t) & b_0'(t) & \dots & b_n'(t) & c_0'(t) & \dots & c_n'(t) \\ a_0''(t) & \dots & a_n''(t) & b_0''(t) & \dots & b_n''(t) & c_0''(t) & \dots & c_n''(t) \end{bmatrix}.$$

Newton form of interpolating polynomial. As before, inverting only one factor of the $\mathcal{M}'_{2n+1}(t)=\mathcal{L}'_{2n+1}(t)\,\boldsymbol{L}\boldsymbol{U}$ mapping yields a triangular basis set $\mathcal{S}'(t)=[\,s_0(t)\;\;s_1(t)\;\;s_2(t)\;\;\dots\,]$

$$\mathcal{M}'_{2n+1}(t)\,\boldsymbol{U}^{-1}=\mathcal{S}'_{2n+1}(t)\,.$$

◦ The first six basis polynomials obtained for $n=2$ are

$$\left\{1,\frac{t-x_0}{x_1-x_0},\frac{(t-x_0)\,(t-x_1)}{(x_2-x_0)(x_2-x_1)},\frac{(t-x_0)\,(t-x_1)\,(t-x_2)}{(x_2-x_0)\,(x_1-x_0)},\frac{(t-x_0)^2\,(t-x_1)\,(t-x_2)}{(x_1-x_0)^2\,(x_1-x_2)},\frac{(t-x_0)^2\,(t-x_1)^2\,(t-x_2)}{(x_2-x_0)^2\,(x_2-x_1)^2}\right\}.$$

◦ A closer link to divided difference and differential calculus is obtained by permuting rows of $\boldsymbol{M}$, e.g., for $n=2$

$$\boldsymbol{P}\boldsymbol{M}=\begin{pmatrix} 1&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&0&1&0 \\ 0&0&1&0&0&0 \\ 0&0&0&0&0&1 \end{pmatrix}\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & x_0^4 & x_0^5 \\ 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ 0 & 1 & 2\,x_0 & 3\,x_0^2 & 4\,x_0^3 & 5\,x_0^4 \\ 0 & 1 & 2\,x_1 & 3\,x_1^2 & 4\,x_1^3 & 5\,x_1^4 \\ 0 & 1 & 2\,x_2 & 3\,x_2^2 & 4\,x_2^3 & 5\,x_2^4 \end{pmatrix}=\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & x_0^4 & x_0^5 \\ 0 & 1 & 2\,x_0 & 3\,x_0^2 & 4\,x_0^3 & 5\,x_0^4 \\ 1 & x_1 & x_1^2 & x_1^3 & x_1^4 & x_1^5 \\ 0 & 1 & 2\,x_1 & 3\,x_1^2 & 4\,x_1^3 & 5\,x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 & x_2^5 \\ 0 & 1 & 2\,x_2 & 3\,x_2^2 & 4\,x_2^3 & 5\,x_2^4 \end{pmatrix}.$$

The first six basis polynomials thus obtained are

$$\left\{1, t-x_0, \frac{(t-x_0)^2}{(x_1-x_0)^2}, \frac{(t-x_0)^2(t-x_1)}{(x_1-x_0)^2}, \frac{(t-x_0)^2(t-x_1)^2}{(x_2-x_0)^2(x_2-x_1)^2}, \frac{(t-x_0)^2(t-x_1)^2(t-x_2)}{(x_2-x_0)^2(x_2-x_1)^2}\right\}.$$

and upon rescaling generalize to the basis set

$$\mathcal{N}'_{2n+1}(t) = [\ n_0(t)\ \ n_1(t)\ \ \dots\ \ n_{2n+1}(t)\ ],$$

with

$$n_{2k}(t) = \prod_{j=0}^{k-1}(t-x_j)^2, n_{2k+1}(t) = (t-x_k)n_{2k}(t), k = 0, 1, \dots, n$$

known as the Newton basis set with repetitions.

The interpolating polynomial in Newton divided difference form is

$$p(t) = [y_0] + [y_0, y_0](t-x_0) + [y_1, y_0, y_0](t-x_0)^2 + \cdots + [y_n, y_n, \dots, y_0, y_0](t-x_0)^2 \dots (t-x_{n-1})^2(t-x_n).$$

Divided difference with repeated values are replaced by their, limits, i.e., the derivatives

$$[y_k, y_k] = \lim_{x_{k-1} \to x_k} \frac{y_k - y_{k-1}}{x_k - x_{k-1}} = y'_k.$$

The forward substitution can again be organized in a table.

i	x_i	$[y_i]$	$[y_i, y_{i-1}]$	$[y_i, y_{i-1}, y_{i-2}]$
0	x_0	y_0	–	–
0	x_0	y_0	y'_0	
1	x_1	y_1	$\frac{y_1-y_0}{x_1-x_0}$	$\frac{1}{x_1-x_0}\left(\frac{y_1-y_0}{x_1-x_0} - y'_0\right)$
1	x_1	y_1	y'_1	$\frac{1}{x_1-x_0}\left(y'_1 - \frac{y_1-y_0}{x_1-x_0}\right)$
2	x_2	y_2	$\frac{y_2-y_1}{x_2-x_1}$	$\frac{1}{x_2-x_1}\left(\frac{y_2-y_1}{x_2-x_1} - y'_1\right)$ $\quad \ddots$
2	x_2	y_2	y'_2	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
n	x_n	y_n	y'_n	$\frac{1}{x_n-x_{n-1}}\left(y'_n - \frac{y_n-y_{n-1}}{x_n-x_{n-1}}\right)$ $\dots$ $\frac{[y_n, \dots, y_1] - [y_{n-1}, \dots, y_0]}{x_n - x_0}$

Table 2.3. Table of repeated divided differences. The Newton basis coefficients are the diagonal terms.

Interpolation of data containing higher derivatives, or differing orders of derivative information at each node are poissible. For multiple repeated values arising in the limit $x_{i+k} \longrightarrow x_i$ of sample points $x_i \leqslant x_{i+1} \leqslant \cdots \leqslant x_{i+k}$ the divided difference is determined by

$$[y_{i+k}, y_{i+k-1}, \dots, y_i] = \begin{cases} \dfrac{[y_{i+k}, y_{i+k-1}, \dots, y_{i+1}] - [y_{i+k-1}, y_{i+k-1}, \dots, y_i]}{x_{i+k} - x_i} & x_i < x_{i+k} \\ \dfrac{y_i^{(k)}}{k!} & x_i = x_{i+k} \end{cases}.$$

LECTURE 16: PIECEWISE INTERPOLATION

1. Splines

Instead of adopting basis functions defined over the entire sampling interval $[x_0, x_n]$ as exemplified by the monomial or Lagrange bases, approximations of $f : \mathbb{R} \longrightarrow \mathbb{R}$ can be constructed with different branches over each subinterval, by introducing $S_i : [x_{i-1}, x_i] \longrightarrow \mathbb{R}$, and the approximation

$$p(t) = \begin{cases} S_1(t) & x_0 \leqslant t < x_1 \\ S_2(t) & x_1 \leqslant t < x_2 \\ \vdots & \vdots \\ S_n(t) & x_{n-1} \leqslant t < x_n \\ S_{n+1}(t) & t = x_n \end{cases}.$$

The interpolation conditions $p(x_i) = y_i$ lead to constraints

$$S_i(x_{i-1}) = y_{i-1}.$$

The form of $S(t)$ can be freely chosen, and though most often $S(t)$ is a low-degree polynomial, the spline functions may have any convenient form, e.g., trigonometric or arcs of circle. The accuracy of the $p(t)$ approximant is determined by the choice of form of $S(t)$, and by the sample points. It is useful to introduce a quantitative measure of the sampling through the following definitions.

DEFINITION. *$\{x_0, x_1, \ldots, x_n\}$ is a partition of the interval $[a, b] \subset \mathbb{R}$ if $x_i \in \mathbb{R}$, $i = 0, 1, \ldots, n$, satisfy*

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b.$$

DEFINITION. *The norm of partition $X = \{x_0, x_1, \ldots, x_n\}$ of the interval $[a, b] \subset \mathbb{R}$ is*

$$\|X\| = \max_{1 \leqslant i \leqslant n} |x_i - x_{i-1}|.$$

Constant splines (degree 0). A simple example is given by the constant functions $S_i(t) = y_{i-1}$. Arbitrary accuracy of the approximation can be achieved in the limit of $n \longrightarrow \infty$, $\|X\| \longrightarrow 0$. Over each subinterval the polynomial error formula gives

$$f(t) - S_i(t) = f'(\xi_t)(t - x_{i-1}),$$

so overall

$$|f(t) - p(t)| \leqslant \|f'\|_\infty \|X\|,$$

which becomes

$$|f(t) - p(t)| \leqslant \|f'\|_\infty h,$$

for equidistant partitions $x_i = x_0 + ih$, $h = (x_n - x_0)/n$. The interpolant $p(t)$ converges to $f(t)$ linearly (order of convergence is 1)

Linear splines (degree 1). A piecewise linear interpolant is obtained by

$$S_i(t) = \frac{t - x_{i-1}}{x_i - x_{i-1}}(y_i - y_{i-1}) + y_{i-1}.$$

The interpolation error is bounded by

$$|f(t) - p(t)| \leqslant \frac{1}{2}\|f'\|_\infty h^2,$$

for an equidistant partition, exhibiting quadratic convergence.

Quadratic splines (degree 2). A piecewise quadratic interpolant is formulated as

$$S_i(t) = b_i(t - x_{i-1})^2 + c_i(t - x_{i-1}) + y_{i-1}.$$

The interpolation conditions are met since $S_i(x_{i-1}) = y_{i-1}$. The additional parameters of this higher order spline interpolant can be determined by enforcing additional conditions, typically continuity of function and derivative at the boundary between two subintervals

$$\begin{array}{ll} S_i(x_i) = b_i h_i^2 + c_i h_i = y_i, & i = 1, 2, \ldots, n \\ S_i'(x_i) = 2b_i h_i + c_i = 2b_{i+1}h_{i+1} + c_{i+1} = S_{i+1}'(x_i) & i = 1, 2, \ldots, n-1 \end{array}.$$

An additional condition is required to close the system, for example $S_n'(x_i) = y_n'$ (known end slope), or $S_n'(x_i) = 0$ (zero end slope), or $S_n'(x_i) = S_n'(x_{i-1})$ (constant end-slope). The coefficients b_i, c_i are conveniently determined by observing that $S_i'(t)$ is linear over interval $[x_{i-1}, x_i]$ of length $h_i = x_i - x_{i-1}$, and is given by

$$S_i'(t) = \frac{t - x_{i-1}}{h_i}(s_i - s_{i-1}) + s_{i-1} = \frac{s_{i-1}}{h_i}(x_i - t) + \frac{s_i}{h_i}(t - x_{i-1}),$$

with $s_i = y_i'$, the slope of the interpolant at x_i. The continuity of first derivative conditions $S_i'(x_i) = S_{i+1}'(x_i)$ are satisfied, and integration gives

$$S_i(t) = \frac{s_i}{2h_i}(t - x_{i-1})^2 - \frac{s_{i-1}}{2h_i}(x_i - t)^2 + A_i.$$

The interpolation condition $S_i(x_{i-1}) = y_{i-1}$, determines the constant of integration A_i

$$A_i - \frac{s_{i-1}h_i}{2} = y_{i-1} \Longrightarrow A_i = y_{i-1} + \frac{s_{i-1}h_i}{2},$$

Imposing the continuity of function condition $S_i(x_i) = S_{i+1}(x_i)$ gives

$$\frac{s_i h_i}{2} + y_{i-1} + \frac{s_{i-1}h_i}{2} = -\frac{s_i h_{i+1}}{2} + y_i + \frac{s_i h_{i+1}}{2},$$

or

$$s_{i-1} + s_i = \frac{2}{h_i}(y_i - y_{i-1}), i = 1, 2, \ldots, n,$$

a bidiagonal system for the slopes that is solved by backward substituion in $O(2n)$ operations. For $i = 1$, the s_0 value arising in the system has to be given by an end condition, and the overall system $\boldsymbol{Bs} = \boldsymbol{d}$ is defined by

$$\boldsymbol{B} = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \end{bmatrix}, \boldsymbol{d} = \begin{bmatrix} \frac{2}{h_1}(y_1 - y_0) - s_0 \\ \frac{2}{h_2}(y_2 - y_1) \\ \vdots \\ \frac{2}{h_n}(y_n - y_{n-1}) \end{bmatrix}, \boldsymbol{s} \in \mathbb{R}^n, \boldsymbol{B} \in \mathbb{R}^{n \times n}.$$

The interpolation error is bounded by

$$|f(t) - p(t)| \leqslant \frac{1}{2}\|f'\|_\infty h^2,$$

for an equidistant partition, exhibiting quadratic convergence.

Cubic splines (degree 3). The approach outlined above can be extended to cubic splines, of special interest since continuity of curvature is achieved at the nodes, a desirable feature in many applications. The second derivative is linear

$$S_i''(t) = \frac{z_{i-1}}{h_i}(x_i - t) + \frac{z_i}{h_i}(t - x_{i-1}),$$

with $z_{i-1} = S_i''(x_{i-1})$, $z_i = S_i''(x_i)$ the curvature at the endpoints of the $[x_{i-1}, x_i]$ subinterval. Double integration gives

$$S_i(t) = \frac{z_{i-1}}{6h_i}(x_i - t)^3 + \frac{z_i}{6h_i}(t - x_{i-1})^3 + A_i(t - x_{i-1}) + B_i(x_i - t).$$

The interpolation conditions $S_i(x_{i-1}) = y_{i-1}$, $S_{i-1}(x_i) = y_i$, gives the integration constants

$$A_i = \frac{y_i}{h_i} - \frac{z_i h_i}{6}, B_i = \frac{y_{i-1}}{h_i} - \frac{z_{i-1} h_i}{6}$$

and continuity of first derivative, $S_i'(x_i) = S_{i+1}'(x_i)$, subsequently leads to a tridiagonal system for the curvatures

$$h_i z_{i-1} + 2(h_i + h_{i-1})z_i + h_{i+1} z_{i+1} = \frac{6(y_{i+1} - y_i)}{h_{i+1}} - \frac{6(y_i - y_{i-1})}{h_i}, i = 1, 2, \ldots, n-1.$$

End conditions are required to close the system. Common choices include:

1. Zero end-curvature, also known as the natural end conditions: $z_0 = z_n = 0$.

2. Curvature extrapolation: $z_0 = z_1$, $z_n = z_{n-1}$

3. Analytical end conditions given by the function curvature: $z_0 = f''(x_0)$, $z_n = f''(x_n)$.

1.1. *B*-splines

The above analytical approach becomes increasingly unwieldy for higher degree piecewise polynomials. An alternative approach is to systematically generate basis sets of desired polynomial degree over each subinterval. The starting point in this basis-spline ($B-$spline) approach is the piecewise constant functions

$$B_{j,0}(t) = \begin{cases} 1 & x_j \leqslant t < x_{j+1} \\ 0 & \text{otherwise} \end{cases},$$

leading to the interpolant

$$f(t) \cong p(t) = \sum_{j=0}^{n} y_j B_{j,0}(t), \tag{2.17}$$

of $f : \mathbb{R} \longrightarrow \mathbb{R}$, as sampled by data set $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0, 1, \ldots, n\}$, $a = x_0 < x_1 < \cdots < x_n = b$. The set

$$\mathscr{B}_0(t; \boldsymbol{x}) = \{B_{0,0}(t), B_{1,0}(t), \ldots, B_{n,0}(t)\}$$

constitutes a basis for all piecewise constant approximants of real functions on the interval $[x_0, x_n]$. Higher degree basis sets $\mathscr{B}_k(t; \boldsymbol{x})$, $k > 0$, are defined recursively through

$$B_{j,k}(t) = w_{j,k}(t) B_{j,k-1}(t) + (1 - w_{j+1,k}(t)) B_{j+1,k-1}(t),$$

with the weight function

$$w_{j,k}(t) = \frac{t - x_j}{x_{j+k} - x_j}.$$

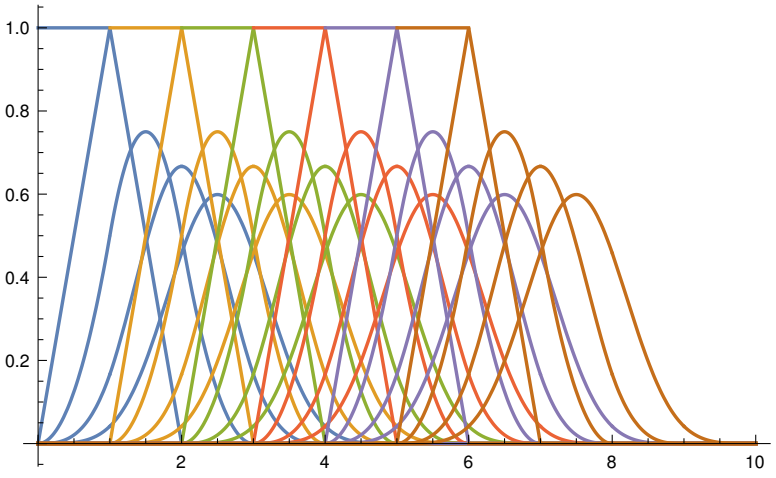


Figure 2.6. *B*-spline sets $\mathscr{B}_0, \mathscr{B}_1, \mathscr{B}_2, \mathscr{B}_3, \mathscr{B}_4$ with $\boldsymbol{x} = [\ 0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ]$

As the degree k increases, the support of $B_{j,k}(t)$ increases to the interval $[x_j, x_{j+k+1}]$. This is the B-spline analog of the additional end conditions in traditional spline formulations, and leads to the set

$$\mathcal{B}_k(t;\boldsymbol{x}) = \{B_{0,k}(t), B_{1,k}(t), \ldots, B_{n,k}(t)\}$$

defining a basis for splines of degree k only on a subinterval within $[x_0, x_n]$. Consider the piecewise linear case $k=1$, (Fig. 2.7). The set $\mathcal{B}_1$ forms a basis for piecewise linear functions if over each subinterval $[x_j, x_{j+1}]$ an arbitrary linear function $S_1(t)$ can be expressed as a linear combination

$$S_1(t) = a + bt = \sum_{i=0}^{n} c_i B_{i,1}(t).$$

Over $[x_j, x_{j+1}]$ only $B_{j-1,1}(t), B_j(t)$ are not identically zero, hence

$$S_1(t) = c_{j-1} B_{j-1,1}(t) + c_j B_{j,1}(t).$$

For the end interval $[x_0, x_1]$, a definition of $B_{-1,1}(t)$ would be required,

$$S_1(t) = c_{-1} B_{-1,1}(t) + c_0 B_{0,1}(t),$$

not available within the chosen $\boldsymbol{x}$ data set. At the other end interval $[x_{n-1}, x_n]$,

$$S_1(t) = c_{n-1} B_{n-1,1}(t) + c_n B_{n,1}(t),$$

invokes $B_{n,1}$ which requires $B_{n+1,0}(t)$, again not available within the chosen data set. One can either include samples outside the $[a, b]$ interval or restrict the spline domain of definition. Again, this is analogous with the treatment of end conditions in traditional splines:

1. Sampling outside of the $[a, b]$ range seeks additional information on the function being interpolated f, as for instance imposed by the condition $S'(a) = f(a)$ in traditional splines;

2. Restricting the definition domain corresponds to inferring information on the behavior of f in the end intervals as in the condition $S'(x_0) = S'(x_1)$ in traditional splines.

Denote by $\mathcal{S}_k(t;\boldsymbol{x})$ the set of splines $S\colon [x_0, x_n] \longrightarrow \mathbb{R}$, that are piecewise polynomials of degree k on the partition $\boldsymbol{x}$ of $[x_0, x_n]$. The $k=0$, piecewise constant interpolant (2.17) is specified by $n+1$ coefficients, the components of $\boldsymbol{y} \in \mathbb{R}^{n+1}$, hence

$$\dim \mathcal{S}_0(t;\boldsymbol{x}) = n+1,$$

i.e., the dimension of the space of piecewise-constant splines is equal to the number of sample points. As the degree k increases, additional end conditions are required to specify a spline interpolation and

$$\dim \mathcal{S}_k(t;\boldsymbol{x}) = n+1+k,$$

requiring a basis set

$$\mathcal{B}_.(t;\boldsymbol{x}) = \{B_{-k,k}(t), \ldots, B_{0,k}(t), B_{1,k}(t), \ldots, B_{n,k}(t)\}.$$

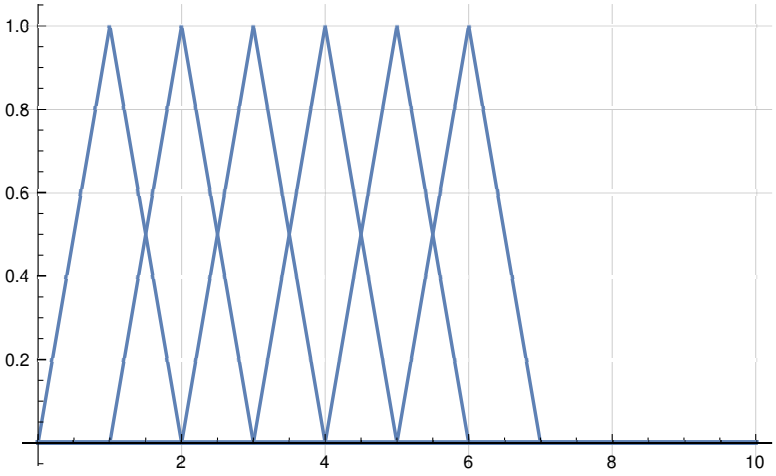


Figure 2.7. B-spline set $\mathcal{B}_1$ for $\boldsymbol{x} = [\ 0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ]$

- **Algorithm B-spline evaluation (inefficient, does not account for known zero values of B)**

  Input: $K \in \mathbb{N}$, $\boldsymbol{t} \in \mathbb{R}^m$, $\boldsymbol{x} \in \mathbb{R}^{k+n+1}$
  $\boldsymbol{B} = \boldsymbol{0} \in \mathbb{R}^{m \times (k+n+1)}$
  for $i = 1{:}\,m$
   for $j = 1{:}\,k+n$
    if $x_j \leqslant t_i < x_{j+1}$ then $B[i,j] = 1$ end
   end
   if $t_i \approx x_{k+n+1}$ then $B[i, k+n+1] = 1$ end
  end
  for $k = 1{:}\,K$
   for $j = 1{:}\,k+n$
    $w = (t - x_j)/(x_{j+k} - x_j)$
    $B[:,j] = wB[:,j] + (1-w)B[:,j+1]$
   end
  end
  return $\boldsymbol{B}$

A B-spline interpolant of degree k is given by a linear combination of the basis set $\mathcal{B}_k(t;\boldsymbol{x})$

$$f(t) \cong p_k(t) = \sum_{j=-k}^{n} c_i B_{j,k}(t)\,.$$

- The interpolation conditions $y_i = p(x_i)$ lead to an underdetermined linear system for $k > 0$

  $$\boldsymbol{Bc} = \boldsymbol{y}, \boldsymbol{B} = [\ B_{-k,k}(\boldsymbol{x})\ \ \ldots\ \ B_{0,k}(\boldsymbol{x})\ \ \ldots\ \ B_{n,k}(\boldsymbol{x})\ ] \in \mathbb{R}^{(n+1)\times(k+n+1)},$$

  analogous to the k degrees of freedom in specification of end conditions for $\mathcal{S}_k(\boldsymbol{x})$.

LECTURE 17: SPECTRAL APPROXIMATIONS

1. Trigonometric basis

The monomial basis $\{1, t, t^2, \ldots\}$ for the vector space of all polynomials $P(\mathbb{R})$, and its derivatives (Lagrange, Newton, $B-$spline) allow the definition of an approximant $p \in P(\mathbb{R})$ for real functions $f: \mathbb{R} \longrightarrow \mathbb{R}$, e.g., for smooth functions $f \in C^\infty(\mathbb{R})$. A different approach to approximation in infinite-dimensional vector spaces such as $P(\mathbb{R})$ or $C^\infty(\mathbb{R})$ is to endow the vector space with a scalar product (f, g) and associated norm $\|f\| = (f, f)^{1/2}$. The availability of a norm allows definition of convergence of sequences and series.

DEFINITION. *A sequence $\{f_n\}_{n\in\mathbb{N}}$ of elements of the normed vector space $\mathcal{X} = (F, \mathbb{C}, +, \cdot)$ converges to f, $f_n \longrightarrow f$ if $\forall \varepsilon > 0$, $\exists N(\varepsilon)$ such that $\|f_n - f\| < \varepsilon$ for all $n > N(\varepsilon)$.*

DEFINITION. *The vector space $\mathcal{X} = (F, \mathbb{C}, +, \cdot)$ with a scalar product $(,): F \times F \longrightarrow \mathbb{C}$ is a Hilbert space if the limit of all Cauchy sequences is an element of F.*

All Hilbert spaces have orthonormal bases, and of special interest are bases that arise Sturm-Liouville problems of relevance to the approximation task.

1.1. Fourier series - Fast Fourier transform

The $L^2([0, 2\pi])$ space of periodic, square-integrable functions is a Hilbert space (L^2 is the only Hilbert space among the L^p function spaces), and has a basis

$$\left\{\frac{1}{2}, \cos t, \sin t, \ldots, \cos kt, \sin kt, \ldots\right\}$$

that is orthonormal with respect to the scalar product

$$(f, g) = \frac{1}{\pi}\int_0^{2\pi} f(t)\,\overline{g(t)}\,\mathrm{d}t.$$

An element $f \in L^2([0, 2\pi])$ can be expressed as the linear combination

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos kt + b_k \sin kt].$$

An alternative orthonormal basis is formed by the exponentials

$$\{e^{\pm int}\}, n \in \mathbb{N},$$

with respect to the scalar product

$$(f, g) = \frac{1}{2\pi}\int_0^{2\pi} f(t)\,\overline{g(t)}\,\mathrm{d}t.$$

The partial sum

$$S_N f(t) = \sum_{k=-N}^{N} c_k e^{ikt}$$

has coefficients c_k determined by projection

$$c_k = (f, e^{ikt}) = \frac{1}{2\pi}\int_0^{2\pi} f(t)\, e^{-ikt}\, \mathrm{d}t,$$

that can be approximated by the Darboux sum on the partition $t_j = 2\pi j/N$

$$c_k \cong \frac{1}{N}\sum_{j=1}^{N} f_j\, e^{-ikt_j} = \frac{1}{N}\sum_{j=1}^{N} f_j \omega_N^{-jk}$$

with

$$\omega = \exp\left[\frac{2\pi i}{N}\right],$$

denoting the N^{th} root of unity. The Fourier coefficients are obtained through a linear mapping

$$\boldsymbol{c} = \boldsymbol{W}\boldsymbol{f},$$

with $\boldsymbol{c}, \boldsymbol{f} \in \mathbb{C}^N$, and $\boldsymbol{W} \in \mathbb{C}^{N\times N}$ with elements

$$\boldsymbol{W} = \left[\omega^{-jk}\right]_{1\leq j,k\leq N}.$$

The above discrete Fourier transform can be seen as a change of basis from the basis $\boldsymbol{I}$ in which the coefficients of f are $\boldsymbol{c}$ to the basis $\boldsymbol{W}$ in which the coefficients are $\boldsymbol{f}$.

1.2. Fast Fourier transform

Carrying out the matrix vector product $\boldsymbol{W}\boldsymbol{f}$ directly would require $O(N^2)$ operations, but the cyclic structure of the $\boldsymbol{W}$ matrix arising from the exponentiation of ω can be exploited to reduce the computational effort. Assume $N = 2P$ and separate even and odd indexed components of $\boldsymbol{f}$

$$c_k = \sum_{j=1}^{N} f_j \omega_N^{-jk} = \sum_{j=1}^{P}\left[f_{2j-1}\omega_N^{-(2j-1)k} + f_{2j}\omega_N^{-2jk}\right] = \sum_{j=1}^{P} f_{2j}\omega_P^{-jk} + \omega^k \sum_{j=1}^{P} f_{2j-1}\omega_P^{-jk}.$$

Through the above, the $O(N^2)$ matrix-vector product is reduced to two smaller matrix-vector products, each requiring $O(N^2/4)$ operations. For $N = 2^q$, recursion of the above procedure reduces the overall operation count to $O(qN)$, or in general for N composed of a small numer of prime factors, $O(N \log N)$. The overall algorithm is known as the fast Fourier transform or FFT.

1.3. Data-sparse matrices from Sturm-Liouville problems

One step of the FFT can be understood as a special matrix factorization

$$\boldsymbol{W}_N = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{D}_N \\ \boldsymbol{I} & -\boldsymbol{D}_N \end{bmatrix}\begin{bmatrix} \boldsymbol{W}_P & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_P \end{bmatrix}\boldsymbol{P}_N$$

where $\boldsymbol{D}_N$ is diagonal and $\boldsymbol{P}_N$ is the even-odd permutation matrix. Though the matrix $\boldsymbol{W}_N$ is full (all elements are non-zero), its factors are sparse, with many zero elements. The matrix $\boldsymbol{W}_N$ is said to be *data sparse*, in the sense that its specification requires many fewer than N^2 numbers. Other examples of data sparse matrices include:

Toeplitz matrices. $\boldsymbol{A} \in \mathbb{C}^{m\times m}$ has constant diagonal terms, e.g., for $m = 4$

$$\boldsymbol{A} = \begin{bmatrix} a & b & c & d \\ e & a & b & c \\ f & e & a & b \\ g & f & e & a \end{bmatrix},$$

or in general the elements of $\boldsymbol{A} = [a_{ij}]_{1\leqslant i,j\leqslant m}$ can be specified in terms of $2m-1$ numbers $a_{1-n},\ldots,a_{n-1}$ through $a_{ij} = a_{i-j}$.

Exterior products. Rank-1 updates arising in the singular value or eigenvalue decompositions have the form

$$\boldsymbol{A} = \boldsymbol{u}\boldsymbol{v}^T = \begin{bmatrix} v_1\boldsymbol{u} & v_2\boldsymbol{u} & \ldots & v_m\boldsymbol{u} \end{bmatrix},$$

and the $2m$ components of $\boldsymbol{u}, \boldsymbol{v}$ are suficient to specify the matrix $\boldsymbol{A}$ with m^2 components. This can be generalized to any exterior product of matrices $\boldsymbol{B} \in \mathbb{C}^{n\times n}$, $\boldsymbol{C} \in \mathbb{C}^{p\times p}$ through

$$\boldsymbol{A} = \boldsymbol{B} \otimes \boldsymbol{C} = \begin{bmatrix} b_1 \otimes \boldsymbol{C} & b_2 \otimes \boldsymbol{C} & \ldots & b_n \otimes \boldsymbol{C} \end{bmatrix} = \begin{bmatrix} b_{11}\boldsymbol{C} & b_{12}\boldsymbol{C} & \ldots & b_{1n}\boldsymbol{C} \\ b_{21}\boldsymbol{C} & b_{22}\boldsymbol{C} & \ldots & b_{2n}\boldsymbol{C} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1}\boldsymbol{C} & b_{n2}\boldsymbol{C} & \ldots & b_{nn}\boldsymbol{C} \end{bmatrix}.$$

The $m^2 = (np)^2$ components of $\boldsymbol{A}$ are specified through only $n^2 + p^2$ components of $\boldsymbol{B}, \boldsymbol{C}$.

The relevance to approximation of functions typically arises due basis sets that are solutions to Sturm-Liouville problems. In the case of the Fourier transform $e^{\pm ikt}$ are eigenfunctions of the Sturm-Liouville problem

$$w'' + \lambda w = 0,\ w = u + iv,\ u'(0) = u'(\pi) = 0,\ v(0) = v(\pi) = 0,$$

with eigenvalues $\lambda_n = k^2$. The solution set $\{\varphi_1, \varphi_2, \ldots\}$ to a general Sturm-Liouville problem to find $f : [a, b] \longrightarrow \mathbb{R}$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[p(t)\frac{\mathrm{d}f}{\mathrm{d}t}\right] + q(t)\, f = -\lambda w(t)\, f,$$

form an orthonormal basis under the scalar product

$$(f, g) = \int_a^b f(t)\, g(t)\, w(t)\, \mathrm{d}t,$$

and approximations of the form

$$\Phi_N f(t) = \sum_{k=1}^{N} c_k \varphi_k(t),$$

and Parseval's theorem states that

$$\|\boldsymbol{c}\|_2^2 = \sum_{k=1}^{\infty} c_k \bar{c}_k = \|f\|_2^2 = (f,f) = \int_a^b f(t)\, f(t)\, w(t)\, \mathrm{d}t,$$

read as an equality between the energy of f and that of $\boldsymbol{c}$. By analogy to the finite-dimensional case, the Fourier transform is unitary in that it preserves lengths in the $\|f\| + (f,f)^{1/2}$ norm with weight function $w(t) = 1$.

2. Wavelet approximations

The bases $\{\varphi_1, \varphi_2, \dots\}$ arising from Sturm-Liouville problems are single-indexed, giving functions of increasing resolution over the entire definition domain. For example $\sin kx$ resolves ever finer features over $[0, 2\pi]$. When applied to a function with localized features, k must be increased with increased resolution in the entire $[0, 2\pi]$ domain. This leads to uneconomical approximation series $S_N f(t)$ with many terms, as exemplified by the Gibbs phenomenon in approximation of a step function, $f(t) = H(t-\pi/2) - H(t-3\pi/2)$ for $t \in [0, 2\pi]$, and $f(t+2\pi) = f(t)$. The approach can be represented as the decomposition of a space of functions by the direct sum

$$F = \Phi_1 \oplus \Phi_2 \oplus \dots,$$

with $\Phi_k = \mathrm{span}(\varphi_k)$, for example

$$L^2 = E_0 \oplus E_1 \oplus E_{-1} \oplus E_2 \oplus E_{-2} \oplus \dots,$$

with $E_k = \mathrm{span}\{e^{ikt}\}$ for the Fourier series.

Approximation of functions with localized features is more efficiently accomplished by choosing some generating function $\psi(t)$ and then defining a set of functions through translation and scaling, say

$$\psi_{jk}(t) = 2^{-j/2}\, \psi(2^{-j} t - k).$$

Such systems are known as *wavelets*, and the simplest example is the step function

$$\psi(t) = \begin{cases} 1 & 0 \leqslant t < 1/2 \\ -1 & 1/2 \leqslant t < 1 \\ 0 & \text{otherwise} \end{cases},$$

with ψ_{jk} having support on the half-open interval $h_{jk} = [k2^{-j}, (k+1)2^{-j})$. The set $\{\psi_{00}, \psi_{01}, \dots\}$ is known as an Haar orthonormal basis for $L^2(\mathbb{R})$ since

$$(\psi_{jk}, \psi_{lm}) = \int_{-\infty}^{\infty} \psi_{jk}(t)\, \psi_{lm}(t)\, \mathrm{d}t = \delta_{jl}\delta_{km}.$$

Approximations based upon a wavelet basis

$$f(t) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} (f, \psi_{jk})\, \psi_{jk}(t),$$

allow identification of localized features in f.

The costly evaluation of scalar products (f, ψ_{jk}) in the double summation can be avoided by a reformulation of the expansion as

$$f(t) = \sum_k c_{l,k}\, \varphi_l(t) + \sum_{j \leq l} \sum_k d_{j,k}\, \psi_{jk}(t), \tag{2.22}$$

with . In addition to the ψ ("mother" wavelet), an auxilliary φ scaling function ("father" wavelet) is defined, for example

$$\varphi(t) = \begin{cases} 1 & 0 \leqslant t < 1 \\ 0 & \text{otherwise} \end{cases},$$

for the Haar wavelet system.

The above approach is known as a *multiresolution* representation and is based upon a hierarchical decomposition of the space of functions, e.g.,

$$L^2 = V_l \oplus W_l \oplus W_{l-1} \oplus W_{l-2} \oplus \ldots$$

with

$$V_j = \text{span}\,\{\varphi_{jk} |\, k \in \mathbb{Z}\}, W_j = \text{span}\,\{\psi_{jk} \,|\, k \in \mathbb{Z}\}.$$

The hierarchical decomposition is based upon the vector subspace inclusions

$$\{0\} < \cdots < V_1 < V_0 < V_{-1} < V_{-2} < \cdots < L^2(\mathbb{R}),$$

and the relations

$$V_m \oplus W_m = V_{m-1},$$

that state that the orthogonal complement of V_m within V_{m-1} is W_m. Analogous to the FFT, a fast wavelet transformation can be defined to compute coefficients of (2.22).

LECTURE 18: BEST APPROXIMANT

1. Best approximants

Interpolation of data $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0, \ldots, n\}$ by an approximant $p(t)$ corresponds to the minimization problem

$$\min_p \|f - p\|,$$

in the discrete one-norm at the sample points x_i

$$\|f\| = \|\boldsymbol{f}\|_1 = \sum_{i=0}^{n} |f(x_i)|.$$

Different approximants are obtained upon changing the norm.

THEOREM (EXISTENCE OF BEST APPROXIMANT. *For any element $f \in F$ in a normed vector space $\mathcal{F} = (F, S, +, \cdot)$, there exists a best approximant $g \in G$ within a finite dimensional subspace $G \subset F$ that is a solution of*

$$\min_{g \in G} \|f - g\|.$$

The argument underlying the above theorem is based upon constructing the closed and bounded subset of G

$$K = \{g \in G \mid \|g - f\| \leqslant \|0 - f\| = \|f\|\} \subset G.$$

Since G is finite dimensional, K is compact, and the continuous mapping $g \longrightarrow \|g - f\|$ attains is extrema.

The two main classes of approximants g of real functions $f : [a, b] \longrightarrow \mathbb{R}$ that arise are:

Approximants based upon sampling. The vectors $\boldsymbol{f} = f(\boldsymbol{x}), \boldsymbol{g} = g(\boldsymbol{x})$ are constructed at sample points $\boldsymbol{x} \in \mathbb{R}^m$ and the best approximant solves the problem

$$\min_{g \in G} \|\boldsymbol{f} - \boldsymbol{g}\|.$$

Note that the minimization is carried out over the members of the subset G, not over the vectors $\boldsymbol{g}$. The norm can include information on derivatives as in the norm

$$\|f\|_H = \|\boldsymbol{f}\|_1 + \|\boldsymbol{f}'\|_1,$$

arising in Hermite interpolation.

Approximants over the function domain. The norm is now expressed through an integral such as the p-norms

$$\|f\|_p = \left(\int_a^b |f(t)|^p \, \mathrm{d}t \right)^{1/p}.$$

In general, the best approximant in a normed space is not unique. However, the best approximant is unique in a Hilbert space, and is further characterized by orthogonality of the residual to the approximation subspace.

THEOREM (BEST APPROXIMANT IN HILBERT SPACE). *For any element $f \in F$ in a Hilbert space $\mathcal{F} = (F, S, +, \cdot)$, there exists a unique approximant $g \in G$ within a finite dimensional subspace $G \subset F$ that is a solution of*

$$\min_{g \in G} \|f - g\|,$$

and the residual $f - g$ is orthogonal to G, $\forall h \in G$

$$(f - g, h) = 0.$$

Note that orthogonality of the residual $(f - g, h) = 0$ implies $(f, h) = (g, h)$ or that the best approximant is the projection of f onto G.

2. Two-norm approximants in Hilbert spaces

For Hilbert spaces with a norm is induced by the scalar product

$$\|f\| = (f, f)^{1/2},$$

finding the best approximant reduces to a problem within $\mathbb{R}^m$ (or $\mathbb{C}^m$). Introduce a basis $\mathcal{B} = \{b_1, b_2, \dots\}$ for $\mathcal{F}$ such that any $f \in F$ has an expansion

$$f(t) = \sum_{j=1}^{\infty} f_j b_j(t), f_j = (f, b_j)$$

Since G is finite dimensional, say $n = \dim(G)$, an approximant has expansion

$$g(t) = \sum_{j=1}^{n} g_j b_{s(j)}(t).$$

Note that the approximation may lie in an arbitrary finite-dimensional subspace of $\mathcal{F}$. Choosing the appropriate subset through the function $s: \mathbb{N} \longrightarrow \mathbb{N}$ is an interesting problem in itself, leading to the goal of selecting those basis functions that capture the largest components of f, i.e., the solution of

$$\min_{s \in \mathbb{N}^n} \sum_{j=1}^{n} |(f, b_{s(j)})|.$$

Approximate solutions of the basis component selection are obtained by processes such as greedy approximation or clustering algorithms. The approach typically adopted is to exploit the Bessel inequality

$$\sum_{i=1}^{n} f_{s(i)}^2 \leq \|f\|^2,$$

and select

$$s(1) = \arg\max_{i \in S} f_i^2,$$

eliminate $s(1)$ from S, and search again. The k^{th}-step is

$$s(k) = \arg\max_{i \in S} f_i^2,$$

with $S_k = S - \{s(1), \dots, s(k-1)\}$.

Assuming $s(j) = j$, the orthogonality relation $f - g \perp G$ leads to a linear system

$$(f - g, b_i) = 0 \Longrightarrow \left(\sum_{j=1}^{n} g_j b_j, b_i \right) = \sum_{j=1}^{n} (b_i, b_j)\, g_j = (f, b_i) \Longrightarrow \boldsymbol{B}\boldsymbol{g} = \boldsymbol{f}.$$

If the basis is orthonormal, then $\boldsymbol{B} = \boldsymbol{I}$, and the best approximant is simply given by the projection of f onto the basis elements. Note that the scalar product need not be the Euclidean discrete or continuous versions

$$(f, g) = \sum_{i=1}^{n} f_i g_i, (f, g) = \int_a^b f(t)\, g(t)\, \mathrm{d}t.$$

A weighting function may be present as in

$$(f, g) = \boldsymbol{f}^T \boldsymbol{W}\boldsymbol{g}, (f, g) = \int_a^b f(t)\, g(t)\, w(t) \mathrm{d}t,$$

discrete and continuous versions, respectively. In essense the appropriate measure $\mu(t)$ for some specific problem

$$\mathrm{d}\mu(t) = w(t)\, \mathrm{d}\, t,$$

arises and might not be the Euclidean measure $w(t) = 1$.

3. Inf-norm approximants

In the vector space of continuous functions defined on a topological space X (e.g., a closed and bounded set in $\mathbb{R}^n$), a norm can be defined by

$$\|f\| = \max_{x \in X} |f(x)|,$$

and the best approximant is found by solving the problem

$$\inf_{g \in G} \|f - g\| = \inf_{g \in G} \max_{x \in X} |f(x) - g(x)|.$$

The fact that g is the best approximant of f can be restated as 0 being the approximant of $f - g$ since

$$\|f - g - 0\| \leqslant \|f - (g + h)\|.$$

A key role is played by the points where $f(x) = g(x)$ leading to the definition of a critical set as

$$\mathrm{crit}(f) = \mathcal{Z}(f) = \{x \in X : |f(x)| = \|f\|\}.$$

When $G = P_{n-1}$, the space of polynomials of degree at most $n - 1$, with $\dim P_{n-1} = n$, the best approximant can be charaterized by the number of sign changes of $f(x) - g(x)$.

THEOREM (CHEBYSHEV ALTERNATION). *The polynomial $p \in P_{n-1}$ is the best approximant of $f : [a, b] \longrightarrow \mathbb{R}$ in the inf-norm*

$$\|f - p\|_\infty = \max_{a \leq x \leq b} |f(x) - p(x)|$$

if and only if there exist $n+1$ points $a \leqslant x_0 < x_1 < \cdots < x_n \leqslant b$ such that

$$f(x_i) - p(x_i) = s \cdot (-1)^i \|f - p\|_\infty,$$

where $|s| = 1$.

Recall that choosing $x_i = \cos[(2i-1)\pi/(2n)]$, the roots of the $T_n(\theta) = \cos(n\theta)$ Chebyshev polynomial (with $x = \cos\theta$, $a = -1$, $b = 1$), leads to the optimal error bound in polynomial interpolation

$$|f(t) - p(t)| \leqslant \frac{\|f^{(n+1)}\|_\infty}{(n+1)!\,2^n}.$$

The error bound came about from consideration of the alternation of signs of $p(x_j) - q(x_j)$ at the extrema of the Chebyshev polynomial T_n, $x_i = \cos(i\pi/n)$, $i = 0, 1, \ldots n$, with p, q monic polynomials. The Cebyshev alternation theorem generalizes this observation and allows the formulation of a general approach to finding the best inf-norm approximant known as the Remez algorithm. The idea is that rather than seeking to satisfy the interpolation conditions

$$\boldsymbol{M}\boldsymbol{a} = \boldsymbol{y}$$

in the monomial basis

$$\boldsymbol{M} = \mathscr{M}_{n-1}(\mathrm{x}) = [\ \mathbf{1}\ \ \boldsymbol{x}\ \ \ldots\ \ \boldsymbol{x}^{n-1}\ ] \in \mathbb{R}^{n \times n},$$

attempt to find n alternating-sign extrema points by considering the basis set

$$\boldsymbol{R} = \mathscr{R}_n(\boldsymbol{x}) = [\ \mathbf{1}\ \ \boldsymbol{x}\ \ \ldots\ \ \boldsymbol{x}^{n-1}\ \ \pm\mathbf{1}\,] \in \mathbb{R}^{(n+1)\times(n+1)}$$

with $\pm\mathbf{1} = [\ +1\ \ -1\ \ +1\ \ \ldots\ ]$.

Algorithm (Remez)

1. Initialize $\boldsymbol{x} \in \mathbb{R}^{n+1}$ to Chebyshev maxima on interval $[a, b]$
2. Solve $\boldsymbol{R}\boldsymbol{c} = f(\boldsymbol{x})\ \mathscr{R}(\boldsymbol{x})$, $\boldsymbol{c}^T = [\ \boldsymbol{a}^T\ \ c_{n+1}\ ]$, $\boldsymbol{a} \in \mathbb{R}^n$
3. Find the extrema $\boldsymbol{y}$ of $p(t) - f(t)$ with $p(t) = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1}$
4. If $p(y_i) - f(y_i)$ are approximately equal in absolute value and of opposite signs, return $\boldsymbol{x}$
5. Otherwise set $\boldsymbol{x} = \boldsymbol{y}$, repeat

CHAPTER 3

LINEAR OPERATOR APPROXIMATION

LECTURE 19: DERIVATIVE APPROXIMATION

1. Linear operator approximation

An operator is understood here as a mapping from a domain vector space $\mathcal{U} = (U, S, +, \cdot)$ to a co-domain vector space $\mathcal{V} = (V, S, +, \cdot)$, and the operator $\mathcal{L}: U \longrightarrow V$ is said to be linear if for any scalars $c_1, c_2 \in S$ and vectors $u_1, u_2 \in U$,

$$\mathcal{L}(c_1 u_1 + c_2 u_2) = c_1 \mathcal{L}(u_1) + c_2 \mathcal{L}(u_2),$$

i.e., the image of a linear combination is the linear combination of the images. Linear algebra considers the case of finite dimensional vector spaces, such as $U = \mathbb{R}^m$, $V = \mathbb{R}^n$, in which case a linear operator is represented by a matrix $\boldsymbol{L} \in \mathbb{R}^{m \times n}$, and satisfies

$$\boldsymbol{L}(c_1 \boldsymbol{u}_1 + c_2 \boldsymbol{u}_2) = c_1 \boldsymbol{L}\boldsymbol{u}_1 + c_2 \boldsymbol{L}\boldsymbol{u}_2 .$$

In contrast, the focus here is on infinite-dimensional function spaces such as $C^r(\mathbb{R})$ (cf. Tab. 1, L18), the space of functions with continuous derivatives up to order r. Common linear operator examples include:

Differentiation. $\mathcal{L}f = \partial^k f / \partial t^k$, $\mathcal{L}: C^r(\mathbb{R}) \longrightarrow C^{r-k}(\mathbb{R})$.

Riemann integration. $\mathcal{L}f = \int_a^b \omega(t)\, f(t)\, \mathrm{d}t$, $\mathcal{L}: C(\mathbb{R}\backslash\Delta) \longrightarrow \mathbb{R}$, where Δ is a set of measure zero.

Linear differential equation. $\mathcal{L}y = \sum_{j=0}^{k} a_j(t)\, y^{(j)} = f(t)$, $\mathcal{L}: C^r(\mathbb{R}) \longrightarrow C^{r-k}(\mathbb{R})$.

1.1. Numerical differentiation

A general approach to operator approximation is to simply introduce an approximation of the function the operator acts upon, $f \cong p$,

$$\mathcal{L}f \cong \mathcal{L}p.$$

Monomial basis. As an example consider the polynomial interpolant of f based upon data $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0, \ldots, n\}$,

$$p(t) = \begin{bmatrix} 1 & t & t^2 & \ldots & t^n \end{bmatrix} \boldsymbol{c},$$

with coeffcients $\boldsymbol{c}$ determined as the solution of the interpolation conditions

$$\boldsymbol{M}\boldsymbol{c} = \boldsymbol{y},$$

with notations

$$\boldsymbol{M} = [\ \mathbf{1}\ \ \boldsymbol{x}\ \ \boldsymbol{x}^2\ \ \ldots\ \ \boldsymbol{x}^n\ ],\ \boldsymbol{x}^k = [\ x_0^k\ \ \ldots\ \ x_n^k\ ]^T,\ \boldsymbol{y} = [\ y_0\ \ \ldots\ \ y_n\ ]^T.$$

Differentiation of f ($\mathcal{L} = \mathrm{d}/\mathrm{d}t$) can be approximated as

$$\frac{\mathrm{d}}{\mathrm{d}t} f \cong \frac{\mathrm{d}}{\mathrm{d}t} p = [\ 0\ \ 1\ \ 2t\ \ \ldots\ \ nt^{n-1}\ ]\boldsymbol{c}.$$

It is often of interest to express the result of applying an operator directly in terms of known information on f. Formally, in the case of differentiation,

$$\frac{\mathrm{d}}{\mathrm{d}t} f \cong [\ 0\ \ 1\ \ 2t\ \ \ldots\ \ nt^{n-1}\ ]\boldsymbol{M}^{-1}\boldsymbol{y},$$

allowing the identification of a differentiation approximation operator $\mathcal{D}$

$$\frac{\mathrm{d}}{\mathrm{d}t} f \cong \mathcal{D}(\boldsymbol{y}), \mathcal{D} = [\ 0\ \ 1\ \ 2t\ \ \ldots\ \ nt^{n-1}\ ]\boldsymbol{M}^{-1}.$$

This formulation explicitly includes the inversion of the sampled basis matrix $\boldsymbol{M}$, and is hence not computationally efficient. Alternative formulations can be constructed that carry out some of the steps in computing $\boldsymbol{M}^{-1}$ analytically.

Newton basis (finite difference calculus). An especially useful formulation for numerical differentiation arises from the Newton interpolant of data $\mathcal{D} = \{(x_i = ih, y_i = f(x_i)), i = 0, \ldots, n\}$, $f : \mathbb{R} \longrightarrow \mathbb{R}$, $f \in C^{(n+1)}(\mathbb{R})$,

$$f(t) \cong p(t) = [y_0] + [y_1, y_0](t - x_0) + \cdots + [y_n, y_{n-1}, \ldots, y_0](t - x_0) \cdot (t - x_1) \cdot \ldots \cdot (t - x_{n-1}).$$

For equidistant sample points $x_i = ih$, the Newton interpolant can be expressed as an operator acting upon the data. Introduce the translation operator

$$Ef(t) = f(t + h).$$

Repeated application of the translation operator leads to

$$E^k f(t) = E(\,E^{k-1} f(t)) = \cdots = f(t + kh),$$

and the identity operator is given by

$$If(t) = f(t) = E^0 f(t) \Longrightarrow I = E^0.$$

Finite differences of the function values are expressed through the forward, backward and central operators

$$\Delta = E - I, \nabla = I - E, \delta = E^{1/2} - E^{-1/2},$$

leading to the formulas

$$\Delta f(t) = f(t+h) - f(t), \nabla f(t) = f(t) - f(t-h), \delta f(t) = f(t+h/2) - f(t-h/2).$$

Applying the above to the data set $\mathcal{D}$ leads to

$$\Delta y_i = y_{i+1} - y_i, \nabla y_i = y_i - y_{i-1}, \delta y_i = y_{i+1/2} - y_{i-1/2}.$$

The divided differences arising in the Newton can be expressed in terms of finite difference operators,

$$[y_1, y_0] = \frac{y_1 - y_0}{h} = \frac{1}{h}\Delta y_0, [y_2, y_1, y_0] = \frac{[y_2, y_1] - [y_1, y_0]}{2h} = \frac{\Delta y_1 - \Delta y_0}{2h^2} = \frac{\Delta^2 y_0}{2h^2},$$

or in general

$$[y_k, \ldots, y_1, y_0] = \frac{\Delta^k}{k!\, h^k} y_0.$$

Using the above and rescaling the variable t in the Newton basis $\mathcal{N} = \{1, t - x_0, (t - x_0)(t - x_1), \ldots\}$ in units of the step size $t = \alpha h + x_0$ leads to

$$p(t(\alpha)) = P(\alpha) = \left(I + \alpha \frac{\Delta}{1!} + \alpha(\alpha - 1)\frac{\Delta^2}{2!} + \cdots + \alpha(\alpha - 1) \cdot \ldots \cdot (\alpha - 1 + n)\frac{\Delta^n}{n!} \right) y_0. \tag{3.1}$$

The generalized binomial series states

$$(1 + x)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} x^k, \tag{3.2}$$

with

$$\binom{\alpha}{k} = \frac{\alpha(\alpha - 1)\ldots(\alpha - k + 1)}{k!}$$

the generalized binomial coefficient. The operator acting upon y_0 in (3.1) can be interpreted as the truncation at order n

$$P(\alpha) \cong (I + \Delta)^\alpha y_0 = \mathscr{I}_\alpha y_0,$$

of the operator $(I + \Delta)^\alpha$ defined through (3.2) by the substitutions $1 \longrightarrow I$, $x \longrightarrow \Delta$. The operator $\mathscr{I}_\alpha = (I + \Delta)^\alpha$can be interpreted as the interpolation operator with equidistant sampling points, with $P(\alpha)$ its truncation to order n. Reversing the order of the sampling points leads to the Newton interpolant

$$p(t) = [y_n] + [y_{n-1}, y_n](t - x_n) + \cdots + [y_0, y_1, \ldots, y_n](t - x_n)(t - x_{n-1}) \cdot \ldots \cdot (t - x_1).$$

The divided differences can be expressed in terms of the backward operator as

$$[y_{n-1}, y_n] = \frac{y_{n-1} - y_n}{h} = -\frac{1}{h}\nabla y_n, [y_{n-2}, y_{n-1}, y_n] = \frac{[y_{n-2}, y_{n-1}] - [y_{n-1}, y_n]}{2h} = -\frac{\nabla y_{n-1} - \nabla y_n}{2h^2} = \frac{\nabla^2 y_n}{2h^2},$$

leading to an analogous expression of the interpolation operator in terms backward finite differences

$$p(t(\alpha)) = P(\alpha) = \left(I - \alpha \frac{\nabla}{1!h} + \alpha(\alpha-1)\frac{\nabla^2}{2!h^2} + \cdots + (-1)^n \alpha(\alpha-1)\cdot \ldots \cdot (\alpha-1+n)\frac{\nabla^n}{n!h^n} \right) y_n \cong (I-\nabla)^\alpha y_n = \mathcal{B}_\alpha y_n.$$

Differentiation of the interpolation expressed in terms of forward finite differences gives

$$f'(t) \cong \frac{\mathrm{d}}{\mathrm{d}t} P(\alpha) = \frac{\mathrm{d}\alpha}{\mathrm{d}t} P'(\alpha) \cong \frac{1}{h}\frac{\mathrm{d}}{\mathrm{d}\alpha}\mathcal{I}_\alpha y_0 = \frac{1}{h}\left[\ln(I+\Delta)\right](I+\Delta)^a y_0 \cong \frac{1}{h}\ln(I+\Delta)\, P(\alpha).$$

The particular interpolant $P(\alpha)$ is irrelevant, leading to the operator identity

$$\frac{\mathrm{d}}{\mathrm{d}t} \cong \frac{1}{h}\ln(I+\Delta).$$

For $|x|<1$, the power series expansions are

$$\frac{\mathrm{d}}{\mathrm{d}x}\ln(1+x) = \frac{1}{1+x} = 1 - x + x^2 - \cdots \Longrightarrow \ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots + (-1)^{k+1}\frac{x^k}{k} + \cdots,$$

are uniformly convergent, leading to the expression

$$\frac{\mathrm{d}}{\mathrm{d}t} \cong \frac{1}{h}\left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \ldots + (-1)^k \frac{1}{k}\Delta^k + \cdots \right),$$

stating that the (continuum) differentiation operator can be approximated by an infinite series of finite difference operations, recovered exactly in the $h \longrightarrow 0$ limit. Denote by D_k^+ the truncation at term k of the above operator series such that

$$f'(x_0) \cong D_k^+(f)(x_0) = \frac{1}{h}\left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \ldots + (-1)^k \frac{1}{k}\Delta^k \right) y_0.$$

Truncation at $k = 1, 2, 3$ leads to the expressions

$$D_1^+(f) = \frac{f(h+t) - f(t)}{h}, \qquad D_2^+(f) = \frac{4f(h+t) - f(2h+t) - 3f(t)}{2h}, \qquad D_3^+(f) = \frac{18f(h+t) - 9f(2h+t) + 2f(3h+t) - 11f(t)}{6h}.$$

The $h \longrightarrow 0$ limit of divided differences is given by

$$\lim_{h\to 0}[y_k, y_{k-1}, \ldots, y_0] = \lim_{h\to 0}\left(\frac{1}{k!h^k}\Delta^k y_0 \right) = \frac{1}{k!} f^{(k)}(x_0),$$

such that for small finite $h > 0$,

$$\Delta^k y_0 \cong h^k f^{(k)}(x_0).$$

The resulting derivative approximation error is of order k,

$$e_k^+(t) = D_k^+(f)(t) - f'(t) = \frac{(-1)^{k+1}h^k}{k+1} f^{(k+1)}(t) = O(h^k).$$

The analogous expression for backward differences is

$$\frac{\mathrm{d}}{\mathrm{d}t} \cong -\frac{1}{h}\ln(I-\nabla) = \frac{1}{h}\left(\nabla + \frac{1}{2}\nabla^2 + \frac{1}{3}\nabla^3 + \ldots + \frac{1}{k}\nabla^k + \cdots\right),$$

and the first few truncations are

$$D_{\bar{1}}(f) = \frac{f(t-h)-f(t)}{h}, \quad D_{\bar{2}}(f) = \frac{-f(t-2h)+4f(t-h)-3f(t)}{2h}, \quad D_{\bar{3}}(f) = \frac{2f(t-3h)-9f(t-2h)+18f(t-h)-11f(t)}{6h}$$

with errors

$$e_{\bar{k}}(t) = D_{\bar{k}}(f)(t) - f'(t) = \frac{h^k}{k} f^{(k+1)}(t) = O(h^k).$$

The above operator identities can be inverted to obtain

$$\Delta = E - I = \exp\left(h\frac{\mathrm{d}}{\mathrm{d}t}\right) - I, \nabla = I - E^{-1} = I - \exp\left(-h\frac{\mathrm{d}}{\mathrm{d}t}\right),$$

leading to

$$E = \exp\left(h\frac{\mathrm{d}}{\mathrm{d}t}\right) = 1 + h\frac{\mathrm{d}}{\mathrm{d}t} + \frac{1}{2}\left(h\frac{\mathrm{d}}{\mathrm{d}t}\right)^2 + \ldots + \frac{1}{k!}\left(h\frac{\mathrm{d}}{\mathrm{d}t}\right)^k + \cdots +$$

this time expressing the finite translation operator as an infinite series of continuum differentiation operations. This allows expressing the central difference operator as

$$\delta = E^{1/2} - E^{-1/2} = \exp\left(\frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}t}\right) - \exp\left(-\frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}t}\right) = 2\sinh\left(\frac{h}{2}\frac{\mathrm{d}}{\mathrm{d}t}\right),$$

and approximations of the derivative based on centered differencing are obtained from

$$\frac{\mathrm{d}}{\mathrm{d}t} \cong \frac{2}{h}\operatorname{arcsinh}\left(\frac{\delta}{2}\right) = \frac{1}{h}\left(\delta - \frac{\delta^3}{24} + \frac{3\delta^5}{640} - \frac{5\delta^7}{7168} + \frac{35\delta^9}{294912} - \cdots\right).$$

An advantage of the centered finite differences (surmised from the odd power series) is a higher order of accuracy

$$e_k = D_k f(f) - f'(t) = O(h^{2k}).$$

Higher order derivative are obtained by repeated application of the operator series, e.g.,

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} = \frac{\mathrm{d}}{\mathrm{d}t}\cdot\frac{\mathrm{d}}{\mathrm{d}t} = \frac{1}{h^2}\left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \cdots\right)^2 = \frac{1}{h^2}\left(\Delta^2 - \Delta^3 + \frac{11}{12}\Delta^4 - \cdots\right)^2.$$

Moment method. An alternative derivation of the above finite difference formulas is to construct a linear combination of function values

$$L_m^n f(t) = \sum_{k=-m}^{n} c_k f(t+kh) = \left(\sum_{k=-m}^{n} c_k E^k\right) f(t),$$

and determine the coefficients c_k such that the p^{th} derivative is approximated to order q

$$f^{(p)}(t) = L_m^n f(t) + O(h^q).$$

For example, for $m=0$, $n=1$, carrying out Taylor series expansions gives

$$\begin{aligned} f(t+h) &= f(t) + hf'(t) + \frac{1}{2}h^2 f''(t) + \cdots \\ f(t) &= f(t). \end{aligned}$$

Eliminating $f(t)$ by multiplying the first equation by $c_1 = 1$ and the second by $c_0 = -1$ recovers the forward finite difference formula

$$f'(t) = \frac{f(t+h) - f(t)}{h} + O(h).$$

***B*-spline basis.** The above example used a truncation of the monomial basis $\mathcal{M}_n(t) = \{1, t, ..., t^n\}$. Analogous results are obtained when using a different basis. Consider the equidistant sample points $x_i = ih + x_0$, data $\mathcal{D} = \{(x_i, y_i = f(x_i), i = 0, 1, \ldots, n)\}$and the first-degree *B*-spline basis

$$\mathcal{B}_{n,1}(t) = \{B_{0,1}(t), B_{1,1}(t), \ldots, B_{n,1}(t)\},$$

in which case the linear piecewise interpolant is expressed as

$$p(t) = \sum_{i=0}^{n} y_i B_{i,1}(t),$$

and over interval $[x_{i-1}, x_i]$ reduces to

$$p_i(t) = y_{i-1}B_{i-1,1}(t) + y_i B_{i,1}(t) = y_{i-1}\cdot\frac{x_i - t}{x_i - x_{i-1}} + y_i\cdot\frac{t - x_{i-1}}{x_i - x_{i-1}}.$$

Differentiation recovers the familiar slope expression

$$p_i'(t) = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} = \frac{y_i - y_{i-1}}{3h}.$$

At the nodes, a piecewise linear spline is discontinuous, hence the derivative is not defined, though one could consider the one-sided limits. Evaluation of derivatives at midpoints $t_i = (x_{i-1} + x_i)/2 = (i-1)\,h + h/2 + x_0$, $i = 1, 2, \ldots, n$, leads to

$$\boldsymbol{y}' = \begin{bmatrix} y_1' \\ y_2' \\ \vdots \\ y_n' \end{bmatrix} = p'(\boldsymbol{t}) = \boldsymbol{D}\boldsymbol{x} = \frac{1}{h}\begin{bmatrix} -1 & 1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 1 & 0 & \ldots & 0 \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & -1 & 1 \end{bmatrix}\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix},$$

with $\boldsymbol{D} \in \mathbb{R}^{n \times (n+1)}$.

LECTURE 20: QUADRATURE

0.1. Numerical integration

Numerical integration proceeds along the same lines as numerical differentiation

$$\mathcal{L}f \cong \mathcal{L}p,$$

with a different operator

$$\mathcal{L}f = \int_a^b \omega(t)\, f(t)\, \mathrm{d}t,$$

with $\mathcal{L}: C([a,b] \backslash Z) \longrightarrow \mathbb{R}$ with Z a set of measure zero, e.g., $Z = \emptyset$ for a function continuous at all points in $[a,b]$. It is often useful to explicitly identify a weight function $\omega(t)$ that can attribute higher significance to subsets of $[a,b]$. In the integration case, the approximation basis choice can be combined with decomposition of the domain into m subintervals $[a_k, b_k)$ such that $\bigcup_{k=1}^m [a_k, b_k) = [a,b)$ through

$$\int_a^b \omega(t)\, f(t)\, \mathrm{d}t = \sum_{k=1}^m \int_{a_k}^{b_k} \omega(t)\, f(t)\, \mathrm{d}t,$$

with

$$a = a_1 < b_1 = a_2 < b_2 = a_3 < \cdots < b_m = b.$$

Monomial basis. As for numerical differentiation, an integration operator $\mathcal{I}$ can be obtained from the polynomial interpolant of f based upon data $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0, \ldots, n\}$,

$$p(t) = \begin{bmatrix} 1 & t & t^2 & \ldots & t^n \end{bmatrix} \boldsymbol{c}, \boldsymbol{c} = \boldsymbol{M}^{-1}\boldsymbol{y},$$

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x} & \boldsymbol{x}^2 & \ldots & \boldsymbol{x}^n \end{bmatrix}, \boldsymbol{x}^k = \begin{bmatrix} x_0^k & \ldots & x_n^k \end{bmatrix}^T, \boldsymbol{y} = \begin{bmatrix} y_0 & \ldots & y_n \end{bmatrix}^T.$$

Integration of f for $\omega(t) = 1$ is approximated as

$$\int_a^b f(t)\,\mathrm{d}t \cong \mathcal{I}(f) = \int_a^b p(t)\,\mathrm{d}t = \left[\begin{array}{cccc} b-a & \frac{1}{2}(b^2-a^2) & \frac{1}{3}(b^3-a^3) & \ldots & \frac{1}{n+1}(b^{n+1}-a^{n+1}) \end{array}\right] \boldsymbol{M}^{-1}\boldsymbol{y}.$$

As for numerical differentation, the computational effort in the above formulation can be reduced through alternative basis choices.

Lagrange basis. Assume $n = dm$ in the data set $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0, \ldots, n\}$, and construct an interpolant of degree d in each subinterval $[a_k, b_k) = [x_{d(k-1)}, x_{dk})$. As highlighted by the approximation of the Runge function, the degree d should be small, typically $d \in \{1, 2, 3\}$.

Trapezoid formula. For $d = 1$, a linear approximant is defined over each subinterval $[x_{k-1}, x_k]$, stated in Lagrange form as

$$p_k(t) = \ell_{k-1}(t)\,y_{k-1} + \ell_k(t)\,y_k = \frac{t - x_k}{x_{k-1} - x_k}\,y_{k-1} + \frac{t - x_{k-1}}{x_k - x_{k-1}}\,y_k.$$

The resulting integral approximation

$$\mathcal{L}(f) = \int_{x_{k-1}}^{x_k} f(t)\,\mathrm{d}t \cong \mathcal{I}(p) = \int_{x_{k-1}}^{x_k} p_k(t)\,\mathrm{d}t = (x_k - x_{k-1})\frac{y_{k-1} + y_k}{2}.$$

The approximation error results from the known polynomial interpolation error formula

$$f(t) - p(t) = \frac{f''(\xi_t)}{2}(t - x_k)(t - x_{k-1}),$$

that gives

$$e_k(t) = \left|\int_{x_{k-1}}^{x_k} [f(t) - p(t)]\,\mathrm{d}t\right| \leqslant \|f''\|_\infty \frac{(x_k - x_{k-1})^3}{12}.$$

Using an equidistant partition $x_k = a + kh$, $h = (b-a)/n$, over the entire $[a, b]$ interval gives the composite trapezoid formula

$$\int_a^b f(t)\,\mathrm{d}t = \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} f(t)\,\mathrm{d}t \cong \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} p_k(t)\,\mathrm{d}t = h\left(\frac{y_0}{2} + \sum_{k=1}^{n} y_k + \frac{y_n}{2}\right).$$

The overall approximation error is bounded by the error over each subinterval

$$e \leqslant \sum_{k=1}^{n} e_k \leqslant \|f''\|_\infty\, n\frac{h^3}{12} = (b-a)\frac{h^2\|f''\|_\infty}{12},$$

and the trapezoid formula exhibits quadratic convergence, $e = O(h^2)$.

Simpson formula. A more accurate, quadratic approximant is obtained for $d = 2$ using sample points x_{k-2}, x_{k-1}, x_k

$$p_k(t) = \ell_{k-2}(t)\, y_{k-2} + \ell_{k-1}(t)\, y_{k-1} + \ell_k(t)\, y_k,$$

$$p_k(t) = \frac{(t-x_{k-1})(t-x_k)}{(x_{k-2}-x_{k-1})(x_{k-2}-x_k)}\, y_{k-2} + \frac{(t-x_{k-2})(t-x_k)}{(x_{k-1}-x_{k-2})(x_{k-1}-x_k)}\, y_{k-1} + \frac{(t-x_{k-2})(t-x_{k-1})}{(x_k-x_{k-2})(x_k-x_{k-1})}\, y_k.$$

Assuming $n = 2m$, $x_k = a + kh$, $h = (b-a)/n$, over a subinterval the Simpson approximation is

$$\int_{x_{k-2}}^{x_k} f(t)\,\mathrm{d}t \cong \int_{x_{k-2}}^{x_k} p_k(t)\,\mathrm{d}t = \frac{h}{3}(y_{k-2} + 4y_{k-1} + y_k).$$

Integration of the interpolation error

$$f(t) - p_k(t) = \frac{f^{(3)}(\xi_t)}{3!}(t-x_{k-2})(t-x_{k-1})(t-x_k),$$

leads to calculation of

$$\int_{x_{k-2}}^{x_k} (t-x_{k-2})(t-x_{k-1})(t-x_k)\,\mathrm{d}t = \int_{-h}^{h} (\tau-h)\,\tau\,(\tau+h)\,\mathrm{d}\tau = 0. \tag{3.3}$$

This null result is a feature of even-degree interpolants $d = 2r$. Note that the interpolation error formula contains an evaluation of $f^{(3)}(\xi_t)$ at some point $\xi_t \in [x_{k-2}, x_k]$ that depends on t, so the integral

$$\int_{x_{k-2}}^{x_k} \frac{f^{(3)}(\xi_t)}{3!}(t-x_{k-2})(t-x_{k-1})(t-x_k)\,\mathrm{d}t$$

is not necessarily equal to zero. To obtain an error estimate, rewrite the interpolating polynomial in Newton form

$$p_k(t) = y_{k-1} + [y_{k-1}, y_{k-2}](t-x_{k-2}) + [y_k, y_{k-1}, y_{k-2}](t-x_{k-2})(t-x_{k-1}),$$

The next higher degree interpolant would be

$$q_k(t) = p_k(t) + c_k(t-x_{k-2})(t-x_{k-1})(t-x_k),$$

and (3.3) implies that the integral of q_k is equal to that of p_k

$$\int_{x_{k-2}}^{x_k} q_k(t)\,\mathrm{d}t = \int_{x_{k-2}}^{x_k} p_k(t)\,\mathrm{d}t,$$

hence the Simpson formula based on a quadratic interpolation is as accurate as that based on a cubic interpolation. The error can now be estimated using

$$e_k = \left| \int_{x_{k-2}}^{x_k} [f(t) - q_k(t)]\,\mathrm{d}t \right| \leq \frac{\|f^{(\mathrm{iv})}\|_\infty}{4!} \int_{x_{k-2}}^{x_k} (t-x_{k-2})(t-x_{k-1})(t-x_k)(t-z_k)\,\mathrm{d}t,$$

where z_k is some additional interpolation point within $[x_{k-2}, x_k]$. It is convenient to choose $z_k = x_{k-1}$, which corresponds to a Hermite interpolation condition at the midpoint. This is simply for the purpose of obtaining an error estimate, and does not affect the Simpson estimate of the integral. Carrying out the calculations gives

$$e_k \leq \frac{\|f^{(4)}\|_\infty}{90} h^5.$$

When applied to the overall interval $[a,b]$, the Simpson formula is stated as

$$\int_a^b f(t)\,\mathrm{d}t \cong \frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right],$$

with error

$$e \leqslant \frac{\|f^{(4)}\|_\infty}{2880}(b-a)^5.$$

The composite Simpson formula is

$$\int_a^b f(t)\,\mathrm{d}t = \sum_{k=1}^m \int_{x_{2k-2}}^{x_{2k}} f(t)\,\mathrm{d}t \cong \sum_{k=1}^m \int_{x_{2k-2}}^{x_{2k}} p_k(t)\,\mathrm{d}t = \frac{h}{3}\left(y_0+4\sum_{k=1}^m y_{2k-1}+2\sum_{k=1}^{m-1} y_{2k}+y_n\right),$$

with an overall error bound

$$e \leqslant \sum_{k=1}^m \int_{x_{2k-2}}^{x_{2k}} |f(t)-p_k(t)|\,\mathrm{d}t \leqslant \frac{\|f^{(4)}\|_\infty}{90}h^5\frac{n}{2} = \frac{(b-a)}{180}\|f^{(4)}\|_\infty h^4,$$

that shows fourth-order convergence $e = O(h^4)$.

Moment method. As in numerical differentiation, an alternative derivation is available. Numerical integration is stated as a quadrature formula

$$\mathcal{L}(f) = \int_a^b \omega(t) f(t)\,\mathrm{d}t \cong Q(f) = \sum_{i=0}^n w_i y_i,$$

using data $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0,\ldots,n\}$. Once the sampling points are chosen, there remain $n+1$ weights w_i to be determined. One approach is to enforce exact quadrature for the first $n+1$ monomials

$$\mathcal{L}(x^k) = Q(x^k) \Longrightarrow \int_a^b \omega(t)t^k\,\mathrm{d}t = \sum_{i=0}^n w_i x_i^k, k = 0,1,\ldots,n.$$

A Vandermonde system results whose solution furnishes the appropriate weights. Since the Vandermonde matrix is ill-conditioned, a solution is sought in exact arithmetic, using rational numbers as opposed to floating point approximations.

The principal utility of the moment method is construction of quadrature formulas for singular integrands. For example, in the integral

$$-\int_0^1 \ln t \sin t\,\mathrm{d}t,$$

the $\ln t$ is an integrable singularity, and accurate quadrature rules can be constructed by the method of moments for

$$\mathcal{L}(f) = \int_a^b \omega(t) f(t)\,\mathrm{d}t,$$

with $\omega(t) = -\ln t$.

0.2. Gauss quadrature

Recall that the method of moments approach to numerical integration based upon sampling $\mathcal{D} = \{(x_i, y_i = f(x_i)), i = 0,\ldots,n\}$,

$$\int_a^b \omega(t) f(t)\,\mathrm{d}t = \sum_{i=0}^n w_i y_i + e \cong \sum_{i=0}^n w_i y_i,$$

imposes exact results for a finite number of members of a basis set $\{\phi_0, \ldots, \phi_n, \ldots\}$

$$\int_a^b \omega(t)\phi_k(t)\,\mathrm{d}t = \sum_{i=0}^{n} w_i \phi_k(x_i), k = 0, 1, \ldots, n.$$

The trapezoid, Simpson formulas arise from the monomial basis set $\{1, t, t^2, \ldots\}$, in which case

$$\int_a^b \omega(t)t^k\,\mathrm{d}t = \sum_{i=0}^{n} w_i x_i^k, k = 0, 1, \ldots, n,$$

but any basis set can be chosen. Instead of prescribing the sampling points x_i *a priori*, which typically leads to an error $e = \mathcal{O}(\phi_{n+1}(t))$, the sampling points can be chosen to minimize the error e. For the monomial basis this leads to a system of $2(n+1)$ equations

$$\int_a^b \omega(t)t^k\,\mathrm{d}t = \sum_{i=0}^{n} w_i x_i^k, k = 0, 1, \ldots, 2n+1,$$

for the unknown $n+1$ quadrature weights w_i and the $n+1$ sampling points x_i. The system is nonlinear, but can be solved in an insightful manner exploiting the properties of orthogonal polynomials known as Gauss quadrature.

The basic idea is to consider a Hilbert function space with the scalar product

$$(f, g) = \int_a^b \omega(t) f(t)\, g(t)\,\mathrm{d}t,$$

and orthonormal basis set $\{\phi_0(t), \phi_1(t), \phi_2(t),, \ldots\}$,

$$(\phi_j, \phi_k) = \int_a^b \omega(t)\phi_j(t)\,\phi_k(t)\,\mathrm{d}t = \delta_{jk}.$$

Assume that $\phi_k(t)$ are polynomials of degree k. A polynomial p_{2n+1} of degree $2n+1$ can be factored as

$$p_{2n+1}(t) = q_n(t)\,\phi_{n+1}(t) + r_n(t),$$

where $q_n(t)$ is the quotient polynomial of degree n, and r_n is the remainder polynomial of degree n. The weighted integral of p_{2n+1} is therefore

$$\int_a^b \omega(t)p_{2n+1}(t)\,\mathrm{d}t = \int_a^b \omega(t)[q_n(t)\,\phi_{n+1}(t) + r_n(t)]\,\mathrm{d}t = (q_n, \phi_{n+1}) + \int_a^b \omega(t)r_n(t)\,\mathrm{d}t.$$

Since $\{\phi_0, \ldots, \phi_{n+1}\}$ is an orthonormal set, $(q_n, \phi_{n+1}) = 0$, and the integral becomes

$$\int_a^b \omega(t)p_{2n+1}(t)\,\mathrm{d}t = \int_a^b \omega(t)r_n(t)\,\mathrm{d}t.$$

The integral of the n^{th} remainder polynomial can be exactly evaluated through an $n+1$ point quadrature

$$\int_a^b \omega(t)r_n(t)\,\mathrm{d}t = \sum_{i=0}^{n} w_i\, r(x_i),$$

that however evaluates $r(t)$ rather than the original integrand $p_{2n+1}(t)$. However, evaluation of the factorization (0.2) at the roots x_i of ϕ_{n+1}, $\phi_{n+1}(x_i) = 0$, $i = 0, 1, \ldots, n$, gives

$$p_{2n+1}(x_i) = q_n(x_i)\,\phi_{n+1}(x_i) + r_n(x_i) = r_n(x_i),$$

stating that the values of the remainder at these nodes are the same as those of the p_{2n+1} polynomial. This implies that

$$\int_a^b \omega(t) p_{2n+1}(t)\,\mathrm{d}t = \sum_{i=0}^{n} w_i\, p_{2n+1}(x_i),$$

is an exact quadrature of order $2n+1$, $e = O(t^{2n+1})$. The weights w_i can be determined through any of the previously outlined methods, e.g., method of moments

$$\int_a^b \omega(t) t^k\,\mathrm{d}t = \sum_{i=0}^{n} w_i x_i^k, k = 0,\ldots,n,$$

which is now a linear system that can be readily solved. Alternatively, the weights are also directly given as integrals of the Lagrange polynomials based upon the nodes that are roots of ϕ_{n+1}

$$w_i = \int_a^b \omega(t)\ell_i(t)\,\mathrm{d}t.$$

LECTURE 21: ORDINARY DIFFERENTIAL EQUATIONS - SINGLE STEP METHODS

1. Ordinary differential equations

An n^{th}-order ordinary differential equation given in explicit form

$$y^{(n)} = f(t, y, y', \ldots, y^{(n-1)}) \tag{3.4}$$

is a statement of equality between the action of two operators. On the left hand side the linear differential operator

$$\mathscr{L} = \frac{\mathrm{d}}{\mathrm{d}t^n}$$

acts upon a sufficiently smooth function, $y \in C^{(n)}(\mathbb{R})$, $\mathscr{L}: C^{(n)}(\mathbb{R}) \longrightarrow C(\mathbb{R})$. On the right hand side, a nonlinear operator $\mathscr{F}$ acts upon the independent variable t and the first $n-1$ derivatives

$$\mathscr{F}: \mathbb{R} \times C(\mathbb{R}) \times \cdots \times C^{(n-1)}(\mathbb{R}).$$

An associated function $f: \mathbb{R}^{n+1} \longrightarrow \mathbb{R}$ has values given by

$$f(t) = f(t, y(t), y'(t), \ldots, y^{(n-1)}(t)).$$

The numerical solution of (3.4) seeks to find an approximant of y through:

1. Approximation of the differentiation operator $\mathscr{L}$;
2. Approximation of the nonlinear operator $\mathscr{F}$;
3. Approximation of the equality between the effect of the two operators

$$\mathscr{L}(y) = \mathscr{F}(t, y, \ldots, y^{(n-1)}).$$

These approximation problems shall be considered one-by-one, starting with approximation of $\mathscr{L}$ assuming that the action of $\mathscr{F}$ is exactly represented through knowledge of f.

Note that an n^{th}-order differential equation can be restated as a system of n first-order equations

$$\boldsymbol{z}' = \boldsymbol{F}(t, \boldsymbol{z}) \tag{3.5}$$

by introducing

$$\boldsymbol{z} = [\ z_1\ \ z_2\ \ \dots\ \ z_{n-1}\ \ z_n\ ]^T = [\ y\ \ y'\ \ \dots\ \ y^{(n-2)}\ \ y^{(n-1)}\ ]^T,$$

$$\boldsymbol{F}(t, \boldsymbol{z}) = [\ z_2(t)\ \ z_3(t)\ \ \dots\ \ z_n(t)\ \ f(t, z_1(t), \dots, z_n(t))\ ]^T.$$

Approximation of the differentiation operator for the problem

$$y' = f(t, y) \tag{3.6}$$

can readily be extended to the individual equations of system (3.5).

Construction of approximants to (3.6) is first considered for the initial value problem (IVP)

$$y' = f(t, y), y(0) = y_0. \tag{3.7}$$

The two procedures are:

1. Approximation of the differentiation operator;
2. Differentiation of an approximation of y.

Often the two approaches leads to the same algorithm. The problem (3.7) has a unique solution over some rectangle $R = [0, T] \times [y_1, y_2]$ in the ty-plane if f is Lipschitz-continuous, stated as the existence of $K \in \mathbb{R}_+$ such that

$$|f(t, y_2) - f(t, y_1)| \leqslant K |y_2 - y_1|.$$

Note that Lipschitz continuity is a stronger condition than standard continuity in that it states $|f(t, y_2) - f(t, y_1)| = O(|y_2 - y_1|)$. Differentiability implies Lipschitz continuity.

Consider approximation of $\mathrm{d}/\mathrm{d}t$ through forward finite differences

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{1}{h}\left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \cdots - \right), \tag{3.8}$$

and denote by y_i the approximation of $y(t)$, $y_i \cong y(t_i)$ at the equidistant sample points $t_i = ih$. Evaluation of (3.5) with a k^{th} order truncation of (3.8) then gives

$$f(t_i, y(t_i)) = \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)(t_i) \cong \frac{1}{h}\left(\Delta - \frac{1}{2}\Delta^2 + \frac{1}{3}\Delta^3 - \cdots - (-1)^k \frac{1}{k}\Delta^k\right).$$

Euler forward scheme. For $k = 1$, the resulting scheme is

$$\frac{1}{h}\Delta y_i = \frac{y_{i+1} - y_i}{h} = f(t_i, y_i) = f_i \Longrightarrow y_{i+1} = y_i + h f_i,$$

where $f_i \cong f(t_i, y(t_i))$, and is known as the Euler forward scheme. New values are obtained from previous values. Such methods are said to be *explicit schemes*. As to be expected from the truncation of (3.8) to the first term in the series, the scheme is first-order accurate. This can be formally established by evaluation of the error at step i

$$e_i = y(t_i) - y_i.$$

At the next step, $e_{i+1} = y(t_{i+1}) - y_i$, and subtraction of the two errors gives upon Taylor-series expansion

$$e_{i+1} - e_i = y(t_{i+1}) - y(t_i) - (y_{i+1} - y_i) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(\xi_i) - y(t_i) - hf_i.$$

Since $f_i = f(t_i, y(t_i))$, the one-step error is given by

$$\tau_i = e_{i+1} - e_i = \frac{h^2}{2}y''(\xi_i).$$

After N steps,

$$e_N - e_0 = \frac{h^2}{2}\sum_{i=1}^{N} y''(\xi_i).$$

Assuming $e_0 = 0$ (exact representation of the initial condition),

$$e_N \leqslant \frac{Nh^2}{2}\|y''\|_\infty.$$

Numerical solution of the initial value problem is carried out over some finite interval $[0, T]$, with $T = Nh$, hence

$$e_N \leqslant h\frac{T}{2}\|y''\|_\infty = O(h), \tag{3.9}$$

indeed with first-order convergence.

Alternatively, one could use the backward or centered finite difference approximations of the derivative

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{1}{h}\left(\nabla + \frac{1}{2}\nabla^2 + \frac{1}{3}\nabla^3 + \cdots\right) = \frac{1}{h}\left(\delta - \frac{1}{24}\delta^3 + \frac{3}{640}\delta^5 - \cdots -\right). \tag{3.10}$$

Backward Euler scheme. Truncation of the backward operator at first order gives

$$f(t_i, y(t_i)) = \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)_i \cong \frac{1}{h}(\nabla y)_i = \frac{y_i - y_{i-1}}{h} \Longrightarrow y_i = y_{i-1} + hf_i = y_{i-1} + hf(t_i, y_i).$$

Note now that the unknown value y_i appears as an argument to f, with $f_i = f(t_i, y_i)$, the approximation of the exact slope $f(t_i, y(t_i))$. Some procedure to solve the equation

$$y_i - y_{i-1} - hf(t_i, y_i) = 0,$$

must be introduced in order to advance the solution from t_{i-1} to t_i. Such methods are said to be *implicit schemes*. The same type of error analysis as in the forward Euler case again leads to the conclusion that the one-step error is $O(h^2)$, while the overall error over a finite interval $[0, T]$ satisfies (3.9), and is first-order.

Leapfrog scheme. Truncation of the centered operator at first order gives

$$f(t_i, y(t_i)) = \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)_i \cong \frac{1}{h}(\delta y)_i = \frac{y_{i+1/2} - y_{i-1/2}}{h} \Longrightarrow y_{i+1/2} = y_{i-1/2} + hf_i = y_{i-1/2} + hf(t_i, y_i).$$

The higher-order accuracy of the centered finite differences leads to a more accurate numerical solution of the problem (3.7). The one-step error is third-order accurate,

$$e_{i+1/2} - e_{i-1/2} = y(t_{i+1/2}) - y(t_{i-1/2}) + hf(t_i, y_i) = \frac{h^3}{3}y'''(\xi_i),$$

and the overall error over interval $[0, T = Nh]$ is second-order accurate

$$e_N \leqslant \frac{h^2}{3} T \|y'''\|_\infty .$$

LECTURE 22: ORDINARY DIFFERENTIAL EQUATIONS - MULTISTEP METHODS

1. Adams-Bashforth and Adams-Moulton schemes

Consider now the approximation of $\mathcal{I} = f$ in the first-order differential equation

$$y' = f(t, y). \tag{3.11}$$

Integration over a time step $[t_i, t_{i+1}]$ gives

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t))\, \mathrm{d}t,$$

and use of quadrature formulas leads to numerical solutions for solving (3.11). Consider for instance data $\mathcal{D} = \{(t_{i+1-k}, f_{i+1-k}), k = 1, \ldots, s\}$ going back s intervals of size h, $t_{i+1-k} = t_{i+1} - kh$. Any quadrature formula based on this data could be used, but the most often encountered approach is to use a polynomial approximant. This can be stated in Lagrange form as

$$f(t, y(t)) \cong \sum_{k=1}^{s} \ell_k(t)\, f_k,\ f_k = f(t_{i+1-k}, y(t_{i+1-k})) \cong f(t_{i+1-k}, y_{i+1-k}).$$

The last approximate equality arises from replacing the exact value $y(t_{i+1-k})$ by its approximation $y_k \cong y(t_{i+1-k})$. The result is known as an Adams-Bashforth scheme

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} \sum_{k=1}^{s} \ell_k(t)\, f_k\, \mathrm{d}t = y_i + \sum_{k=1}^{s} \Big(\int_{t_i}^{t_{i+1}} \ell_k(t)\, \mathrm{d}t \Big) f_k = y_i + h \sum_{k=1}^{s} b_k f_{i+1-k},$$

with coefficients that are readily computed (cf. Table 1).

$$b_k = \frac{1}{h} \Big(\int_{t_i}^{t_{i+1}} \ell_k(t)\, \mathrm{d}t \Big).$$

s	b_1	b_2	b_3	b_4
1	1			
2	$\frac{3}{2}$	$-\frac{1}{2}$		
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$

Table 3.1. Adams-Bashforth scheme coefficients.

The $s = 1$ Adams-Bashforth scheme is identical to forward Euler and the above approach yields schemes that are explicit, i.e., the new value is directly obtained from knowledge of previous values.

Choosing data $\mathcal{D} = \{(t_{i+1-k}, f_{i+1-k}), k = 0, \ldots, s-1\}$ that contains the point yet to be computed (t_{i+1}, y_{i+1}) gives rise to a class of implicit schemes known as the Adams-Moulton schemes (Table 2)

$$y_{i+1} = y_i + h \sum_{k=0}^{s-1} b_k f_{i+1-k},$$

s	b_0	b_1	b_2	b_3
1	1			
2	$\frac{1}{2}$	$\frac{1}{2}$		
3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$	
4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$

Table 3.2. Adams-Moulton scheme coefficients.

2. Simultaneous operator approximation - linear multistep methods

Approximation of both operators $\mathcal{L} = \mathrm{d}/\mathrm{d}t$ and $\mathcal{F} = f$ arising in $\mathcal{L}y = \mathcal{F}y$, or $y' = f(t, y)$ is possible. Combining previous computations, the resulting schemes can be stated as

$$\sum_{k=0}^{s} a_k y_{i+k} = h \sum_{k=0}^{s} b_k f_{i+k},\ f_{i+k} = f(t_{i+k}, y_{i+k}). \tag{3.12}$$

Both sides arise from linear approximants: of the derivative on the left, and of f on the right.

3. Consistency, convergence, stability

Any of the above schemes defines a sequence $\{y_n\}_{n\in\mathbb{N}}$ that approximates the solution $y(t_n)$ of the initial value problem

$$y' = f(t, y), y(0) = y_0,$$

over a time interval $[0, T]$, $t_n = nh$, $h = T/N$. A scheme is said to be convergent if

$$\lim_{\substack{h\to 0 \\ Nh=T}} y_N = y(T).$$

The above states that in the limit of taking small step sizes while maintaining $Nh = T$ for some finite time T, the estimate at the endpoint converges to the exact value. Such a definition is rather difficult to apply directly, and an alternative characterization of convergence is desirable.

To motivate the overall approach, consider first the following model problem

$$y' = \lambda y, y(0) = y_0, \lambda \leqslant 0 \tag{3.13}$$

The model problem arises from truncation of the general non-linear function f to first order

$$y' = f(y) = f(0) + f'(0)\, y + \cdots.$$

Since $f(0)$ is a constant that simply leads to linear growth, and the model problem captures the lowest-order non-trivial behavior. The exact solution is

$$y(t) = e^{\lambda t} y_0 \Longrightarrow y(t_n) = e^{n\lambda h} y_0,$$

giving $y(T) = e^{\lambda T} y_0$. The restriction of $\lambda \leqslant 0$ in the model problem arises from consideration of the effect of a small perturbation in the initial condition representative of floating point representation errors. This leads to $\tilde{y}(T) = e^{\lambda T}(y_0 + \delta)$, and the error $\varepsilon = \tilde{y}(T) - y(T) = e^{\lambda T}\delta$ can only be maintained small if $\lambda \leqslant 0$.

Applying the forward Euler scheme to the model problem (3.13) gives

$$y_{n+1} = y_n + \lambda h y_n = (1+z) y_n,$$

with $z = \lambda h$. After N steps the numerical approximation is

$$y_N = (1+z)^N y_0.$$

The exponential decay of the exact solution can only be recovered if which leads to a restriction on the allowable step size

$$-\frac{2}{\lambda} > h > 0.$$

If the step size is too large, $h > -2/\lambda$, inherent floating point errors are amplified by the forward Euler method, and the scheme is said to be *unstable*. This is avoided by choosing a subunitary parameter z , $|z| = |\lambda h| \leqslant 1$, which leads to a step size restriction $h < 1/|\lambda|$.

These observations on the simple case of the Euler forward method generalize to linear multistep methods. Applying (3.12) to the model problem (3.13) leads to the following linear finite difference equation

$$\sum_{k=0}^{s} a_k y_{i+k} = z \sum_{k=0}^{s} b_k y_{i+k}\,. \tag{3.14}$$

The above is solved using a procedure analogous to that for differential equations by hypothesizing solutions of the form

$$y_n = r^n,$$

to obtain a characteristic equation

$$\pi(r;z) = \rho(r) - z\sigma(r) = 0,$$

where $\rho(r), \sigma(r)$ are polynomials

$$\rho(r) = \sum_{k=0}^{s} a_k r^k, \sigma(r) = \sum_{k=0}^{s} b_k r^k.$$

The above polynomials allow an operational assessment of algorithms of form (3.12). An algorithm (3.12) that recovers the ordinary differential equation (3.11) in the limit of $h \longrightarrow 0$ is said to be *consistent*, which occurs if and only if

$$\rho(1) = 0, \rho'(1) - \sigma(1) = 0.$$

Furthermore an algorithm of form (3.12) that does not amplify inherent floating point errors is said to be stable, which occurs if the roots of $\pi(r;z)$ are subunitary in absolute value

$$|r_j| < 1, \pi(r_j; z) = 0.$$

THEOREM. *An algorithm to solve (3.11) that is consistent and stable is convergent.*

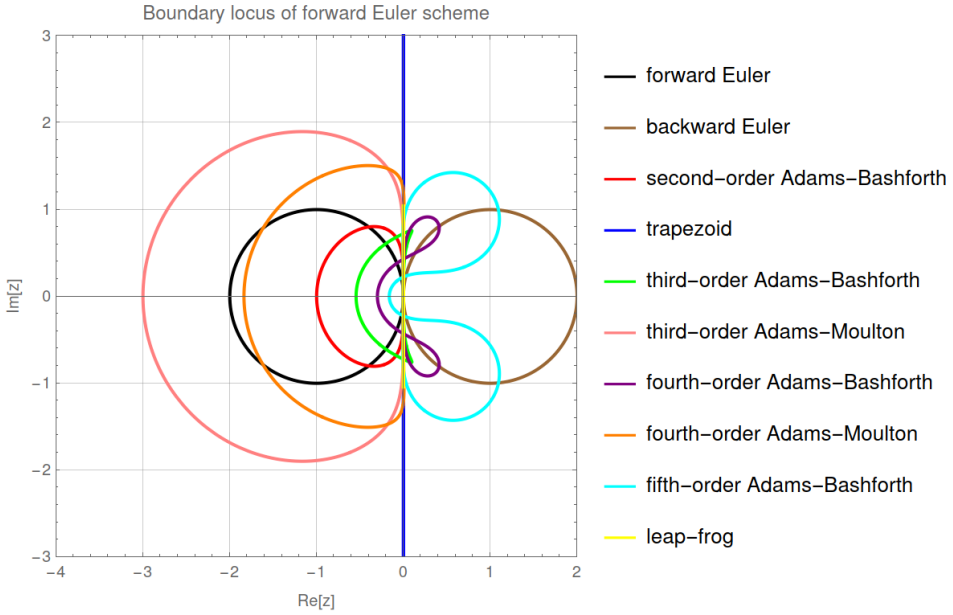
Boundary locus method. A convenient procedure to determine the stable range of step sizes is to consider r of unit absolute value

$$r = e^{i\theta},$$

and evaluate the characteristic equation

$$\pi(e^{i\theta};z)=\rho(e^{i\theta})-z\sigma(e^{i\theta})=0\Longrightarrow z(\theta)=\frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})},$$

where $z(\theta)$ is the boundary locus delimiting zones of stability in the complex plane (Fig 1).



A method is said to be A-stable if its region of stability contains the entire left half-plane in $\mathbb{C}$, and is said to be L-stable if $\lim_{\omega\to\infty}\rho(\omega e^{i\theta})/\sigma(\omega e^{i\theta})=0$.

LECTURE 23: NONLINEAR SCALAR OPERATOR EQUATIONS

1. Root-finding algorithms

The null space of a linear mapping represented through matrix $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ is defined as $N(\boldsymbol{A})=\{\boldsymbol{x}|\,\boldsymbol{A}\boldsymbol{x}=\boldsymbol{0}\}$, the set of all points that have a null image through the mapping. The null space is a vector subspace of the domain of the linear mapping. A first step in the study of nonlinear mappings is to consider the generalization of the concept of a null set, starting with the simplest case,

$$f(x)=0 \tag{3.15}$$

where $f:\mathbb{R}\longrightarrow\mathbb{R}$, $f\in C^p(\mathbb{R})$, $p\geqslant 0$, i.e., f has p continuous derivatives. It is assumed that a closed form analytical solution is not available, and algorithms are sought to construct an approximating sequence $\{x_n\}_{n\in\mathbb{N}}$ whose limit is a root of (3.15). The general approach is to replace (3.15) with

$$g_n(x)=0, \tag{3.16}$$

where g_n is some approximation of f, and x_n the root of (3.16) can be easily determined.

1.1. First-degree polynomial approximants

Secant method. Consider $g(x) = ax + b$ (a linear function, but not a linear mapping for $b \neq 0$), an approximant of f based upon data $\{(x_{n-2}, f_{n-2} = f(x_{n-2})), (x_{n-1}, f_{n-1} = f(x_{n-1}))\}$, given in Newton interpolant form by

$$g_n(x) = f_{n-2} + \frac{f_{n-1} - f_{n-2}}{x_{n-1} - x_{n-2}}(x - x_{n-2}). \tag{3.17}$$

The solution of (3.17) is

$$x_n = x_{n-2} - \frac{f_{n-2}}{f_{n-1} - f_{n-2}}(x_{n-1} - x_{n-2}) = \frac{x_{n-2} f_{n-1} - x_{n-1} f_{n-2}}{f_{n-1} - f_{n-2}},$$

an iteration known as the secant method. The error in root approximation is

$$e_n = x_n - x = e_{n-2} - \frac{f_{n-2}}{f_{n-1} - f_{n-2}}(e_{n-1} - e_{n-2}),$$

and can be estimated by Taylor series expansions around the root x for which $f(x) = 0$,

$$f_{n-k} = f(x_{n-k}) = f'(x)(x_{n-k} - x) + \frac{1}{2}f''(x)(x_{n-k} - x)^2 + \cdots = f'\, e_{n-k} + \frac{1}{2}f''\, e_{n-k}^2 + \cdots,$$

where derivatives f', f'' are assumed to be evaluated at x. In the result

$$e_n = e_{n-2} - \frac{f'\, e_{n-2} + \frac{1}{2}f''\, e_{n-2}^2 + \cdots}{f' \cdot (e_{n-1} - e_{n-2}) + \frac{1}{2}f'' \cdot (e_{n-1}^2 - e_{n-2}^2) + \cdots}(e_{n-1} - e_{n-2}) = e_{n-2}\left[1 - \frac{f' + \frac{1}{2}f'' \cdot e_{n-2} + \cdots}{f' + \frac{1}{2}f'' \cdot (e_{n-1} + e_{n-2}) + \cdots}\right],$$

assuming $f'(x) \neq 0$, (i.e., x is a simple root) gives

$$e_n = e_{n-2}\left[1 - \frac{1 + c \cdot e_{n-2} + \cdots}{1 + c \cdot (e_{n-1} + e_{n-2}) + \cdots}\right], c = \frac{1}{2}(f''/f').$$

For small errors, to first order the above can be written as

$$e_n = e_{n-2}[1 - (1 + c \cdot e_{n-2})(1 - c \cdot (e_{n-1} + e_{n-2}))] = c e_{n-1} e_{n-2}.$$

Assuming p-order convergence of e_n,

$$|e_n| \sim A\,|e_{n-1}|^p,$$

leads to

$$A^{p+1}\,|e_{n-2}|^{p^2} \sim cA\,|e_{n-2}|^{p+1} \Longrightarrow |e_{n-2}|^{p^2-p-1} \sim cA^{-p}.$$

Since c, A are finite while $e_n \longrightarrow 0$, the above asymptotic relation can only be satisfied if

$$p^2 - p - 1 = 0 \Longrightarrow p = \frac{1 + \sqrt{5}}{2} \cong 1.62,$$

hence the secant method exhibits superlinear, but subquadratic convergence.

Newton-Raphson method. A different linear approximant arises from the Hermite interpolant based on data

$$\{(x_{n-1}, f_{n-1} = f(x_{n-1}), f'_{n-1} = f'(x_{n-1}))\},$$

which is given in Newton form as

$$g_n(x) = f_{n-1} + f'_{n-1} \cdot (x - x_{n-1}),$$

with root

$$x_n = x_{n-1} - \frac{f_{n-1}}{f'_{n-1}}, \tag{3.18}$$

an iteration known as the Newton-Raphson method. The error is given by

$$e_n = x_n - x = e_{n-1} - \frac{f_{n-1}}{f'_{n-1}}. \tag{3.19}$$

Taylor series exapnsion around the root gives for small e_{n-1},

$$e_n = e_{n-1} - \frac{f' \cdot e_{n-1} + \frac{1}{2} f'' \cdot e_{n-1}^2 + \cdots}{f' + f'' e_{n-1} + \cdots} = e_{n-1}\left[1 - \frac{1 + c e_{n-1} + \cdots}{1 + 2c e_{n-1} + \cdots}\right] \approx e_{n-1}[1 - (1 + c e_{n-1})(1 - 2c e_{n-1})].$$

The resulting expression

$$e_n \approx c e_{n-1}^2 = \frac{1}{2}\frac{f''}{f'} e_{n-1}^2, \tag{3.20}$$

states quadratic convergence for Newton's method. This faster convergence than the secant method requires however knowledge of the derivative, and the computational expense of evaluating it.

The above estimate assumes convergence of $\{x_n\}_{n\in\mathbb{N}}$, but this is not guaranteed in general. Newton's method requires an accurate initial approximation x_0, within a neighborhood of the root in which f is increasing, $f' > 0$, and convex, $f'' > 0$. Equivalently, since roots of f are also roots of $-f$, Newton's method converges when f', $f'' < 0$. In both cases (3.20) in the prior iteration states that $e_{n-1} = x_{n-1} - r > 0$, hence $x_{n-1} > r$. Since f is increasing $f(x_{n-1}) > f(r) = 0$, hence (3.19) implies $e_n < e_{n-1}$. Thus the sequence $\{e_n\}_{n\in\mathbb{N}}$ is decreasing and bounded below by zero, hence $\lim_{n\to\infty} e_n = 0$, and Newton's method converges.

1.2. Second-degree polynomial approximants

An immediate extension of the above approach is to increase the accuracy of the approximant by seeking a higher-degree polynomial interpolant. The expense of the resulting algorithm increases rapidly though, and in practice linear and quadratic approximants are the most widely used. Consider the Hermite interpolant based on data

$$\{(x_{n-1}, f_{n-1} = f(x_{n-1}), f'_{n-1} = f'(x_{n-1}),, f''_{n-1} = f''(x_{n-1}))\},$$

given in Newton form as

$$g_n(x) = f_{n-1} + f'_{n-1} \cdot (x - x_{n-1}) + \frac{1}{2} f''_{n-1} \cdot (x - x_{n-1})^2 = C + Bs + A s^2,$$

with roots

$$x_n = x_{n-1} + \frac{-f'_{n-1} \pm \sqrt{(f'_{n-1})^2 - 2 f_{n-1} f''_{n-1}}}{f''_{n-1}}.$$

Tha above exhibits the difficulties arising in higher-order interpolants. The iteration requires evaluation of a square root, and checking for a positive discriminant.

Halley's method. Algebraic manipulations can avoid the appearance of radicals in a root-finding iteration. As an example, Halley's method

$$x_n = x_{n-1} - \frac{2 f_{n-1} f'_{n-1}}{2(f'_{n-1})^2 - f_{n-1} f''_{n-1}},$$

exhibits cubic convergence.

2. Composite approximations

The secant iteration

$$x_n = x_{n-2} - \frac{f_{n-2}}{f_{n-1} - f_{n-2}}(x_{n-1} - x_{n-2}) = x_{n-2} - \frac{f_{n-2}}{\dfrac{f_{n-1} - f_{n-2}}{x_{n-1} - x_{n-2}}},$$

in the limit of $x_{n-2} \longrightarrow x_{n-1}$ recovers Newton's method

$$x_n = x_{n-1} - \frac{f_{n-1}}{f'_{n-1}}.$$

This suggests seeking advantageous approximations of the derivative

$$x_n = x_{n-1} - \frac{f_{n-1}}{\dfrac{f(x_{n-1} + h_{n-1}) - f(x_{n-1})}{h_{n-1}}},$$

based upon some step-size sequence $\{h_n\}$. Since $f(x_n) \longrightarrow 0$, the choice $h_{n-1} = f(x_{n-1})$ suggests itself, leading to Steffensen's method

$$x_n = x_{n-1} - \frac{f_{n-1}}{\dfrac{f(x_{n-1} + f(x_{n-1})) - f(x_{n-1})}{f(x_{n-1})}} = x_{n-1} - \frac{f_{n-1}}{g_{n-1}}, g_{n-1} = \frac{f(x_{n-1} + f(x_{n-1}))}{f(x_{n-1})} - 1.$$

Steffensen's method exhibits quadratic convergence, just like Newton's method, but does not require knowledge of the derivative. The higher order by comparison to the secant method is a direct result of the derivative approximation

$$f'(x_{n-1}) \cong \frac{f(x_{n-1} + f(x_{n-1})) - f(x_{n-1})}{f(x_{n-1})},$$

which, remarkably, utilizes a composite approximation

$$f(x_{n-1} + f(x_{n-1})) = (f \circ (1 + f))(x_{n-1}).$$

Such composite techniques are a prominent feature of various nonlinear approximations such as a k-layer deep neural network $\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{I}_k \circ \boldsymbol{I}_{k-1} \circ \ldots \circ \boldsymbol{I}_1)(\boldsymbol{x})$.

3. Fixed-point iteration

The above iterative sequences have the form

$$x_n = F(x_{n-1}),$$

and the root is a fixed point of the iteration

$$x = F(x).$$

For example, in Newton's method

$$F(x) = x - \frac{f(x)}{f'(x)},$$

and indeed at a root $x = F(x)$. Characterization of mappings F that lead to convergent approximation sequences is of interest and leads to the following definition and theorem.

DEFINITION. *A function $F\colon [a,b] \longrightarrow [a,b]$ is said to be a contractive mapping if $\forall x, y \in [a,b]$ there exists $c \in (0,1)$ such that*

$$|F(x) - F(y)| \leqslant c\,|x - y|.$$

THEOREM. *(Contractive Mapping theorem). If $F\colon [a,b] \longrightarrow [a,b]$ is a contractive mapping then F has a unique fixed point $x \in [a,b]$, $x = F(x)$.*

The fixed point theorem is an entry point to the study of non-additive approximation sequences.

Example 3.1. The sequence

$$x_1 = \sqrt{p}, x_2 = \sqrt{p + \sqrt{p}}, \ldots \ (p > 0)$$

is expressed recursively as

$$x_{n+1} = \sqrt{p + x_n},$$

and has the limit

$$x = \sqrt{p + \sqrt{p + \sqrt{p + \cdots}}},$$

that is the fixed point of F,

$$x = F(x) = \sqrt{p + x} = \frac{1 + \sqrt{1 + p}}{2}.$$

Over the interval $[0, p+1]$, F is a contraction since

$$F'(x) = \frac{1}{2\sqrt{p + x}} \leqslant \frac{1}{2\sqrt{p}} < 1.$$

Example 3.2. The sequence

$$x_1 = \frac{1}{p}, x_2 = \frac{1}{p + \frac{1}{p}}, \dots \ (p > 0)$$

is expressed recursively as

$$x_{n+1} = \frac{1}{p + x_n},$$

and has the limit

$$x = \frac{1}{p + \frac{1}{p + \cdots}},$$

that is the fixed point of F,

$$x = F(x) = \frac{1}{p + x} = \frac{-p + \sqrt{p^2 + 1}}{2}.$$

Over the interval $[0, 1]$, F is a contraction since

$$|F'(x)| = \frac{1}{(p + x)^2} \leqslant \frac{1}{p^2} < 1.$$

CHAPTER 4

NONLINEAR OPERATOR APPROXIMATION

LECTURE 24: NONLINEAR VECTOR OPERATOR EQUATIONS

1. Multivariate root-finding algorithms

Consider now nonlinear finite-dimensional mappings $\boldsymbol{f}:\mathbb{R}^d \longrightarrow \mathbb{R}^d$, and the root-finding problem

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \tag{4.1}$$

whose set of solutions generalize the linear mapping concept of a null space, $N(\boldsymbol{A}) = \{\boldsymbol{x} | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{A} \in \mathbb{C}^{d\times d}\}$. As in the scalar-valued case, algorithms are sought to construct an approximating sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ whose limit is a root of (4.1), by approximating $\boldsymbol{f}$ with $\boldsymbol{g}_k$, and solving

$$\boldsymbol{g}_k(\boldsymbol{x}) = 0. \tag{4.2}$$

Multivariate approximation is however considerably more complex than univariate approximation. For example, consider $d = 2$, $\boldsymbol{f}:\mathbb{R}^2 \longrightarrow \mathbb{R}^2$, and the univariate monomial interpolants in Lagrange form

$$\mathcal{L}_t \boldsymbol{f}(s,t) = \sum_{i=0}^{m} \boldsymbol{f}(x_i,t)\, l_i^x(s), \mathcal{L}_s \boldsymbol{f}(s,t) = \sum_{j=0}^{n} \boldsymbol{f}(s,y_j)\, l_j^y(t),$$

with

$$l_i^x(s) = \prod_{k=0}^{m\,'} \frac{s-x_k}{x_i-x_k}, l_j^y(s) = \prod_{l=0}^{n\,'} \frac{t-y_l}{y_j-y_l}.$$

The operator $\mathcal{L}_t$ carries out interpolation at fixed t value of the data set $\mathcal{D}_x = \{(x_i, \boldsymbol{f}(x_i,t)), i = 0,\ldots,m\}$. Similarly, operator $\mathcal{L}_s$ carries out interpolation at fixed s value of the data set $\mathcal{D}_y = \{(y_j, \boldsymbol{f}(s,y_j)), j = 0,\ldots,n\}$. Multivariate interpolation of the data set

$$\mathcal{D} = \{(x_i, y_j, \boldsymbol{f}(x_i,y_j)), i = 0,\ldots,m, j = 0,\ldots,n\},$$

can be carried out through multiple operator composition procedures.

Operator product. Define $\mathcal{L} = \mathcal{L}_t \otimes \mathcal{L}_s$ as

$$\mathcal{L}\boldsymbol{f}(s,t) = (\mathcal{L}_t\mathcal{L}_s)\boldsymbol{f}(s,t) = \mathcal{L}_t(\mathcal{L}_s\boldsymbol{f}(s,t)) = \mathcal{L}_t\left(\sum_{i=0}^{m} \boldsymbol{f}(x_i,t)\, l_i^x(s)\right) = \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{f}(x_i,y_j)\, l_i^x(s)\, l_j^y(t).$$

Operator Boolean sum. Define $\mathcal{L} = \mathcal{L}_t \oplus \mathcal{L}_s$ as $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s - \mathcal{L}_t \mathcal{L}_s$

$$\mathcal{L}\boldsymbol{f}(s,t) = \sum_{i=0}^{m} \boldsymbol{f}(x_i,t)\, l_i^x(s) + \sum_{j=0}^{n} \boldsymbol{f}(s,y_j)\, l_j^y(t) - \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{f}(x_i,y_j)\, l_i^x(s)\, l_j^y(t).$$

1.1. First-degree polynomial approximants

Secant method. Bivariate ($d = 2$) root-finding algorithms already exemplifies the additional complexity in constructing root finding algorithms. The goal is to determine a new approximation (x_k, y_k) from the prior approximants

$$(x_0,y_0),\ldots,(x_{k-2},y_{k-2}),(x_{k-1},y_{k-1})\,.$$

Whereas in the scalar case two prior points allowed construction of a linear approximant, the two points in data

$$\mathcal{D} = \{(x_{k-2},y_{k-2}),(x_{k-1},y_{k-1})\}$$

are insufficient to determine

$$\mathcal{L}\boldsymbol{f} = \sum_{i=k-2}^{k-1}\sum_{j=k-2}^{k-1} \boldsymbol{f}(x_i,y_j)\, l_i^x(s)\, l_j^y(t),$$

which requires four data points. Various approaches to exploit the additional degrees of freedom are available, of which the class of quasi-Newton methods finds widespread applicability.

Newton, quasi-Newton methods. A linear multivariate approximant in d dimensions requires 2^d data. A Hermite interpolant based upon function and partial derivative values can be constructed, but it is more direct to truncate the multivariate Taylor series

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_k) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_k)(\boldsymbol{x}-\boldsymbol{x}_k) + \cdots,$$

where

$$\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_d} \\ \vdots & \vdots & \ddots & \\ \frac{\partial f_d}{\partial x_1} & \frac{\partial f_d}{\partial x_2} & \cdots & \frac{\partial f_d}{\partial x_d} \end{bmatrix} = \nabla \boldsymbol{f},$$

is the Jacobian matrix of $\boldsymbol{f}$. Setting $\boldsymbol{f}(\boldsymbol{x}_{k+1}) = \boldsymbol{0}$, as the condition for the next iterate leads to the update

$$\boldsymbol{J}(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1}-\boldsymbol{x}_k) = -\boldsymbol{f}(\boldsymbol{x}_k),$$

a linear system that is solved at each iteration. Computation of the multiple partial derivatives arising in the Jacobian might not be possible or too expensive, hence approximations are sought $\boldsymbol{B}_k \cong \boldsymbol{J}(\boldsymbol{x}_k)$, similar in principle to the approximation of a tangent by a secant. In such quasi-Newton methods, a secant condition on $\boldsymbol{B}_k$ is stated as

$$\boldsymbol{B}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) = \boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_{k-1}),$$

and corresponds to a truncation of the Taylor series expansion around $\boldsymbol{x}_{k-1}$. The above secant condition is not sufficient by itself to determine $\boldsymbol{B}_k$, hence additional considerations can be imposed.

1. Recalling that the scalar Newton method for finding roots of $f(x) = 0$ converges in a region where $f', f'' > 0$, imposing analogous behavior for $\boldsymbol{B}_k$ suggests itself. This is typically done by requiring $\boldsymbol{B}_k$ to be symmetric positive definite.

2. Assuming convergence of the approximating sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ to a root, $\boldsymbol{B}_{k+1}$ should be close to the previous approximation suggesting the condtion

$$\min_{\boldsymbol{B}_{k+1}} \|\boldsymbol{B}_{k+1} - \boldsymbol{B}_k\|.$$

   Various algorithms arise from a particular choice of norm and procedure to apply (2).

One widely used quasi-Newton method, arising from a rank-two update at each iteration to maintain positive definiteness, is the Broyden-Fletcher-Goldfard-Shanno update

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} - \frac{\boldsymbol{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \boldsymbol{B}_k^T}{\boldsymbol{s}_k^T \boldsymbol{B}_k \boldsymbol{s}_k},$$

where the updates are determined by

1. Solving $\boldsymbol{B}_k \boldsymbol{p}_k = -[\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_{k-1})]$ to find a search direction $\boldsymbol{p}_k$;

2. Finding the distance along the search direction by $\alpha_k = \operatorname{argmin} \|\boldsymbol{f}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)\|_2$;

3. Updating the approximation $\boldsymbol{s}_k = \alpha_k \boldsymbol{p}_k$, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

4. Computing $\boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}(\boldsymbol{x}_k)$.

LECTURE 25: INTRODUCTION TO NONLINEAR APPROXIMATION

4.1. HISTORICAL ANALOGUES

4.1.1. Operator calculus

4.1.1.1. Heavisde study of telgraphist equation

In late nineteenth century, telegrapher's equations, a system of linear PDEs for current $I(x,t)$ and voltage $V(x,t)$

$$\frac{\partial}{\partial x}V(x,t) = -L\frac{\partial}{\partial t}I(x,t) - RI(x,t)$$

$$\frac{\partial}{\partial x}I(x,t) = -C\frac{\partial}{\partial t}C(x,t) - GV(x,t)$$

Heaviside avoided solution of the PDEs by reduction to an algebraic formulation historical formulation, e.g., for the ODE for $y(t)$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + ay = b$$

Heaviside considered the associated algebraic problem for $Y(s)$

$$sY + aY = b \Longrightarrow Y(s) = \frac{b}{a+s} \Longrightarrow y(t) = \mathscr{L}^{-1}[Y(s)]$$

4.1.1.2. Development of mathematical theory of operator calculus

> Why should I refuse a good dinner simply because I don't understand the digestive processes involved? (Heaviside, ?)

Heaviside's formal framework (1890's) for solving ODEs was discounted since it lacked mathematical rigour.

- Russian mathematician 1920's established first results (Vladimirov)
- Theory of Distributions (Schwartz, 1950s)

4.2. BASIC APPROXIMATION THEORY

4.2.1. Problem definition

Consider function $f:\mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g:\mathbb{R}^n \longrightarrow \mathbb{R}$, $T:\mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\|f - g \circ T\| < \varepsilon$$

for some $\varepsilon > 0$.

4.2.1.1. Linear approximation example

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} (f, \phi_j)\phi_j = \sum_{j=1}^{n} c_j \phi_j$$

The approximation is convergent if

$$\lim_{n\to\infty} \|f - g \circ T\| = 0,$$

This assumes $c_j = (f, \phi_j)$ rapidly decrease.

THEOREM. *(Parseval) The Fourier transform is unitary. For $A, B\colon \mathbb{R} \to \mathbb{C}$, square integrable, 2π-periodic with Fourier series*

$$A(t) = \sum_{n=-\infty}^{\infty} a_n e^{int}, B(t) = \sum_{n=-\infty}^{\infty} b_n e^{int},$$

$$\sum_{n=-\infty}^{\infty} a_n \bar{b}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} A(t)\bar{B}(t)\,\mathrm{d}t.$$

Bessel inequality:

$$\sum_{j=1}^{n} |(f, \phi_j)|^2 \leq \|f\|_2.$$

Fourier coefficient decay: for $f \in C^{(k-1)}(\mathbb{R})$, $f^{(k-1)}$ absolutely continuous,

$$|c_n| \leq \min_{0\leq j\leq k} \frac{\|f^{(j)}\|_1}{|n|^j}.$$

In practice: coefficients decay as

- $1/n$ for functions with discontinuities on a set of Lebesgue measure 0;
- $1/n^2$ for functions with discontinuous first derivative on a set of Lebesgue measure 0;

- $1/n^3$ for functions with discontinuous second derivative on a set of Lebesgue measure 0.

Fourier coefficients for analytic functions decay faster than any monomial power $c_n = o(n^{-p}), \forall p \in \mathbb{N}$, a property known as exponential convergence.

Denote such approximations by $\mathcal{L}$, and they are linear

$$\mathcal{L}(\alpha f + \beta g) = \alpha \mathcal{L}(f) + \beta \mathcal{L}(g)$$

-

4.2.1.2. Non-Linear approximation example

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j$$

Let $\Phi_n = \{\varphi_{k(1)}, \varphi_{k(2)}, \dots, \varphi_{k(n)}\}$ such

$$(f, \varphi_{k(1)}) \geqslant (f, \varphi_{k(2)}) \geqslant \cdots \geqslant (f, \varphi_{k(n)}).$$

Choose $c_j = (f, \varphi_{k(j)})$, and

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j .$$

Denote such approximations by $\mathcal{G}$, and they are non-linear.

4.2.2. Nonlinear approximation by composition

Consider function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g : \mathbb{R}^n \longrightarrow \mathbb{R}$, $T : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\| f - g \circ T \| < \varepsilon$$

for some $\varepsilon > 0$.

What questions do you ask?

Does T exist?. $\forall f, \varepsilon, \exists T, \text{such that} \|f - g \circ T\| < \varepsilon$

Can arbitrary ε be achieved?.

Can we construct T?.

→ By what procedure?

$$T = T_1 \circ T_2 \circ \ldots \circ T_J$$

with T_i simple modifications of identity (ReLU)

$$\min_{T_1, \ldots T_J} \|f - g \circ T_1 \circ T_2 \circ \ldots \circ T_J\|$$

$$\boldsymbol{T}_j(\boldsymbol{x}) = \eta(\boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{b}_j)$$

$$\eta(t) = \begin{cases} 0 & t < 0 \\ t & t \geqslant 0 \end{cases}$$

→ At what cost?

How big is n?.

.

LECTURE 26: DATA-DRIVEN BASES

CHAPTER 5
DIFFERENTIAL CONSERVATION LAWS

LECTURE 27: DIFFERENTIAL CONSERVATION LAWS

1. The relevance of physics for scientific computation

Efficient algorithms often arise from the specifities of an underlying application domain, perhaps none more so than those inspired from physics. Classical physics can be derived from a remarkably small set of experimentally verified postulates.

- The *least action principle* asserts that a physical system can be described by a function $L(t, q, \dot{q})$ of the system generalized coordinates $q(t)$ and velocities $\dot{q}(t) = \mathrm{d}q/\mathrm{d}t$, known as the *Lagrangian*, itself the difference of the system's kinetic and potential energy $L = K - U$. The time evolution of the system is known as the system's trajectory $(q(t), \dot{q}(t))$, and of all possible trajectories consistent with system constraints the trajectory actually followed by the system from initial time t_0 to final time t_1 minimizes a functional known as the *action S*

$$S(q, \dot{q}) = \int_{t_0}^{t_1} L(t, q(t), \dot{q}(t))\, \mathrm{d}t.$$

Example. However complex a physical system might be, application of the least action principle follows the procedure exemplified here for a simple mass-spring system. A point mass m attached to a spring of stiffness k is at distance $q(t)$ away from the equilibrium position $q = 0$. For constant m, this harmonic oscillator motion is described by the differential system

$$\frac{\mathrm{d}}{\mathrm{d}t}(mv) = -kq \Longrightarrow \frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{k}{m}q, \frac{\mathrm{d}q}{\mathrm{d}t} = v.$$

A state of this system is given by the values for position and velocity (q, v), and the above equations specify the time evolution of the system. Denoting the velocity as $v = \dot{q}$, $\mathrm{d}v/\mathrm{d}t = \ddot{q}$, and eliminating v gives the familiar

$$m\ddot{q} + kq = 0. \tag{5.1}$$

The same equation also results from the minimization of the action $S(q, \dot{q})$ of the Lagrangian

$$L(q, \dot{q}) = \frac{1}{2}m\dot{q}^2 - \frac{1}{2}kq^2, \tag{5.2}$$

with $K = m\dot{q}^2/2$, $U = kq^2/2$. The minimization is performed over all trajectories $(q(t), \dot{q}(t))$ with the same end-point values at t_0, t_1. Let the δ operator denote a small change in a trajectory. Since all trajectories have the same endpoints $\delta q(t_0) = \delta q(t_1) = 0$. The change in the action is

$$\delta S = \int_{t_0}^{t_1} \delta L(t, q(t), \dot{q}(t))\, \mathrm{d}t = \int_{t_0}^{t_1} \left(\frac{\partial L}{\partial q}\delta q + \frac{\partial L}{\partial \dot{q}}\delta \dot{q} \right) \mathrm{d}t.$$

Consider changes in overall trajectory to be independent of time such that the δ and $\mathrm{d}/\mathrm{d}t$ operators commute, and apply integration by parts

$$\int_{t_0}^{t_1}\left[\frac{\partial L}{\partial \dot{q}}\,\delta\left(\frac{\mathrm{d}q}{\mathrm{d}t}\right)\right]\mathrm{d}t=\int_{t_0}^{t_1}\left[\frac{\partial L}{\partial \dot{q}}\,\frac{\mathrm{d}}{\mathrm{d}t}(\delta q)\right]\mathrm{d}t=\left[\frac{\partial L}{\partial \dot{q}}\,\delta q\right]_{t_0}^{t_1}-\int_{t_0}^{t_1}\left[\delta q\,\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right)\right]\mathrm{d}t=-\int_{t_0}^{t_1}\left[\delta q\,\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right)\right]\mathrm{d}t.$$

For S to be at a minimum the change in the action must be stationary $\delta S=0$,

$$\int_{t_0}^{t_1}\left[\frac{\partial L}{\partial q}-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right)\right]\delta q\,\mathrm{d}t=0.$$

For the above to be valid for all δq the equation

$$\frac{\partial L}{\partial q}-\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right)=0$$

must hold, and replacing (5.2) recovers (5.1).

- One class of constraints are the *conservation laws*, the experimental observation that certain quantities remain constant during the system's evolution. Classical mechanics identifies three conserved quantities: mass, momentum, and energy. It is a matter of personal preference to also consider conservation of angular momentum as fundamental or as a consequence of conservation of linear momentum. Classical electrodynamics adds conservation of electric charge, while quantum mechanics also defines conservation of certain microscopic quantities known as *quantum numbers* such as the baryon or lepton numbers.

Other constraints often refer to allowed spatial positions and are known generically as geometric constraints. Note that this is an idealization: in reality some other physical system M is interacting with the one being considered P and it is assumed that the system M is so much larger that its position does not change. Such idealization or modeling assumptions are often encountered. As another important example, the system P may exhibit energy dissipation, such as the decrease of an object's momentum due to friction. Energy is indeed lost from system P to the surrounding medium M, but the overall energy of the combined system $M+P$ is conserved.

Scientific computation uses many concepts and terms from physics, such as the characterization of a numerical scheme for differential equations as being “conservative”, in the sense of maintaining the conserved physical quantities. Also, a remarkably large number of efficient algorithms arise from the desire to mimic physical properties. Conservation laws can be stated for a small enough spatial domain that it can be considered to be infinitesimal in the sense of calculus. In this case differential conservation laws are obtained. Alternatively, consideration of a finite-sized spatial domain leads to integral formulation of the conservation laws. In a large class of physical systems of current research interest models are constructed in which the evolution of the system P depends on the history of interactions with the surrounding medium M. Such systems are described by integro-differential laws, elegantly expressed through fractional derivatives.

2. Conservation laws

Banking example. Conservation of some physical quantity is stated for a hypothesized *isolated* system. In reality no system is truly isolated and the most interesting applications come about from the study of interaction between two or more systems. This leads to the question of how one can follow the changes in physical quantities of the separate systems. An extremely useful procedure is to set up an accounting procedure. A mundane but illuminating example is quantity of Euros E in a building B. If the building is a commonplace one, it is to be expected that when completely isolated, the amount of currency in the building is fixed

$$E=E_0\,. \tag{5.3}$$

E_0 is some constant. Equation (5.3) is self-evident but not particularly illuminating – of course the amount of money is constant if nothing goes in or out! Similar physics statements such as "the total mass-energy of the universe is constant" are again not terribly useful, though one should note this particular statement is not obviously true. Things get more interesting when we consider a more realistic scenario in which the system is not isolated. People might be coming and going from building B and some might actually have money in their pockets. In more leisurely economic times, one might be interested just in the amount of money in the building at the end of the day. Just a bit of thought leads to

$$E_n = E_{n-1} + \Delta E_{n-1,n}$$

where E_n is the amount of money at the end of day n, E_{n-1} that from the previous day and $\Delta E_{n-1,n}$ the difference between money received and that payed in the building during day n

$$\Delta E_{n-1,n} = R_{n-1,n} - P_{n-1,n}.$$

As economic activity picks up and we take building B to mean "bank" it becomes important to keep track of the money at all times, not just at the end of the day. It then makes sense to think of the rate at which money is moving in or out of the building so we can not only track the amount of currency at any given time, but also be able to make future predictions. Instead of separate receipts R and payments P, use a single quantity F to denote the amount of money leaving or entering building B during time interval Δt with the understanding that positive values of F represent incomes and negative ones expenditures. Such understandings go by the name of sign conventions. They're not especially meaningful but it aids communication if a single convention is adopted. The amount of euros in the building then changes in accordance to

$$E(t + \Delta t) = E(t) + F\Delta t, \tag{5.4}$$

and F is known as a *flux*, the Latin term for flow.

While (5.4) is a good approximation for small intervals, errors arise when Δt is large since economic acitvity might change from hour to hour. Better accounting is obtained by considering F as defined at any given time t, such that $F(t)$ is the instantaneous flux of euros at time t. The fundamental theorem of calculus then states

$$E(t + \Delta t) = E(t) + \int_t^{t+\Delta t} F(\tau)\,\mathrm{d}\tau, \tag{5.5}$$

with the same significance as (5.4).

In a large bank one keeps track of the amount of money in individual rooms and the inflows and outflows through individual doors. A room or door can be identified by its spatial position $\boldsymbol{x} = (x_1, x_2, x_3)$, but $\boldsymbol{x}$ refers to a single point and physical currency occupies some space. The conceptual difficulty is overcome by introducing a fictitious *density of currency* at time t denoted by $e(\mathbf{x}, t)$. The only real meaning associated with this density is that the sum of all values of $e(\mathbf{x}, t)$ in some volume ω is the amount of currency in that volume

$$E(\omega, t) = \int_\omega e(\boldsymbol{x}, t)\,\mathrm{d}\boldsymbol{x}. \tag{5.6}$$

On afterthought, the same sort of question should have arisen when $E(t)$ was defined at one instant in time. Ingrained psychological perspectives make $E(t)$ more plausible, but were we to live our lives such that quantum fluctuations are observable, $E(t)$ would be much more questionable.

By an analogous procedure, define $\boldsymbol{f}(\boldsymbol{x},\tau)$ as the instantaneous flux density of euros in a small region around ($\boldsymbol{x}$, τ). This flux is a vector quantity to distinguish fluxes along different spatial directions. The flux density along direction $\boldsymbol{n}(\boldsymbol{x})$ is given by $\boldsymbol{f}(\boldsymbol{x},\tau)\cdot\boldsymbol{n}(x)$. Consider $\boldsymbol{n}(\boldsymbol{x})$ as the inward pointing unit vector normal to the surface ∂B that bounds the bank. The total flux is again obtained by integrating flux densities

$$F(\tau)=\int_{\partial B}\boldsymbol{f}(\boldsymbol{x},\tau)\cdot\boldsymbol{n}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{5.7}$$

Gathering the above leads to re-expressing (5.4) or (5.5) gives

$$E(B,t+\Delta t)=E(B,t)+\int_t^{t+\Delta t}\int_{\partial B}\boldsymbol{f}(\boldsymbol{x},\tau)\cdot\boldsymbol{n}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\tau \tag{5.8}$$

Using (5.6) leads to the statement,

$$\int_B e(\boldsymbol{x},t+\Delta t)\,\mathrm{d}\boldsymbol{x}=\int_B e(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}+\int_t^{t+\Delta t}\int_{\partial B}\boldsymbol{f}(\boldsymbol{x},\tau)\cdot\boldsymbol{n}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\tau.$$

There are special cases in which additional events affecting the balance of E can occur. When B is a reserve bank money might be (legally) printed and destroyed in the building. Again by analogy with fluid dynamics, such events are said to be *sources* of E within B, much like a underground spring is a source of surface water. Let $\Sigma(t)$ be the total sources at time t. As before, $\Sigma(t)$ might actually be obtained by summing over several sources placed in a number of positions, for instance the separate printing presses and furnaces that exist in B. It is useful to introduce a spatial density of sources $\sigma(\mathbf{x},t)$. The conservation statement now becomes

$$\int_B e(\boldsymbol{x},t+\Delta t)\,\mathrm{d}\boldsymbol{x}-\int_B e(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}=\int_t^{t+\Delta t}\int_{\partial B}\boldsymbol{f}(\boldsymbol{x},\tau)\cdot\boldsymbol{n}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\tau+\int_t^{t+\Delta t}\int_{\partial B}\sigma(\boldsymbol{x},\tau)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\tau. \tag{5.9}$$

The above encompasses all physical conservation laws, and is quite straightforward in interpretation:

change in Euros in B = net Euros coming in or going out of B + net Euros produced or destroyed in B.

It should be emphasized that the above statement has true physical meaning and is referred to as an *integral formulation of a conservation law*. The key term is "integral" and refers to the integration over some spatial domain.

Local formulations. Equation (5.9) is useful and often applied directly in the analysis of physical systems. From an operational point of view it does have some inconveniences though. These have mainly to do with the integration domains B, sometimes difficult to describe and to perform integrations over. Avoid this by considering $\boldsymbol{f}(\boldsymbol{x},t)$ defined everywhere, not only on ∂B (the doors and windows of B). These internal fluxes can be shown to have a proper physical interpretation. Assuming that $\boldsymbol{f}(\boldsymbol{x},t)$ is smooth allows use of the Gauss theorem to transform the surface integral over ∂B into a volume integral over B

$$\int_{\partial B}\boldsymbol{f}(\boldsymbol{x},\tau)\cdot\boldsymbol{n}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}=-\int_B\nabla\cdot\boldsymbol{f}(\boldsymbol{x},\tau)\,\mathrm{d}\boldsymbol{x} \tag{5.10}$$

The minus assign arises from the convention of an inward pointing normal. Applying (5.10) to (5.9) leads to

$$\int_B\Big[e(\mathbf{x},t+\Delta t)-e(\mathbf{x},t)+\int_t^{t+\Delta t}\nabla\cdot\boldsymbol{f}(\boldsymbol{x},\tau)\,\mathrm{d}\tau\Big]\,d\mathbf{x}=\int_t^{t+\Delta t}\int_{\partial B}\sigma(\boldsymbol{x},\tau)\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\tau. \tag{5.11}$$

There was nothing special about the shape of the building B or the length of the time interval Δt we used in deriving (5.11), hence the equality should hold for infinitesimal domains

$$\frac{\partial e}{\partial t} + \nabla \cdot \boldsymbol{f} = \sigma, \tag{5.12}$$

where, as is customary, the dependence of $e, \boldsymbol{f}, \sigma$ on space and time is understood but not written out explicitly. Equation (5.12) is known as the *local* or *differential form* of the conservation law for E.

3. Special forms of conservation laws

Second law of dynamics. The full general form (5.12) often arises in applications, but simplifications can arise from specific system properties. As a simple example, the dynamics of a point mass m which has no internal structure is described by the conservation of momentum statement

$$\frac{\mathrm{d}}{\mathrm{d}t}(m\boldsymbol{v}) = \sum \boldsymbol{F}. \tag{5.13}$$

The correspondence with (5.12) is given by $e \longleftrightarrow (m\mathbf{v})$, $\sigma \longleftrightarrow \sum \boldsymbol{F}$, hence the statement: "external forces are sources of momentum". Instead of a PDE, the lack of internal structure has led to an ODE.

Advection equation. Other special forms of (5.12) are not quite so trivial. Often $\boldsymbol{f}, \sigma$ depend on e, with the specific form of this dependence is given by physical analysis. Accounting for all physical effects is so difficult that simple approximations are often used. For instance if $\boldsymbol{f}(e)$ is sufficiently smooth Taylor series expansion gives

$$\boldsymbol{f}(e) = \boldsymbol{f}_0 + \boldsymbol{f}'(e_0)\,(e - e_0) + \ldots. \tag{5.14}$$

Choosing the origin such that $\boldsymbol{f}_0 = 0$ and $e_0 = 0$, the simplest truncation is

$$\boldsymbol{f}(e) \cong \boldsymbol{f}'(0)\, e = \boldsymbol{u} e, \tag{5.15}$$

and the $\sigma = 0$ form of (5.12) is

$$\frac{\partial e}{\partial t} + \nabla \cdot (\boldsymbol{u} e) = 0. \tag{5.16}$$

In this approximation $\boldsymbol{u}$ is a constant giving

$$\frac{\partial e}{\partial t} + \boldsymbol{u} \cdot \nabla e = 0 \tag{5.17}$$

known as the *constant velocity advection equation*. Its one-dimensional form is the basis of much development in numerical methods for PDE's

$$\frac{\partial e}{\partial t} + u \frac{\partial e}{\partial x} = 0 \tag{5.18}$$

Diffusion equation. Another widely encountered dependence of $\boldsymbol{f}$ on e is of the form

$$\boldsymbol{f}(e) = -\alpha \nabla e \tag{5.19}$$

and this leads to

$$\frac{\partial e}{\partial t} - \nabla \cdot (\alpha \nabla e) = \sigma(e). \tag{5.20}$$

If there are no sources and α is a constant we have

$$\frac{\partial e}{\partial t} = \alpha \nabla^2 e \tag{5.21}$$

the heat or diffusion equation.

Combined effects. Both above flux types can appear in which case the associated conservation law is

$$\frac{\partial e}{\partial t} + \nabla \cdot (\boldsymbol{u} e) = \alpha \nabla^2 e, \tag{5.22}$$

known as the advection-diffusion equation, and is a linear PDE. If sources σ exist the above becomes

$$\frac{\partial e}{\partial t} + \nabla \cdot (\boldsymbol{u} e) = \alpha \nabla^2 e + \sigma, \tag{5.23}$$

or for constant advection velocity $\boldsymbol{u}$

$$\frac{\partial e}{\partial t} + \boldsymbol{u} \cdot \nabla e = \alpha \nabla^2 e + \sigma. \tag{5.24}$$

It is often the case that the flux depends on the conserved quantity itself, $\boldsymbol{f}(e) = \boldsymbol{u}(e)\, e$, in which case (5.22) becomes a non-linear PDE.

Steady-state transport. Various effects can balance leading to no observable time dependence, $\partial e/\partial t = 0$. If there is no overall diffusive flux $\boldsymbol{f}(e) = -\alpha \nabla e$ within an infinitesimal volume, then $\nabla \cdot \boldsymbol{f} = -\alpha \nabla^2 e = 0$ leads to the Laplace equation

$$\nabla^2 e = 0.$$

If the infinitesimal volume contains sources the Poisson equation

$$\nabla^2 e = \sigma,$$

is obtained.

Separation of variables. Often, the time dependence can be isolated from the spatial dependence, $e(\boldsymbol{x}, t) = X(\boldsymbol{x}) T(t)$, in which case the diffusion equation for constant α leads to

$$\frac{\dot{T}}{T} = \alpha \frac{\nabla^2 X}{X} = -\lambda,$$

with λ a positive constant to avoid unphysical exponential growth. The spatial part of the solution statisfies the Helmholtz equation

$$\nabla^2 X = -\kappa^2 X,$$

with $\kappa^2 = \lambda/\alpha$. The above is interpreted as an eigenproblem for the Laplacian operator $\Delta = \nabla^2$.

The above special forms of differential conservation laws play an important role in scientific computation. Numerical techniques have been developed to capture the underlying physical behavior expressed in say the diffusion equation of the Helmholtz equation. These equations were first studied within physics, but they reflect universal behavior. Consider the Black-Scholes financial model for the price of an option $V(S,t)$ on an asset $S(t)$

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} = rV - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2},$$

with σ the standard deviation of stock market returns and r the annualized risk-free interest rate. The terminology might be totally different, but the same patterns emerge and the Black Scholes model can be interpreted as an advection diffusion equation with non-constant advection velocity rS, negative diffusion coefficient $-\sigma^2 S^2/2$ and source term rV. The similarity to the physics advection and diffusion equations arises from the same type of modeling assumptions relating fluxes to state variables.

LECTURE 28: LINEAR OPERATOR SPLITTING

1. Finite difference Poisson equation

Consider now the approximation of $\mathscr{I} = f$ in the first-order differential equation

$$y' = f(t, y). \tag{5.25}$$

Integration over a time step $[t_i, t_{i+1}]$ gives

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t))\,\mathrm{d}t,$$

and use of quadrature formulas leads to numerical solutions for solving (5.25). Consider for instance data $\mathscr{D} = \{(t_{i+1-k}, f_{i+1-k}), k = 1, \ldots, s\}$ going back s intervals of size h, $t_{i+1-k} = t_{i+1} - kh$. Any quadrature formula based on this data could be used, but the most often encountered approach is to use a polynomial approximant. This can be stated in Lagrange form as

$$f(t, y(t)) \cong \sum_{k=1}^{s} \ell_k(t)\, f_k,\ f_k = f(t_{i+1-k}, y(t_{i+1-k})) \cong f(t_{i+1-k}, y_{i+1-k}).$$

The last approximate equality arises from replacing the exact value $y(t_{i+1-k})$ by its approximation $y_k \cong y(t_{i+1-k})$. The result is known as an Adams-Bashforth scheme

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} \sum_{k=1}^{s} \ell_k(t)\, f_k\,\mathrm{d}t = y_i + \sum_{k=1}^{s} \Big(\int_{t_i}^{t_{i+1}} \ell_k(t)\,\mathrm{d}t\Big) f_k = y_i + h \sum_{k=1}^{s} b_k f_{i+1-k},$$

with coefficients that are readily computed (cf. Table 1).

$$b_k = \frac{1}{h}\Big(\int_{t_i}^{t_{i+1}} \ell_k(t)\,\mathrm{d}t\Big).$$

s	b_1	b_2	b_3	b_4
1	1			
2	$\frac{3}{2}$	$-\frac{1}{2}$		
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$

Table 5.1. Adams-Bashforth scheme coefficients.

The $s = 1$ Adams-Bashforth scheme is identical to forward Euler and the above approach yields schemes that are explicit, i.e., the new value is directly obtained from knowledge of previous values.

Choosing data $\mathcal{D} = \{(t_{i+1-k}, f_{i+1-k}), k = 0, \ldots, s-1\}$ that contains the point yet to be computed (t_{i+1}, y_{i+1}) gives rise to a class of implicit schemes known as the Adams-Moulton schemes (Table 2)

$$y_{i+1} = y_i + h \sum_{k=0}^{s-1} b_k f_{i+1-k},$$

s	b_0	b_1	b_2	b_3
1	1			
2	$\frac{1}{2}$	$\frac{1}{2}$		
3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$	
4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$

Table 5.2. Adams-Moulton scheme coefficients.

2. Matrix splitting iteration

Approximation of both operators $\mathscr{L} = \mathrm{d}/\mathrm{d}t$ and $\mathscr{R} = f$ arising in $\mathscr{L}y = \mathscr{R}y$, or $y' = f(t, y)$ is possible. Combining previous computations, the resulting schemes can be stated as

$$\sum_{k=0}^{s} a_k y_{i+k} = h \sum_{k=0}^{s} b_k f_{i+k}, \; f_{i+k} = f(t_{i+k}, y_{i+k}). \tag{5.26}$$

Both sides arise from linear approximants: of the derivative on the left, and of f on the right.

3. Convergence analysis

Any of the above schemes defines a sequence $\{y_n\}_{n \in \mathbb{N}}$ that approximates the solution $y(t_n)$ of the initial value problem

$$y' = f(t, y), y(0) = y_0,$$

over a time interval $[0, T]$, $t_n = nh$, $h = T/N$. A scheme is said to be convergent if

$$\lim_{\substack{h \to 0 \\ Nh = T}} y_N = y(T).$$

The above states that in the limit of taking small step sizes while maintaining $Nh = T$ for some finite time T, the estimate at the endpoint converges to the exact value. Such a definition is rather difficult to apply directly, and an alternative characterization of convergence is desirable.

To motivate the overall approach, consider first the following model problem

$$y' = \lambda y, y(0) = y_0, \lambda \leqslant 0 \tag{5.27}$$

The model problem arises from truncation of the general non-linear function f to first order

$$y' = f(y) = f(0) + f'(0)\, y + \cdots.$$

Since $f(0)$ is a constant that simply leads to linear growth, and the model problem captures the lowest-order non-trivial behavior. The exact solution is

$$y(t) = e^{\lambda t} y_0 \Longrightarrow y(t_n) = e^{n\lambda h} y_0,$$

giving $y(T) = e^{\lambda T} y_0$. The restriction of $\lambda \leqslant 0$ in the model problem arises from consideration of the effect of a small perturbation in the initial condition representative of floating point representation errors. This leads to $\tilde{y}(T) = e^{\lambda T}(y_0 + \delta)$, and the error $\varepsilon = \tilde{y}(T) - y(T) = e^{\lambda T}\delta$ can only be maintained small if $\lambda \leqslant 0$.

Applying the forward Euler scheme to the model problem (5.27) gives

$$y_{n+1} = y_n + \lambda h y_n = (1 + z) y_n,$$

with $z = \lambda h$. After N steps the numerical approximation is

$$y_N = (1 + z)^N y_0.$$

The exponential decay of the exact solution can only be recovered if which leads to a restriction on the allowable step size

$$-\frac{2}{\lambda} > h > 0.$$

If the step size is too large, $h > -2/\lambda$, inherent floating point errors are amplified by the forward Euler method, and the scheme is said to be *unstable*. This is avoided by choosing a subunitary parameter z , $|z| = |\lambda h| \leqslant 1$, which leads to a step size restriction $h < 1/|\lambda|$.

These observations on the simple case of the Euler forward method generalize to linear multistep methods. Applying (5.26) to the model problem (5.27) leads to the following linear finite difference equation

$$\sum_{k=0}^{s} a_k y_{i+k} = z \sum_{k=0}^{s} b_k y_{i+k}\,. \tag{5.28}$$

The above is solved using a procedure analogous to that for differential equations by hypothesizing solutions of the form

$$y_n = r^n,$$

to obtain a characteristic equation

$$\pi(r; z) = \rho(r) - z\sigma(r) = 0,$$

where $\rho(r), \sigma(r)$ are polynomials

$$\rho(r) = \sum_{k=0}^{s} a_k r^k, \sigma(r) = \sum_{k=0}^{s} b_k r^k.$$

The above polynomials allow an operational assessment of algorithms of form (5.26). An algorithm (5.26) that recovers the ordinary differential equation (5.25) in the limit of $h \longrightarrow 0$ is said to be *consistent*, which occurs if and only if

$$\rho(1) = 0, \rho'(1) - \sigma(1) = 0.$$

Furthermore an algorithm of form (5.26) that does not amplify inherent floating point errors is said to be stable, which occurs if the roots of $\pi(r; z)$ are subunitary in absolute value

$$|r_j| < 1, \pi(r_j; z) = 0.$$

THEOREM. *An algorithm to solve (5.25) that is consistent and stable is convergent.*

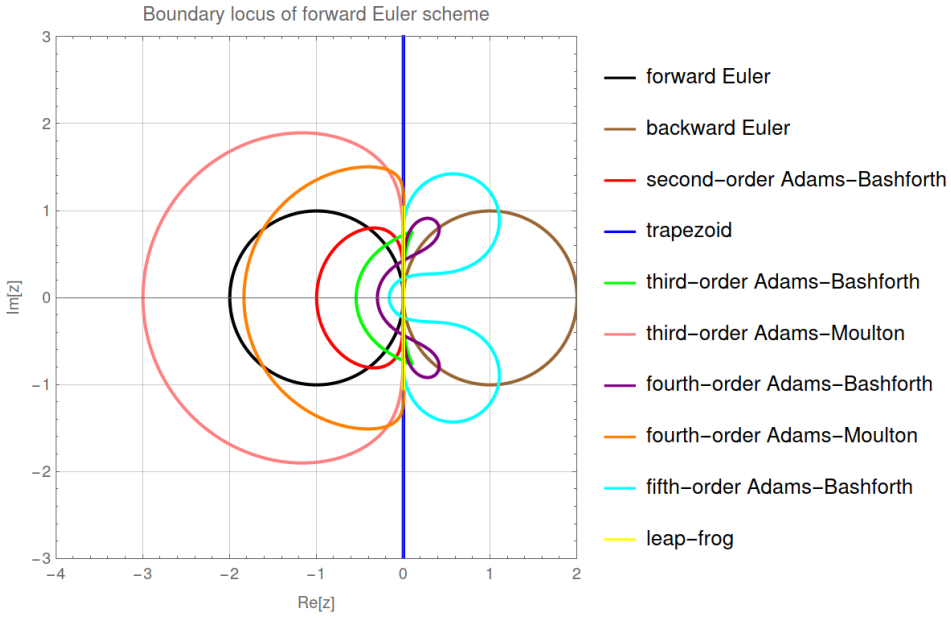
A convenient procedure to determine the stable range of step sizes is to consider r of unit absolute value

$$r = e^{i\theta},$$

and evaluate the characteristic equation

$$\pi(e^{i\theta}; z) = \rho(e^{i\theta}) - z\sigma(e^{i\theta}) = 0 \Longrightarrow z(\theta) = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})},$$

where $z(\theta)$ is the boundary locus delimiting zones of stability in the complex plane (Fig 1).



A method is said to be A-stable if its region of stability contains the entire left half-plane in $\mathbb{C}$, and is said to be L-stable if $\lim_{\omega\to\infty}\rho(\omega e^{i\theta})/\sigma(\omega e^{i\theta}) = 0$.

LECTURE 29: SPLITTING FOR VARIABLE COEFFICIENT LINEAR OPERATORS

1. Spatially dependent diffusivity

The null space of a linear mapping represented through matrix $\boldsymbol{A}\in\mathbb{C}^{m\times n}$ is defined as $N(\boldsymbol{A}) = \{\boldsymbol{x}\,|\,\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$, the set of all points that have a null image through the mapping. The null space is a vector subspace of the domain of the linear mapping. A first step in the study of nonlinear mappings is to consider the generalization of the concept of a null set, starting with the simplest case,

$$f(x) = 0 \tag{5.29}$$

where $f:\mathbb{R}\longrightarrow\mathbb{R}$, $f\in C^p(\mathbb{R})$, $p\geqslant 0$, i.e., f has p continuous derivatives. It is assumed that a closed form analytical solution is not available, and algorithms are sought to construct an approximating sequence $\{x_n\}_{n\in\mathbb{N}}$ whose limit is a root of (5.29). The general approach is to replace (5.29) with

$$g_n(x) = 0, \tag{5.30}$$

where g_n is some approximation of f, and x_n the root of (5.30) can be easily determined.

Consider $g(x) = ax + b$ (a first-degree polynomial, but not a linear mapping for $b\neq 0$), an approximant of f based upon data $\{(x_{n-2}, f_{n-2} = f(x_{n-2})), (x_{n-1}, f_{n-1} = f(x_{n-1}))\}$, given in Newton interpolant form by

$$g_n(x) = f_{n-2} + \frac{f_{n-1} - f_{n-2}}{x_{n-1} - x_{n-2}}(x - x_{n-2}). \tag{5.31}$$

The root of (5.31) is

$$x_n = x_{n-2} - \frac{f_{n-2}}{f_{n-1} - f_{n-2}}(x_{n-1} - x_{n-2}) = \frac{x_{n-2}\,f_{n-1} - x_{n-1}\,f_{n-2}}{f_{n-1} - f_{n-2}},$$

an iteration known as the secant method. The error in root approximation is

$$e_n = x_n - x = e_{n-2} - \frac{f_{n-2}}{f_{n-1} - f_{n-2}}(e_{n-1} - e_{n-2}),$$

and can be estimated by Taylor series expansions around the root x for which $f(x) = 0$,

$$f_{n-k} = f(x_{n-k}) = f'(x)(x_{n-k} - x) + \frac{1}{2}f''(x)(x_{n-k} - x)^2 + \cdots = f'\,e_{n-k} + \frac{1}{2}f''\,e_{n-k}^2 + \cdots,$$

where derivatives f', f'' are assumed to be evaluated at x. In the result

$$e_n = e_{n-2} - \frac{f'\,e_{n-2} + \frac{1}{2}f''\,e_{n-2}^2 + \cdots}{f'\cdot(e_{n-1} - e_{n-2}) + \frac{1}{2}f''\cdot(e_{n-1}^2 - e_{n-2}^2) + \cdots}(e_{n-1} - e_{n-2}) = e_{n-2}\left[1 - \frac{f' + \frac{1}{2}f''\cdot e_{n-2} + \cdots}{f' + \frac{1}{2}f''\cdot(e_{n-1} + e_{n-2}) + \cdots}\right],$$

assuming $f'(x) \neq 0$, (i.e., x is a simple root) gives

$$e_n = e_{n-2}\left[1 - \frac{1 + c \cdot e_{n-2} + \cdots}{1 + c \cdot (e_{n-1} + e_{n-2}) + \cdots}\right], c = \frac{1}{2}(f''/f').$$

For small errors, to first order the above can be written as

$$e_n = e_{n-2}[1 - (1 + c \cdot e_{n-2})(1 - c \cdot (e_{n-1} + e_{n-2}))] = c e_{n-1} e_{n-2}.$$

Assuming p-order convergence of e_n,

$$|e_n| \sim A |e_{n-1}|^p,$$

leads to

$$A^{p+1} |e_{n-2}|^{p^2} \sim cA |e_{n-2}|^{p+1} \Longrightarrow |e_{n-2}|^{p^2-p-1} \sim cA^{-p}.$$

Since c, A are finite while $e_n \longrightarrow 0$, the above asymptotic relation can only be satisfied if

$$p^2 - p - 1 = 0 \Longrightarrow p = \frac{1 + \sqrt{5}}{2} \cong 1.62,$$

hence the secant method exhibits superlinear, but subquadratic convergence.

A different linear approximant arises from the Hermite interpolant based on data

$$\{(x_{n-1}, f_{n-1} = f(x_{n-1}), f'_{n-1} = f'(x_{n-1}))\},$$

which is given in Newton form as

$$g_n(x) = f_{n-1} + f'_{n-1} \cdot (x - x_{n-1}),$$

with root

$$x_n = x_{n-1} - \frac{f_{n-1}}{f'_{n-1}}, \tag{5.32}$$

an iteration known as the Newton-Raphson method. The error is given by

$$e_n = x_n - x = e_{n-1} - \frac{f_{n-1}}{f'_{n-1}}. \tag{5.33}$$

Taylor series exapnsion around the root gives for small e_{n-1},

$$e_n = e_{n-1} - \frac{f' \cdot e_{n-1} + \frac{1}{2} f'' \cdot e_{n-1}^2 + \cdots}{f' + f'' e_{n-1} + \cdots} = e_{n-1}\left[1 - \frac{1 + c e_{n-1} + \cdots}{1 + 2 c e_{n-1} + \cdots}\right] \approx e_{n-1}[1 - (1 + c e_{n-1})(1 - 2 c e_{n-1})].$$

The resulting expression

$$e_n \approx c e_{n-1}^2 = \frac{1}{2}\frac{f''}{f'} e_{n-1}^2, \tag{5.34}$$

states quadratic convergence for Newton's method. This faster convergence than the secant method requires however knowledge of the derivative, and the computational expense of evaluating it.

The above estimate assumes convergence of $\{x_n\}_{n\in\mathbb{N}}$, but this is not guaranteed in general. Newton's method requires an accurate initial approximation x_0, within a neighborhood of the root in which f is increasing, $f' > 0$, and convex, $f'' > 0$. Equivalently, since roots of f are also roots of $-f$, Newton's method converges when f', $f'' < 0$. In both cases (5.34) in the prior iteration states that $e_{n-1} = x_{n-1} - r > 0$, hence $x_{n-1} > r$. Since f is increasing $f(x_{n-1}) > f(r) = 0$, hence (5.33) implies $e_n < e_{n-1}$. Thus the sequence $\{e_n\}_{n\in\mathbb{N}}$ is decreasing and bounded below by zero, hence $\lim_{n\to\infty} e_n = 0$, and Newton's method converges.

An immediate extension of the above approach is to increase the accuracy of the approximant by seeking a higher-degree polynomial interpolant. The expense of the resulting algorithm increases rapidly though, and in practice linear and quadratic approximants are the most widely used. Consider the Hermite interpolant based on data

$$\{(x_{n-1}, f_{n-1} = f(x_{n-1}), f'_{n-1} = f'(x_{n-1}),, f''_{n-1} = f''(x_{n-1}))\},$$

given in Newton form as

$$g_n(x) = f_{n-1} + f'_{n-1}\cdot(x - x_{n-1}) + \frac{1}{2} f''_{n-1}\cdot(x - x_{n-1})^2 = C + Bs + A s^2,$$

with roots

$$x_n = x_{n-1} + \frac{-f'_{n-1} \pm \sqrt{(f'_{n-1})^2 - 2 f_{n-1} f''_{n-1}}}{f''_{n-1}}.$$

Tha above exhibits the difficulties arising in higher-order interpolants. The iteration requires evaluation of a square root, and checking for a positive discriminant.

Algebraic manipulations can avoid the appearance of radicals in a root-finding iteration. As an example, Halley's method

$$x_n = x_{n-1} - \frac{2 f_{n-1} f'_{n-1}}{2(f'_{n-1})^2 - f_{n-1} f''_{n-1}},$$

exhibits cubic convergence.

2. Gradient descent

The secant iteration

$$x_n = x_{n-2} - \frac{f_{n-2}}{f_{n-1} - f_{n-2}}(x_{n-1} - x_{n-2}) = x_{n-2} - \frac{f_{n-2}}{\dfrac{f_{n-1} - f_{n-2}}{x_{n-1} - x_{n-2}}},$$

in the limit of $x_{n-2} \longrightarrow x_{n-1}$ recovers Newton's method

$$x_n = x_{n-1} - \frac{f_{n-1}}{f'_{n-1}}.$$

This suggests seeking advantageous approximations of the derivative

$$x_n = x_{n-1} - \frac{f_{n-1}}{\dfrac{f(x_{n-1}+h_{n-1}) - f(x_{n-1})}{h_{n-1}}},$$

based upon some step-size sequence $\{h_n\}$. Since $f(x_n) \longrightarrow 0$, the choice $h_{n-1} = f(x_{n-1})$ suggests itself, leading to Steffensen's method

$$x_n = x_{n-1} - \frac{f_{n-1}}{\dfrac{f(x_{n-1}+f(x_{n-1})) - f(x_{n-1})}{f(x_{n-1})}} = x_{n-1} - \frac{f_{n-1}}{g_{n-1}}, g_{n-1} = \frac{f(x_{n-1}+f(x_{n-1}))}{f(x_{n-1})} - 1.$$

Steffensen's method exhibits quadratic convergence, just like Newton's method, but does not require knowledge of the derivative. The higher order by comparison to the secant method is a direct result of the derivative approximation

$$f'(x_{n-1}) \cong \frac{f(x_{n-1}+f(x_{n-1})) - f(x_{n-1})}{f(x_{n-1})},$$

which, remarkably, utilizes a composite approximation

$$f(x_{n-1}+f(x_{n-1})) = (f \circ (1+f))(x_{n-1}).$$

Such composite techniques are a prominent feature of various nonlinear approximations such as a k-layer deep neural network $\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{I}_k \circ \boldsymbol{I}_{k-1} \circ \ldots \circ \boldsymbol{I}_1)(\boldsymbol{x})$.

3. Conjugate gradient

The above iterative sequences have the form

$$x_n = F(x_{n-1}),$$

and the root is a fixed point of the iteration

$$x = F(x).$$

For example, in Newton's method

$$F(x) = x - \frac{f(x)}{f'(x)},$$

and indeed at a root $x = F(x)$. Characterization of mappings F that lead to convergent approximation sequences is of interest and leads to the following definition and theorem.

DEFINITION. *A function $F:[a,b] \longrightarrow [a,b]$ is said to be a contractive mapping if $\forall x, y \in [a,b]$ there exists $c \in (0,1)$ such that*

$$|F(x) - F(y)| \leqslant c\,|x - y|\,.$$

THEOREM. *(Contractive Mapping theorem). If $F:[a,b] \longrightarrow [a,b]$ is a contractive mapping then F has a unique fixed point $x \in [a,b]$, $x = F(x)$.*

The fixed point theorem is an entry point to the study of non-additive approximation sequences.

Example 5.1. The sequence

$$x_1 = \sqrt{p}, x_2 = \sqrt{p + \sqrt{p}}, \ldots \ (p > 0)$$

is expressed recursively as

$$x_{n+1} = \sqrt{p + x_n},$$

and has the limit

$$x = \sqrt{p + \sqrt{p + \sqrt{p + \cdots}}},$$

that is the fixed point of F,

$$x = F(x) = \sqrt{p + x} = \frac{1 + \sqrt{1 + p}}{2}.$$

Over the interval $[0, p+1]$, F is a contraction since

$$F'(x) = \frac{1}{2\sqrt{p + x}} \leqslant \frac{1}{2\sqrt{p}} < 1.$$

Example 5.2. The sequence

$$x_1 = \frac{1}{p}, x_2 = \frac{1}{p + \frac{1}{p}}, \ldots \ (p > 0)$$

is expressed recursively as

$$x_{n+1} = \frac{1}{p + x_n},$$

and has the limit

$$x = \frac{1}{p + \frac{1}{p + \cdots}},$$

that is the fixed point of F,

$$x = F(x) = \frac{1}{p+x} = \frac{-p+\sqrt{p^2+1}}{2}.$$

Over the interval $[0,1]$, F is a contraction since

$$|F'(x)| = \frac{1}{(p+x)^2} \leqslant \frac{1}{p^2} < 1.$$

LECTURE 30: NONSYMMETRIC LINEAR OPERATORS, IRREGULAR SPARSITY

1. Finite element discretization

Consider now nonlinear finite-dimensional mappings $\boldsymbol{f}:\mathbb{R}^d \longrightarrow \mathbb{R}^d$, and the root-finding problem

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \tag{5.35}$$

whose set of solutions generalize the linear mapping concept of a null space, $N(\boldsymbol{A}) = \{\boldsymbol{x} | \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{A} \in \mathbb{C}^{d\times d}\}$. As in the scalar-valued case, algorithms are sought to construct an approximating sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ whose limit is a root of (5.35), by approximating $\boldsymbol{f}$ with $\boldsymbol{g}_k$, and solving

$$\boldsymbol{g}_k(\boldsymbol{x}) = 0. \tag{5.36}$$

Multivariate approximation is however considerably more complex than univariate approximation. For example, consider $d = 2$, $\boldsymbol{f}:\mathbb{R}^2 \longrightarrow \mathbb{R}^2$, and the univariate monomial interpolants in Lagrange form

$$\mathcal{L}_t \boldsymbol{f}(s,t) = \sum_{i=0}^{m} \boldsymbol{f}(x_i,t)\, l_i^x(s), \mathcal{L}_s \boldsymbol{f}(s,t) = \sum_{j=0}^{n} \boldsymbol{f}(s,y_j)\, l_j^y(t),$$

with

$$l_i^x(s) = \prod_{k=0}^{m\,'} \frac{s-x_k}{x_i-x_k}, l_j^y(s) = \prod_{l=0}^{n\,'} \frac{t-y_l}{y_j-y_l}.$$

The operator $\mathcal{L}_t$ carries out interpolation at fixed t value of the data set $\mathcal{D}_x = \{(x_i, \boldsymbol{f}(x_i,t)), i = 0,\ldots,m\}$. Similarly, operator $\mathcal{L}_s$ carries out interpolation at fixed s value of the data set $\mathcal{D}_y = \{(y_j, \boldsymbol{f}(s,y_j)), j = 0,\ldots,n\}$. Multivariate interpolation of the data set

$$\mathcal{D} = \{(x_i, y_j, \boldsymbol{f}(x_i,y_j)), i = 0,\ldots,m, j = 0,\ldots,n\},$$

can be carried out through multiple operator composition procedures.

Bivariate ($d = 2$) root-finding algorithms already exemplifies the additional complexity in constructing root finding algorithms. The goal is to determine a new approximation (x_k, y_k) from the prior approximants

$$(x_0, y_0), \ldots, (x_{k-2}, y_{k-2}), (x_{k-1}, y_{k-1}) .$$

Whereas in the scalar case two prior points allowed construction of a linear approximant, the two points in data

$$\mathcal{D} = \{(x_{k-2}, y_{k-2}), (x_{k-1}, y_{k-1})\}$$

are insufficient to determine

$$\mathcal{L}\boldsymbol{f} = \sum_{i=k-2}^{k-1} \sum_{j=k-2}^{k-1} \boldsymbol{f}(x_i, y_j)\, l_i^x(s)\, l_j^y(t),$$

which requires four data points. Various approaches to exploit the additional degrees of freedom are available, of which the class of quasi-Newton methods finds widespread applicability.

A linear multivariate approximant in d dimensions requires 2^d data. A Hermite interpolant based upon function and partial derivative values can be constructed, but it is more direct to truncate the multivariate Taylor series

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_k) + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) + \cdots,$$

where

$$\boldsymbol{J} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_d} \\ \vdots & \vdots & \ddots & \\ \frac{\partial f_d}{\partial x_1} & \frac{\partial f_d}{\partial x_2} & \cdots & \frac{\partial f_d}{\partial x_d} \end{bmatrix} = \nabla \boldsymbol{f},$$

is the Jacobian matrix of $\boldsymbol{f}$. Setting $\boldsymbol{f}(\boldsymbol{x}_{k+1}) = \boldsymbol{0}$, as the condition for the next iterate leads to the update

$$\boldsymbol{J}(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) = -\boldsymbol{f}(\boldsymbol{x}_k),$$

a linear system that is solved at each iteration. Computation of the multiple partial derivatives arising in the Jacobian might not be possible or too expensive, hence approximations are sought $\boldsymbol{B}_k \cong \boldsymbol{J}(\boldsymbol{x}_k)$, similar in principle to the approximation of a tangent by a secant. In such quasi-Newton methods, a secant condition on $\boldsymbol{B}_k$ is stated as

$$\boldsymbol{B}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) = \boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_{k-1}),$$

and corresponds to a truncation of the Taylor series expansion around $\boldsymbol{x}_{k-1}$. The above secant condition is not sufficient by itself to determine $\boldsymbol{B}_k$, hence additional considerations can be imposed.

1. Recalling that the scalar Newton method for finding roots of $f(x) = 0$ converges in a region where $f', f'' > 0$, imposing analogous behavior for $\boldsymbol{B}_k$ suggests itself. This is typically done by requiring $\boldsymbol{B}_k$ to be symmetric positive definite.

2. Assuming convergence of the approximating sequence $\{\boldsymbol{x}_k\}_{k\in\mathbb{N}}$ to a root, $\boldsymbol{B}_{k+1}$ should be close to the previous approximation suggesting the condtion

$$\min_{\boldsymbol{B}_{k+1}} \|\boldsymbol{B}_{k+1} - \boldsymbol{B}_k\|.$$

Various algorithms arise from a particular choice of norm and procedure to apply (2).

One widely used quasi-Newton method, arising from a rank-two update at each iteration to maintain positive definiteness, is the Broyden-Fletcher-Goldfard-Shanno update

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} - \frac{\boldsymbol{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \boldsymbol{B}_k^T}{\boldsymbol{s}_k^T \boldsymbol{B}_k \boldsymbol{s}_k},$$

where the updates are determined by

1. Solving $\boldsymbol{B}_k \boldsymbol{p}_k = -[\boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{f}(\boldsymbol{x}_{k-1})]$ to find a search direction $\boldsymbol{p}_k$;
2. Finding the distance along the search direction by $\alpha_k = \operatorname{argmin} \|\boldsymbol{f}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k)\|_2$;
3. Updating the approximation $\boldsymbol{s}_k = \alpha_k \boldsymbol{p}_k$, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$
4. Computing $\boldsymbol{y}_k = \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \boldsymbol{f}(\boldsymbol{x}_k)$.

2. Krylov iteration

3. GMRES and biconjugate gradient

LECTURE 31: INCOMPLETE OPERATOR DECOMPOSITION

1. Finite difference Helmholtz equation

In late nineteenth century, telegrapher's equations, a system of linear PDEs for current $I(x,t)$ and voltage $V(x,t)$

$$\frac{\partial}{\partial x} V(x,t) = -L \frac{\partial}{\partial t} I(x,t) - RI(x,t)$$

$$\frac{\partial}{\partial x} I(x,t) = -C \frac{\partial}{\partial t} C(x,t) - GV(x,t)$$

Heaviside avoided solution of the PDEs by reduction to an algebraic formulation historical formulation, e.g., for the ODE for $y(t)$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + ay = b$$

Heaviside considered the associated algebraic problem for $Y(s)$

$$sY + aY = b \Longrightarrow Y(s) = \frac{b}{a+s} \Longrightarrow y(t) = \mathcal{L}^{-1}[Y(s)]$$

> Why should I refuse a good dinner simply because I don't understand the digestive processes involved? (Heaviside, ?)

Heaviside's formal framework (1890's) for solving ODEs was discounted since it lacked mathematical rigour.

- Russian mathematician 1920's established first results (Vladimirov)
- Theory of Distributions (Schwartz, 1950s)

2. Arnoldi iteration

Consider function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g : \mathbb{R}^n \longrightarrow \mathbb{R}$, $T : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\| f - g \circ T \| < \varepsilon$$

for some $\varepsilon > 0$.

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} (f, \phi_j)\phi_j = \sum_{j=1}^{n} c_j \phi_j$$

The approximation is convergent if

$$\lim_{n\to\infty} \|f - g\circ T\| = 0,$$

This assumes $c_j = (f, \phi_j)$ rapidly decrease.

THEOREM. *(Parseval) The Fourier transform is unitary. For $A, B\colon \mathbb{R}\to\mathbb{C}$, square integrable, 2π-periodic with Fourier series*

$$A(t) = \sum_{n=-\infty}^{\infty} a_n e^{int}, B(t) = \sum_{n=-\infty}^{\infty} b_n e^{int},$$

$$\sum_{n=-\infty}^{\infty} a_n \bar{b}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} A(t)\bar{B}(t)\,\mathrm{d}t.$$

Bessel inequality:

$$\sum_{j=1}^{n} |(f, \phi_j)|^2 \leq \|f\|_2.$$

Fourier coefficient decay: for $f \in C^{(k-1)}(\mathbb{R})$, $f^{(k-1)}$ absolutely continuous,

$$|c_n| \leq \min_{0\leq j\leq k} \frac{\|f^{(j)}\|_1}{|n|^j}.$$

In practice: coefficients decay as

- $1/n$ for functions with discontinuities on a set of Lebesgue measure 0;
- $1/n^2$ for functions with discontinuous first derivative on a set of Lebesgue measure 0;
- $1/n^3$ for functions with discontinuous second derivative on a set of Lebesgue measure 0.

Fourier coefficients for analytic functions decay faster than any monomial power $c_n = o(n^{-p}), \forall p \in \mathbb{N}$, a property known as exponential convergence.

Denote such approximations by $\mathcal{L}$, and they are linear

$$\mathcal{L}(\alpha f + \beta g) = \alpha\mathcal{L}(f) + \beta\mathcal{L}(g)$$

-

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j$$

Let $\Phi_n = \{\varphi_{k(1)}, \varphi_{k(2)}, \dots, \varphi_{k(n)}\}$ such

$$(f, \varphi_{k(1)}) \geqslant (f, \varphi_{k(2)}) \geqslant \cdots \geqslant (f, \varphi_{k(n)}).$$

Choose $c_j = (f, \varphi_{k(j)})$, and

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j \,.$$

Denote such approximations by $\mathcal{G}$, and they are non-linear.

3. Lanczos iteration

Consider function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g : \mathbb{R}^n \longrightarrow \mathbb{R}$, $T : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\| f - g \circ T \| < \varepsilon$$

for some $\varepsilon > 0$.

What questions do you ask?

Does T exist?. $\forall f, \varepsilon, \exists T,$ such that $\| f - g \circ T \| < \varepsilon$

Can arbitrary ε be achieved?.

Can we construct T?.

$\longrightarrow$ By what procedure?

$$T = T_1 \circ T_2 \circ \dots \circ T_J$$

with T_i simple modifications of identity (ReLU)

$$\min_{T_1,\dots T_J} \|f - g \circ T_1 \circ T_2 \circ \dots \circ T_J\|$$

$$\boldsymbol{T}_j(\boldsymbol{x}) = \eta(\boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{b}_j)$$

$$\eta(t) = \begin{cases} 0 & t < 0 \\ t & t \geqslant 0 \end{cases}$$

$\longrightarrow$ At what cost?

How big is n?.

.

LECTURE 32: BASES FOR INCOMPLETE DECOMPOSITION

1. Preconditioning

In late nineteenth century, telegrapher's equations, a system of linear PDEs for current $I(x,t)$ and voltage $V(x,t)$

$$\frac{\partial}{\partial x} V(x,t) = -L \frac{\partial}{\partial t} I(x,t) - RI(x,t)$$

$$\frac{\partial}{\partial x} I(x,t) = -C \frac{\partial}{\partial t} C(x,t) - GV(x,t)$$

Heaviside avoided solution of the PDEs by reduction to an algebraic formulation historical formulation, e.g., for the ODE for $y(t)$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + ay = b$$

Heaviside considered the associated algebraic problem for $Y(s)$

$$sY + aY = b \Longrightarrow Y(s) = \frac{b}{a+s} \Longrightarrow y(t) = \mathscr{L}^{-1}[Y(s)]$$

> Why should I refuse a good dinner simply because I don't understand the digestive processes involved? (Heaviside, ?)

Heaviside's formal framework (1890's) for solving ODEs was discounted since it lacked mathematical rigour.

- Russian mathematician 1920's established first results (Vladimirov)
- Theory of Distributions (Schwartz, 1950s)

2. Multigrid

Consider function $f:\mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g:\mathbb{R}^n \longrightarrow \mathbb{R}$, $T:\mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\|f - g \circ T\| < \varepsilon$$

for some $\varepsilon > 0$.

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} (f, \phi_j)\phi_j = \sum_{j=1}^{n} c_j \phi_j$$

The approximation is convergent if

$$\lim_{n\to\infty} \|f - g \circ T\| = 0,$$

This assumes $c_j = (f, \phi_j)$ rapidly decrease.

THEOREM. *(Parseval) The Fourier transform is unitary. For $A, B:\mathbb{R} \longrightarrow \mathbb{C}$, square integrable, 2π-periodic with Fourier series*

$$A(t) = \sum_{n=-\infty}^{\infty} a_n e^{int}, B(t) = \sum_{n=-\infty}^{\infty} b_n e^{int},$$

$$\sum_{n=-\infty}^{\infty} a_n \bar{b}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} A(t)\bar{B}(t)\,\mathrm{d}t.$$

Bessel inequality:

$$\sum_{j=1}^{n} |(f, \phi_j)|^2 \leqslant \|f\|_2 .$$

Fourier coefficient decay: for $f \in C^{(k-1)}(\mathbb{R})$, $f^{(k-1)}$ absolutely continuous,

$$|c_n| \leqslant \min_{0 \leqslant j \leqslant k} \frac{\|f^{(j)}\|_1}{|n|^j}.$$

In practice: coefficients decay as

- $1/n$ for functions with discontinuities on a set of Lebesgue measure 0;
- $1/n^2$ for functions with discontinuous first derivative on a set of Lebesgue measure 0;
- $1/n^3$ for functions with discontinuous second derivative on a set of Lebesgue measure 0.

Fourier coefficients for analytic functions decay faster than any monomial power $c_n = o(n^{-p}), \forall p \in \mathbb{N}$, a property known as exponential convergence.

Denote such approximations by $\mathcal{L}$, and they are linear

$$\mathcal{L}(\alpha f + \beta g) = \alpha \mathcal{L}(f) + \beta \mathcal{L}(g)$$

-

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j$$

Let $\Phi_n = \{\varphi_{k(1)}, \varphi_{k(2)}, \dots, \varphi_{k(n)}\}$ such

$$(f, \varphi_{k(1)}) \geqslant (f, \varphi_{k(2)}) \geqslant \cdots \geqslant (f, \varphi_{k(n)}).$$

Choose $c_j = (f, \varphi_{k(j)})$, and

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j .$$

Denote such approximations by $\mathcal{G}$, and they are non-linear.

3. Random multigrid and stochastic descent

Consider function $f:\mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g:\mathbb{R}^n \longrightarrow \mathbb{R}$, $T:\mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\|f - g \circ T\| < \varepsilon$$

for some $\varepsilon > 0$.

What questions do you ask?

Does T exist?. $\forall f, \varepsilon, \exists T,$ such that $\|f - g \circ T\| < \varepsilon$

Can arbitrary ε be achieved?.

Can we construct T?.

→ By what procedure?

$$T = T_1 \circ T_2 \circ \ldots \circ T_J$$

with T_i simple modifications of identity (ReLU)

$$\min_{T_1,\ldots T_J} \|f - g \circ T_1 \circ T_2 \circ \ldots \circ T_J\|$$

$$\boldsymbol{T}_j(\boldsymbol{x}) = \eta(\boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{b}_j)$$

$$\eta(t) = \begin{cases} 0 & t < 0 \\ t & t \geqslant 0 \end{cases}$$

→ At what cost?

How big is n?.

.

LECTURE 33: MULTIPLE OPERATORS

1. Semi-discretization

In late nineteenth century, telegrapher's equations, a system of linear PDEs for current $I(x,t)$ and voltage $V(x,t)$

$$\frac{\partial}{\partial x}V(x,t) = -L\frac{\partial}{\partial t}I(x,t) - RI(x,t)$$

$$\frac{\partial}{\partial x}I(x,t) = -C\frac{\partial}{\partial t}C(x,t) - GV(x,t)$$

Heaviside avoided solution of the PDEs by reduction to an algebraic formulation historical formulation, e.g., for the ODE for $y(t)$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + ay = b$$

Heaviside considered the associated algebraic problem for $Y(s)$

$$sY + aY = b \Longrightarrow Y(s) = \frac{b}{a+s} \Longrightarrow y(t) = \mathscr{L}^{-1}[Y(s)]$$

> Why should I refuse a good dinner simply because I don't understand the digestive processes involved? (Heaviside, ?)

Heaviside's formal framework (1890's) for solving ODEs was discounted since it lacked mathematical rigour.

- Russian mathematician 1920's established first results (Vladimirov)
- Theory of Distributions (Schwartz, 1950s)

2. Method of lines

Consider function $f:\mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g:\mathbb{R}^n \longrightarrow \mathbb{R}$, $T:\mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\|f - g \circ T\| < \varepsilon$$

for some $\varepsilon > 0$.

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} (f, \phi_j)\phi_j = \sum_{j=1}^{n} c_j \phi_j$$

The approximation is convergent if

$$\lim_{n\longrightarrow\infty} \|f - g \circ T\| = 0,$$

This assumes $c_j = (f, \phi_j)$ rapidly decrease.

THEOREM. *(Parseval) The Fourier transform is unitary. For $A, B:\mathbb{R} \longrightarrow \mathbb{C}$, square integrable, 2π-periodic with Fourier series*

$$A(t) = \sum_{n=-\infty}^{\infty} a_n e^{int}, B(t) = \sum_{n=-\infty}^{\infty} b_n e^{int},$$

$$\sum_{n=-\infty}^{\infty} a_n \bar{b}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} A(t)\bar{B}(t)\,\mathrm{d}t.$$

Bessel inequality:

$$\sum_{j=1}^{n} |(f, \phi_j)|^2 \leqslant \|f\|_2.$$

Fourier coefficient decay: for $f \in C^{(k-1)}(\mathbb{R})$, $f^{(k-1)}$ absolutely continuous,

$$|c_n| \leqslant \min_{0\leqslant j\leqslant k} \frac{\|f^{(j)}\|_1}{|n|^j}.$$

In practice: coefficients decay as

- $1/n$ for functions with discontinuities on a set of Lebesgue measure 0;
- $1/n^2$ for functions with discontinuous first derivative on a set of Lebesgue measure 0;
- $1/n^3$ for functions with discontinuous second derivative on a set of Lebesgue measure 0.

Fourier coefficients for analytic functions decay faster than any monomial power $c_n = o(n^{-p}), \forall p \in \mathbb{N}$, a property known as exponential convergence.

Denote such approximations by $\mathcal{L}$, and they are linear

$$\mathcal{L}(\alpha f + \beta g) = \alpha \mathcal{L}(f) + \beta \mathcal{L}(g)$$

-

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j$$

Let $\Phi_n = \{\varphi_{k(1)}, \varphi_{k(2)}, \dots, \varphi_{k(n)}\}$ such

$$(f, \varphi_{k(1)}) \geqslant (f, \varphi_{k(2)}) \geqslant \cdots \geqslant (f, \varphi_{k(n)}).$$

Choose $c_j = (f, \varphi_{k(j)})$, and

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j .$$

Denote such approximations by $\mathcal{G}$, and they are non-linear.

3. Implicit-explicit methods

Consider function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g : \mathbb{R}^n \longrightarrow \mathbb{R}$, $T : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\| f - g \circ T \| < \varepsilon$$

for some $\varepsilon > 0$.

What questions do you ask?

Does T exist?. $\forall f, \varepsilon, \exists T, \text{such that} \|f - g \circ T\| < \varepsilon$

Can arbitrary ε be achieved?.

Can we construct T?.

→ By what procedure?

$$T = T_1 \circ T_2 \circ \ldots \circ T_J$$

with T_i simple modifications of identity (ReLU)

$$\min_{T_1, \ldots T_J} \|f - g \circ T_1 \circ T_2 \circ \ldots \circ T_J\|$$

$$\boldsymbol{T}_j(\boldsymbol{x}) = \eta(\boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{b}_j)$$

$$\eta(t) = \begin{cases} 0 & t < 0 \\ t & t \geqslant 0 \end{cases}$$

→ At what cost?

How big is n?.

.

LECTURE 34: OPERATOR-INDUCED BASES

1. Spectral methods

In late nineteenth century, telegrapher's equations, a system of linear PDEs for current $I(x,t)$ and voltage $V(x,t)$

$$\frac{\partial}{\partial x}V(x,t) = -L\frac{\partial}{\partial t}I(x,t) - RI(x,t)$$

$$\frac{\partial}{\partial x}I(x,t) = -C\frac{\partial}{\partial t}C(x,t) - GV(x,t)$$

Heaviside avoided solution of the PDEs by reduction to an algebraic formulation historical formulation, e.g., for the ODE for $y(t)$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + ay = b$$

Heaviside considered the associated algebraic problem for $Y(s)$

$$sY + aY = b \Longrightarrow Y(s) = \frac{b}{a+s} \Longrightarrow y(t) = \mathscr{L}^{-1}[Y(s)]$$

> Why should I refuse a good dinner simply because I don't understand the digestive processes involved? (Heaviside, ?)

Heaviside's formal framework (1890's) for solving ODEs was discounted since it lacked mathematical rigour.

- Russian mathematician 1920's established first results (Vladimirov)
- Theory of Distributions (Schwartz, 1950s)

2. Quasi-spectral methods

Consider function $f:\mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g:\mathbb{R}^n \longrightarrow \mathbb{R}$, $T:\mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\|f - g \circ T\| < \varepsilon$$

for some $\varepsilon > 0$.

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} (f, \phi_j)\phi_j = \sum_{j=1}^{n} c_j \phi_j$$

The approximation is convergent if

$$\lim_{n\to\infty} \|f - g \circ T\| = 0,$$

This assumes $c_j = (f, \phi_j)$ rapidly decrease.

THEOREM. *(Parseval) The Fourier transform is unitary. For $A, B\colon \mathbb{R} \to \mathbb{C}$, square integrable, 2π-periodic with Fourier series*

$$A(t) = \sum_{n=-\infty}^{\infty} a_n e^{int}, B(t) = \sum_{n=-\infty}^{\infty} b_n e^{int},$$

$$\sum_{n=-\infty}^{\infty} a_n \bar{b}_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} A(t)\bar{B}(t)\,\mathrm{d}t.$$

Bessel inequality:

$$\sum_{j=1}^{n} |(f, \phi_j)|^2 \leqslant \|f\|_2.$$

Fourier coefficient decay: for $f \in C^{(k-1)}(\mathbb{R})$, $f^{(k-1)}$ absolutely continuous,

$$|c_n| \leqslant \min_{0\leqslant j\leqslant k} \frac{\|f^{(j)}\|_1}{|n|^j}.$$

In practice: coefficients decay as

- $1/n$ for functions with discontinuities on a set of Lebesgue measure 0;
- $1/n^2$ for functions with discontinuous first derivative on a set of Lebesgue measure 0;
- $1/n^3$ for functions with discontinuous second derivative on a set of Lebesgue measure 0.

Fourier coefficients for analytic functions decay faster than any monomial power $c_n = o(n^{-p}), \forall p \in \mathbb{N}$, a property known as exponential convergence.

Denote such approximations by $\mathcal{L}$, and they are linear

$$\mathcal{L}(\alpha f + \beta g) = \alpha \mathcal{L}(f) + \beta \mathcal{L}(g)$$

•

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j$$

Let $\Phi_n = \{\varphi_{k(1)}, \varphi_{k(2)}, \dots, \varphi_{k(n)}\}$ such

$$(f, \varphi_{k(1)}) \geq (f, \varphi_{k(2)}) \geq \cdots \geq (f, \varphi_{k(n)}).$$

Choose $c_j = (f, \varphi_{k(j)})$, and

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j .$$

Denote such approximations by $\mathcal{G}$, and they are non-linear.

3. Fast transforms

Consider function $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g : \mathbb{R}^n \longrightarrow \mathbb{R}$, $T : \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\| f - g \circ T \| < \varepsilon$$

for some $\varepsilon > 0$.

What questions do you ask?

Does T exist?. $\forall f, \varepsilon, \exists T,$ such that $\| f - g \circ T \| < \varepsilon$

Can arbitrary ε be achieved?.

Can we construct T?.

→ By what procedure?

$$T = T_1 \circ T_2 \circ \ldots \circ T_J$$

with T_i simple modifications of identity (ReLU)

$$\min_{T_1,\ldots T_J} \|f - g \circ T_1 \circ T_2 \circ \ldots \circ T_J\|$$

$$\boldsymbol{T}_j(\boldsymbol{x}) = \eta(\boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{b}_j)$$

$$\eta(t) = \begin{cases} 0 & t < 0 \\ t & t \geqslant 0 \end{cases}$$

→ At what cost?

How big is n?.

.

LECTURE 35: NONLINEAR OPERATORS

1. Advection equation

In late nineteenth century, telegrapher's equations, a system of linear PDEs for current $I(x,t)$ and voltage $V(x,t)$

$$\frac{\partial}{\partial x} V(x,t) = -L \frac{\partial}{\partial t} I(x,t) - RI(x,t)$$

$$\frac{\partial}{\partial x} I(x,t) = -C \frac{\partial}{\partial t} C(x,t) - GV(x,t)$$

Heaviside avoided solution of the PDEs by reduction to an algebraic formulation historical formulation, e.g., for the ODE for $y(t)$

$$\frac{\mathrm{d}y}{\mathrm{d}t} + ay = b$$

Heaviside considered the associated algebraic problem for $Y(s)$

$$sY + aY = b \Longrightarrow Y(s) = \frac{b}{a+s} \Longrightarrow y(t) = \mathcal{L}^{-1}[Y(s)]$$

> Why should I refuse a good dinner simply because I don't understand the digestive processes involved? (Heaviside, ?)

Heaviside's formal framework (1890's) for solving ODEs was discounted since it lacked mathematical rigour.

- Russian mathematician 1920's established first results (Vladimirov)
- Theory of Distributions (Schwartz, 1950s)

2. Convection equation

Consider function $f: \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g: \mathbb{R}^n \longrightarrow \mathbb{R}$, $T: \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\| f - g \circ T \| < \varepsilon$$

for some $\varepsilon > 0$.

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} (f, \phi_j)\phi_j = \sum_{j=1}^{n} c_j \phi_j$$

The approximation is convergent if

$$\lim_{n \to \infty} \| f - g \circ T \| = 0,$$

This assumes $c_j = (f, \phi_j)$ rapidly decrease.

THEOREM. *(Parseval) The Fourier transform is unitary. For $A, B: \mathbb{R} \to \mathbb{C}$, square integrable, 2π-periodic with Fourier series*

$$A(t) = \sum_{n=-\infty}^{\infty} a_n e^{int}, B(t) = \sum_{n=-\infty}^{\infty} b_n e^{int},$$

$$\sum_{n=-\infty}^{\infty} a_n \bar{b}_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} A(t) \bar{B}(t) \, \mathrm{d}t.$$

Bessel inequality:

$$\sum_{j=1}^{n} |(f, \phi_j)|^2 \leqslant \|f\|_2 .$$

Fourier coefficient decay: for $f \in C^{(k-1)}(\mathbb{R})$, $f^{(k-1)}$ absolutely continuous,

$$|c_n| \leqslant \min_{0 \leqslant j \leqslant k} \frac{\|f^{(j)}\|_1}{|n|^j}.$$

In practice: coefficients decay as

- $1/n$ for functions with discontinuities on a set of Lebesgue measure 0;
- $1/n^2$ for functions with discontinuous first derivative on a set of Lebesgue measure 0;
- $1/n^3$ for functions with discontinuous second derivative on a set of Lebesgue measure 0.

Fourier coefficients for analytic functions decay faster than any monomial power $c_n = o(n^{-p}), \forall p \in \mathbb{N}$, a property known as exponential convergence.

Denote such approximations by $\mathcal{L}$, and they are linear

$$\mathcal{L}(\alpha f + \beta g) = \alpha \mathcal{L}(f) + \beta \mathcal{L}(g)$$

-

Choose a basis set (Monomials, Exponentials, Wavelets) $\{\phi_1, \phi_2, \dots\}$ to approximation of $L^2(\mathbb{R})$ functions in Hibert space

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j$$

Let $\Phi_n = \{\varphi_{k(1)}, \varphi_{k(2)}, \ldots, \varphi_{k(n)}\}$ such

$$(f, \varphi_{k(1)}) \geqslant (f, \varphi_{k(2)}) \geqslant \cdots \geqslant (f, \varphi_{k(n)}).$$

Choose $c_j = (f, \varphi_{k(j)})$, and

$$g_n(t) = \sum_{j=1}^{n} c_j \phi_j .$$

Denote such approximations by $\mathcal{G}$, and they are non-linear.

3. Discontinuous solutions

Consider function $f: \mathbb{R}^d \longrightarrow \mathbb{R}$, $d \gg 1$ assumed large, f of unknown form, difficult to compute for general input. Seek $g: \mathbb{R}^n \longrightarrow \mathbb{R}$, $T: \mathbb{R}^d \longrightarrow \mathbb{R}^n$ such that

$$\|f - g \circ T\| < \varepsilon$$

for some $\varepsilon > 0$.

What questions do you ask?

Does T exist?. $\forall f, \varepsilon, \exists T,$ such that $\|f - g \circ T\| < \varepsilon$

Can arbitrary ε be achieved?.

Can we construct T?.

$\longrightarrow$ By what procedure?

$$T = T_1 \circ T_2 \circ \ldots \circ T_J$$

with T_i simple modifications of identity (ReLU)

$$\min_{T_1, \ldots T_J} \|f - g \circ T_1 \circ T_2 \circ \ldots \circ T_J\|$$

$$\boldsymbol{T}_j(\boldsymbol{x}) = \eta(\boldsymbol{A}_j \boldsymbol{x} + \boldsymbol{b}_j)$$

$$\eta(t) = \begin{cases} 0 & t < 0 \\ t & t \geqslant 0 \end{cases}$$

$\longrightarrow$ At what cost?

How big is n?.

.

CHAPTER 6

INTEGRAL CONSERVATION LAWS

Part III

Nonlinear Approximation

CHAPTER 7

COMPUTATIONAL TOPOLOGY

CHAPTER 8

COMPUTATIONAL GEOMETRY

CHAPTER 9

STOCHASTIC DIFFERENTIAL EQUATIONS

CHAPTER 10

RANDOMIZED LINEAR ALGEBRA