Math Foundations

of engineering analysis

Rico A. R. Picone
Department of Mechanical Engineering
Saint Martin's University

Sunday 5 December, 2021

Mathematics itself

itself.tru Truth

1 Before we can discuss mathematical truth, we should begin with a discussion of truth itself.[1] It is important to note that this is obviously extremely incomplete. My aim is to give a sense of the subject via brutal (mis)abbreviation.

2 Of course, the study of truth cannot but be entangled with the study of the world as such (metaphysics) and of knowledge (epistemology). Some of the following theories presuppose or imply a certain metaphysical and/or epistemological theory, but which these are is controversial.

Neo-classical theories of truth

3 The neo-classical theories of truth take for granted that there is truth and attempt to explain what its precise nature is (Michael Glanzberg. ?Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018). What are provided here are modern understandings of theories developed primarily in the early 20th century.

The correspondence theory

4 A version of what is called the correspondence theory of truth is the following.

correspondence theory

> A proposition is true iff there is an existing entity in the world that corresponds with it.

Such existing entities are called facts. Facts are relational in that their parts (e.g. subject, predicate, etc.) are related in a certain way.

facts

5 Under this theory, then, if a proposition does not correspond to a fact, it is false.

false

[1] For much of this lecture I rely on the thorough overview of Glanzberg. (Michael Glanzberg. ?Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018)

6 This theory of truth is rather intuitive and consistently popular (Marian David. ?The Correspondence Theory of Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2016. Metaphysics Research Lab, Stanford University, 2016. A detailed overview of the correspondence theory of truth.).

The coherence theory

coherence theory

7 The coherence theory of truth is adamant that the truth of any given proposition is only as good as its holistic system of propositions.[2] This includes (but typically goes beyond) a requirement for consistency of a given proposition with the whole and the self-consistency of the whole, itself—sometimes called coherence.

coherence

8 For parallelism, let's attempt a succinct formulation of this theory, cast in terms of propositions.

> A proposition is true iff it is has coherence with a system of propositions.

9 Note that this has no reference to facts, whatsoever. However, it need not necessarily preclude them.

The pragmatic theory

10 Of the neo-classical theories of truth, this is probably the least agreed upon as having a single clear statement (Glanzberg, ?Truth?). However, as with pragmatism in general,[3] the pragmatic truth is oriented practically.

pragmatism

11 Perhaps the most important aspect of this theory is that it is thoroughly a correspondence theory, agreeing that true propositions are those that correspond to the world. However, there is a different focus here that differentiates it from

[2] This is typically put in terms of "beliefs" or "judgments," but for brevity and parallelism I have cast it in terms of propositions. It is to this theory I have probably committed the most violence.

[3] Pragmatism was an American philosophical movement of the early 20th century that valued the success of "practical" application of theories. For an introduction, see Legg and Hookway. (Catherine Legg and Christopher Hookway. ?Pragmatism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019. An introductory article on the philsophical movement "pragmatism." It includes an important clarification of the pragmatic slogan, "truth is the end of inquiry.")

correspondence theory, proper: it values as more true that which has some sort of practical use in human life.

12 We'll try to summarize pragmatism in two slogans with slightly different emphases; here's the first, again cast in propositional parallel.

> A proposition is true iff it works.[4]

Now, there are two ways this can be understood: (a) the proposition "works" in that it empirically corresponds to the world or (b) the proposition "works" in that it has an effect that some agent intends. The former is pretty standard correspondence theory. The latter is new and fairly obviously has ethical implications, especially today.

13 Let us turn to a second formulation.

> A proposition is true if it corresponds with a process of inquiry.[5]

inquiry

This has two interesting facets: (a) an agent's active inquiry creates truth and (b) it is a sort of correspondence theory that requires a correspondence of a proposition with a process of inquiry, not, as in the correspondence theory, with a fact about the world. The latter has shades of both correspondence theory and coherence theory.

The picture theory

Before we delve into this theory, we must take a moment to clarify some terminology.

States of affairs and facts

facts

14 When discussing the correspondence theory, we have used the term fact to mean an actual state of things in the world. A problem arises in the correspondence theory, here. It says that a proposition is true iff there is a fact

[4] This is especially congruent with the work of William James (Legg and Hookway, ?Pragmatism?).

[5] This is especially congruent with the work of Charles Sanders Peirce (ibidem).

that corresponds with it. What of a negative proposition like “there are no cows in Antarctica”? We would seem to need a corresponding “negative fact” in the world to make this true. If a fact is taken to be composed of a complex of actual objects and relations, it is hard to imagine such facts.[6]

15 Furthermore, if a proposition is true, it seems that it is the corresponding fact that makes it so; what, then, makes a proposition false, since there is no fact to support the falsity? (Mark Textor. ?States of Affairs? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University, 2016)

16 And what of nonsense? There are some propositions like “there is a round cube” that are neither true nor false. However, the preceding correspondence theory cannot differentiate between false and nonsensical propositions.

17 A state of affairs is something possible that may or may not be actual (ibidem). If a state of affairs is actual, it is said to obtain. The picture theory will make central this concept instead of that of the fact.

state of affairs
obtain
model
picture

The picture theory of meaning (and truth)

18 The picture theory of meaning uses the analogy of the model or picture to explain the meaningfulness of propositions.[7]

> A proposition names possible objects and arranges these names to correspond to a state of affairs.

See Fig. tru.1. This also allows for an easy account of truth, falsity, and nonsense.

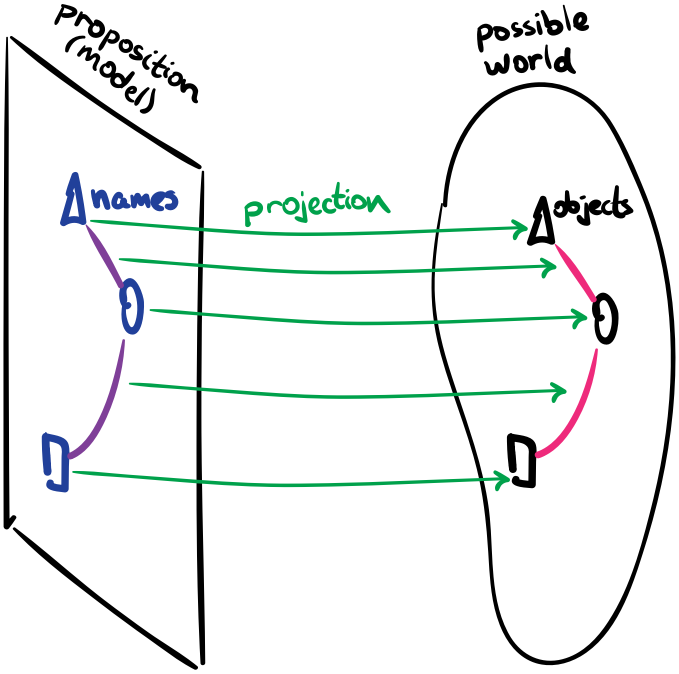


Figure tru.1: a representation of the picture theory.

nonsense A sentence that appears to be a proposition is actually not if the arrangement of named objects is

[6] But Barker and Jago (Stephen Barker and Mark Jago. ?Being Positive About Negative Facts? inPhilosophy and Phenomenological Research: 85.1 [2012], pages 117–138. DOI: 10.1111/j.1933-1592.2010.00479.x) have attempted just that.

[7] See Wittgenstein, (Ludwig Wittgenstein. Tractatus Logico-Philosophicus. byeditorC.}, familyi=C., given=K. Ogden, giveni=. O. Project Gutenberg. International Library of Psychology Philosophy and Scientific Method. Kegan Paul, Trench, Trubner & Co., Ltd., 1922. A brilliant work on what is possible to express in language—and what is not. As Wittgenstein puts it, “What can be said at all can be said clearly; and whereof one cannot speak thereof one must be silent.”) Biletzki and Matar, (Anat Biletzki and Anat Matar. ?Ludwig Wittgenstein? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford University, 2018. An introduction to Wittgenstein and his thought.) glock2016 (), and Dolby. (David Dolby. ?Wittgenstein on Truth? inA Companion to Wittgenstein: John Wiley & Sons, Ltd, 2016. chapter 27, pages 433–442. ISBN: 9781118884607. DOI: 10.1002/9781118884607.ch27)

impossible. Such a sentence is simply nonsense.

nonsense

truth A proposition is true if the state of affairs it depicts obtains.

falsity A proposition is false if the state of affairs it depicts does not obtain.

19 Now, some (Hans Johann Glock. ?Truth in the Tractatus? inSynthese: 148.2 [january 2006], pages 345–368. ISSN: 1573-0964. DOI: 10.1007/s11229-004-6226-2) argue this is a correspondence theory and others (David Dolby. ?Wittgenstein on Truth? inA Companion to Wittgenstein: John Wiley & Sons, Ltd, 2016. chapter 27, pages 433–442. ISBN: 9781118884607. DOI: 10.1002/9781118884607.ch27) that it is not. In any case, it certainly solves some issues that have plagued the correspondence theory.

“What cannot be said must be shown”

20 Something the picture theory does is declare a limit on what can meaningfully be said. A proposition (as defined above) must be potentially true or false. Therefore, something that cannot be false (something necessarily true) cannot be a proposition (ibidem). And there are certain things that are necessarily true for language itself to be meaningful—paradigmatically, the logical structure of the world. What a proposition does, then, is show, via its own logical structure, the necessary (for there to be meaningful propositions at all) logical structure of the world.[8]

21 An interesting feature of this perspective is that it opens up language itself to analysis and limitation.[9] And, furthermore, it suggests that the set of what is, is smaller than the set of what can be meaningfully spoken about.

[8] See, also, (Slavoj i ek. Less Than Nothing: Hegel and the Shadow of Dialectical Materialism. Verso, 2012. ISBN: 9781844678976. This is one of the most interesting presentations of Hegel and Lacan by one of the most exciting contemporary philosophers. pp. 25-6), from whom I stole the section title.

language itself

[9] This was one of the contributions to the “linguistic turn” (Wikipedia. Linguistic turn—Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Linguistic%20turn&oldid=922305269. [Online; accessed 23-October-2019]. 2019. Hey, we all do it.) of philosophy in the early 20th century.

The relativity of truth

22 Each subject (i.e. agent) in the world, with their propositions, has a perspective: a given moment, a given place, an historical-cultural-linguistic situation. At the very least, the truth of propositions must account for this. Just how a theory of truth should do so is a matter of significant debate (Maria Baghramian and J. Adam Carter. ?Relativism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2019. Metaphysics Research Lab, Stanford University, 2019).

perspective

23 Some go so far as to be skeptical about truth (Peter Klein. ?Skepticism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Summer 2015. Metaphysics Research Lab, Stanford University, 2015), regarding it to be entirely impossible. Others say that while a proposition may or may not be true, we could never come to know this.

skepticism

24 Often underlying this conversation is the question of there being a common world in which we all participate, and, if so, whether or not we can properly represent this world in language such that multiple subjects could come to justifiably agree or disagree on the truth of a proposition. If every proposition is so relative that it is relevant to only the proposer, truth would seem of little value. On the other hand, if truth is understood to be "objective"—independent of subjective perspective—a number of objections can be made (Baghramian and Carter, ?Relativism?), such as that there is no non-subjective perspective from which to judge truth.

Other ideas about truth

25 There are too many credible ideas about truth to attempt a reasonable summary;

however, I will attempt to highlight a few important ones.

Formal methods

26 A set of tools was developed for exploring theories of truth, especially correspondence theories.[10] Focus turned from beliefs to sentences, which are akin to propositions. (Recall that the above theories have already been recast in the more modern language of propositions.) Another aspect of these sentences under consideration is that they begin to be taken as interpreted sentences: they are already have meaning.

beliefs
sentences

interpreted sentences

27 Beyond this, several technical apparatus are introduced that formalize criteria for truth. For instance, a sentence is given a sign ϕ. A need arises to distinguish between the quotation of sentence ϕ and the unqoted sentence ϕ, which is then given the quasi-quotation notation $\ulcorner\phi\urcorner$. For instance, let ϕ stand for snow is white; then $\phi \to$ snow is white and $\ulcorner\phi\urcorner \to$ 'snow is white'. Tarski introduces Convention T, which states that for a fixed language **L** with fully interpreted sentences, (Glanzberg, ?Truth?)

quasi−quotation

Convention T

> An adequate theory of truth for **L** must imply for each sentence ϕ of **L** $\ulcorner\phi\urcorner$ is true if and only if ϕ.

Using the same example, then,

> 'snow is white' if and only if snow is white.

Convention T states a general rule for the adequacy of a theory of truth and is used in several contemporary theories.

28 We can see that these formal methods get quite technical and fun! For more, see Hodges, Gómez-Torrente and Hylton and Kemp.[11]

[10] Especially notable here is the work of Alfred Tarski in the mid-20^{t}h century.

[11] Wilfrid Hodges. ?Tarski's Truth Definitions? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018; Mario Gómez-Torrente. ?Alfred Tarski? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019; Peter Hylton and Gary Kemp. ?Willard Van Orman Quine? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019.

Deflationary theories

29 Deflationary theories of truth try to minimize or eliminate altogether the concept of or use of the term ‘truth’. For instance, the redundancy theory claim that (Glanzberg, ?Truth?):

redundancy theory

> To assert that ⌜ϕ⌝ is true is just to assert that ϕ.

Therefore, we can eliminate the use of ‘is true’.
30 For more of less, see Stoljar and Damnjanovic.[12]

Language

31 It is important to recognize that language mediates truth; that is, truth is embedded in language. The way language in general affects theories of truth has been studied extensively. For instance, whether the truth-bearer is a belief or a proposition or a sentence—or something else—has been much discussed. The importance of the meaning of truth-bearers like sentences has played another large role. Theories of meaning, like the picture theory presented above, are often closely intertwined with theories of truth.

truth−bearer

meaning

32 One of the most popular theories of meaning is called the theory of use:

theory of use

> For a large class of cases of the employment of the word “meaning” – though not for all – this word can be explained in this way: the meaning of a word is its use in the language. (L. Wittgenstein, P.M.S. Hacker and J. Schulte. Philosophical Investigations. Wiley, 2010. ISBN: 9781444307979)

This theory is accompanied by the concept of language-games, which are loosely defined as

language−games

[12] Daniel Stoljar and Nic Damnjanovic. ?The Deflationary Theory of Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2014. Metaphysics Research Lab, Stanford University, 2014.

rule-based contexts within which sentences have uses. The idea is that the meaning of a given sentence is its use in a network of meaning that is constantly evolving. This view tends to be understood as deflationary or relativistic about truth.

Metaphysical and epistemological considerations

33 We began with the recognition that truth is intertwined with metaphysics and epistemology. Let's consider a few such topics.

34 The first is metaphysical realism, which claims that there is a world existing objectively: independently of how we think about or describe it. This "realism" tends to be closely tied to, yet distinct from, scientific realism, which goes further, claiming the world is "actually" as science describes, independently of the scientific descriptions (e.g. there are actual objects corresponding to the phenomena we call atoms, molecules, light particles, etc.).

metaphysical realism

scientific realism

35 There have been many challenges to the realist claim (for some recent versions, see Khlentzos[13]) put forth by what is broadly called anti-realism. These vary, but often challenge the ability of realists to properly link language to supposed objects in the world.

anti−realism

36 Metaphysical idealism has been characterized as claiming that "mind" or "subjectivity" generate or completely compose the world, which has no being outside mind. Epistemological idealism, on the other hand, while perhaps conceding that there is a world independent of mind, claims all knowledge of the world is created through mind and for mind and therefore can never escape a sort of mind-world gap.[14] This epistemological idealism has been highly influential since the work of Immanuel Kant (I. Kant, P. Guyer and

Metaphysical idealism

Epistemological idealism

[13] Drew Khlentzos. ?Challenges to Metaphysical Realism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University, 2016.

[14] These definitions are explicated by Guyer and Horstmann. (Paul Guyer and Rolf-Peter Horstmann. ?Idealism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2018. Metaphysics Research Lab, Stanford University, 2018)

A.W. Wood. Critique of Pure Reason. The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press, 1999. ISBN: 9781107268333) in the late 18th century, which ushered in the idea of the noumenal world in-itself and the phenomenal world, which is how the noumenal world presents to us. Many have held that phenomena can be known through inquiry, whereas noumena are inaccessible. Furthermore, what can be known is restricted by the categories pre-existent in the knower.

noumenal world

phenomenal world

37 Another approach, taken by Georg Wilhelm Friedrich Hegel (Paul Redding. ?Georg Wilhelm Friedrich Hegel? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford University, 2018) and other German idealists following Kant, is to reframe reality as thoroughly integrating subjectivity (G.W.F. Hegel and A.V. Miller. Phenomenology of Spirit. Motilal Banarsidass, 1998. ISBN: 9788120814738; Slavoj i ek. Less Than Nothing: Hegel and the Shadow of Dialectical Materialism. Verso, 2012. ISBN: 9781844678976. This is one of the most interesting presentations of Hegel and Lacan by one of the most exciting contemporary philosophers.); that is, "everything turns on grasping and expressing the True, not only as Substance, but equally as Subject." A subject's proposition is true inasmuch as it corresponds with its Notion (approximately: the idea or meaning for the subject). Some hold that this idealism is compatible with a sort of metaphysical realism, at least as far as understanding is not independent of but rather beholden to reality (i ek, Less Than Nothing: Hegel and the Shadow of Dialectical Materialism, p. 906 ff.).

Notion

38 Clearly, all these ideas have many implications for theories of truth and vice versa.

Where this leaves us

39 The truth is hard. What may at first appear to be a simple concept becomes complex upon analysis. It is important to recognize that we have only sampled some highlights of the theories of truth. I recommend further study of this fascinating topic.

40 Despite the difficulties of finding definitive grounds for understanding truth, we are faced with the task of provisionally forging ahead. Much of what follows in the study of mathematics makes certain implicit and explicit assumptions about truth. However, we have found that the foundations of these assumptions may themselves be problematic. It is my contention that, despite the lack of clear foundations, it is still worth studying engineering analysis, its mathematical foundations, and the foundations of truth itself. My justification for this claim is that I find the utility and the beauty of this study highly rewarding.

itself.found The foundations of mathematics

1 Mathematics has long been considered exemplary for establishing truth. Primarily, it uses a method that begins with axioms—unproven propositions that include undefined terms—and uses logical deduction to prove other propositions (theorems): to show that they are necessarily true if the axioms are.

axioms

deduction

proof

theorems

2 It may seem obvious that truth established in this way would always be relative to the truth of the axioms, but throughout history this footnote was often obscured by the "obvious" or "intuitive" universal truth of the axioms.[15] For instance, Euclid (Wikipedia. Euclid — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Euclid&oldid=923031048. [Online; accessed 26-October-2019]. 2019) founded geometry—the study of mathematical objects traditionally considered to represent physical space, like points, lines, etc.—on axioms thought so solid that it was not until the early 19th century that Carl Friedrich Gauss (Wikipedia. Carl Friedrich Gauss — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Carl%20Friedrich%20Gauss&oldid=922692291. [Online; accessed 26-October-2019]. 2019) and others recognized this was only one among many possible geometries (M. Kline. Mathematics: The Loss of Certainty. A Galaxy book. Oxford University Press, 1982. ISBN: 9780195030853. A detailed account of the "illogical" development of mathematics and an exposition of its therefore remarkable utility in describing the world.) resting on different axioms. Furthermore, Aristotle (Christopher Shields. ?Aristotle? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University,

Euclid

geometry

Gauss

Aristotle

[15] Throughout this section, for the history of mathematics I rely heavily on Kline. (M. Kline. Mathematics: The Loss of Certainty. A Galaxy book. Oxford University Press, 1982. ISBN: 9780195030853. A detailed account of the "illogical" development of mathematics and an exposition of its therefore remarkable utility in describing the world.)

2016) had acknowledged that reasoning must begin with undefined terms; however, even Euclid (presumably aware of Aristotle's work) seemed to forget this and provided definitions, obscuring the foundations of his work and starting mathematics on a path that for over 2,000 years would forget its own relativity (Kline, Mathematics: The Loss of Certainty, p. 101-2).

3 The foundations of Euclid were even shakier than its murky starting point: several unstated axioms were used in proofs and some proofs were otherwise erroneous. However, for two millennia, mathematics was seen as the field wherein truth could be established beyond doubt.

Algebra ex nihilo

4 Although not much work new geometry appeared during this period, the field of algebra (Wikipedia. Algebra — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Algebra&oldid=920573802. [Online; accessed 26-October-2019]. 2019)—the study of manipulations of symbols standing for numbers in general—began with no axiomatic foundation whatsoever. The Greeks had a notion of rational numbers, ratios of natural numbers (positive integers), and it was known that many solutions to algebraic equations were irrational (could not be expressed as a ratio of integers). But these irrational numbers, like virtually everything else in algebra, were gradually accepted because they were so useful in solving practical problems (they could be approximated by rational numbers and this seemed good enough). The rules of basic arithmetic were accepted as applying to these and other forms of new numbers that arose in algebraic solutions: negative, imaginary, and

algebra

rational numbers
natural numbers
integers

irrational numbers

negative numbers
imaginary numbers

complex numbers.

complex numbers

The application of mathematics to science

5 During this time, mathematics was being applied to optics and astronomy. Sir Isaac Newton then built calculus upon algebra, applying it to what is now known as Newtonian mechanics, which was really more the product of Leonhard Euler (George Smith. ?Isaac Newton? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2008. Metaphysics Research Lab, Stanford University, 2008; Wikipedia. Leonhard Euler — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Leonhard%20Euler&oldid=921824700. [Online; accessed 26-October-2019]. 2019). Calculus introduced its own dubious operations, but the success of mechanics in describing and predicting physical phenomena was astounding. Mathematics was hailed as the language of God (later, Nature).

optics
astronomy
calculus
Newtonian mechanics

The rigorization of mathematics

6 It was not until Gauss created non-Euclidean geometry, in which Euclid's were shown to be one of many possible axioms compatible with the world, and William Rowan Hamilton (Wikipedia. William Rowan Hamilton — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=William%20Rowan%20Hamilton&oldid=923163451. [Online; accessed 26-October-2019]. 2019) created quaternions (Wikipedia. Quaternion — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Quaternion&oldid=920710557. [Online; accessed 26-October-2019]. 2019), a number system in which multiplication is noncommunicative, that it became apparent something was

non−Euclidean geometry

quaternions

fundamentally wrong with the way truth in mathematics had been understood. This started a period of rigorization in mathematics that set about axiomatizing and proving 19th century mathematics. This included the development of symbolic logic, which aided in the process of deductive reasoning.

rigorization

symbolic logic

7 An aspect of this rigorization is that mathematicians came to terms with the axioms that include undefined terms. For instance, a "point" might be such an undefined term in an axiom. A mathematical model is what we create when we attach these undefined terms to objects, which can be anything consistent with the axioms.[16] The system that results from proving theorems would then apply to anything "properly" described by the axioms. So two masses might be assigned "points" in a Euclidean geometric space, from which we could be confident that, for instance, the "distance" between these masses is the Euclidean norm of the line drawn between the points. It could be said, then, that a "point" in Euclidean geometry is implicitly defined by its axioms and theorems, and nothing else. That is, mathematical objects are not inherently tied to the physical objects to which we tend to apply them. Euclidean geometry is not the study of physical space, as it was long considered—it is the study of the objects implicitly defined by its axioms and theorems.

mathematical model

implicit definition

The foundations of mathematics are built

8 The building of the modern foundations mathematics began with clear axioms, solid reasoning (with symbolic logic), and lofty yet seemingly attainable goals: prove theorems to support the already ubiquitous mathematical techniques in geometry, algebra, and calculus from axioms; furthermore, prove that these

[16] The branch of mathematics called model theory concerns itself with general types of models that can be made from a given formal system, like an axiomatic mathematical system. For more on model theory, see Hodges. (Wilfrid Hodges. ?Model Theory? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018) It is noteworthy that the engineering/science use of the term "mathematical model" is only loosely a "model" in the sense of model theory.

axioms (and things they imply) do not contradict each other, i.e. are consistent, and that the axioms are not results of each other (one that can be derived from others is a theorem, not an axiom).

consistent

theorem

9 Set theory is a type of formal axiomatic system that all modern mathematics is expressed with, so set theory is often called the foundation of mathematics (Joan Bagaria. ?Set Theory? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2019. Metaphysics Research Lab, Stanford University, 2019). We will study the basics in ??. The primary objects in set theory are sets: informally, collections of mathematical objects. There is not just one a single set of axioms that is used as the foundation of all mathematics for reasons will review in a moment. However, the most popular set theory is Zermelo-Fraenkel set theory with the axiom of choice (ZFC). The axioms of ZF sans C are as follows. (ibidem)

Set theory

foundation

sets

ZFC set theory

extensionality If two sets A and B have the same elements, then they are equal.

empty set There exists a set, denoted by $\varnothing$ and called the empty set, which has no elements.

pair Given any sets A and B, there exists a set, denoted by $\{A, B\}$, which contains A and B as its only elements. In particular, there exists the set $\{A\}$ which has A as its only element.

power set For every set A there exists a set, denoted by $\mathcal{P}(A)$ and called the power set of A, whose elements are all the subsets of A.

union For every set A, there exists a set, denoted by $\bigcup A$ and called the union of A, whose elements are all the elements of the elements of A.

infinity There exists an infinite set. In particular, there exists a set Z that contains $\varnothing$ and such that if $A \in Z$, then $\bigcup\{A, \{A\}\} \in Z$.

separation For every set A and every given property, there is a set containing exactly the elements of A that have that property. A property is given by a formula φ of the first-order language of set theory. Thus, separation is not a single axiom but an axiom schema, that is, an infinite list of axioms, one for each formula φ.

replacement For every given definable function with domain a set A, there is a set whose elements are all the values of the function.

10 ZFC also has the axiom of choice. (Bagaria, ?Set Theory?)

choice For every set A of pairwise-disjoint non-empty sets, there exists a set that contains exactly one element from each set in A.

The foundations have cracks

11 The foundationalists' goal was to prove that some set of axioms from which all of mathematics can be derived is both consistent (contains no contradictions) and complete (every true statement is provable). The work of Kurt Gödel (Juliette Kennedy. ?Kurt Gödel? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2018. Metaphysics Research Lab, Stanford University, 2018) in the mid 20th century shattered this dream by proving in his first incompleteness theorem that any consistent formal system within which one can do some amount of basic arithmetic is incomplete! His argument is worth reviewing (see Raatikainen[17]), but at its heart is an undecidable statement like "This sentence is

first incompleteness theorem

incomplete

[17] Panu Raatikainen. ?Gödel's Incompleteness Theorems? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018. A through and contemporary description of Gödel's incompleteness theorems, which have significant implications for the foundations and function of mathematics and mathematical truth.

undecidable

unprovable." Let U stand for this statement. If it is true it is unprovable. If it is provable it is false. Therefore, it is true iff it is provable. Then he shows that if a statement A that essentially says "arithmetic is consistent" is provable, then so is the undecidable statement U. But if U is to be consistent, it cannot be provable, and, therefore neither can A be provable!

12 This is problematic. It tells us virtually any conceivable axiomatic foundation of mathematics is incomplete. If one is complete, it is inconsistent (and therefore worthless). One problem this introduces is that a true theorem may be impossible to prove; but, it turns out, we can never know that in advance of its proof if it is provable.

second incompleteness theorem

13 But it gets worse: Gödel's second incompleteness theorem shows that such systems cannot even be shown to be consistent! This means, at any moment, someone could find an inconsistency in mathematics, and not only would we lose some of the theorems: we would lose them all. This is because, by what is called the material implication (Kline, Mathematics: The Loss of Certainty, pp. 187-8, 264), if one contradiction can be found, every proposition can be proven from it. And if this is the case, all (even proven) theorems in the system would be suspect.

material implication

Banach–Tarski paradox

14 Even though no contradiction has yet appeared in ZFC, its axiom of choice, which is required for the proof of most of what has thus far been proven, generates the Banach-Tarski paradox that says a sphere of diameter x can be partitioned into a finite number of pieces and recombined to form two spheres of diameter x. Troubling, to say the least! Attempts were made for a while to eliminate the use of the axiom of choice, but our buddy Gödel later proved that if ZF is consistent, so is ZFC (ibidem, p. 267).

Mathematics is considered empirical

15 Since its inception, mathematics has been applied extensively to the modeling of the world. Despite its cracked foundations, it has striking utility. Many recent leading minds of mathematics, philosophy, and science suggest we treat mathematics as empirical, like any science, subject to its success in describing and predicting events in the world. As Kline[18] summarizes,

empirical

> The upshot […] is that sound mathematics must be determined not by any one foundation which may some day prove to be right. The "correctness" of mathematics must be judged by its application to the physical world. Mathematics is an empirical science much as Newtonian mechanics. It is correct only to the extent that it works and when it does not, it must be modified. It is not a priori knowledge even though it was so regarded for two thousand years. It is not absolute or unchangeable.

[18] Kline, Mathematics: The Loss of Certainty.

itself.reason Mathematical reasoning

itself.overview Mathematical topics in overview

itself.engmath What is mathematics for engineering?

language

itself.exe Exercises for Chapter itself

Mathematical reasoning, logic, and set theory

In order to communicate mathematical ideas effectively, formal languages have been developed within which logic, i.e. deductive (mathematical) reasoning, can proceed.

formal languages

logic

reasoning

Propositions are statements that can be either true ⊤ or false ⊥. Axiomatic systems begin with statements (axioms) assumed true. Theorems are proven by deduction. In many forms of logic, like propositional calculus (Wikipedia. Propositional calculus — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Propositional%20calculus&oldid=914757384. [Online; accessed 29-October-2019]. 2019), compound propositions are constructed via logical connectives like "and" and "or" of atomic propositions (see Lec. sets.logic). In others, like first-order logic (Wikipedia. First-order logic — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=First-order%20logic&oldid=921437906. [Online; accessed 29-October-2019]. 2019), there are also logical quantifiers like "for every" and "there exists."

propositions

theorems

proof

propositional calculus

logical connectives

first–order logic

quantifiers

The mathematical objects and operations about which most propositions are made are expressed in terms of set theory, which was introduced in Lec. itself.found and will be expanded upon in Lec. sets.setintro. We can say that mathematical reasoning is comprised of

set theory

mathematical objects and operations expressed in set theory and logic allows us to reason therewith.

sets.setintro Introduction to set theory

set theory

Set theory is the language of the modern foundation of mathematics, as discussed in Lec. itself.found. It is unsurprising, then, that it arises throughout the study of mathematics. We will use set theory extensively in ?? on probability theory.

The axioms of ZFC set theory were introduced in Lec. itself.found. Instead of proceeding in the pure mathematics way of introducing and proving theorems, we will opt for a more applied approach in which we begin with some simple definitions and include basic operations. A more thorough and still readable treatment is given by Ciesielski[1] and a very gentle version by Enderton.[2]

set

A set is a collection of objects. Set theory gives us a way to describe these collections. Often, the objects in a set are numbers or sets of numbers. However, a set could represent collections of zebras and trees and hairballs. For instance, here are some sets:

field

addition

multiplication

subtraction

division

real numbers

A field is a set with special structure. This structure is provided by the addition $(+)$ and multiplication $(\times)$ operators and their inverses subtraction $(-)$ and division $(\div)$. The quintessential example of a field is the set of real numbers $\mathbb{R}$, which admits these operators, making it a field. The reals $\mathbb{R}$, the complex numbers $\mathbb{C}$, the integers $\mathbb{Z}$, and the natural numbers[3] $\mathbb{N}$ are the fields we typically consider.

set membership

Set membership is the belonging of an object to a set. It is denoted with the symbol $\in$, which can be read "is an element of," for element x and set X:

[1] K. Ciesielski. Set Theory for the Working Mathematician. London Mathematical Society Student Texts. Cambridge University Press, 1997. ISBN: 9780521594653. A readable introduction to set theory.

[2] H.B. Enderton. Elements of Set Theory. Elsevier Science, 1977. ISBN: 9780080570426. A gentle introduction to set theory and mathematical reasoning—a great place to start.

[3] When the natural numbers include zero, we write $\mathbb{N}_0$.

For instance, we might say $7 \in \{1, 7, 2\}$ or $4 \notin \{1, 7, 2\}$. Or, we might declare that a is a real number by stating: $x \in \mathbb{R}$.

set operations

Set operations can be used to construct new sets from established sets. We consider a few common set operations, now.

union

The union $\cup$ of sets is the set containing all the elements of the original sets (no repetition allowed). The union of sets A and B is denoted $A \cup B$. For instance, let $A = \{1, 2, 3\}$ and $B = \{-1, 3\}$; then

intersection

The intersection $\cap$ of sets is a set containing the elements common to all the original sets. The intersection of sets A and B is denoted $A \cap B$. For instance, let $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$; then

empty set

If two sets have no elements in common, the intersection is the empty set $\varnothing = \{\}$, the unique set with no elements.

set difference

The set difference of two sets A and B is the set of elements in A that aren't also in B. It is denoted $A \setminus B$. For instance, let $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$. Then

subset

proper subset

A subset $\subseteq$ of a set is a set, the elements of which are contained in the original set. If the two sets are equal, one is still considered a subset of the other. We call a subset that is not equal to the other set a proper subset $\subset$. For instance, let $A = \{1, 2, 3\}$ and $B = \{1, 2\}$. Then

The complement of a subset is a set of elements of the original set that aren't in the subset. For instance, if $B \subseteq A$, then the complement of B, denoted $\overline{B}$ is

complement

The cartesian product of two sets A and B is denoted $A \times B$ and is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$. It's worthwhile considering the following notation for this definition:

cartesian product

which means "the cartesian product of A and B is the ordered pair (a, b) such that $a \in A$ and $b \in B$" in set-builder notation (Wikipedia. Set-builder notation — Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/w/index.php?title=Set-builder%20notation&oldid=917328223`. [Online; accessed 29-October-2019]. 2019).

set−builder notation

Let A and B be sets. A map or function f from A to B is an assignment of some element $a \in A$ to each element $b \in B$. The function is denoted $f : A \to B$ and we say that f maps each element $a \in A$ to an element $f(a) \in B$ called the value of a under f, or $a \mapsto f(a)$. We say that f has domain A and codomain B. The image of f is the subset of its codomain B that contains the values of all elements mapped by f from its domain A.

map
function
value
domain
codomain
image

sets.logic Logical connectives and quantifiers

In order to make compound propositions, we need to define logical connectives. In order to specify quantities of variables, we need to define logical quantifiers. The following is a form of first-order logic (Wikipedia, First-order logic — Wikipedia, The Free Encyclopedia).

first−order logic

Logical connectives

A proposition can be either true $\top$ and false $\bot$. When it does not contain a logical connective, it is called an atomistic proposition. To combine propositions into a compound proposition, we require logical connectives. They are not ($\neg$), and ($\wedge$), and or ($\vee$). Table logic.1 is a truth table for a number of connectives.

atomistic proposition
compound proposition
logical connectives
not
and
or
truth table

Quantifiers

Logical quantifiers allow us to indicate the quantity of a variable. The universal quantifier symbol $\forall$ means "for all". For instance, let A be a set; then $\forall a \in A$ means "for all elements in A" and gives this quantity variable a. The existential quantifier $\exists$ means "there exists at least one" or "for some". For instance, let A be a set; then $\exists a \in A \ldots$ means "there exists at least one element a in A ...."

universal quantifier symbol

existential quantifier

Table logic.1: a truth table for logical connectives. The first two columns are the truth values of propositions p and q; the rest are outputs.

		not	and	or	nand	nor	xor	xnor
p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \uparrow q$	$p \downarrow q$	$p \veebar q$	$p \Leftrightarrow q$
$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\bot$
$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$

sets.exe Exercises for Chapter sets

Exercise sets.hardhat

For the following, write the set described in set-builder notation.

a. $A = \{2, 3, 5, 9, 17, 33, \cdots\}$.
b. B is the set of integers divisible by 11.
c. $C = \{1/3, 1/4, 1/5, \cdots\}$.
d. D is the set of reals between -3 and 42.

Exercise sets.2

Let $x, y \in \mathbb{R}^n$. Prove the Cauchy-Schwarz Inequality

$$|\boldsymbol{x} \cdot \boldsymbol{y}| \leqslant \|\boldsymbol{x}\| \|\boldsymbol{y}\|. \tag{1}$$

Hint: you may find the geometric definition of the dot product helpful.

Exercise sets.3

Let $x \in \mathbb{R}^n$. Prove that

$$\boldsymbol{x} \cdot \boldsymbol{x} = \|\boldsymbol{x}\|^2. \tag{2}$$

Hint: you may find the geometric definition of the dot product helpful.

Exercise sets.4

Let $x, y \in \mathbb{R}^n$. Prove the Triangle Inequality

$$\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|. \tag{3}$$

Hint: you may find the Cauchy-Schwarz Inequality helpful.

prob

Probability

prob.meas Probability and measurement

Probability theory

Probability theory is a well-defined branch of mathematics. Andrey Kolmogorov described a set of axioms in 1933 that are still in use today as the foundation of probability theory.[1]

interpretation of probability

event

We will implicitly use these axioms in our analysis. The interpretation of probability is a contentious matter. Some believe probability quantifies the frequency of the occurrence of some event that is repeated in a large number of trials. Others believe it quantifies the state of our knowledge or belief that some event will occur. In experiments, our measurements are tightly coupled to probability. This is apparent in the questions we ask. Here are some examples.

1. How common is a given event?
2. What is the probability we will reject a good theory based on experimental results?
3. How repeatable are the results?
4. How confident are we in the results?
5. What is the character of the fluctuations and drift in the data?
6. How much data do we need?

[1] For a good introduction to probability theory, see Ash (Robert B. Ash. Basic Probability Theory. Dover Publications, Inc., 2008) or Jaynes andothers. (E.T. Jaynes andothers. Probability Theory: The Logic of Science. Cambridge University Press, 2003. ISBN: 9780521592710. An excellent and comprehensive introduction to probability theory.)

prob.prob Basic probability theory

The mathematical model for a class of measurements is called the probability space and is composed of a mathematical triple of a sample space Ω, σ-algebra $\mathcal{F}$, and probability measure P, typically denoted $(\Omega, \mathcal{F}, P)$, each of which we will consider in turn (Wikipedia. Probability space — Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/w/index.php?title=Probability%20space&oldid=914939789`. [Online; accessed 31-October-2019]. 2019).

probability space

The sample space Ω of an experiment is the set representing all possible outcomes of the experiment. If a coin is flipped, the sample space is $\Omega = \{H, T\}$, where H is heads and T is tails. If a coin is flipped twice, the sample space could be

sample space

outcomes

However, the same experiment can have different sample spaces. For instance, for two coin flips, we could also choose

We base our choice of Ω on the problem at hand. An event is a subset of the sample space. That is, an event corresponds to a yes-or-no question about the experiment. For instance, event A (remember: $A \subseteq \Omega$) in the coin flipping experiment (two flips) might be $A = \{HT, TH\}$. A is an event that corresponds to the question, "Is the second flip different than the first?" A is the event for which the answer is "yes."

event

Algebra of events

Because events are sets, we can perform the usual set operations with them.

Example prob.prob–1 **re: set operations with events**

Consider a toss of a single die. We choose the sample space to be $\Omega = \{1, 2, 3, 4, 5, 6\}$. Let the following define events.

$$A \equiv \{\text{the result is even}\} = \{2, 4, 6\}$$
$$B \equiv \{\text{the result is greater than 2}\} = \{3, 4, 5, 6\}.$$

Find the following event combinations:

$$A \cup B \qquad A \cap B \qquad A \setminus B \qquad B \setminus A \qquad \overline{A} \setminus B.$$

The σ-algebra $\mathcal{F}$ is the collection of events of interest. Often, $\mathcal{F}$ is the set of all possible events given a sample space Ω, which is just the power set of Ω (Wikipedia, Probability space — Wikipedia, The Free Encyclopedia). When referring to an event, we often state that it is an element of $\mathcal{F}$. For instance, we might say an event $A \in \mathcal{F}$.

σ–algebra

We're finally ready to assign probabilities to events. We define the probability measure $P : \mathcal{F} \to [0, 1]$ to be a function satisfying the following conditions.

probability measure

1. For every event $A \in \mathcal{F}$, the probability measure of A is greater than or equal to zero—i.e. $P(A) \geqslant 0$.
2. If an event is the entire sample space, its probability measure is unity—i.e. if $A = \Omega$, $P(A) = 1$.

3. If events $A_1, A_2, \cdots$ are disjoint sets (no elements in common), then $P(A_1 \cup A_2 \cup \cdots) = P(A_1) + P(A_2) + \cdots$.

We conclude the basics by observing four facts that can be proven from the definitions above.

1.
2.
3.

4.

prob.condition Independence and conditional probability

Two events A and B are independent if and only if

independent

$$P(A \cap B) = P(A)P(B).$$

If an experimenter must make a judgment without data about the independence of events, they base it on their knowledge of the events, as discussed in the following example.

Example prob.condition−1

re: independence

Answer the following questions and imperatives.

1. Consider a single fair die rolled twice. What is the probability that both rolls are 6?
2. What changes if the die is biased by a weight such that $P(\{6\}) = 1/7$?
3. What changes if the die is biased by a magnet, rolled on a magnetic dice-rolling tray such that $P(\{6\}) = 1/7$?
4. What changes if there are two dice, biased by weights such that for each $P(\{6\}) = 1/7$, rolled once, both resulting in 6?
5. What changes if there are two dice, biased by magnets, rolled together?

Conditional probability

If events A and B are somehow dependent, we need a way to compute the probability of B occurring given that A occurs. This is called the conditional probability of B given A, and is denoted $P(B \mid A)$. For $P(A) > 0$, it is defined as

dependent

conditional probability

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}. \tag{1}$$

We can interpret this as a restriction of the sample space Ω to A; i.e. the new sample space $\Omega' = A \subseteq \Omega$. Note that if A and B are independent, we obtain the obvious result:

Example prob.condition–2

re: dependence

Consider two unbiased dice rolled once. Let events $A = \{\text{sum of faces} = 8\}$ and $B = \{\text{faces are equal}\}$. What is the probability the faces are equal given that their sum is 8?

prob.bay Bayes' theorem

Bayes' theorem

Given two events A and B, Bayes' theorem (aka Bayes' rule) states that

$$P(A \mid B) = P(B \mid A)\frac{P(A)}{P(B)}. \tag{1}$$

Sometimes this is written

$$\begin{aligned} P(A \mid B) &= \frac{P(B \mid A)P(A)}{P(B \mid A)P(A) + P(B \mid \neg A)P(\neg A)} \qquad (2) \\ &= \frac{1}{1 + \frac{P(B \mid \neg A)}{P(B \mid A)} \cdot \frac{P(\neg A)}{P(A)}}. \qquad (3) \end{aligned}$$

This is a useful theorem for determining a test's effectiveness. If a test is performed to determine whether an event has occurred, we might as questions like "if the test indicates that the event has occurred, what is the probability it has actually occurred?" Bayes' theorem can help compute an answer.

Testing outcomes

The test can be either positive 🟥 or negative 🟦, meaning it can either indicate or not indicate that A has occurred. Furthermore, this result can be either true ☺ or false ☹.
There are four options, then. Consider an event A and an event that is a test result B indicating that event A has occurred. Table bay.1 shows these four possible test outcomes. The event A occurring can lead to a true positive or a false negative, whereas $\neg A$ can lead to a true negative or a false positive.
Terminology is important, here.

		A	$\neg A$
positive (B)	🟥	true ☺ 🟥	false ☹ 🟥
negative ($\neg B$)	🟦	false ☹ 🟦	true ☺ 🟦

Table bay.1: test outcome B for event A.

- $P(\{\text{true positive}\}) = P(B \mid A)$, aka sensitivity or detection rate,

sensitivity

detection rate

- $P(\{\text{true negative}\}) = P(\neg B \mid \neg A)$, aka specificity,
- $P(\{\text{false positive}\}) = P(B \mid \neg A)$,
- $P(\{\text{false negative}\}) = P(\neg B \mid A)$.

specificity

Clearly, the desirable result for any test is that it is true. However, no test is true 100 percent of the time. So sometimes it is desirable to err on the side of the false positive, as in the case of a medical diagnostic. Other times, it is more desirable to err on the side of a false negative, as in the case of testing for defects in manufactured balloons (when a false negative isn't a big deal).

Posterior probabilities

Returning to Bayes' theorem, we can evaluate the posterior probability $P(A \mid B)$ of the event A having occurred given that the test B is positive, given information that includes the prior probability $P(A)$ of A. The form in Eq. 2 or (3) is typically useful because it uses commonly known test probabilities: of the true positive $P(B \mid A)$ and of the false positive $P(B \mid \neg A)$. We calculate $P(A \mid B)$ when we want to interpret test results.

posterior probability

prior probability

Some interesting results can be found from this. For instance, if we let $P(B \mid A) = P(\neg B \mid \neg A)$ (sensitivity equal specificity) and realize that $P(B \mid \neg A) + P(\neg B \mid \neg A) = 1$ (when $\neg A$, either B or $\neg B$), we can derive the expression

$$P(B \mid \neg A) = 1 - P(B \mid A). \tag{4}$$

Using this and $P(\neg A) = 1 - P(A)$ in Eq. 3 gives (recall we've assumed sensitivity equals specificity!)

$$\begin{aligned} P(A \mid B) &= \frac{1}{1 + \dfrac{1 - P(B \mid A)}{P(B \mid A)} \cdot \dfrac{1 - P(A)}{P(A)}} && (5) \\ &= \frac{1}{1 + \left(\dfrac{1}{P(B \mid A)} - 1\right)\left(\dfrac{1}{P(A)} - 1\right)} && (6) \end{aligned}$$

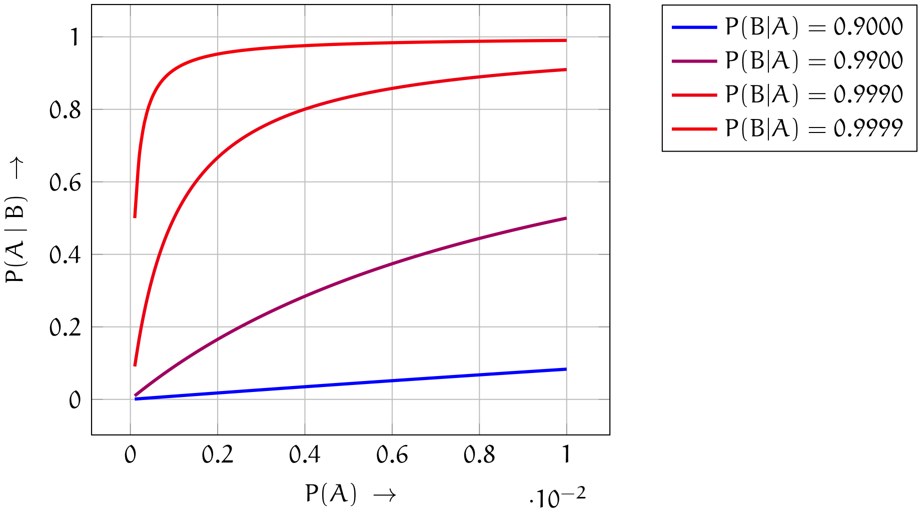


Figure bay.1: for different high-sensitivities, the probability that an event A occurred given that a test for it B is positive versus the probability that the event A occurs, under the assumption the specificity equals the sensitivity.

This expression is plotted in Fig. bay.1. See that a positive result for a rare event (small $P(A)$) is hard to trust unless the sensitivity $P(B \mid A)$ and specificity $P(\neg B \mid \neg A)$ are very high, indeed!

Example prob.bay–1 re: Bayes' theorem

Suppose 0.1% of springs manufactured at a given plant are defective. Suppose you need to design a test that, when it indicates a deffective part, the part is actually defective 99% of the time. What sensitivity should your test have assuming it can be made equal to its specificity?

The following was generated from a Jupyter notebook with the following filename and kernel.

```
notebook filename: bayes_theorem_example_01.ipynb
notebook kernel: python3
```

```
from sympy import * # for symbolics
import numpy as np # for numerics
import matplotlib.pyplot as plt # for plots!
from IPython.display import display, Markdown, Latex
```

Define symbolic variables.

```
var('p_A,p_nA,p_B,p_nB,p_B_A,p_B_nA,p_A_B',real=True)
```

```
(p_A, p_nA, p_B, p_nB, p_B_A, p_B_nA, p_A_B)
```

Beginning with Bayes’ theorem and assuming the sensitivity and specificity are equal by Eq. 4, we can derive the following expression for the posterior probability $P(A \mid B)$.

```
p_A_B_e1 = Eq(p_A_B,p_B_A*p_A/p_B).subs(
  {
    p_B: p_B_A*p_A+p_B_nA*p_nA, # conditional prob
    p_B_nA: 1-p_B_A, # Eq (3.5)
    p_nA: 1-p_A
  }
)
display(p_A_B_e1)
```

$$p_{AB} = \frac{p_A p_{BA}}{p_A p_{BA} + (1 - p_A)(1 - p_{BA})}$$

Solve this for $P(B \mid A)$, the quantity we seek.

```
p_B_A_sol = solve(p_A_B_e1,p_B_A,dict=True)
p_B_A_eq1 = Eq(p_B_A,p_B_A_sol[0][p_B_A])
display(p_B_A_eq1)
```

$$p_{BA} = \frac{p_{AB}(1 - p_A)}{-2p_A p_{AB} + p_A + p_{AB}}$$

Now let’s substitute the given probabilities.

```
p_B_A_spec = p_B_A_eq1.subs(
  {
    p_A: 0.001,
    p_A_B: 0.99,
  }
)
display(p_B_A_spec)
```

$$p_{BA} = 0.999989888981011$$

That’s a tall order!

prob.rando Random variables

Probabilities are useful even when they do not deal strictly with events. It often occurs that we measure something that has randomness associated with it. We use random variables to represent these measurements.

random variable

A random variable $X : \Omega \to \mathbb{R}$ is a function that maps an outcome ω from the sample space Ω to a real number $x \in \mathbb{R}$, as shown in Fig. rando.1. A random variable will be denoted with a capital letter (e.g. X and K) and a specific value that it maps to (the value) will be denoted with a lowercase letter (e.g. x and k).

discrete random variable

continuous random variable

A discrete random variable K is one that takes on discrete values. A continuous random variable X is one that takes on continuous values.

Example prob.rando−1 re: dice again

Roll two unbiased dice. Let K be a random variable representing the sum of the two. Let $P(k)$ be the probability of the result $k \in K$. Plot and interpret $P(k)$.

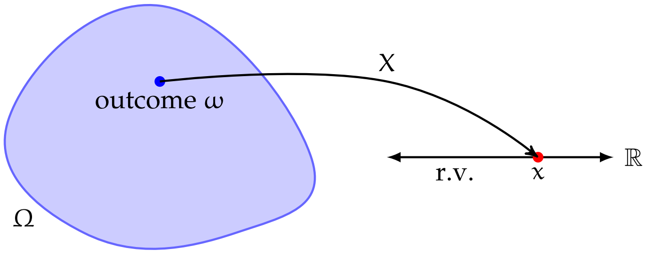


Figure rando.1: a random variable X maps an outcome $\omega \in \Omega$ to an $x \in \mathbb{R}$.

Example prob.rando−2 re: Johnson−Nyquist noise

A resistor at nonzero temperature without any applied voltage exhibits an interesting phenomenon: its voltage randomly fluctuates. This is called Johnson-Nyquist noise and is a result of thermal excitation of charge carriers (electrons, typically). For a given resistor and measurement system, let the probability density function f_V of the voltage V across an unrealistically hot resistor be

$$f_V(V) = \frac{1}{\sqrt{\pi}} e^{-V^2}.$$

Plot and interpret the meaning of this function.

prob.pxf Probability density and mass functions

Consider an experiment that measures a random variable. We can plot the relative frequency of the measurand landing in different "bins" (ranges of values). This is called a frequency distribution or a probability mass function (PMF).

frequency distribution

probability mass function

Consider, for instance, a probability mass function as plotted in Fig. pxf.1, where a frequency a_i can be interpreted as an estimate of the probability of the measurand being in the ith interval. The sum of the frequencies must be unity:

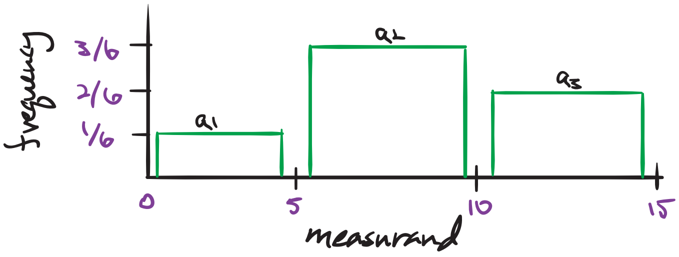


Figure pxf.1: plot of a probability mass function.

with k being the number of bins.

The frequency density distribution is similar to the frequency distribution, but with $a_i \mapsto a_i/\Delta x$, where Δx is the bin width.

frequency density distribution

If we let the bin width approach zero, we derive the probability density function (PDF)

probability density function

$$f(x) = \lim_{\substack{k\to\infty \\ \Delta x \to 0}} \sum_{j=1}^{k} a_j/\Delta x. \qquad (1)$$

We typically think of a probability density function f, like the one in Fig. pxf.2 as a function that can be integrated over to find the probability of the random variable (measurand) being in an interval $[a, b]$:

$$P(x \in [a, b]) = \int_a^b f(\chi) d\chi. \qquad (2)$$

Of course,

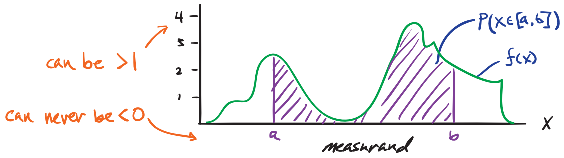


Figure pxf.2: plot of a probability density function.

We now consider a common PMF and a common PDF.

Binomial PMF

Consider a random binary sequence of length n such that each element is a random 0 or 1, generated independently, like

$$(1, 0, 1, 1, 0, \cdots, 1, 1). \tag{3}$$

Let events $\{1\}$ and $\{0\}$ be mutually exclusive and exhaustive and $P(\{1\}) = p$. The probability of the sequence above occurring is

There are n choose k,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, \tag{4}$$

possible combinations of k ones for n bits. Therefore, the probability of any combination of k ones in a series is

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}. \tag{5}$$

We call Eq. 5 the binomial distribution PDF.

binomial distribution PDF

Example prob.pxf−1

re: Binomial PMF

Consider a field sensor that fails for a given measurement with probability p. Given n measurements, plot the binomial PMF as a function of k failed measurements for a few different probabilities of failure $p \in [0.04, 0.25, 0.5, 0.75, 0.96]$.

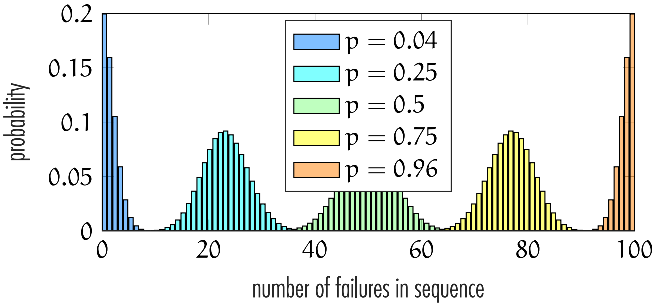


Figure pxf.3: binomial PDF for $n = 100$ measurements and different values of $P(\{1\}) = p$, the probability of a measurement error. The plot is generated by the Matlab code of Fig. pxf.4.

Fig. pxf.4 shows Matlab code for constructing the PDFs plotted in Fig. pxf.3. Note that the symmetry is due to the fact that events $\{1\}$ and $\{0\}$ are mutually exclusive and exhaustive.

Example prob.pxf−2 re: hi

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

```
%% parameters
n = 100;
k_a = linspace(1,n,n);
p_a = [.04,.25,.5,.75,.96];

%% binomial function
f = @(n,k,p) nchoosek(n,k)*p^k*(1-p)^(n-k);

% loop through to construct an array
f_a = NaN*ones(length(k_a),length(p_a));
for i = 1:length(k_a)
    for j = 1:length(p_a)
        f_a(i,j) = f(n,k_a(i),p_a(j));
    end
end

%% plot
figure
colors = jet(length(p_a));
for j = 1:length(p_a)
    bar(...
        k_a,f_a(:,j),...
        'facecolor',colors(j,:),...
        'facealpha',0.5,...
        'displayname', ['$p = ',num2str(p_a(j)),'$']...
    ); hold on
end
leg = legend('show','location','north');
set(leg,'interpreter','latex')
hold off
xlabel('number of ones in sequence k')
ylabel('probability')
xlim([0,100])
```

Figure pxf.4: a Matlab script for generating binomial PMFs.

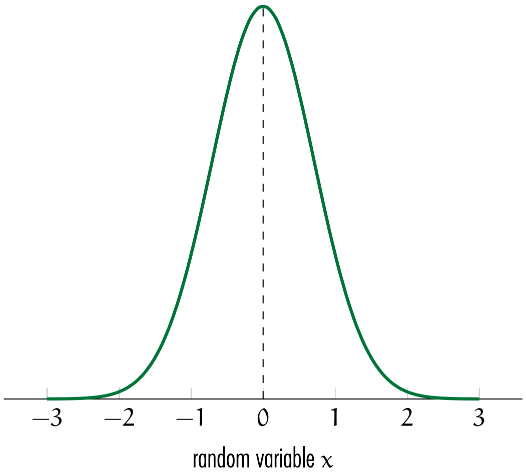


Figure pxf.5: PDF for Gaussian random variable x, mean $\mu = 0$, and standard deviation $\sigma = 1/\sqrt{2}$.

Gaussian PDF

The Gaussian or normal random variable x has PDF

Gaussian or normal random variable

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(x-\mu)^2}{2\sigma^2}. \qquad (6)$$

Although we're not quite ready to understand these quantities in detail, it can be shown that the parameters μ and σ have the following meanings:

- μ is the mean of x,
- σ is the standard deviation of x, and
- σ^2 is the variance of x.

mean

standard deviation

variance

Consider the "bell-shaped" Gaussian PDF in Fig. pxf.5. It is always symmetric. The mean μ is its central value and the standard deviation σ is directly related to its width. We will continue to explore the Gaussian distribution in the following lectures, especially in Lec. stats.confidence.

prob.E Expectation

Recall that a random variable is a function $X : \Omega \to \mathbb{R}$ that maps from the sample space to the reals. Random variables are the arguments of probability mass functions (PMFs) and probability density functions (PDFs).

expected value
expectation

The expected value (or expectation) of a random variable is akin to its "average value" and depends on its PMF or PDF. The expected value of a random variable X is denoted $\langle X \rangle$ or $\mathrm{E}[X]$. There are two definitions of the expectation, one for a discrete random variable, the other for a continuous random variable. Before we define, them, however, it is useful to predefine the most fundamental property of a random variable, its mean.

mean

Definition prob.1: mean

The mean of a random variable X is defined as

$$m_X = \mathrm{E}[X] \, .$$

Let's begin with a discrete random variable.

Definition prob.2: expectation of a discrete random variable

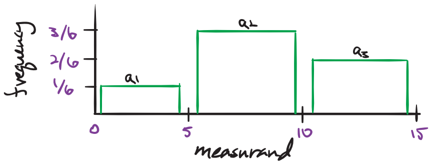
Let K be a discrete random variable and f its PMF. The expected value of K is defined as

$$\mathrm{E}[K] = \sum_{\forall k} k f(k).$$

Example prob.E–1

re: expectation of a discrete random variable

Given a discrete random variable K with PMF shown below, what is its mean m_K?



Let us now turn to the expectation of a continuous random variable.

Definition prob.3: expectation of a continuous random variable

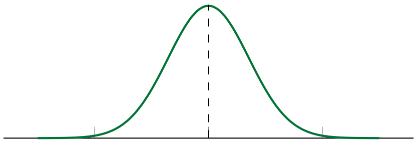
Let X be a continuous random variable and f its PDF. The expected value of X is defined as

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx.$$

Example prob.E–2

Given a continuous random variable X with Gaussian PDF f, what is the expected value of X?

re: expectation of a continuous random variable



random variable x

Due to its sum or integral form, the expected value $\mathrm{E}\left[\cdot\right]$ has some familiar properties for random variables X and Y and reals a and b.

$$\begin{aligned}
\mathrm{E}\left[a\right] &= a && \text{(1a)}\\
\mathrm{E}\left[X+a\right] &= \mathrm{E}\left[X\right] + a && \text{(1b)}\\
\mathrm{E}\left[aX\right] &= a\,\mathrm{E}\left[X\right] && \text{(1c)}\\
\mathrm{E}\left[\mathrm{E}\left[X\right]\right] &= \mathrm{E}\left[X\right] && \text{(1d)}\\
\mathrm{E}\left[aX+bY\right] &= a\,\mathrm{E}\left[X\right] + b\,\mathrm{E}\left[Y\right]. && \text{(1e)}
\end{aligned}$$

prob.moments Central moments

Given a probability mass function (PMF) or probability density function (PDF) of a random variable, several useful parameters of the random variable can be computed. These are called central moments, which quantify parameters relative to its mean.

central moments

Definition prob.4: central moments

The nth central moment of random variable X, with PDF f, is defined as

$$E\left[(X-\mu_X)^n\right] = \int_{-\infty}^{\infty} (x-\mu_X)^n f(x)\,dx.$$

For discrete random variable K with PMF f,

$$E\left[(K-\mu_K)^n\right] = \sum_{\forall k}^{\infty} (k-\mu_K)^n f(k).$$

Example prob.moments–1

re: first moment

Prove that the first moment of continuous random variable X is zero.

The second central moment of random variable X is called the variance and is denoted

variance

$$\sigma_X^2 \quad \text{or} \quad \text{Var}\left[X\right] \quad \text{or} \quad E\left[(X-\mu_X)^2\right]. \qquad (1)$$

The variance is a measure of the width or spread of the PMF or PDF. We usually compute the variance with the formula

Other properties of variance include, for real constant c,

$$\mathrm{Var}\left[c\right] = 0 \tag{2}$$

$$\mathrm{Var}\left[X + c\right] = \mathrm{Var}\left[X\right] \tag{3}$$

$$\mathrm{Var}\left[cX\right] = c^2\,\mathrm{Var}\left[X\right]. \tag{4}$$

standard deviation

The standard deviation is defined as

Although the variance is mathematically more convenient, the standard deviation has the same physical units as X, so it is often the more physically meaningful quantity. Due to its meaning as the width or spread of the probability distribution, and its sharing of physical units, it is a convenient choice for error bars on plots of a random variable.

skewness

The skewness $\mathrm{Skew}\left[X\right]$ is a normalized third central moment:

$$\mathrm{Skew}\left[X\right] = \frac{\mathrm{E}\left[(X - \mu_X)^3\right]}{\sigma_X^3}. \tag{5}$$

asymmetry

Skewness is a measure of asymmetry of a random variable's PDF or PMF. For a symmetric PMF or PDF, such as the Gaussian PDF, $\mathrm{Skew}\left[X\right] = 0$.

kurtosis

The kurtosis $\mathrm{Kurt}\left[X\right]$ is a normalized fourth central moment:

$$\mathrm{Kurt}\left[X\right] = \frac{\mathrm{E}\left[(X - \mu_X)^4\right]}{\sigma_X^4}. \tag{6}$$

tailedness

Kurtosis is a measure of the tailedness of a random variable's PDF or PMF. "Heavier" tails yield higher kurtosis.

A Gaussian random variable has PDF with kurtosis 3. Given that for Gaussians both skewness and kurtosis have nice values (0 and 3), we can think of skewness and and kurtosis as measures of similarity to the Gaussian PDF.

prob.exe Exercises for Chapter prob

Exercise prob.5

Several physical processes can be modeled with a random walk: a process of interatively changing a quantity by some random amount. Infinitely many variations are possible, but common factors of variation include probability distribution, step size, dimensionality (e.g. one-dimensional, two-dimensional, etc.), and coordinate system. Graphical representations of these walks can be beautiful. Develop a computer program that generates random walks and corresponding graphics. Do it well and call it art because it is.

stats

Statistics

Whereas probability theory is primarily focused on the relations among mathematical objects, statistics is concerned with making sense of the outcomes of observation (Steven S. Skiena. Calculated Bets: Computers, Gambling, and Mathematical Modeling to Win. Outlooks. Cambridge University Press, 2001. DOI: 10.1017/CBO9780511547089. This includes a lucid section on probability versus statistics, also available here: https://www3.cs.stonybrook.edu/~skiena/jaialai/excerpts/node12.html.). However, we frequently use statistical methods to estimate probabilistic models. For instance, we will learn how to estimate the standard deviation of a random process we have some reason to expect has a Gaussian probability distribution.

estimation

Statistics has applications in nearly every applied science and engineering discipline. Any time measurements are made, statistical analysis is how one makes sense of the results. For instance, determining a reasonable level of confidence in a measured parameter requires statistics.

A particularly hot topic nowadays is machine learning, which seems to be a field with applications that continue to expand. This field is fundamentally built on statistics.

machine learning

A good introduction to statistics appears at the end of Ash.[1] A more involved introduction is given by Jaynes andothers.[2] The treatment by

[1] Ash, Basic Probability Theory.

[2] Jaynes andothers, Probability Theory: The Logic of Science.

Kreyszig[3] is rather incomplete, as will be our own.

[3] Erwin Kreyszig. Advanced Engineering Mathematics. 10th. John Wiley & Sons, Limited, 2011. ISBN: 9781119571094. The authoritative resource for engineering mathematics. It includes detailed accounts of probability, statistics, vector calculus, linear algebra, fourier analysis, ordinary and partial differential equations, and complex analysis. It also includes several other topics with varying degrees of depth. Overall, it is the best place to start when seeking mathematical guidance.

stats.terms Populations, samples, and machine learning

An experiment's population is a complete collection of objects that we would like to study. These objects can be people, machines, processes, or anything else we would like to understand experimentally.

population

Of course, we typically can't measure all of the population. Instead, we take a subset of the population—called a sample—and infer the characteristics of the entire population from this sample.

sample

However, this inference that the sample is somehow representative of the population assumes the sample size is sufficiently large and that the sampling is random. This means selection of the sample should be such that no one group within a population are systematically over- or under-represented in the sample.

random

machine learning

Machine learning is a field that makes extensive use of measurements and statistical inference. In it, an algorithm is trained by exposure to sample data, which is called a training set. The variables measured are called features. Typically, a predictive model is developed that can be used to extrapolate from the data to a new situation. The methods of statistical analysis we introduce in this chapter are the foundation of most machine learning methods.

training

training set

features

predictive model

Example stats.terms–1

re: combat boots

Consider a robot, Pierre, with a particular gravitas and sense of style. He seeks just the right-looking pair of combat boots for wearing in the autumn rains. Pierre is to purchase the boots online via image recognition, and decides to gather data by visiting a hipster hangout one evening to train his style. For contrast, he also watches footage of a White Nationalist rally, focusing special attention on the boots of wearers of khakis and polos. Comment on Pierre's methods.

stats.sample Estimation of sample mean and variance

Estimation and sample statistics

The mean and variance definitions of Lec. prob.E and Lec. prob.moments apply only to a random variable for which we have a theoretical probability distribution. Typically, it is not until after having performed many measurements of a random variable that we can assign a good distribution model. Until then, measurements can help us estimate aspects of the data. We usually start by estimating basic parameters such as mean and variance before estimating a probability distribution.

There are two key aspects to randomness in the measurement of a random variable. First, of course, there is the underlying randomness with its probability distribution, mean, standard deviation, etc., which we call the population statistics. Second, there is the statistical variability that is due to the fact that we are estimating the random variable's statistics—called its sample statistics—from some sample. Statistical variability is decreased with greater sample size and number of samples, whereas the underlying randomness of the random variable does not decrease. Instead, our estimates of its probability distribution and statistics improve.

Sample mean, variance, and standard deviation

sample mean

The arithmetic mean or sample mean of a measurand with sample size N, represented by random variable X, is defined as

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{1}$$

population mean

If the sample size is large, $\overline{x} \to m_X$ (the sample mean approaches the mean). The population mean is another name for the mean m_X, which is equal to

$$m_X = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{2}$$

Recall that the definition of the mean is $m_X = E[x]$.

The sample variance of a measurand represented by random variable X is defined as

sample variance

$$S_X^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2. \tag{3}$$

If the sample size is large, $S_X^2 \to \sigma_X^2$ (the sample variance approaches the variance). The population variance is another term for the variance σ_X^2, and can be expressed as

population variance

$$\sigma_X^2 = \lim_{N\to\infty} \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2. \tag{4}$$

Recall that the definition of the variance is $\sigma_X^2 = E\left[(X - m_X)^2\right]$.

The sample standard deviation of a measurand represented by random variable X is defined as

$$S_X = \sqrt{S_X^2}. \tag{5}$$

If the sample size is large, $S_X \to \sigma_X$ (the sample standard deviation approaches the standard deviation). The population standard deviation is another term for the standard deviation σ_X, and can be expressed as

$$\sigma_X = \lim_{N\to\infty} \sqrt{S_X^2}. \tag{6}$$

Recall that the definition of the standard deviation is $\sigma_X = \sqrt{\sigma_X^2}$.

Sample statistics as random variables

There is an ambiguity in our usage of the term “sample.” It can mean just one measurement or it can mean a collection of measurements gathered together. Hopefully, it is clear from context.

In the latter sense, often we collect multiple samples, each of which has its own sample mean $\overline{X}_i$ and standard deviation S_{X_i}. In this situation, $\overline{X}_i$ and S_{X_i} are themselves random variables (meta af, I know). This means they have their own sample means $\overline{\overline{X}}_i$ and $\overline{S_{X_i}}$ and standard deviations $S_{\overline{X}_i}$ and $S_{S_{X_i}}$.

The mean of means $\overline{\overline{X}}_i$ is equivalent to a mean with a larger sample size and is therefore our best estimate of the mean of the underlying random process. The mean of standard deviations $\overline{S_{X_i}}$ is our best estimate of the standard deviation of the underlying random process. The standard deviation of means $S_{\overline{X}_i}$ is a measure of the spread in our estimates of the mean. It is our best estimate of the standard deviation of the statistical variation and should therefore tend to zero as sample size and number of samples increases. The standard deviation of standard deviations $S_{S_{X_i}}$ is a measure of the spread in our estimates of the standard deviation of the underlying process. It should also tend to zero as sample size and number of samples increases.

mean of means $\overline{\overline{X}}_i$

mean of standard deviations $\overline{S_{X_i}}$

standard deviation of means $S_{\overline{X}_i}$

standard deviation of standard deviations $S_{S_{X_i}}$

Let N be the size of each sample. It can be shown that the standard deviation of the means $S_{\overline{X}_i}$ can be estimated from a single sample standard deviation:

$$S_{\overline{X}_i} \approx \frac{S_{X_i}}{\sqrt{N}}. \tag{7}$$

This shows that as the sample size N increases, the statistical variability of the mean decreases (and in the limit approaches zero).

Nonstationary signal statistics

The sample mean, variance, and standard deviation definitions, above, assume the random process is stationary—that is, its population mean does not vary with time. However, a great many measurement signals have populations that do vary with time, i.e. they are nonstationary. Sometimes the nonstationarity arises from a "drift" in the dc value of a signal or some other slowly changing variable. But dynamic signals can also change in a recognizable and predictable manner, as when, say, the temperature of a room changes when a window is opened or when a water level changes with the tide.

Typically, we would like to minimize the effect of nonstationarity on the signal statistics. In certain cases, such as drift, the variation is a nuissance only, but other times it is the point of the measurement.

Two common techniques are used, depending on the overall type of nonstationarity. If it is periodic with some known or estimated period, the measurement data series can be "folded" or "reshaped" such that the ith measurement of each period corresponds to the ith measurement of all other periods. In this case, somewhat counterintuitively, we can consider the ith measurements to correspond to a sample of size N, where N is the number of periods over which measurements are made.

When the signal is aperiodic, we often simply divide it into "small" (relative to its overall trend) intervals over which statistics are computed, separately.

Note that in this discussion, we have assumed that the nonstationarity of the signal is due to a variable that is deterministic (not random).

Example stats.sample−1 re: measurement of gaussian noise on nonstationary signal

Consider the measurement of the temperature inside a desktop computer chassis via an inexpensive thermistor, a resistor that changes resistance with temperature. The processor and power supply heat the chassis in a manner that depends on processing demand. For the test protocol, the processors are cycled sinusoidally through processing power levels at a frequency of 50 mHz for $n_T = 12$ periods and sampled at 1 Hz. Assume a temperature fluctuation between about 20 and 50 C and gaussian noise with standard deviation 4 C. Consider a sample to be the multiple measurements of a certain instant in the period.

1. Generate and plot simulated temperature data as a time series and as a histogram or frequency distribution. Comment on why the frequency distribution sucks.
2. Compute the sample mean and standard deviation for each sample in the cycle.
3. Subtract the mean from each sample in the period such that each sample distribution is centered at zero. Plot the composite frequency distribution of all samples, together. This represents our best estimate of the frequency distribution of the underlying process.
4. Plot a comparison of the theoretical mean, which is 35, and the sample mean of means with an error bar. Vary the number of samples n_T and comment on its effect on the estimate.
5. Plot a comparison of the theoretical standard deviation and the sample mean of sample standard deviations with an error bar. Vary the number of samples n_T and comment on its effect on the estimate.
6. Plot the sample means over a single period with error bars of ± one sample standard deviation of the means. This represents our best estimate of the sinusoidal heating temperature. Vary the number of samples n_T and comment on

the estimate.

```
clear; close all; % clear kernel
```

Generate the temperature data

The temperature data can be generated by constructing an array that is passed to a sinusoid, then “randomized” by gaussian random numbers. Note that we add `1` to `np` and `n` to avoid the sneaky fencepost error.

```
f = 50e-3; % Hz ... sinusoid frequency
a = 15; % C ... amplitude of oscillation
dc = 35; % C ... dc offset of oscillation
fs = 1; % Hz ... sampling frequency
nT = 12; % number of sinusoid periods
s = 4; % C ... standard deviation
np = fs/f+1; % number of samples per period
n = nT*np+1; % total number of samples

t_a = linspace(0,nT/f,n); % time array
sin_a = dc + a*sin(2*pi*f*t_a); % sinusoidal array
rng(43); % seed the random number generator
noise_a = s*randn(size(t_a)); % gaussian noise
signal_a = sin_a + noise_a; % sinusoid + noise
```

Now that we have an array of “data,” we’re ready to plot.

```
h = figure;
p = plot(t_a,signal_a,'o-',...
    'Color',[.8,.8,.8],...
    'MarkerFaceColor','b',...
    'MarkerEdgeColor','none',...
    'MarkerSize',3);
xlabel('time (s)');
ylabel('temperature (C)');
hgsave(h,'figures/temp');
```

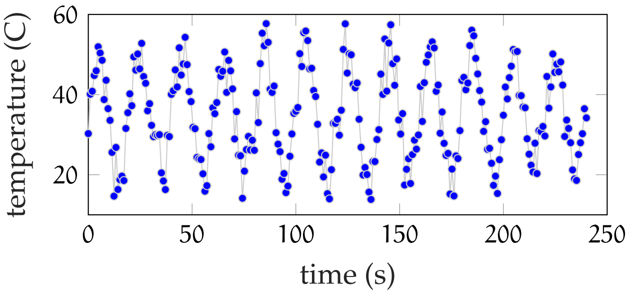


Figure sample.1: temperature over time

This is something like what we might see for continuous measurement data. Now, the histogram.

```
h = figure;
histogram(signal_a,...
    30, ... % number of bins
    'normalization','probability'... % for PMF
);
xlabel('temperature (C)')
ylabel('probability')
hgsave(h,'figures/temp');
```

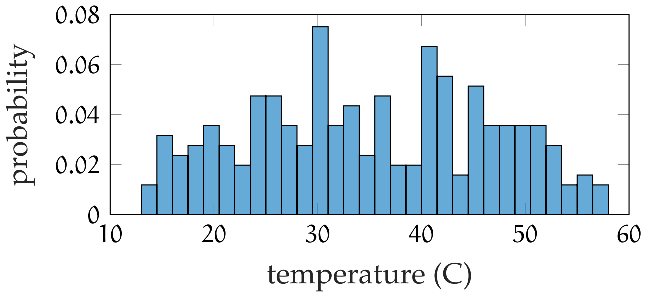


Figure sample.2: a poor histogram due to unstationarity of the signal.

This sucks because we plot a frequency distribution to tell us about the random variation, but this data includes the sinusoid.

Sample mean, variance, and standard deviation

To compute the sample mean μ and standard deviation s for each sample in the period, we must "pick out" the `nT` data points that correspond to each other. Currently, they're in one long `1` × `n` array `signal_a`. It is helpful to reshape the data so it is in an `nT` × `np` array, which each row corresponding to a new period. This leaves the correct points aligned in columns. It is important to note that we can do this "folding" operation only when we know rather precisely the period of the underlying sinusoid. It is given in the problem that it is a controlled experiment variable. If we did not know it, we would have to estimate it, too, from the data.

```
signal_ar = reshape(signal_a(1:end-1)',[np,nT])';
size(signal_ar) % check size
signal_ar(1:3,1:4) % print three rows, four columns
```

```
ans =

    12    21


ans =

   30.2718   40.0946   40.8341   44.7662
   40.1836   37.2245   49.4076   46.1137
   40.0571   40.9718   46.1627   41.9145
```

Define the mean, variance, and standard deviation functions as “anonmymous functions.” We “roll our own.” These are not as efficient or flexible as the built-in Matlab functions `mean`, `var`, and `std`, which should typically be used.

```
my_mean = @(vec) sum(vec)/length(vec);
my_var = @(vec) sum((vec-my_mean(vec)).^2)/...
  (length(vec)-1);
my_std = @(vec) sqrt(my_var(vec));
```

Now the sample mean, variance, and standard deviations can be computed. We proceed by looping through each column of the reshaped signal array.

```
mu_a = NaN*ones(1,np); % initialize mean array
var_a = NaN*ones(1,np); % initialize var array
s_a = NaN*ones(1,np); % initialize std array

for i = 1:np % for each column
    mu_a(i) = my_mean(signal_ar(:,i));
    var_a(i) = my_var(signal_ar(:,i));
    s_a(i) = sqrt(var_a(i)); % touch of speed
end
```

Composite frequency distribution

The columns represent samples. We want to subtract the mean from each column. We use `repmat` to reproduce `mu_a` in `nT` rows so it can be easily subtracted.

```
signal_arz = signal_ar - repmat(mu_a,[nT,1]);
size(signal_arz) % check size
signal_arz(1:3,1:4) % print three rows, four columns
```

```
ans =

    12    21


ans =

   -5.0881    0.9525   -0.2909   -1.5700
    4.8237   -1.9176    8.2826   -0.2225
    4.6972    1.8297    5.0377   -4.4216
```

Now that all samples have the same mean, we can lump them into one big bin for the frequency distribution. There are some nice built-in functions to do a quick reshape and fit.

```
% resize
signal_arzr = reshape(signal_arz,[1,nT*np]);
size(signal_arzr) % check size
% fit
pdfit_model = fitdist(signal_arzr','normal'); % fit
x_a = linspace(-15,15,100);
pdfit_a = pdf(pdfit_model,x_a);
pdf_a = normpdf(x_a,0,s); % theoretical pdf
```

```
ans =

     1   252
```

Plot!

```
h = figure;
histogram(signal_arzr,...
    round(s*sqrt(nT)), ... % number of bins
    'normalization','probability'... % for PMF
);
hold on
plot(x_a,pdfit_a,'b-','linewidth',2); hold on
plot(x_a,pdf_a,'g--','linewidth',2);
```

```
legend('pmf','pdf est.','pdf')
xlabel('zero-mean temperature (C)')
ylabel('probability mass/density')
hgsave(h,'figures/temp');
```

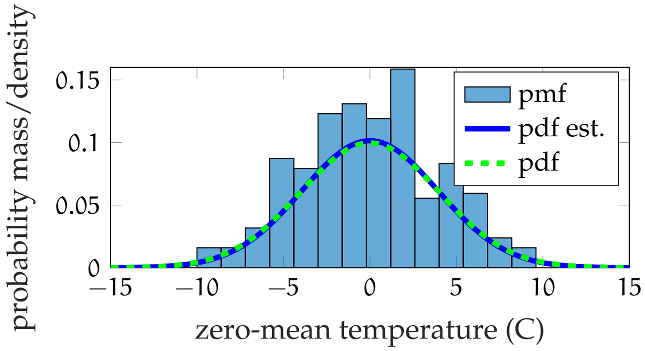


Figure sample.3: PMF and estimated and theoretical PDFs.

Means comparison

The sample mean of means is simply the following.

```
mu_mu = my_mean(mu_a)
```

```
mu_mu =

   35.1175
```

The standard deviation that works as an error bar, which should reflect how well we can estimate the point plotted, is the standard deviation of the means. It is difficult to compute this directly for a nonstationary process. We use the estimate given above and improve upon it by using the mean of standard deviations instead of a single sample's standard deviation.

```
s_mu = mean(s_a)/sqrt(nT)
```

```
s_mu =

    1.1580
```

Now, for the simple plot.

```
h = figure;
bar(mu_mu); hold on % gives bar
errorbar(mu_mu,s_mu,'r','linewidth',2) % error bar
ax = gca; % current axis
ax.XTickLabels = {'$\overline{\overline{X}}$'};
ax.TickLabelInterpreter = 'latex';
hgsave(h,'figures/temp');
```

Standard deviations comparison

The sample mean of standard deviations is simply the following.

```
mu_s = my_mean(s_a)
```

```
mu_s =

    4.0114
```

The standard deviation that works as an error bar, which should reflect how well we can estimate the point plotted, is the standard deviation of the standard deviations. We can compute this directly.

```
s_s = my_std(s_a)
```

```
s_s =

    0.8495
```

Now, for the simple plot.

```
h = figure;
bar(mu_s); hold on % gives bar
errorbar(mu_s,s_s,'r','linewidth',2) % error bars
ax = gca; % current axis
ax.XTickLabels = {'$\overline{S_X}$'};
ax.TickLabelInterpreter = 'latex';
hgsave(h,'figures/temp');
```

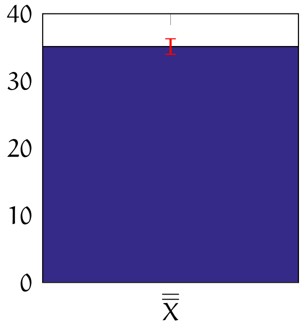


Figure sample.4: sample mean of sample means.

Plot a period with error bars

Plotting the data with error bars is fairly straightforward with the built-in `errorbar` function. The main question is "which standard deviation?" Since we're plotting the means, it makes sense to plot the error bars as a single sample standard deviation of the means.

```
h = figure;
e1 = errorbar(t_a(1:np),mu_a,s_mu*ones(1,np),'b');
hold on
t_a2 = linspace(0,1/f,101);
e2 = plot(t_a2,dc + a*sin(2*pi*f*t_a2),'r-');
xlim([t_a(1),t_a(np)])
grid on
xlabel('folded time (s)')
ylabel('temperature (C)')
legend([e1 e2],'sample mean','population mean',...
'Location','NorthEast')
hgsave(h,'figures/temp');
```

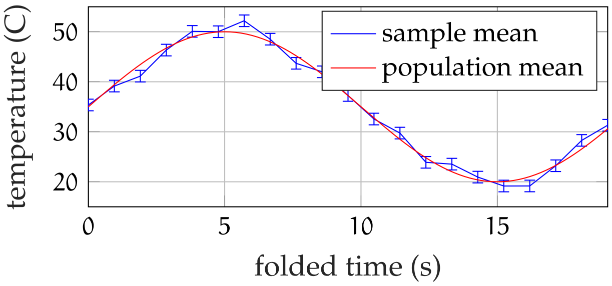


Figure sample.6: sample means over a period.

stats.confidence Confidence

One really ought to have it to give a lecture named it, but we'll give it a try anyway.

confidence

Confidence is used in the common sense, although we do endow it with a mathematical definition to scare business majors, who aren't actually impressed, but indifferent.

Approximately: if, under some reasonable assumptions (probabilistic model), we estimate the probability of some event to be $P\%$, we say we have $P\%$ confidence in it. I mean, business majors are all, "Supply and demand? Let's call that a 'law,' " so I think we're even.

So we're back to computing probability from distributions—probability density functions (PDFs) and probability mass functions (PMFs).

Usually we care most about estimating the mean of our distribution. Recall from the previous lecture that when several samples are taken, each with its own mean, the mean is itself a random variable—with a mean, of course. Meanception.

central limit theorem

But the mean has a probability distribution of its own. The central limit theorem has as one of its implications that, as the sample size N gets large, regardless of the sample distributions, this distribution of means approaches the Gaussian distribution.

But sometimes I always worry I'm being lied to, so let's check.

```
clear; close all; % clear kernel
```

Generate some data to test the central limit theorem

Data can be generated by constructing an array using a (seeded for consistency) random number generator. Let's use a uniformly distributed PDF between 0 and 1.

```
N = 150; % sample size (number of measurements per sample)
M = 120; % number of samples
```

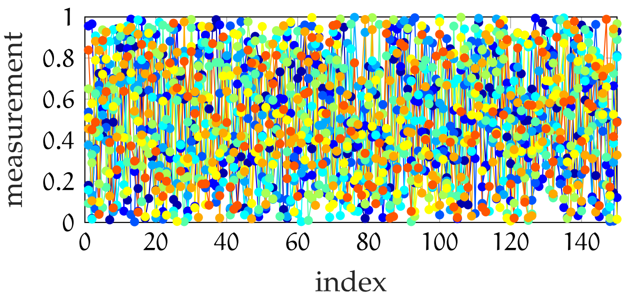


Figure confidence.1: raw data with colors corresponding to samples.

```
n = N*M; % total number of measurements
mu_pop = 0.5; % because it's a uniform PDF between 0 and 1

rng(11); % seed the random number generator
signal_a = rand(N,M); % uniform PDF
size(signal_a) % check the size
```

```
ans =

   150   120
```

Let's take a look at the data by plotting the first ten samples (columns), as shown in Fig. confidence.1.
This is something like what we might see for continuous measurement data. Now, the histogram, shown in ??.

```
samples_to_plot = 10;
h = figure;
c = jet(samples_to_plot); % color array
for j=1:samples_to_plot
    histogram(signal_a(:,j),...
        30, ... % number of bins
        'facecolor',c(j,:),...
        'facealpha',.3,...
        'normalization','probability'... % for PMF
    );
    hold on;
end
hold off;
xlim([-.05,1.05])
xlabel('measurement')
ylabel('probability')
hgsave(h,'figures/temp');
```

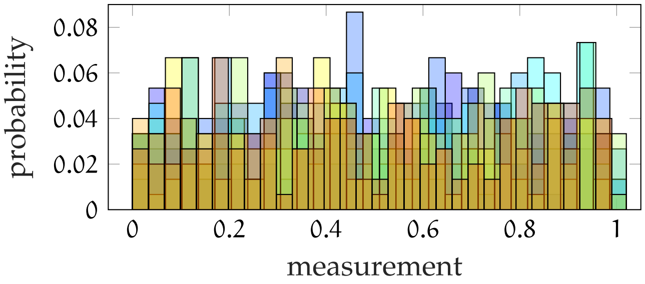


Figure confidence.2: a histogram showing the approximately uniform distribution of each sample (color).

This isn't a great plot, but it shows roughly that each sample is fairly uniformly distributed.

Sample statistics

Now let's check out the sample statistics. We want the sample mean and standard deviation of each column. Let's use the built-in functions `mean` and `std`.

```
mu_a = mean(signal_a,1); % mean of each column
s_a = std(signal_a,1); % std of each column
```

Now we can compute the mean statistics, both the mean of the mean $\overline{\overline{X}}$ and the standard deviation of the mean $s_{\overline{X}}$, which we don't strictly need for this part, but we're curious. We choose to use the direct estimate instead of the $s_X/\sqrt{N}$ formula, but they should be close.

```
mu_mu = mean(mu_a)
s_mu = std(mu_a)
```

```
mu_mu =

    0.4987


s_mu =

    0.0236
```

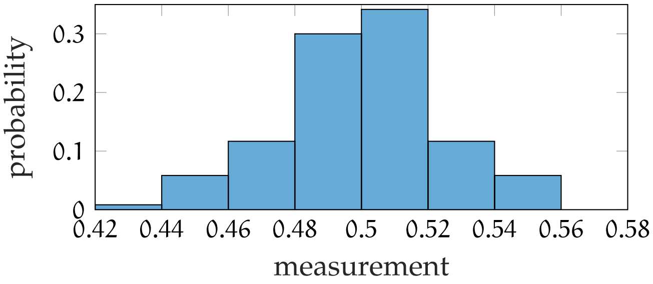


Figure confidence.3: a histogram showing the approximately normal distribution of the means.

The truth about sample means

It's the moment of truth. Let's look at the distribution, shown in Fig. confidence.3.

```
h = figure;
histogram(mu_a,...
    'normalization','probability'... % for PMF
);
hold off;
xlabel('measurement')
ylabel('probability')
hgsave(h,'figures/temp');
```

This looks like a Gaussian distribution about the mean of means, so I guess the central limit theorem is legit.

Gaussian and probability

We already know how to compute the probability P a value of a random variable X lies in a certain interval from a PMF or PDF (the sum or the integral, respectively). This means that, for sufficiently large sample size N such that we can assume from the central limit theorem that the sample means $\overline{x_i}$ are normally distributed, the probability a sample mean value $\overline{x_i}$ is in a certain interval is given by integrating the Gaussian PDF. The Gaussian PDF for random variable Y representing the sample means is

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(y-\mu)^2}{2\sigma^2}. \tag{1}$$

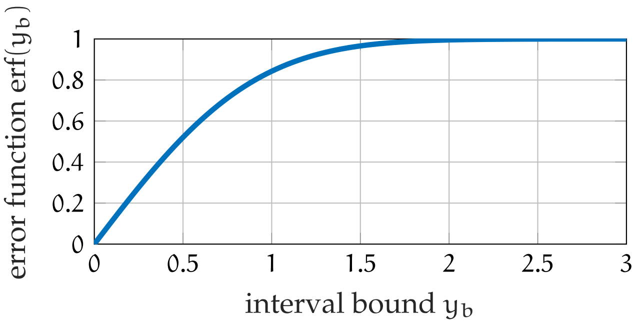


Figure confidence.4: the error function.

where μ is the population mean and σ is the population standard deviation.
The integral of f over some interval is the probability a value will be in that interval. Unfortunately, that integral is uncool. It gives rise to the definition of the error function, which, for the Gaussian random variable Y, is

$$\mathrm{erf}(y_b) = \frac{1}{\sqrt{\pi}} \int_{-y_b}^{y_b} e^{-t^2} dt. \tag{2}$$

This expresses the probability a sample mean being in the interval $[-y_b, y_b]$ if Y has mean 0 and variance $1/2$.
Matlab has this built-in as `erf`, shown in Fig. confidence.4.

```
y_a = linspace(0,3,100);
h = figure;
p1 = plot(y_a,erf(y_a));
p1.LineWidth = 2;
grid on
xlabel('interval bound $y_b$','interpreter','latex')
ylabel('error function $\textrm{erf}(y_b)$',...
  'interpreter','latex')
hgsave(h,'figures/temp');
```

We could deal directly with the error function, but most people don't and we're weird enough, as it is. Instead, people use the Gaussian cumulative distribution function (CDF)

Gaussian cumulative distribution function

$\Phi : \mathbb{R} \to \mathbb{R}$, which is defined as

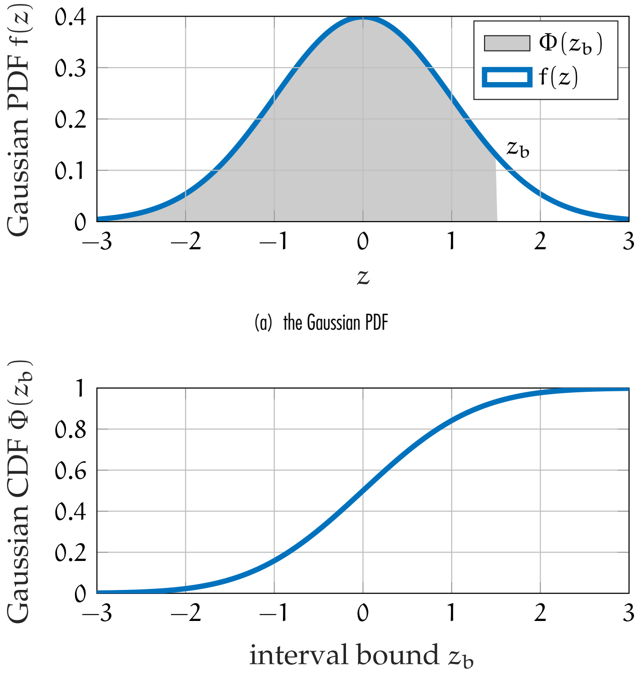
$$\Phi(z) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right)\right) \tag{3}$$

and which expresses the probability of a Gaussian random variable Z with mean 0 and standard deviation 1 taking on a value in the interval $(-\infty, z]$. The Gaussian CDF and PDF are shown in Fig. confidence.5. Values can be taken directly from the graph, but it's more accurate to use the table of values in Appendix A.01.
That's great and all, but occasionally (always) we have Gaussian random variables with nonzero means and nonunity standard deviations. It turns out we can shift any Gaussian random variable by its mean and scale it by its standard deviation to make it have zero mean and standard deviation. We can then use Φ and interpret the results as being relative to the mean and standard deviation, using phrases like "the probability it is within two standard deviations of its mean." The transformed random variable Z and its values z are sometimes called the z-score. For a particular value x of a random variable X, we can compute its z-score (or value z of random variable Z) with the formula

$$z = \frac{x - \mu_X}{\sigma_X} \tag{4}$$

and compute the probability of X taking on a value within the interval, say, $x \in [x_{b-}, x_{b+}]$ from the table. (Sample statistics $\overline{X}$ and S_X are appropriate when population statistics are unknown.)
For instance, compute the probability a Gaussian random variable X with $\mu_X = 5$ and $\sigma_X = 2.34$ takes on a value within the interval $x \in [3, 6]$.

1. Compute the z-score of each endpoint of the interval:



(a) the Gaussian PDF

(b) the Gaussian CDF

Figure confidence.5: the Gaussian PDF and CDF for z-scores.

$$z_3 = \frac{3 - \mu_X}{\sigma_X} \approx -0.85 \tag{5}$$

$$z_6 = \frac{6 - \mu_X}{\sigma_X} \approx 0.43. \tag{6}$$

2. Look up the CDF values for z_3 and z_6, which are $\Phi(z_3) = 0.1977$ and $\Phi(z_6) = 0.6664$. 3. The CDF values correspond to the probabilities $x < 3$ and $x < 6$. Therefore, to find the probability x lies in that interval, we subtract the lower bound probability:

$$\begin{aligned} P(x \in [3,6]) &= P(x < 6) - P(x < 3) \quad (7) \\ &= \Phi(6) - \Phi(3) \quad (8) \\ &\approx 0.6664 - 0.1977 \quad (9) \\ &\approx 0.4689. \quad (10) \end{aligned}$$

So there is a 46.89% probability, and therefore we have 46.89% confidence, that $x \in [3,6]$.

Often we want to go the other way, estimating the symmetric interval $[x_{b-}, x_{b+}]$ for which there is a given probability. In this case, we first look up the z-score corresponding to a certain probability. For concreteness, given the same population statistics above, let's find the symmetric interval $[x_{b-}, x_{b+}]$ over which we have 90% confidence. From the table, we want two, symmetric z-scores that have CDF-value difference 0.9. Or, in maths,

$$\Phi(z_{b+}) - \Phi(z_{b-}) = 0.9 \quad \text{and} \quad z_{b+} = -z_{b-}. \tag{11}$$

Due to the latter relation and the additional fact that the Gaussian CDF has antisymmetry,

$$\Phi(z_{b+}) + \Phi(z_{b-}) = 1. \tag{12}$$

Adding the two Φ equations,

$$\begin{aligned} \Phi(z_{b+}) &= 1.9/2 \qquad (13)\\ &= 0.95 \qquad (14) \end{aligned}$$

and $\Phi(z_{b-}) = 0.05$. From the table, these correspond (with a linear interpolation) to $z_b = z_{b+} = -z_{b-} \approx 1.645$. All that remains is to solve the z-score formula for x:

$$x = \mu_X + z\sigma_X. \tag{15}$$

From this,

$$\begin{aligned} x_{b+} &= \mu_X + z_{b+}\sigma_X \approx 8.849 \qquad (16)\\ x_{b-} &= \mu_X + z_{b-}\sigma_X \approx 1.151. \qquad (17) \end{aligned}$$

and X has a 90% confidence interval $[1.151, 8.849]$.

Example stats.confidence–1 re: gaussian confidence for a mean

Consider the data set generated above. What is our 95% confidence interval in our estimate of the mean?

Assuming we have a sufficiently large data set, the distribution of means is approximately Gaussian. Following the same logic as above, we need z-score that gives an upper CDF value of . From the table, we obtain the $z_b = z_{b+} = -z_{b-}$, below.

```
z_b = 1.96;
```

Now we can estimate the mean using our sample and mean statistics,

$$\overline{X} = \overline{\overline{X}} \pm z_b S_{\overline{X}}. \tag{18}$$

```
mu_x_95 = mu_mu + [-z_b,z_b]*s_mu
```

```
mu_x_95 =

    0.4526    0.5449
```

This is our 95% confidence interval in our estimate of the mean.

stats.student Student confidence

The central limit theorem tells us that, for large sample size N, the distribution of the means is Gaussian. However, for small sample size, the Gaussian isn't as good of an estimate. Student's t-distribution is superior for lower sample size and equivalent at higher sample size.

Student's t–distribution

Technically, if the population standard deviation σ_X is known, even for low sample size we should use the Gaussian distribution. However, this rarely arises in practice, so we can usually get away with an "always t" approach.

A way that the t-distribution accounts for low-N is by having an entirely different distribution for each N (seems a bit of a cheat, to me). Actually, instead of N, it uses the degrees of freedom ν, which is N minus the number of parameters required to compute the statistic. Since the standard deviation requires only the mean, for most of our cases, $\nu = N - 1$.

degrees of freedom

As with the Gaussian distribution, the t-distribution's integral is difficult to calculate. Typically, we will use a t-table, such as the one given in Appendix A.02. There are three points of note.

1. Since we are primarily concerned with going from probability/confidence values (e.g. $P\%$ probability/confidence) to intervals, typically there is a column for each probability.
2. The extra parameter ν takes over one of the dimensions of the table because three-dimensional tables are illegal.
3. Many of these tables are "two-sided," meaning their t-scores and probabilities assume you want the symmetric probability about the mean over the interval $[-t_b, t_b]$, where t_b is your t-score bound.

Consider the following example.

Example stats.student−1 re: student confidence interval

Write a Matlab script to generate a data set with 200 samples and sample sizes $N \in \{10, 20, 100\}$ using any old distribution. Compare the distribution of the means for the different N. Use the sample distributions and a t-table to compute 99% confidence intervals.

Generate the data set.

```
confidence = 0.99; % requirement

M = 200; % # of samples
N_a = [10,20,100]; % sample sizes

mu = 27; % population mean
sigma = 9; % population std

rng(1) % seed random number generator
data_a = mu + sigma*randn(N_a(end),M); % normal
size(data_a) % check size
data_a(1:10,1:5) % check 10 rows and five columns
```

```
ans =

   100   200


ans =

   21.1589   30.2894   27.8705   30.7835   28.3662
   37.6305   17.1264   28.2973   24.0811   34.3486
   20.1739   44.3719   43.7059   39.0699   32.2002
   17.0135   32.6064   36.9030   37.9230   36.5747
   19.3900   32.9156   23.7230   22.4749   19.7709
   21.8460   13.8295   31.2479   16.9527   34.1876
   21.9719   34.6854   19.4480   18.7014   24.1642
   28.6054   32.2244   22.2873   26.9906   37.6746
   25.2282   18.7326   14.5011   28.3814   27.7645
   32.2780   34.1538   27.0382   18.8643   14.1752
```

Compute the means for different sample sizes.

```
mu_a = NaN*ones(length(N_a),M);
for i = 1:length(N_a)
    mu_a(i,:) = mean(data_a(1:N_a(i),1:M),1);
end
```

Plotting the distribution of the means yields Figure student.1.

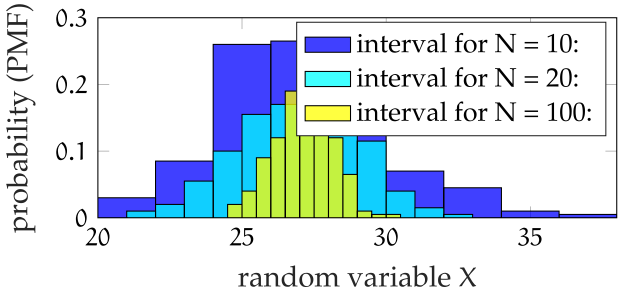


Figure student.1: a histogram showing the distribution of the means for each sample size.

It makes sense that the larger the sample size, the smaller the spread. A quantitative metric for the spread is, of course, the standard deviation of the means for each sample size.

```
S_mu = std(mu_a,0,2)
```

```
S_mu =

    2.8365
    2.0918
    1.0097
```

Look up t-table values or use Matlab's `tinv` for different sample sizes and 99% confidence. Use these, the mean of means, and the standard deviation of means to compute the 99% confidence interval for each N.

```
t_a = tinv(confidence,N_a-1)
for i = 1:length(N_a)
    interval = mean(mu_a(i,:)) + ...
      [-1,1]*t_a(i)*S_mu(i);
    disp(sprintf('interval for N = %i: ',N_a(i)))
    disp(interval)
end
```

```
t_a =

    2.8214    2.5395    2.3646

interval for N = 10:
   19.0942   35.1000
```

```
interval for N = 20:
   21.6292   32.2535

interval for N = 100:
   24.7036   29.4787
```

As expected, the larger the sample size, the smaller the interval over which we have 99% confidence in the estimate.

stats.multivar Multivariate probability and correlation

Thus far, we have considered probability density and mass functions (PDFs and PMFs) of only one random variable. But, of course, often we measure multiple random variables $X_1, X_2, \dots, X_n$ during a single event, meaning a measurement consists of determining values $x_1, x_2, \dots, x_n$ of these random variables.

We can consider an n-tuple of continuous random variables to form a sample space $\Omega = \mathbb{R}^n$ on which a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, called the joint PDF assigns a probability density to each outcome $\boldsymbol{x} \in \mathbb{R}^n$. The joint PDF must be greater than or equal to zero for all $\boldsymbol{x} \in \mathbb{R}^n$, the multiple integral over Ω must be unity, and the multiple integral over a subset of the sample space $A \subset \Omega$ is the probability of the event A.

joint PDF

We can consider an n-tuple of discrete random variables to form a sample space $\mathbb{N}_0^n$ on which a multivariate function $f : \mathbb{N}_0^n \to \mathbb{R}$, called the joint PMF assigns a probability to each outcome $\boldsymbol{x} \in \mathbb{N}_0^n$. The joint PMF must be greater than or equal to zero for all $\boldsymbol{x} \in \mathbb{N}_0^n$, the multiple sum over Ω must be unity, and the multiple sum over a subset of the sample space $A \subset \Omega$ is the probability of the event A.

joint PMF

Example stats.multivar−1 re: bivariate gaussian pdf

Let's visualize multivariate PDFs by plotting a bivariate gaussian using Matlab's `mvnpdf` function.

```
mu = [10,20]; % means
Sigma = [1,0;0,.2]; % cov ... we'll talk about this
x1_a = linspace(...
    mu(1)-5*sqrt(Sigma(1,1)),...
    mu(1)+5*sqrt(Sigma(1,1)),...
    30);
x2_a = linspace(...
    mu(2)-5*sqrt(Sigma(2,2)),...
    mu(2)+5*sqrt(Sigma(2,2)),...
    30);
[X1,X2] = meshgrid(x1_a,x2_a);
f = mvnpdf([X1(:) X2(:)],mu,Sigma);
f = reshape(f,length(x2_a),length(x1_a));

h = figure;
p = surf(x1_a,x2_a,f);
xlabel('$x_1$','interpreter','latex')
ylabel('$x_2$','interpreter','latex')
zlabel('$f(x_1,x_2)$','interpreter','latex')
shading interp
hgsave(h,'figures/temp');
```

The result is Fig. multivar.1. Note how the means and standard deviations affect the distribution.

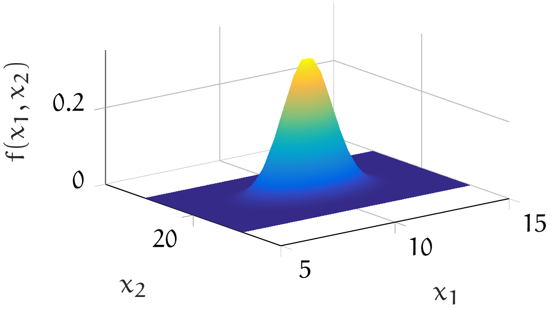


Figure multivar.1: two-variable gaussian PDF.

Marginal probability

marginal PDF

The marginal PDF of a multivariate PDF is the PDF of some subspace of Ω after one or more variables have been "integrated out," such that a fewer number of random variables remain. Of course, these marginal PDFs must have the same properties of any PDF, such as integrating to unity.

Example stats.multivar–2 re: bivariate gaussian marginal probability

Let's demonstrate this by numerically integrating over x_2 in the bivariate Gaussian, above.

Continuing from where we left off, let's integrate.

```
f1 = trapz(x2_a,f',2); % trapezoidal integration
```

Plotting this yields Fig. multivar.2.

```
h = figure;
p = plot(x1_a,f1);
p.LineWidth = 2;
xlabel('$x_1$','interpreter','latex')
ylabel(...
'$g(x_1)=\int_{-\infty}^\infty f(x_1,x_2) d x_2$',...
'interpreter','latex'...
)
hgsave(h,'figures/temp');
```

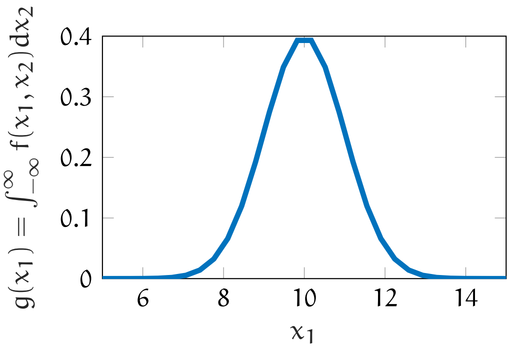


Figure multivar.2: marginal Gaussian PDF $g(x_1)$.

We should probably verify that this integrates to one.

```
disp(['integral over x_1 = ',...
  sprintf('%0.7f',trapz(x1_a,f1))] ...
)
```

```
integral over x_1 = 0.9999986
```

Not bad.

Covariance

Very often, especially in machine learning or artificial intelligence applications, the question about two random variables X and Y is: how do they co-vary? That is what is their covariance,

machine learning

artificial intelligence

covariance

defined as

$$\begin{aligned}\mathrm{Cov}\left[X,Y\right] &\equiv E\left((X-\mu_X)(Y-\mu_Y)\right)\\ &= E(XY)-\mu_X\mu_Y.\end{aligned}$$

Note that when $X = Y$, the covariance is just the variance. When a covariance is large and positive, it is an indication that the random variables are strongly correlated. When it is large and negative, they are strongly anti-correlated. Zero covariance means the variables are uncorrelated. In fact, correlation is defined as

correlation

$$\mathrm{Cor}\left[X,Y\right] = \frac{\mathrm{Cov}\left[X,Y\right]}{\sqrt{\mathrm{Var}\left[X\right]\mathrm{Var}\left[Y\right]}}.$$

This is essentially the covariance "normalized" to the interval $[-1, 1]$.

Sample covariance

As with the other statistics we've considered, covariance can be estimated from measurement. The estimate, called the sample covariance q_{XY}, of random variables X and Y with sample size N is given by

sample covariance

$$q_{XY} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\overline{X})(y_i-\overline{Y}).$$

Multivariate covariance

With n random variables X_i, one can compute the covariance of each pair. It is common practice to define an $n \times n$ matrix of covariances called the covariance matrix Σ such that each pair's covariance

covariance matrix

$$\mathrm{Cov}\left[X_i, X_j\right] \tag{1}$$

appears in its row-column combination (making it symmetric), as shown below.

$$\Sigma = \begin{bmatrix} \mathrm{Cov}\left[X_1,X_1\right] & \mathrm{Cov}\left[X_1,X_2\right] & \cdots & \mathrm{Cov}\left[X_1,X_n\right] \\ \mathrm{Cov}\left[X_2,X_1\right] & \mathrm{Cov}\left[X_2,X_2\right] & & \mathrm{Cov}\left[X_2,X_n\right] \\ \vdots & & \ddots & \vdots \\ \mathrm{Cov}\left[X_n,X_1\right] & \mathrm{Cov}\left[X_n,X_2\right] & \cdots & \mathrm{Cov}\left[X_n,X_n\right] \end{bmatrix}$$

The multivariate sample covariance matrix Q is the same as above, but with sample covariances $q_{x_i x_j}$.

sample covariance matrix

Both covariance matrices have correlation analogs.

Example stats.multivar–3

re: car data sample covariance and correlation

Let's use a built-in multivariate data set that describes different features of cars, listed below.

```
d = load('carsmall.mat') % this is a "struct"
```

Let's compute the sample covariance and correlation matrices.

```
variables = {...
    'MPG','Cylinders',...
    'Displacement','Horsepower',...
    'Weight','Acceleration',...
    'Model_Year'};
n = length(variables);
m = length(d.MPG);

data = NaN*ones(m,n); % preallocate
for i = 1:n
    data(:,i) = d.(variables{i});
end

cov_d = nancov(data); % sample covariance
cor_d = corrcov(cov_d) % sample correlation
```

This is highly correlated/anticorrelated data! Let's plot each variable versus each other variable to see the correlations of each. We use a red-to-blue colormap to contrast anticorrelation and correlation. Purple, then, is uncorrelated.

The following builds the red-to-blue colormap.

```
n_colors = 10;
cmap_rb = NaN*ones(n_colors,3);
for i = 1:n_colors
```

```
    a = i/n_colors;
    cmap_rb(i,:) = (1-a)*[1,0,0]+a*[0,0,1];
end
```

```
h = figure;
for i = 1:n
    for j = 1:n
        subplot(n,n,sub2ind([n,n],j,i))
        p = plot(...
          d.(variables{i}),...
          d.(variables{j}),'.'...
        ); hold on
        this_color = cmap_rb(...
          round((cor_d(i,j)+1)*(n_colors-1)/2),...
          :...
        );
        p.MarkerFaceColor = this_color;
        p.MarkerEdgeColor = this_color;
    end
end
hgsave(h,'figures/temp');
```

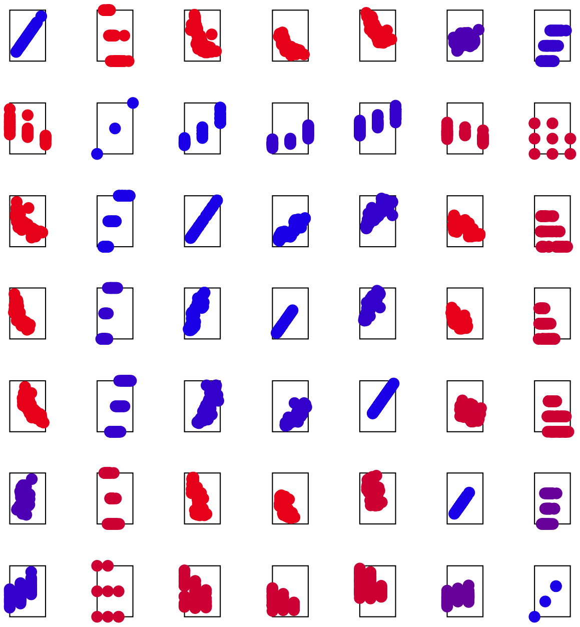


Figure multivar.3: car data correlation.

Conditional probability and dependence

Independent variables are uncorrelated. However, uncorrelated variables may or may not be independent. Therefore, we cannot use correlation alone as a test for independence. For instance, for random variables X and Y, where X has some even distribution and $Y = X^2$, clearly the variables are dependent. However, the are also uncorrelated (due to symmetry).

Example stats.multivar−4

re: car data sample covariance and correlation

Using a uniform distribution $U(-1, 1)$, show that X and Y are uncorrelated (but dependent) with $Y = X^2$ with some sampling. We compute the correlation for different sample sizes.

```
N_a = round(linspace(10,500,100));
qc_a = NaN*ones(size(N_a));
rng(6)
x_a = -1 + 2.*rand(max(N_a),1);
y_a = x_a.^2;
for i = 1:length(N_a)
    % should write incremental algorithm
    % but lazy
    q = cov(x_a(1:N_a(i)),y_a(1:N_a(i)));
    qc = corrcov(q);
    qc_a(i) = qc(2,1); % "cross" correlation
end
```

The absolute values of the correlations are shown in Fig. multivar.4. Note that we should probably average several such curves to estimate how the correlation would drop off with $N$, but the single curve describes our understanding that the correlation, in fact, approaches zero in the large-sample limit.

```
h = figure;
p = plot(N_a,abs(qc_a));
p.LineWidth = 2;
xlabel('sample size $N$','interpreter','latex')
ylabel(...
  'absolute sample correlation',...
  'interpreter','latex'...
)
hgsave(h,'figures/temp');
```

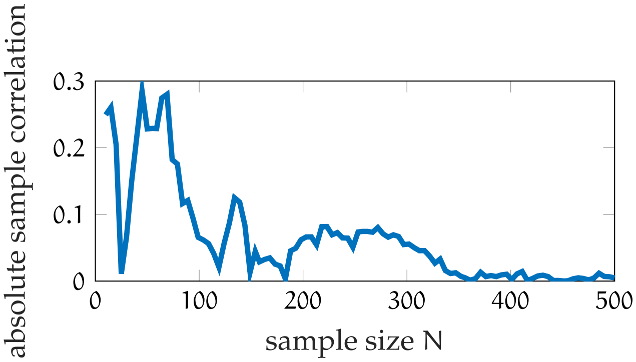


Figure multivar.4: absolute value of the sample correlation between $X \sim U(-1, 1)$ and $Y = X^2$ for different sample size N. In the limit, the population correlation should approach zero and yet X and Y are not independent.

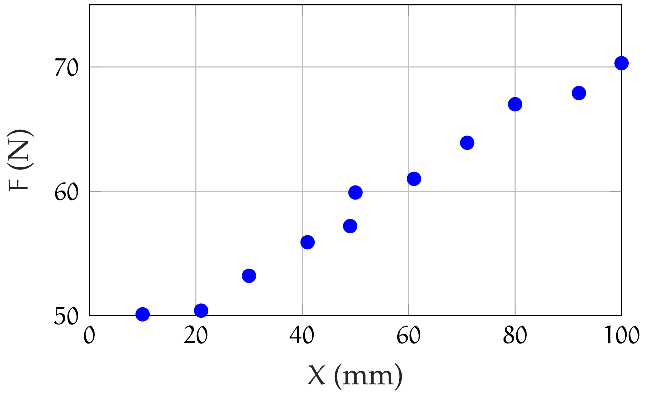


Figure regression.1: force-displacement data.

stats.regression Regression

Suppose we have a sample with two measurands: (1) the force $F$ through a spring and (2) its displacement $X$ (not from equilibrium). We would like to determine an analytic function that relates the variables, perhaps for prediction of the force given another displacement. There is some variation in the measurement. Let's say the following is the sample.

```
X_a = 1e-3*[10,21,30,41,49,50,61,71,80,92,100]'; % m
F_a = [50.1,50.4,53.2,55.9,57.2,59.9,...
  61.0,63.9,67.0,67.9,70.3]'; % N
```

Let's take a look at the data. The result is Figure regression.1.

```
h = figure;
p = plot(X_a*1e3,F_a,'.b','MarkerSize',15);
xlabel('$X$ (mm)','interpreter','latex')
ylabel('$F$ (N)','interpreter','latex')
xlim([0,max(X_a*1e3)])
grid on
hgsave(h,'figures/temp');
```

How might we find an analytic function that agrees with the data? Broadly, our strategy will be to assume a general form of a function and use the data to set the parameters in the function such that the difference between the data and the function is minimal.

Let y be the analytic function that we would like to fit to the data. Let y_i denote the value of $y(x_i)$, where x_i is the ith value of the random variable X from the sample. Then we want to minimize the differences between the force measurements F_i and y_i.

From calculus, recall that we can minimize a function by differentiating it and solving for the zero-crossings (which correspond to local maxima or minima).

First, we need such a function to minimize. Perhaps the simplest, effective function D is constructed by squaring and summing the differences we want to minimize, for sample size N:

$$D(x_i) = \sum_{i=1}^{N} (F_i - y_i)^2 \tag{1}$$

(recall that $y_i = y(x_i)$, which makes D a function of x).

Now the form of y must be chosen. We consider only mth-order polynomial functions y, but others can be used in a similar manner:

$$y(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m. \tag{2}$$

If we treat D as a function of the polynomial coefficients a_j, i.e.

$$D(a_0, a_1, \cdots, a_m), \tag{3}$$

and minimize D for each value of x_i, we must take the partial derivatives of D with respect to each a_j and set each equal to zero:

$$\frac{\partial D}{\partial a_0} = 0, \quad \frac{\partial D}{\partial a_1} = 0, \quad \cdots, \quad \frac{\partial D}{\partial a_m} = 0.$$

This gives us N equations and $m+1$ unknowns

a_j. Writing the system in matrix form,

$$\underbrace{\begin{bmatrix}1 & x_1 & x_1^2 & \cdots & x_1^m\\ 1 & x_2 & x_2^2 & \cdots & x_2^m\\ 1 & x_3 & x_3^2 & \cdots & x_3^m\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 1 & x_N & x_N^2 & \cdots & x_N^m\end{bmatrix}}_{A_{N\times(m+1)}} \underbrace{\begin{bmatrix}a_0\\ a_1\\ a_2\\ \vdots\\ a_m\end{bmatrix}}_{\mathbf{a}_{(m+1)\times 1}} = \underbrace{\begin{bmatrix}F_1\\ F_2\\ F_3\\ \vdots\\ F_N\end{bmatrix}}_{\mathbf{b}_{N\times 1}}. \tag{4}$$

Typically $N > m$ and this is an overdetermined system. Therefore, we usually can't solve by taking A^{-1} because A doesn't have an inverse! Instead, we either find the Moore-Penrose pseudo-inverse $A^\dagger$ and have $\mathbf{a} = A^\dagger \mathbf{b}$ as the solution, which is inefficient—or we can approximate $\mathbf{b}$ with an algorithm such as that used by Matlab's \ operator. In the latter case, `a_a = A\b_a`.

Example stats.regression–1

re: regression

Use Matlab's \ operator to find a good polynomial fit for the sample. There's the sometimes-difficult question "what order should we fit?" Let's try out several and see what the squared-differences function D gives.

```
N = length(X_a); % sample size
m_a = 2:N; % all the order up to N

A = NaN*ones(length(m_a),max(m_a),N);
for k = 1:length(m_a) % each order
    for j = 1:N % each measurement
        for i = 1:( m_a(k) + 1 ) % each coef
            A(k,j,i) = X_a(j)^(i-1);
        end
    end
end
disp(squeeze(A(2,:,1:5)))
```

```
    1.0000    0.0100    0.0001    0.0000       NaN
    1.0000    0.0210    0.0004    0.0000       NaN
    1.0000    0.0300    0.0009    0.0000       NaN
    1.0000    0.0410    0.0017    0.0001       NaN
    1.0000    0.0490    0.0024    0.0001       NaN
    1.0000    0.0500    0.0025    0.0001       NaN
    1.0000    0.0610    0.0037    0.0002       NaN
    1.0000    0.0710    0.0050    0.0004       NaN
    1.0000    0.0800    0.0064    0.0005       NaN
    1.0000    0.0920    0.0085    0.0008       NaN
    1.0000    0.1000    0.0100    0.0010       NaN
```

We've printed the first five columns of the third-order matrix, which only has four columns, so NaNs fill in the rest.

Now we can use the \ operator to solve for the coefficients.

```
a = NaN*ones(length(m_a),max(m_a));

warning('off','all')
for i = 1:length(m_a)
    A_now = squeeze(A(i,:,1:m_a(i)));
    a(i,1:m_a(i)) = (A_now(:,1:m_a(i))\F_a)';
end
warning('on','all')
```

```
n_plot = 100;
x_plot = linspace(min(X_a),max(X_a),n_plot);
y = NaN*ones(n_plot,length(m_a)); % preallocate
for i = 1:length(m_a)
    y(:,i) = polyval(fliplr(a(i,1:m_a(i))),x_plot);
end
```

```
h = figure;
for i = 1:2:length(m_a)-1
    p = plot(x_plot*1e3,y(:,i),'linewidth',1.5);
    hold on
    p.DisplayName = sprintf(...
        'order: %i ',...
        (m_a(i)-1)...
    );
end
p = plot(X_a*1e3,F_a,'.b','MarkerSize',15);
xlabel('$X$ (mm)','interpreter','latex')
ylabel('$F$ (N)','interpreter','latex')
p.DisplayName = 'sample';
legend('show','location','southeast')
grid on
hgsave(h,'figures/temp');
```

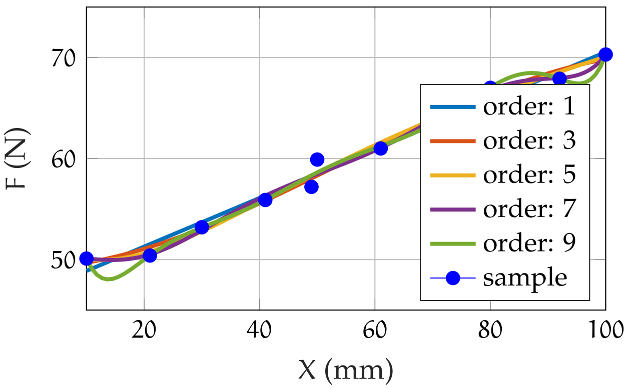


Figure regression.2: force-displacement data with curve fits.

stats.exe Exercises for Chapter stats

Exercise stats.brew

______/20 p.

You need to know the duration of time a certain stage of a brewing process takes. You set up an automated test environment that repeats the test 100 times, recorded in the following JSON[4] data file:

`http://ricopic.one/mathematical_foundations/source/brew.json`

Perform the following analysis.

a. Download and parse the JSON file (it contains a single array).
b. Estimate the duration of the process from the sample.
c. Choose and justify an assumed probability density function for the random variable duration.
d. Use this PDF model to compute a 99 percent confidence interval for your duration estimate.
e. Compute your duration confidence interval for the range of confidence values $[85, 99.99]$ percent.[5]
f. Plot the confidence intervals over the range of confidence in said intervals.

Exercise stats.laboritorium

______/20 p.

Use linear regression techniques to find the values of a, b, c, and d, in a cubic function of the form,

$$f(x) = ax^3 + bx^2 + cx + d,$$

using the data below.

[4] JSON is a simple and common programming language-independent data format. For parsing it with Matlab, see `jsondecode` here: `mathworks.com/help/matlab/ref/jsondecode.html`. For parsing it with Python, see the module `json` here: `docs.python.org/library/json`.

[5] Consider using a z- or t-score inverse CDF lookup function like `t.ppf` from `scipy.stats`.

x	$f(x)$
-2.0	-4.7
-1.5	-1.9
-1.0	1.5
-0.5	1.5
0.0	1.4
0.6	0.3
1.1	-1.5
1.6	0.0
2.1	0.6
2.6	4.2

Exercise stats.robotization

Use linear regression techniques to find the value of τ in the function,

$$f(t) = 1 - e^{\frac{-t^2}{\tau}}$$

Using the data below.

t	$f(t)$
0.1	0.02
0.6	0.34
1.1	0.74
1.6	0.94
2.1	0.98

vecs

Vector calculus

A great many physical situations of interest to engineers can be described by calculus. It can describe how quantities continuously change over (say) time and gives tools for computing other quantities. We assume familiarity with the fundamentals of calculus: limit, series, derivative, and integral. From these and a basic grasp of vectors, we will outline some of the highlights of vector calculus. Vector calculus is particularly useful for describing the physics of, for instance, the following.

calculus

limit
series
derivative
integral

vector calculus

mechanics of particles wherein is studied the motion of particles and the forcing causes thereof

rigid–body mechanics wherein is studied the motion, rotational and translational, and its forcing causes, of bodies considered rigid (undeformable)

solid mechanics wherein is studied the motion and deformation, and their forcing causes, of continuous solid bodies (those that retain a specific resting shape)

fluid mechanics wherein is studied the motion and its forcing causes of fluids (liquids, gases, plasmas)

heat transfer wherein is studied the movement of thermal energy through and among bodies

electromagnetism wherein is studied the motion and its forcing causes of electrically charged particles

This last example was in fact very influential in the original development of both vector calculus and complex analysis.[1] It is not an exaggeration to say that the topics above comprise the majority of physical topics of interest in engineering.

complex analysis

A good introduction to vector calculus is given by Kreyszig.[2] Perhaps the most famous and enjoyable treatment is given by Schey[3] in the adorably titled Div, Grad, Curl and All that.

It is important to note that in much of what follows, we will describe (typically the three-dimensional space of our lived experience) as a euclidean vector space: an n-dimensional vector space isomorphic to $\mathbb{R}^n$. As we know from linear algebra, any vector $\boldsymbol{v} \in \mathbb{R}^n$ can be expressed in any number of bases. That is, the vector $\boldsymbol{v}$ is a basis-free object with multiple basis representations. The components and basis vectors of a vector change with basis changes, but the vector itself is invariant. A coordinate system is in fact just a basis. We are most familiar, of course, with Cartesian coordinates, which is the specific orthonormal basis $\mathbf{b}$ for $\mathbb{R}^n$:

euclidean vector space

bases

components

basis vectors

invariant

coordinate system

Cartesian coordinates

$$\mathbf{b}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \cdots, \quad \mathbf{b}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}. \tag{1}$$

Manifolds are spaces that appear locally as $\mathbb{R}^n$, but can be globally rather different and can describe non-euclidean geometry wherein euclidean geometry's parallel postulate is invalid. Calculus on manifolds is the focus of differential geometry, a subset of which we can consider our current study. A motivation for further study of differential geometry is that it is very convenient when dealing with advanced applications of mechanics, such as rigid-body mechanics of robots and vehicles. A very nice mathematical introduction is given by Lee[4] and

manifolds

non−euclidean geometry

parallel postulate

differential geometry

[1] For an introduction to complex analysis, see Kreyszig. (Kreyszig, Advanced Engineering Mathematics, Part D)

[2] ibidem, Chapters 9, 10.

[3] H.M. Schey. Div, Grad, Curl, and All that: An Informal Text on Vector Calculus. W.W. Norton, 2005. ISBN: 9780393925166.

[4] John M. Lee. Introduction to Smooth Manifolds. second. volume 218. Graduate Texts in Mathematics. Springer, 2012.

Bullo and Lewis[5] give a compact presentation in the context of robotics.

Vector fields have several important properties of interest we'll explore in this chapter. Our goal is to gain an intuition of these properties and be able to perform basic calculation.

[5] Francesco Bullo and Andrew D. Lewis. Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems. byeditorJ.E. Marsden, L. Sirovich and M. Golubitsky. Springer, 2005.

vecs.div Divergence, surface integrals, and flux

Flux and surface integrals

Consider a surface S. Let $\mathbf{r}(u,v) = [x(u,v), y(u,v), z(u,v)]$ be a parametric position vector on a Euclidean vector space $\mathbb{R}^3$. Furthermore, let $\mathbf{F} : \mathbb{R}^3 \to \mathbb{R}^3$ be a vector-valued function of $\mathbf{r}$ and let $\mathbf{n}$ be a unit-normal vector on a surface S. The surface integral

surface integral

$$\iint_S \mathbf{F} \cdot \mathbf{n}\, dS \tag{1}$$

which integrates the normal of $\mathbf{F}$ over the surface. We call this quantity the flux of $\mathbf{F}$ out of the surface S. This terminology comes from fluid flow, for which the flux is the mass flow rate out of S. In general, the flux is a measure of a quantity (or field) passing through a surface. For more on computing surface integrals, see Schey[6] and Kreyszig.[7]

flux

Continuity

Consider the flux out of a surface S that encloses a volume ΔV, divided by that volume:

$$\frac{1}{\Delta V} \iint_S \mathbf{F} \cdot \mathbf{n}\, dS. \tag{2}$$

This gives a measure of flux per unit volume for a volume of space. Consider its physical meaning when we interpret this as fluid flow: all fluid that enters the volume is negative flux and all that leaves is positive. If physical conditions are such that we expect no fluid to enter or exit the volume via what is called a source or a sink, and if we assume the density of the fluid is uniform (this is called an incompressible fluid), then all the fluid that enters the volume must exit and we get

source
sink

incompressible

$$\frac{1}{\Delta V} \iint_S \mathbf{F} \cdot \mathbf{n}\, dS = 0. \tag{3}$$

[6] Schey, Div, Grad, Curl, and All that: An Informal Text on Vector Calculus, pp. 21-30.

[7] Kreyszig, Advanced Engineering Mathematics, § 10.6.

This is called a continuity equation, although typically this name is given to equations of the form in the next section. This equation is one of the governing equations in continuum mechanics.

continuity equation

Divergence

Let's take the flux-per-volume as the volume $\Delta V \to 0$ we obtain the following.

Equation 4 divergence: integral form

$$\lim_{\Delta V \to 0} \frac{1}{\Delta V} \iint_S \mathbf{F} \cdot \mathbf{n} \, dS.$$

This is called the divergence of $\mathbf{F}$ and is defined at each point in $\mathbb{R}^3$ by taking the volume to zero about it. It is given the shorthand $\operatorname{div} \mathbf{F}$.

divergence

What interpretation can we give this quantity? It is a measure of the vector field's flux outward through a surface containing an infinitesimal volume. When we consider a fluid, a positive divergence is a local decrease in density and a negative divergence is a density increase. If the fluid is incompressible and has no sources or sinks, we can write the continuity equation

$$\operatorname{div} \mathbf{F} = 0. \tag{5}$$

In the Cartesian basis, it can be shown that the divergence is easily computed from the field

$$\mathbf{F} = F_x \hat{\mathbf{i}} + F_y \hat{\mathbf{j}} + F_z \hat{\mathbf{k}} \tag{6}$$

as follows.

Equation 7 divergence: differential form

$$\operatorname{div} \mathbf{F} = \partial_x F_x + \partial_y F_y + \partial_z F_z$$

Exploring divergence

Divergence is perhaps best explored by considering it for a vector field in $\mathbb{R}^2$. Such a field $\mathbf{F} = F_x\hat{\imath} + F_y\hat{\jmath}$ can be represented as a "quiver" plot. If we overlay the quiver plot over a "color density" plot representing $\operatorname{div}\mathbf{F}$, we can increase our intuition about the divergence. The following was generated from a Jupyter notebook with the following filename and kernel.

```
notebook filename: div_surface_integrals_flux.ipynb
notebook kernel: python3
```

First, load some Python packages.

```python
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import LogLocator
from matplotlib.colors import *
from sympy.utilities.lambdify import lambdify
from IPython.display import display, Markdown, Latex
```

Now we define some symbolic variables and functions.

```python
var('x,y')
F_x = Function('F_x')(x,y)
F_y = Function('F_y')(x,y)
```

Rather than repeat code, let's write a single function `quiver_plotter_2D` to make several of these plots.

```python
def quiver_plotter_2D(
  field={F_x:x*y,F_y:x*y},
  grid_width=3,
  grid_decimate_x=8,
  grid_decimate_y=8,
  norm=Normalize(),
  density_operation='div',
  print_density=True,
):
  # define symbolics
  var('x,y')
  F_x = Function('F_x')(x,y)
```

```
F_y = Function('F_y')(x,y)
field_sub = field

# compute density
if density_operation is 'div':
  den = F_x.diff(x) + F_y.diff(y)
elif density_operation is 'curl':
  # in the k direction
  den = F_y.diff(x) - F_x.diff(y)
else:
  error('div and curl are the only density operators')
den_simp = den.subs(
  field_sub
).doit().simplify()
if den_simp.is_constant():
  print(
    'Warning: density operator is constant (no density plot)'
  )
if print_density:
  print(f'The {density_operation} is:')
  display(den_simp)

# lambdify for numerics
F_x_sub = F_x.subs(field_sub)
F_y_sub = F_y.subs(field_sub)
F_x_fun = lambdify((x,y),F_x.subs(field_sub),'numpy')
F_y_fun = lambdify((x,y),F_y.subs(field_sub),'numpy')
if F_x_sub.is_constant:
  F_x_fun1 = F_x_fun # dummy
  F_x_fun = lambda x,y: F_x_fun1(x,y)*np.ones(x.shape)
if F_y_sub.is_constant:
  F_y_fun1 = F_y_fun # dummy
  F_y_fun = lambda x,y: F_y_fun1(x,y)*np.ones(x.shape)
if not den_simp.is_constant():
  den_fun = lambdify((x,y), den_simp,'numpy')

# create grid
w = grid_width
Y, X = np.mgrid[-w:w:100j, -w:w:100j]

# evaluate numerically
F_x_num = F_x_fun(X,Y)
F_y_num = F_y_fun(X,Y)
if not den_simp.is_constant():
  den_num = den_fun(X,Y)

# plot
p = plt.figure()
#   colormesh
if not den_simp.is_constant():
  cmap = plt.get_cmap('PiYG')
  im = plt.pcolormesh(X,Y,den_num,cmap=cmap,norm=norm)
  plt.colorbar()
# Abs  quiver
dx = grid_decimate_y
```

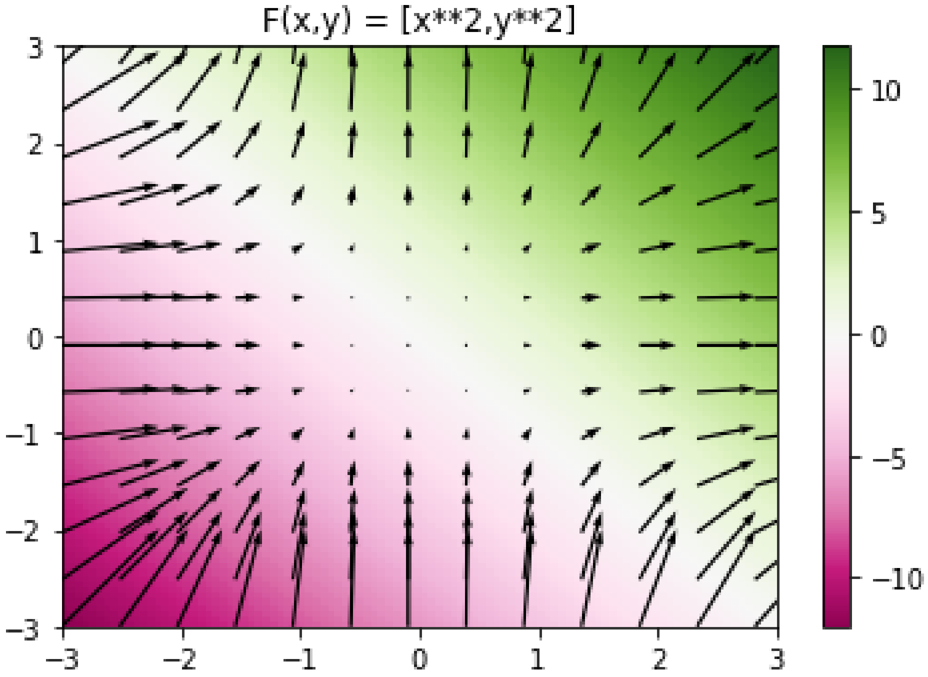


Figure div.1: png

```
  dy = grid_decimate_x
  plt.quiver(
    X[::dx,::dy],Y[::dx,::dy],
    F_x_num[::dx,::dy],F_y_num[::dx,::dy],
    units='xy', scale=10
  );
  plt.title(f'F(x,y) = [{F_x.subs(field_sub)},{F_y.subs(field_sub)}]')
  return p
```

Note that while we're at it, we included a hook for density plots of the curl of F, and we'll return to this in a later lecture.

Let's inspect several cases.

```
p = quiver_plotter_2D(
  field={F_x:x**2,F_y:y**2}
)
```

```
The div is:
```

$2x + 2y$

```
p = quiver_plotter_2D(
  field={F_x:x*y,F_y:x*y}
)
```

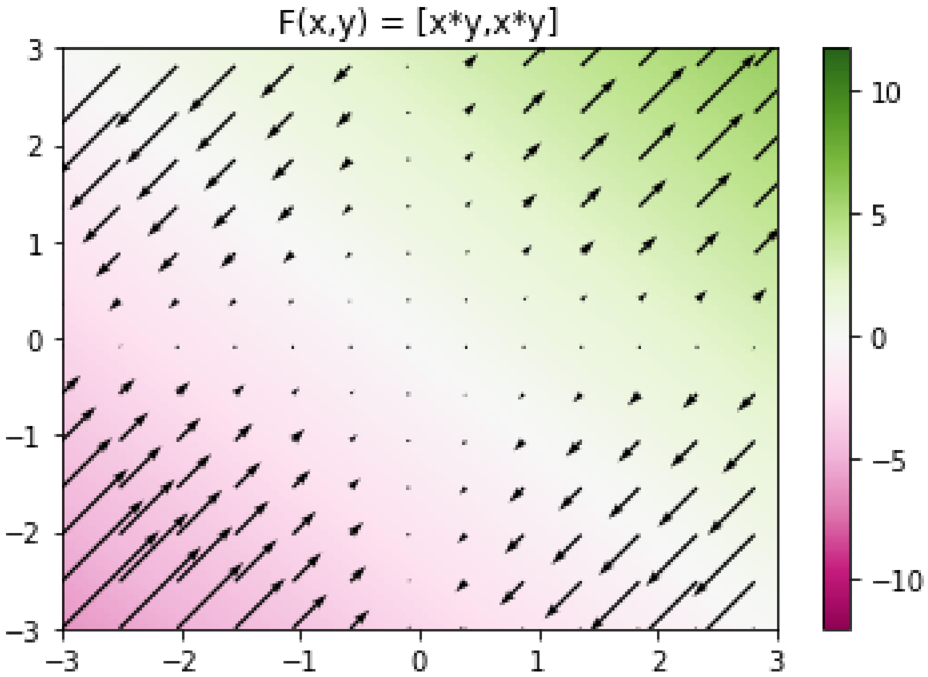


Figure div.2: png

```
The div is:
```

$$x + y$$

```
p = quiver_plotter_2D(
  field={F_x:x**2+y**2,F_y:x**2+y**2}
)
```

```
The div is:
```

$$2x + 2y$$

```
p = quiver_plotter_2D(
  field={F_x:x**2/sqrt(x**2+y**2),F_y:y**2/sqrt(x**2+y**2)},
  norm=SymLogNorm(linthresh=.3, linscale=.3)
)
```

```
The div is:
```

$$\frac{-x^3 - y^3 + 2\left(x + y\right)\left(x^2 + y^2\right)}{\left(x^2 + y^2\right)^{\frac{3}{2}}}$$

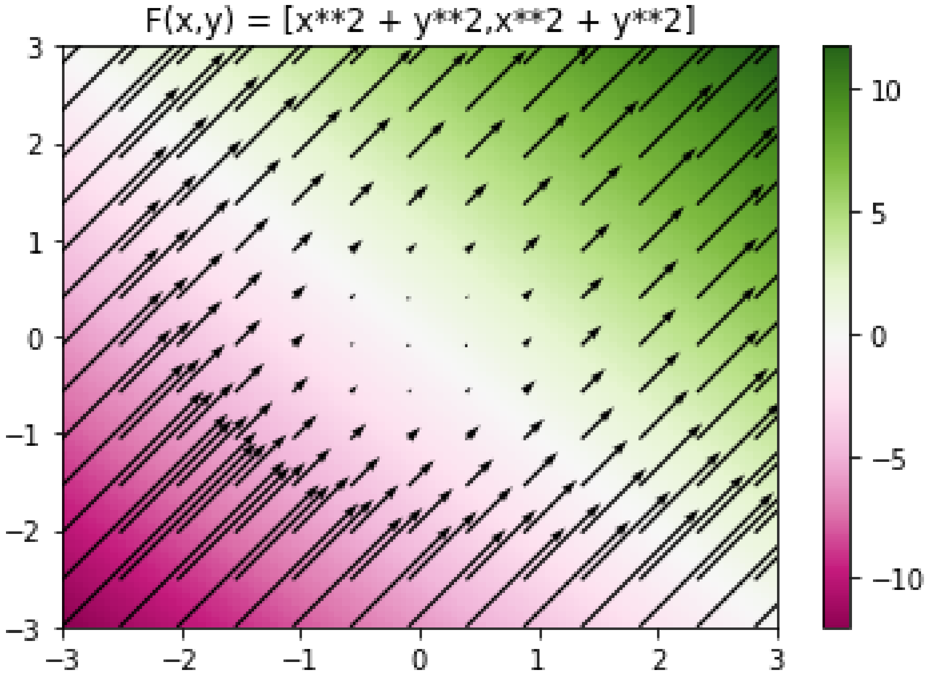


Figure div.3: png

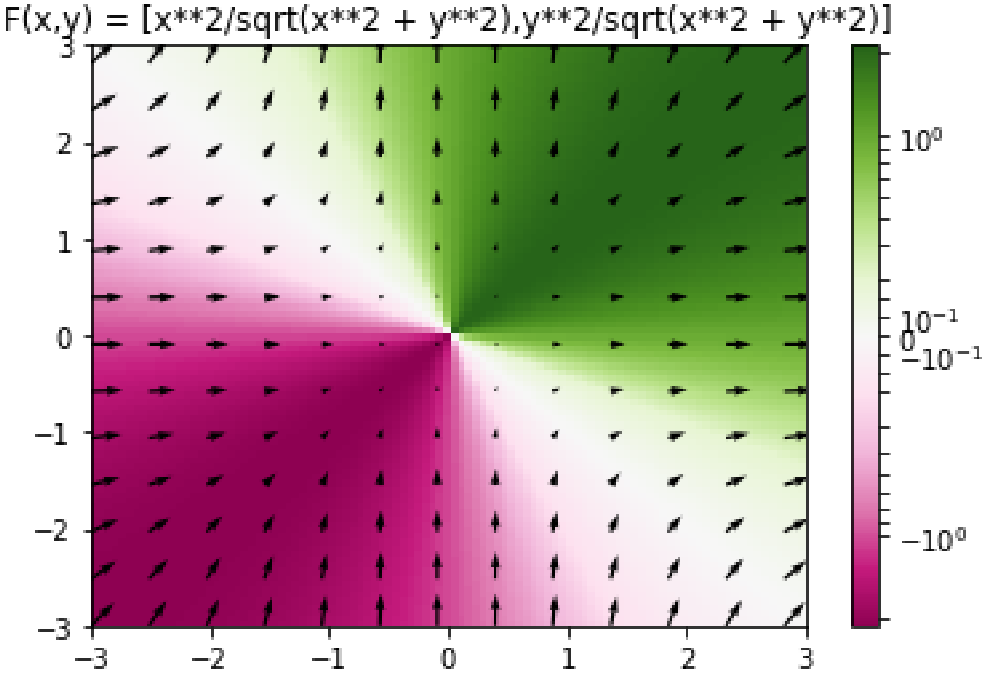


Figure div.4: png

vecs.curl Curl, line integrals, and circulation

Line integrals

Consider a curve C in a Euclidean vector space $\mathbb{R}^3$. Let $\mathbf{r}(t) = [x(t), y(t), z(t)]$ be a parametric representation of C. Furthermore, let $\mathbf{F} : \mathbb{R}^3 \to \mathbb{R}^3$ be a vector-valued function of $\mathbf{r}$ and let $\mathbf{r}'(t)$ be the tangent vector. The line integral is

line integral

$$\int_C \mathbf{F}(\mathbf{r}(t)) \cdot \mathbf{r}'(t)\, dt \tag{1}$$

which integrates $\mathbf{F}$ along the curve. For more on computing line integrals, see Schey[8] and Kreyszig.[9]
If $\mathbf{F}$ is a force being applied to an object moving along the curve C, the line integral is the work done by the force. More generally, the line integral integrates $\mathbf{F}$ along the tangent of C.

force
work

Circulation

Consider the line integral over a closed curve C, denoted by

$$\oint_C \mathbf{F}(\mathbf{r}(t)) \cdot \mathbf{r}'(t)\, dt. \tag{2}$$

We call this quantity the circulation of $\mathbf{F}$ around C.

circulation

For certain vector-valued functions $\mathbf{F}$, the circulation is zero for every curve. In these cases (static electric fields, for instance), this is sometimes called the the law of circulation.

the law of circulation

Curl

Consider the division of the circulation around a curve in $\mathbb{R}^3$ by the surface area it encloses ΔS,

$$\frac{1}{\Delta S} \oint_C \mathbf{F}(\mathbf{r}(t)) \cdot \mathbf{r}'(t)\, dt. \tag{3}$$

[8] Schey, Div, Grad, Curl, and All that: An Informal Text on Vector Calculus, pp. 63-74.

[9] Kreyszig, Advanced Engineering Mathematics, § 10.1, 10.2.

In a manner analogous to the operation that gaves us the divergence, let's consider shrinking this curve to a point and the surface area to zero,

$$\lim_{\Delta S \to 0} \frac{1}{\Delta S} \oint\limits_C \mathbf{F}(\mathbf{r}(t)) \cdot \mathbf{r}'(t)\, dt. \tag{4}$$

We call this quantity the "scalar" curl of $\mathbf{F}$ at each point in $\mathbb{R}^3$ in the direction normal to ΔS as it shrinks to zero. Taking three (or n for $\mathbb{R}^n$) "scalar" curls in indepedent normal directions (enough to span the vector space), we obtain the curl proper, which is a vector-valued function $\operatorname{curl} : \mathbb{R}^3 \to \mathbb{R}^3$.

curl

curl

The curl is coordinate-independent. In cartesian coordinates, it can be shown to be equivalent to the following.

Equation 5 curl: differential form, cartesian coordinates

$$\operatorname{curl} \mathbf{F} = \begin{bmatrix} \partial_y F_z - \partial_z F_y & \partial_z F_x - \partial_x F_z & \partial_x F_y - \partial_y F_x \end{bmatrix}^\top$$

But what does the curl of $\mathbf{F}$ represent? It quantifies the local rotation of $\mathbf{F}$ about each point. If $\mathbf{F}$ represents a fluid's velocity, $\operatorname{curl} \mathbf{F}$ is the local rotation of the fluid about each point and it is called the vorticity.

vorticity

Zero curl, circulation, and path independence

Circulation

It can be shown that if the circulation of $\mathbf{F}$ on all curves is zero, then in each direction $\mathbf{n}$ and at every point $\operatorname{curl} \mathbf{F} = 0$ (i.e. $\mathbf{n} \cdot \operatorname{curl} \mathbf{F} = 0$). Conversely, for $\operatorname{curl} \mathbf{F} = 0$ in a simply connected region[10], $\mathbf{F}$ has zero circulation.

Succinctly, informally, and without the requisite

[10] A region is simply connected if every curve in it can shrink to a point without leaving the region. An example of a region that is not simply connected is the surface of a toroid.

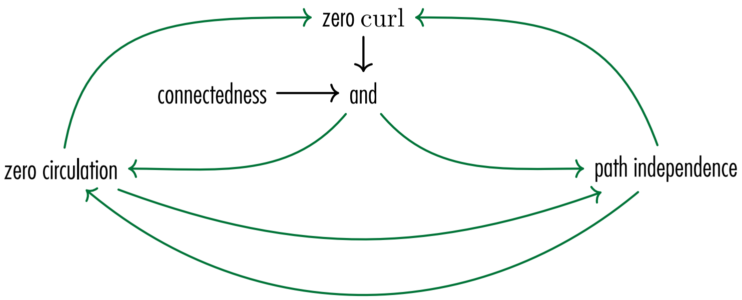


Figure curl.1: an implication graph relating zero curl, zero circulation, path independence, and connectedness. Green edges represent implication (a implies b) and black edges represent logical conjunctions.

qualifiers above,

$$\text{zero circulation} \Rightarrow \text{zero curl} \tag{6}$$

$$\text{zero curl } + \text{simply connected region} \Rightarrow \text{zero circulation.} \tag{7}$$

Path independence

It can be shown that if the path integral of $\mathbf{F}$ on all curves between any two points is path-independent, then in each direction $\mathbf{n}$ and at every point $\operatorname{curl}\mathbf{F} = 0$ (i.e. $\mathbf{n} \cdot \operatorname{curl}\mathbf{F} = 0$). Conversely, for $\operatorname{curl}\mathbf{F} = 0$ in a simply connected region, all line integrals are independent of path. Succinctly, informally, and without the requisite qualifiers above,

path independence

$$\text{path independence} \Rightarrow \text{zero curl} \tag{8}$$

$$\text{zero curl } + \text{simply connected region} \Rightarrow \text{path independence.} \tag{9}$$

… and how they relate

It is also true that

$$\text{path independence} \Leftrightarrow \text{zero circulation.} \tag{10}$$

So, putting it all together, we get Fig. curl.1.

Exploring curl

Curl is perhaps best explored by considering it for a vector field in $\mathbb{R}^2$. Such a field in cartesian coordinates $\mathbf{F} = F_x \hat{\mathbf{i}} + F_y \hat{\mathbf{j}}$ has curl

$$\begin{aligned}
\operatorname{curl} \mathbf{F} &= \begin{bmatrix} \partial_y 0 - \partial_z F_y & \partial_z F_x - \partial_x 0 & \partial_x F_y - \partial_y F_x \end{bmatrix}^\top \\
&= \begin{bmatrix} 0 - 0 & 0 - 0 & \partial_x F_y - \partial_y F_x \end{bmatrix}^\top \\
&= \begin{bmatrix} 0 & 0 & \partial_x F_y - \partial_y F_x \end{bmatrix}^\top .
\end{aligned} \tag{11}$$

That is, $\operatorname{curl} \mathbf{F} = (\partial_x F_y - \partial_y F_x)\hat{\mathbf{k}}$ and the only nonzero component is normal to the xy-plane. If we overlay a quiver plot of $\mathbf{F}$ over a "color density" plot representing the $\hat{\mathbf{k}}$-component of $\operatorname{curl} \mathbf{F}$, we can increase our intuition about the curl.

The following was generated from a Jupyter notebook with the following filename and kernel.

```
notebook filename: curl-and-line-integrals.ipynb
notebook kernel: python3
```

First, load some Python packages.

```python
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import LogLocator
from matplotlib.colors import *
from sympy.utilities.lambdify import lambdify
from IPython.display import display, Markdown, Latex
```

Now we define some symbolic variables and functions.

```python
var('x,y')
F_x = Function('F_x')(x,y)
F_y = Function('F_y')(x,y)
```

We use the same function defined in Lec. vecs.div, `quiver_plotter_2D`, to make several of these plots.

Let's inspect several cases.

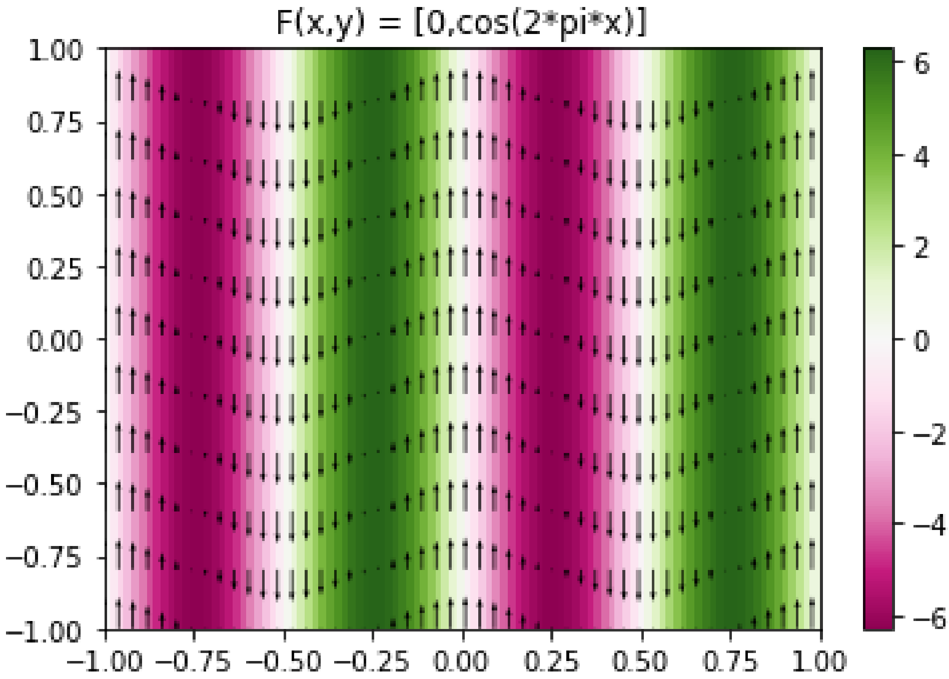


Figure curl.2: png

```
p = quiver_plotter_2D(
  field={F_x:0,F_y:cos(2*pi*x)},
  density_operation='curl',
  grid_decimate_x=2,
  grid_decimate_y=10,
  grid_width=1
)
```

```
The curl is:
```

$$-2\pi \sin\left(2\pi x\right)$$

```
p = quiver_plotter_2D(
  field={F_x:0,F_y:x**2},
  density_operation='curl',
  grid_decimate_x=2,
  grid_decimate_y=20,
)
```

```
The curl is:
```

$$2x$$

```
p = quiver_plotter_2D(
  field={F_x:y**2,F_y:x**2},
```

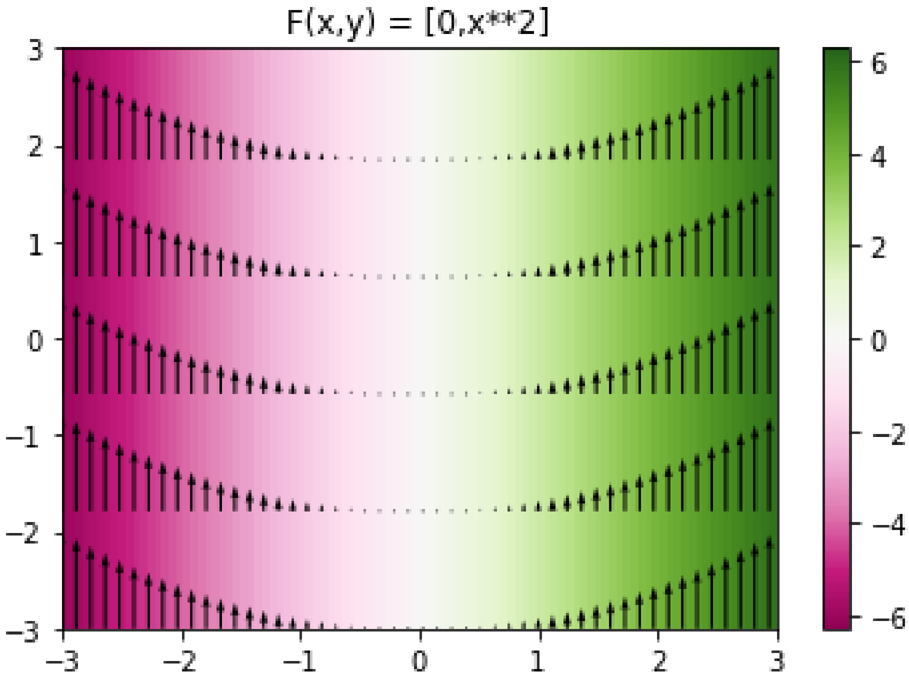


Figure curl.3: png

```
  density_operation='curl',
  grid_decimate_x=2,
  grid_decimate_y=20,
)
```

```
The curl is:
```

$$2x - 2y$$

```
p = quiver_plotter_2D(
  field={F_x:-y,F_y:x},
  density_operation='curl',
  grid_decimate_x=6,
  grid_decimate_y=6,
)
```

```
Warning: density operator is constant (no density
↪  plot)
The curl is:
```

$$2$$

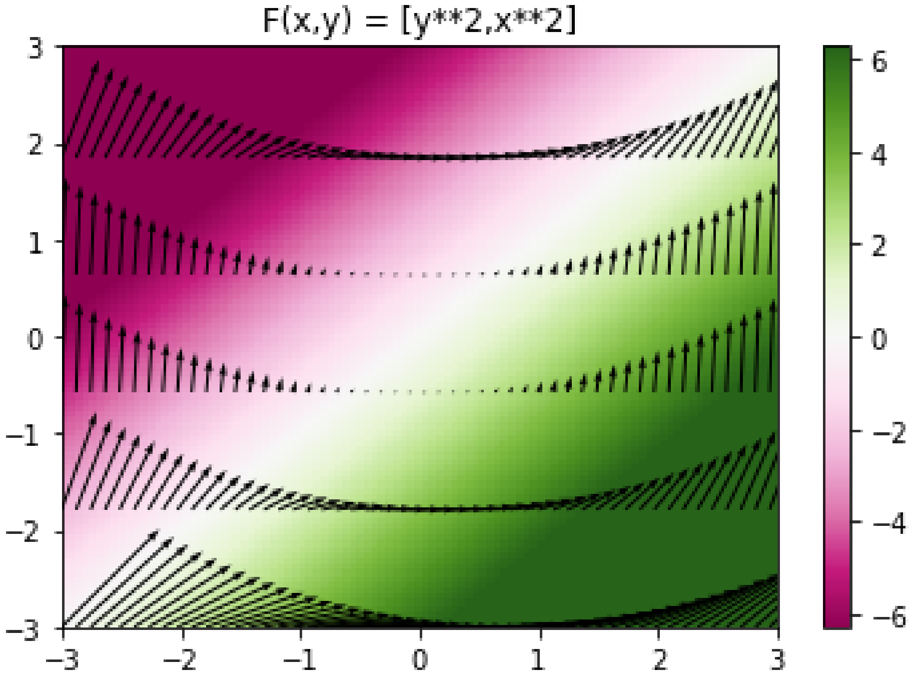


Figure curl.4: png

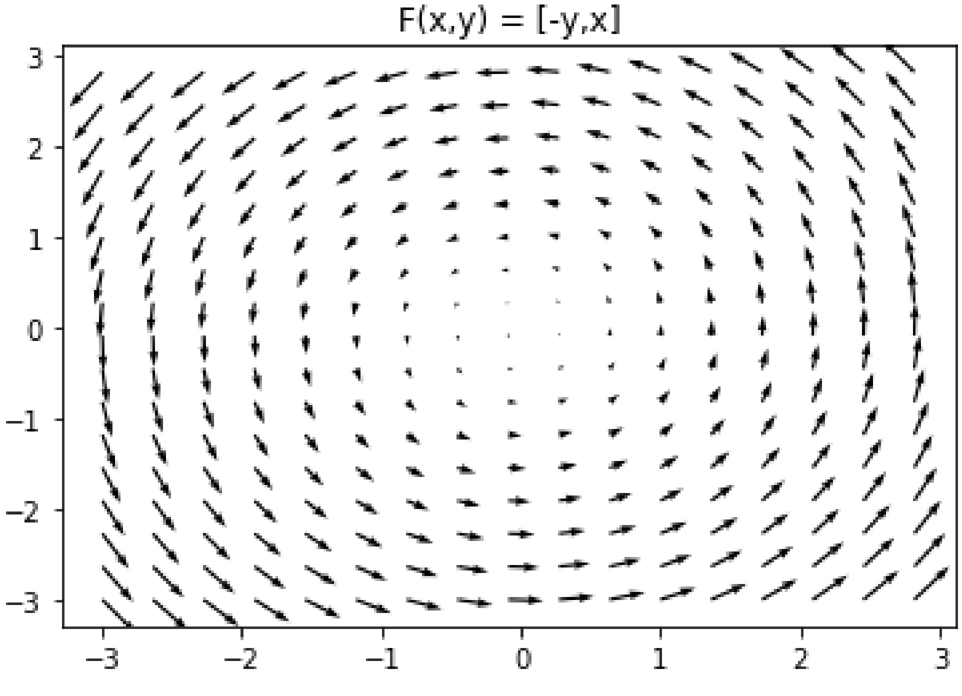


Figure curl.5: png

vecs.grad Gradient

Gradient

The gradient grad of a scalar-valued function $f : \mathbb{R}^3 \to \mathbb{R}$ is a vector field $\mathbf{F} : \mathbb{R}^3 \to \mathbb{R}^3$; that is, $\operatorname{grad} f$ is a vector-valued function on $\mathbb{R}^3$. The gradient's local direction and magnitude are those of the local maximum rate of increase of f. This makes it useful in optimization (e.g. in the method of gradient descent).

gradient

direction
magnitude

In classical mechanics, quantum mechanics, relativity, string theory, thermodynamics, and continuum mechanics (and elsewhere) the principle of least action is taken as fundamental (Richard P. Feynman, Robert B. Leighton and Matthew Sands. The Feynman Lectures on Physics. New Millennium. Perseus Basic Books, 2010). This principle tells us that nature's laws quite frequently seem to be derivable by assuming a certain quantity—called action—is minimized. Considering, then, that the gradient supplies us with a tool for optimizing functions, it is unsurprising that the gradient enters into the equations of motion of many physical quantities.

principle of least action

The gradient is coordinate-independent, but its coordinate-free definitions don't add much to our intuition. In cartesian coordinates, it can be shown to be equivalent to the following.

Equation 1 gradient: cartesian coordinates

$$\operatorname{grad} f = \begin{bmatrix} \partial_x f & \partial_y f & \partial_z f \end{bmatrix}^\top$$

Vector fields from gradients are special

Although all gradients are vector fields, not all vector fields are gradients. That is, given a vector field $\mathbf{F}$, it may or may not be equal to the

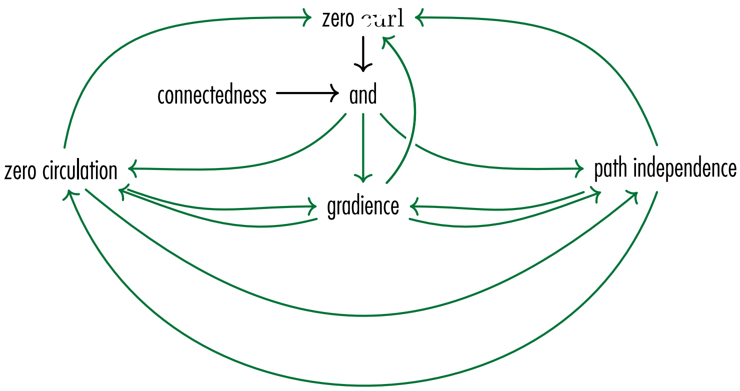


Figure grad.1: an implication graph relating gradience, zero curl, zero circulation, path independence, and connectedness. Green edges represent implication (a implies b) and black edges represent logical conjunctions.

gradient of any scalar-valued function f. Let's say of a vector field that is a gradient that it has gradience.[11] Those vector fields that are gradients have special properties. Surprisingly, those properties are connected to path independence and curl. It can be shown that iff a field is a gradient, line integrals of the field are path independent. That is, for a vector field,

gradience

$$\text{gradience} \Leftrightarrow \text{path independence}. \tag{2}$$

Considering what we know from Lec. vecs.curl about path independence we can expand Fig. curl.1 to obtain Fig. grad.1. One implication is that gradients have zero curl! Many important fields that describe physical interactions (e.g. static electric fields, Newtonian gravitational fields) are gradients of scalar fields called potentials.

potentials

Exploring gradient

Gradient is perhaps best explored by considering it for a scalar field on $\mathbb{R}^2$. Such a field in cartesian coordinates $f(x, y)$ has gradient

$$\operatorname{grad} f = \begin{bmatrix} \partial_x f & \partial_y f \end{bmatrix}^\top \tag{3}$$

[11] This is nonstandard terminology, but we're bold.

That is, $\operatorname{grad} f = \mathbf{F} = \partial_x f\,\hat{\mathbf{i}} + \partial_y f\,\hat{\mathbf{j}}$. If we overlay a quiver plot of **F** over a "color density" plot representing the f, we can increase our intuition about the gradient.

The following was generated from a Jupyter notebook with the following filename and kernel.

```
notebook filename: grad.ipynb
notebook kernel: python3
```

First, load some Python packages.

```
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import LogLocator
from matplotlib.colors import *
from sympy.utilities.lambdify import lambdify
from IPython.display import display, Markdown, Latex
```

Now we define some symbolic variables and functions.

```
var('x,y')
```

```
(x, y)
```

Rather than repeat code, let's write a single function `grad_plotter_2D` to make several of these plots.

Let's inspect several cases. While considering the following plots, remember that they all have zero curl!

```
p = grad_plotter_2D(
  field=x,
)
```

```
The gradient is:
```

$$\begin{bmatrix}1 & 0\end{bmatrix}$$

```
p = grad_plotter_2D(
  field=x+y,
)
```

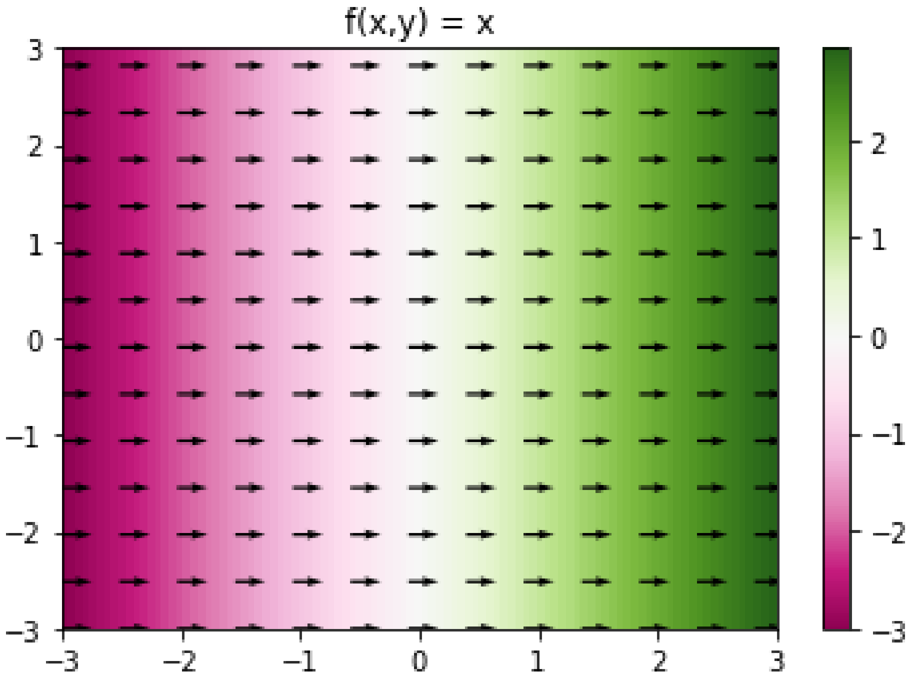


Figure grad.2: png

```
The gradient is:
```

$$\begin{bmatrix}1 & 1\end{bmatrix}$$

```
p = grad_plotter_2D(
  field=1,
)
```

```
Warning: field is constant (no plot)
The gradient is:
```

$$\begin{bmatrix}0 & 0\end{bmatrix}$$

Gravitational potential

Gravitational potentials have the form of 1/distance. Let’s check out the gradient.

```
p = grad_plotter_2D(
  field=1/sqrt(x**2+y**2),
  norm=SymLogNorm(linthresh=.3, linscale=.3),
  mask=True,
)
```

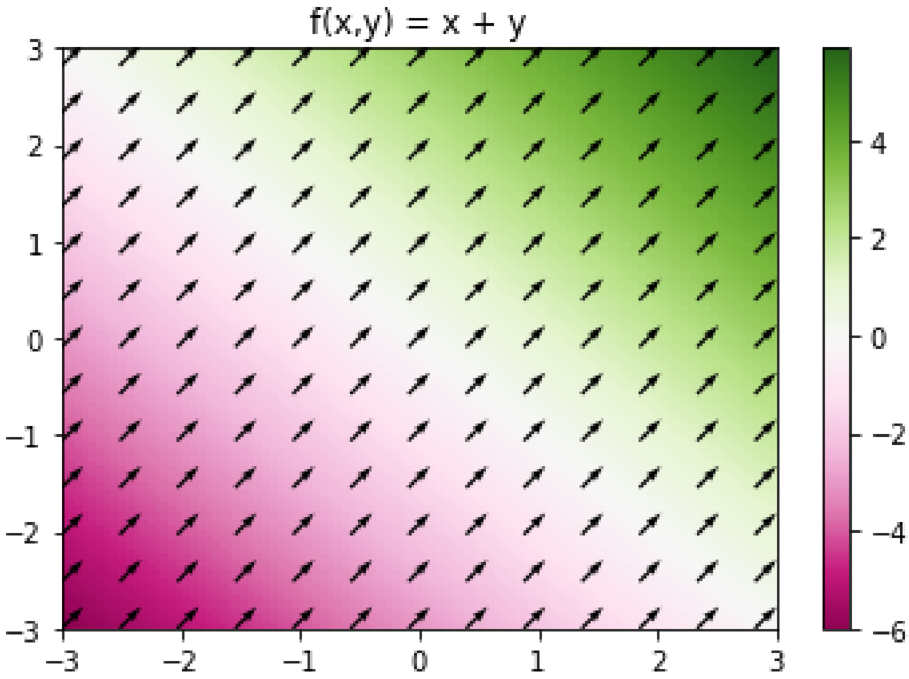


Figure grad.3: png

```
The gradient is:
```

$$\left[\begin{matrix}- \frac{x}{\left(x^{2} + y^{2}\right)^{\frac{3}{2}}} & - \frac{y}{\left(x^{2} + y^{2}\right)^{\frac{3}{2}}}\end{matrix}\right]$$

Conic section fields

Gradients of conic section fields can be explored. The following is called a parabolic field.

conic section

parabolic fields

```
p = grad_plotter_2D(
  field=x**2,
)
```

```
The gradient is:
```

$$\left[\begin{matrix}2 x & 0\end{matrix}\right]$$

The following are called eliptic fields.

eliptic fields

```
p = grad_plotter_2D(
  field=x**2+y**2,
)
```

```
The gradient is:
```

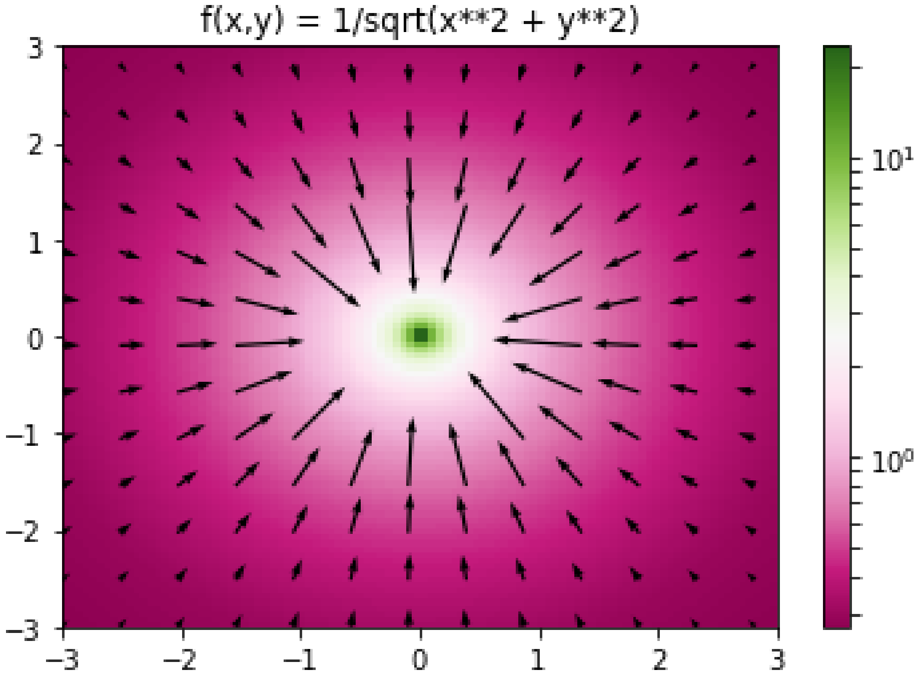


Figure grad.4: png

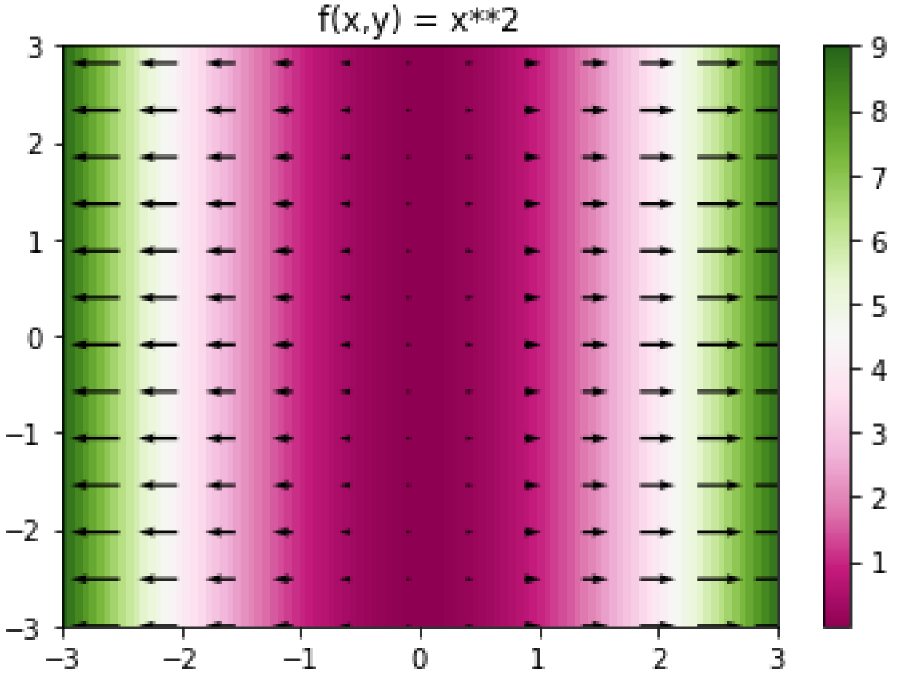


Figure grad.5: png

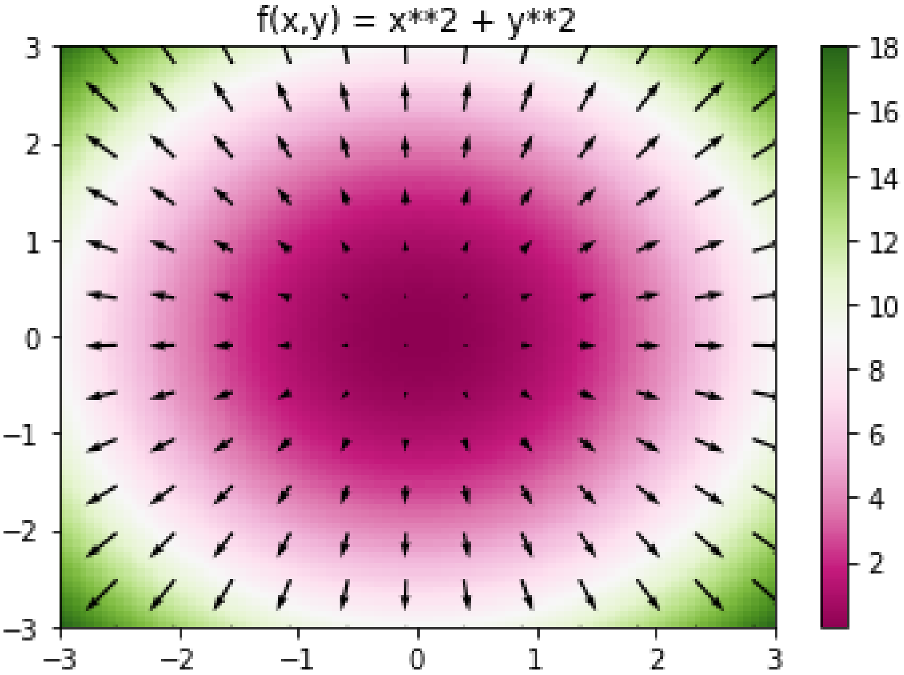


Figure grad.6: png

$$\begin{bmatrix} 2x & 2y \end{bmatrix}$$

```
p = grad_plotter_2D(
  field=-x**2-y**2,
)
```

```
The gradient is:
```

$$\begin{bmatrix} -2x & -2y \end{bmatrix}$$

The following is called a hyperbolic field.

hyperbolic fields

```
p = grad_plotter_2D(
  field=x**2-y**2,
)
```

```
The gradient is:
```

$$\begin{bmatrix} 2x & -2y \end{bmatrix}$$

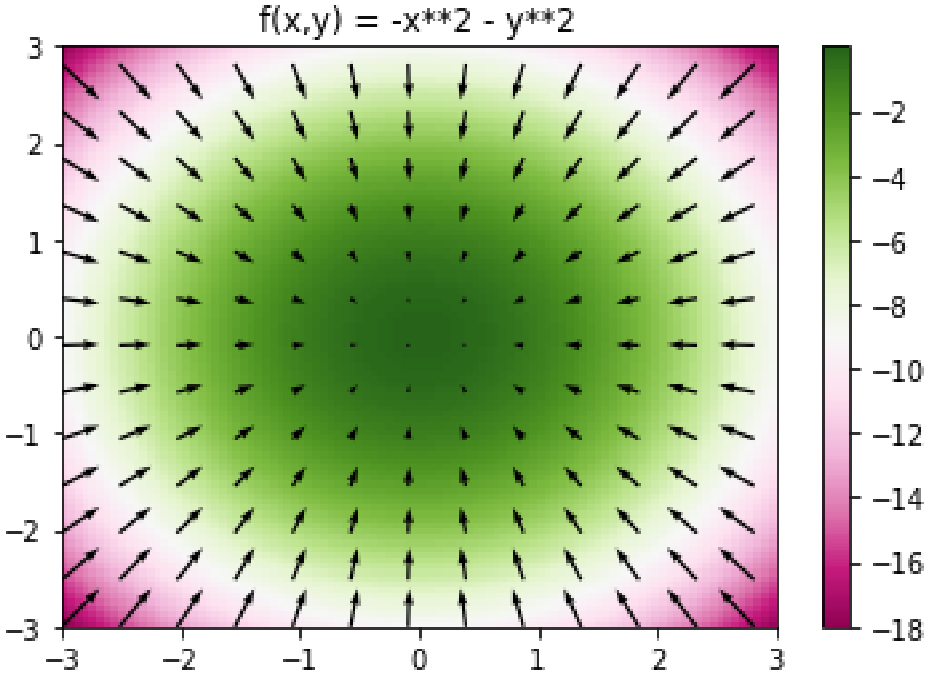


Figure grad.7: png

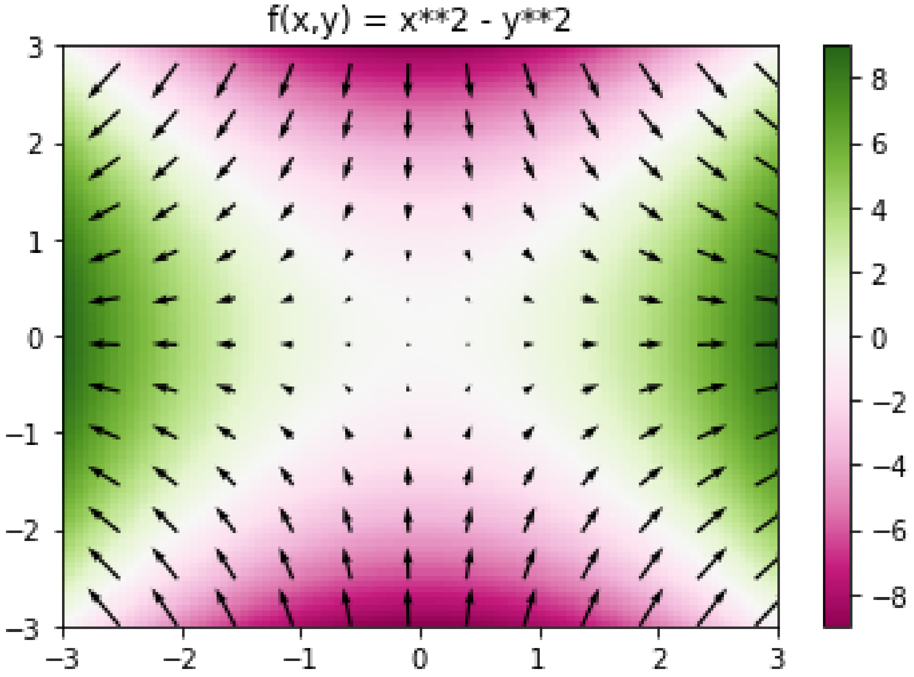


Figure grad.8: png

vecs.stoked Stokes and divergence theorems

Two theorems allow us to exchange certain integrals in $\mathbb{R}^3$ for others that are easier to evaluate.

The divergence theorem

divergence theorem

triple integral

The divergence theorem asserts the equality of the surface integral of a vector field $\mathbf{F}$ and the triple integral of $\operatorname{div}\mathbf{F}$ over the volume V enclosed by the surface S in $\mathbb{R}^3$. That is,

$$\iint_S \mathbf{F}\cdot\mathbf{n}\,dS = \iiint_V \operatorname{div}\mathbf{F}\,dV. \tag{1}$$

orientable

Caveats are that V is a closed region bounded by the orientable[12] surface S and that $\mathbf{F}$ is continuous and continuously differentiable over a region containing V. This theorem makes some intuitive sense: we can think of the divergence inside the volume "accumulating" via the triple integration and equaling the corresponding surface integral. For more on the divergence theorem, see Kreyszig[13] and Schey.[14] A lovely application of the divergence theorem is that, for any quantity of conserved stuff (mass, charge, spin, etc.) distributed in a spatial $\mathbb{R}^3$ with time-dependent density $\rho : \mathbb{R}^4 \to \mathbb{R}$ and velocity field $\boldsymbol{v} : \mathbb{R}^4 \to \mathbb{R}^3$, the divergence theorem can be applied to find that

$$\partial_t \rho = -\operatorname{div}(\rho\boldsymbol{v}), \tag{2}$$

continuity equation

which is a more general form of a continuity equation, one of the governing equations of many physical phenomena. For a derivation of this equation, see Schey.[15]

The Kelvin-Stokes' theorem

Kelvin–Stokes' theorem

The Kelvin-Stokes' theorem asserts the equality of the circulation of a vector field $\mathbf{F}$ over a closed curve C and the surface integral of $\operatorname{curl}\mathbf{F}$ over a

[12] A surface is orientable if a consistent normal direction can be defined. Most surfaces are orientable, but some, notably the Möbius strip, cannot be. See Kreyszig (Kreyszig, Advanced Engineering Mathematics, § 10.6) for more.

[13] ibidem, § 10.7.

[14] Schey, Div, Grad, Curl, and All that: An Informal Text on Vector Calculus, pp. 45-52.

[15] ibidem, pp. 49-52.

surface S that has boundary C. That is, for $\mathbf{r}(t)$ a parameterization of C and surface normal $\mathbf{n}$,

$$\oint_C \mathbf{F}(\mathbf{r}(t)) \cdot \mathbf{r}'(t)\, dt = \iint_S \mathbf{n} \cdot \operatorname{curl} \mathbf{F}\, dS. \tag{3}$$

Caveats are that S is piecewise smooth,[16] its boundary C is a piecewise smooth simple closed curve, and $\mathbf{F}$ is continuous and continuously differentiable over a region containing S. This theorem is also somewhat intuitive: we can think of the divergence over the surface "accumulating" via the surface integration and equaling the corresponding circulation. For more on the Kelvin-Stokes' theorem, see Kreyszig[17] and Schey.[18]

piecewise smooth

Related theorems

Greene's theorem is a two-dimensional special case of the Kelvin-Stokes' theorem. It is described by Kreyszig.[19]

Greene's theorem

It turns out that all of the above theorems (and the fundamental theorem of calculus, which relates the derivative and integral) are special cases of the generalized Stokes' theorem defined by differential geometry. We would need a deeper understanding of differential geometry to understand this theorem. For more, see Lee.[20]

generalized Stokes' theorem

[16] A surface is smooth if its normal is continuous everywhere. It is piecewise smooth if it is composed of a finite number of smooth surfaces.

[17] Kreyszig, Advanced Engineering Mathematics, § 10.9.

[18] Schey, Div, Grad, Curl, and All that: An Informal Text on Vector Calculus, pp. 93-102.

[19] Kreyszig, Advanced Engineering Mathematics, § 10.9.

[20] Lee, Introduction to Smooth Manifolds, Ch. 16.

vecs.exe Exercises for Chapter vecs

Exercise vecs.light

Consider a vector field $\mathsf{F} : \mathbb{R}^3 \to \mathbb{R}^3$ defined in Cartesian coordinates (x, y, z) as

______/20 p.

$$\mathsf{F} = [x^2 - y^2, y^2 - z^2, z^2 - x^2]. \tag{1}$$

a. Compute the divergence of F.
b. Compute the curl of F.
c. Prove that, in a simply connected region of $\mathbf{R}^3$, line integrals of F are path-dependent.
d. Prove that F is not the gradient of a potential (scalar) function (i.e. that it does not have gradience, as we've called it).

four

Fourier and orthogonality

four.series Fourier series

1 Fourier series are mathematical series that can represent a periodic signal as a sum of sinusoids at different amplitudes and frequencies. They are useful for solving for the response of a system to periodic inputs. However, they are probably most important conceptually: they are our gateway to thinking of signals in the frequency domain—that is, as functions of frequency (not time). To represent a function as a Fourier series is to analyze it as a sum of sinusoids at different frequencies[1] ω_n and amplitudes a_n. Its frequency spectrum is the functional representation of amplitudes a_n versus frequency ω_n.

frequency domain

Fourier analysis

frequency spectrum

2 Let's begin with the definition.

Definition four.1: Fourier series: trigonometric form

The Fourier analysis of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$, period T, and angular frequency $\omega_n = 2\pi n/T$,

$$a_n = \frac{2}{T}\int_{-T/2}^{T/2} y(t)\cos(\omega_n t)dt \tag{1}$$

$$b_n = \frac{2}{T}\int_{-T/2}^{T/2} y(t)\sin(\omega_n t)dt. \tag{2}$$

The Fourier synthesis of a periodic function $y(t)$ with analysis components a_n and b_n corresponding to ω_n is

$$y(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n\cos(\omega_n t) + b_n\sin(\omega_n t). \tag{3}$$

3 Let's consider the complex form of the Fourier series, which is analogous to Definition four.1. It may be helpful to review Euler's formula(s) – see Appendix D.01.

[1] It's important to note that the symbol ω_n, in this context, is not the natural frequency, but a frequency indexed by integer n.

Definition four.2: Fourier series: complex form

The Fourier analysis of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$, period T, and angular frequency $\omega_n = 2\pi n/T$,

$$c_{\pm n} = \frac{1}{T}\int_{-T/2}^{T/2} y(t) e^{-j\omega_n t} dt. \tag{4}$$

The Fourier synthesis of a periodic function $y(t)$ with analysis components c_n corresponding to ω_n is

$$y(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\omega_n t}. \tag{5}$$

4 We call the integer n a harmonic and the frequency associated with it,

harmonic

$$\omega_n = 2\pi n/T, \tag{6}$$

the harmonic frequency. There is a special name for the first harmonic ($n = 1$): the fundamental frequency. It is called this because all other frequency components are integer multiples of it.

harmonic frequency

fundamental frequency

5 It is also possible to convert between the two representations above.

Definition four.3: Fourier series: converting between forms

The complex Fourier analysis of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$ and a_n and b_n as defined above,

$$c_{\pm n} = \frac{1}{2}\left(a_{|n|} \mp j b_{|n|}\right) \tag{7}$$

The sinusoidal Fourier analysis of a periodic function $y(t)$ is, for $n \in \mathbb{N}_0$ and c_n as defined above,

$$a_n = c_n + c_{-n} \text{ and} \tag{8}$$

$$b_n = j\left(c_n - c_{-n}\right). \tag{9}$$

6 The harmonic amplitude C_n is

harmonic amplitude

$$C_n = \sqrt{a_n^2 + b_n^2} \tag{10}$$

$$= 2\sqrt{c_n c_{-n}}. \tag{11}$$

A magnitude line spectrum is a graph of the harmonic amplitudes as a function of the harmonic frequencies. The harmonic phase is

magnitude line spectrum

harmonic phase

$$\theta_n = -\arctan_2(b_n, a_n) \quad \text{(see Appendix C.02)}$$

$$= \arctan_2(\operatorname{Im}(c_n), \operatorname{Re}(c_n)). \tag{12}$$

7 The illustration of Fig. series.1 shows how sinusoidal components sum to represent a square wave. A line spectrum is also shown.

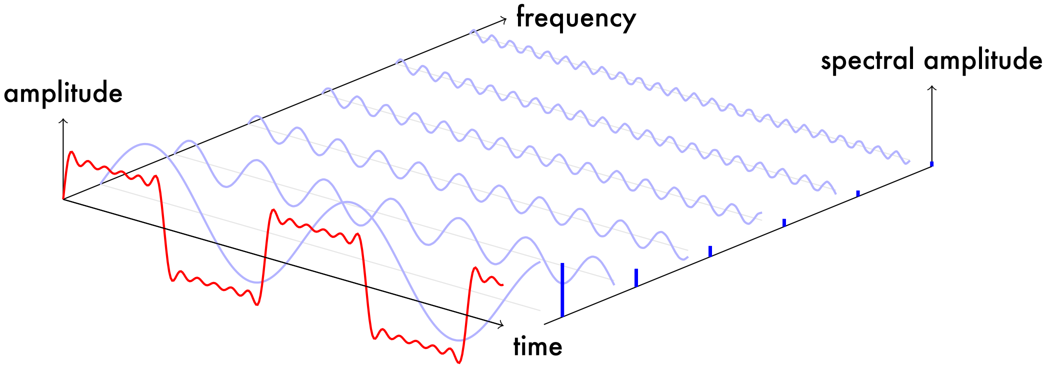


Figure series.1: a partial sum of Fourier components of a square wave shown through time and frequency. The spectral amplitude shows the amplitude of the corresponding Fourier component.

8 Let us compute the associated spectral components in the following example.

Example four.series–1 re: Fourier series analysis: line spectrum

Compute the first five harmonic amplitudes that represent the line spectrum for a square wave in the figure above.

four.trans Fourier transform

We begin with the usual loading of modules.

```
import numpy as np # for numerics
import sympy as sp # for symbolics
import matplotlib.pyplot as plt # for plots!
from IPython.display import display, Markdown, Latex
```

Let's consider a periodic function f with period T (`T`). Each period, the function has a triangular pulse of width δ (`pulse_width`) and height $\delta/2$.

```
period = 15 # period
pulse_width = 2 # pulse width
```

First, we plot the function f in the time domain. Let's begin by defining f.

```
def pulse_train(t,T,pulse_width):
  f = lambda x:pulse_width/2-abs(x) # pulse
  tm = np.mod(t,T)
  if tm <= pulse_width/2:
    return f(tm)
  elif tm >= T-pulse_width/2:
    return f(-(tm-T))
  else:
    return 0
```

Now, we develop a numerical array in time to plot f.

```
N = 201 # number of points to plot
tpp = np.linspace(-period/2,5*period/2,N) # time values
fpp = np.array(np.zeros(tpp.shape))
for i,t_now in enumerate(tpp):
  fpp[i] = pulse_train(t_now,period,pulse_width)
```

```
p = plt.figure(1)
plt.plot(tpp,fpp,'b-',linewidth=2) # plot
plt.xlabel('time (s)')
plt.xlim([-period/2,3*period/2])
plt.xticks(
  [0,period],
  [0,'$T='+str(period)+'$ s']
)
plt.yticks([0,pulse_width/2],['0','$\delta/2$'])
plt.show() # display here
```

[1] Python code in this section was generated from a Jupyter notebook named `fourier_series_to_transform.ipynb` with a `python3` kernel.

For $\delta = 2$ and $T \in [5, 15, 25]$, the left-hand column of Fig. trans.1 shows two triangle pulses for each period T.

Consider the following argument. Just as a Fourier series is a frequency domain representation of a periodic signal, a Fourier transform is a frequency domain representation of an aperiodic signal (we will rigorously define it in a moment). The Fourier series components will have an analog, then, in the Fourier transform. Recall that they can be computed by integrating over a period of the signal. If we increase that period infinitely, the function is effectively aperiodic. The result (within a scaling factor) will be the Fourier transform analog of the Fourier series components.

Let us approach this understanding by actually computing the Fourier series components for increasing period T using ??. We'll use sympy to compute the Fourier series cosine and sine components a_n and b_n for component n (n) and period T (T).

```
sp.var('x,a_0,a_n,b_n',real=True)
sp.var('delta,T',positive=True)
sp.var('n',nonnegative=True)
# a0 = 2/T*sp.integrate(
#   (delta/2-sp.Abs(x)),
#   (x,-delta/2,delta/2) # otherwise zero
# ).simplify()
an = sp.integrate(
  2/T*(delta/2-sp.Abs(x))*sp.cos(2*sp.pi*n/T*x),
  (x,-delta/2,delta/2) # otherwise zero
).simplify()
bn = 2/T*sp.integrate(
  (delta/2-sp.Abs(x))*sp.sin(2*sp.pi*n/T*x),
  (x,-delta/2,delta/2) # otherwise zero
).simplify()
display(sp.Eq(a_n,an),sp.Eq(b_n,bn))
```

$$a_n = \begin{cases} \frac{T\left(1-\cos\left(\frac{\pi\delta n}{T}\right)\right)}{\pi^2 n^2} & \text{for } n \neq 0 \\ \frac{\delta^2}{2T} & \text{otherwise} \end{cases}$$

$$b_n = 0$$

Furthermore, let us compute the harmonic amplitude (`f_harmonic_amplitude`):

$$C_n = \sqrt{a_n^2 + b_n^2} \tag{1}$$

which we have also scaled by a factor T/δ in order to plot it with a convenient scale.

```
sp.var('C_n',positive=True)
cn = sp.sqrt(an**2+bn**2)
display(sp.Eq(C_n,cn))
```

$$C_n = \begin{cases} \frac{T\left|\cos\left(\frac{\pi\delta n}{T}\right)-1\right|}{\pi^2 n^2} & \text{for } n \neq 0 \\ \frac{\delta^2}{2T} & \text{otherwise} \end{cases}$$

Now we lambdify the symbolic expression for a `numpy` function.

```
cn_f = sp.lambdify((n,T,delta),cn)
```

Now we can plot.

```
omega_max = 12 # rad/s max frequency in line spectrum
n_max = round(omega_max*period/(2*np.pi)) # max harmonic
n_a = np.linspace(0,n_max,n_max+1)
omega = 2*np.pi*n_a/period
p = plt.figure(2)
markerline, stemlines, baseline = plt.stem(
    omega, period/pulse_width*cn_f(n_a,period,pulse_width),
    linefmt='b-', markerfmt='bo', basefmt='r-',
    use_line_collection=True,
)
plt.xlabel('frequency $\omega$ (rad/s)')
plt.xlim([0,omega_max])
plt.ylim([0,pulse_width/2])
plt.yticks([0,pulse_width/2],['0','$\delta/2$'])
plt.show() # show here
```

The line spectra are shown in the right-hand column of Fig. trans.1. Note that with our chosen scaling, as T increases, the line spectra reveal a distinct waveform.

Let F be the continuous function of angular frequency ω

$$F(\omega) = \frac{\delta}{2} \cdot \frac{\sin^2(\omega\delta/4)}{(\omega\delta/4)^2}. \tag{2}$$

First, we plot it.

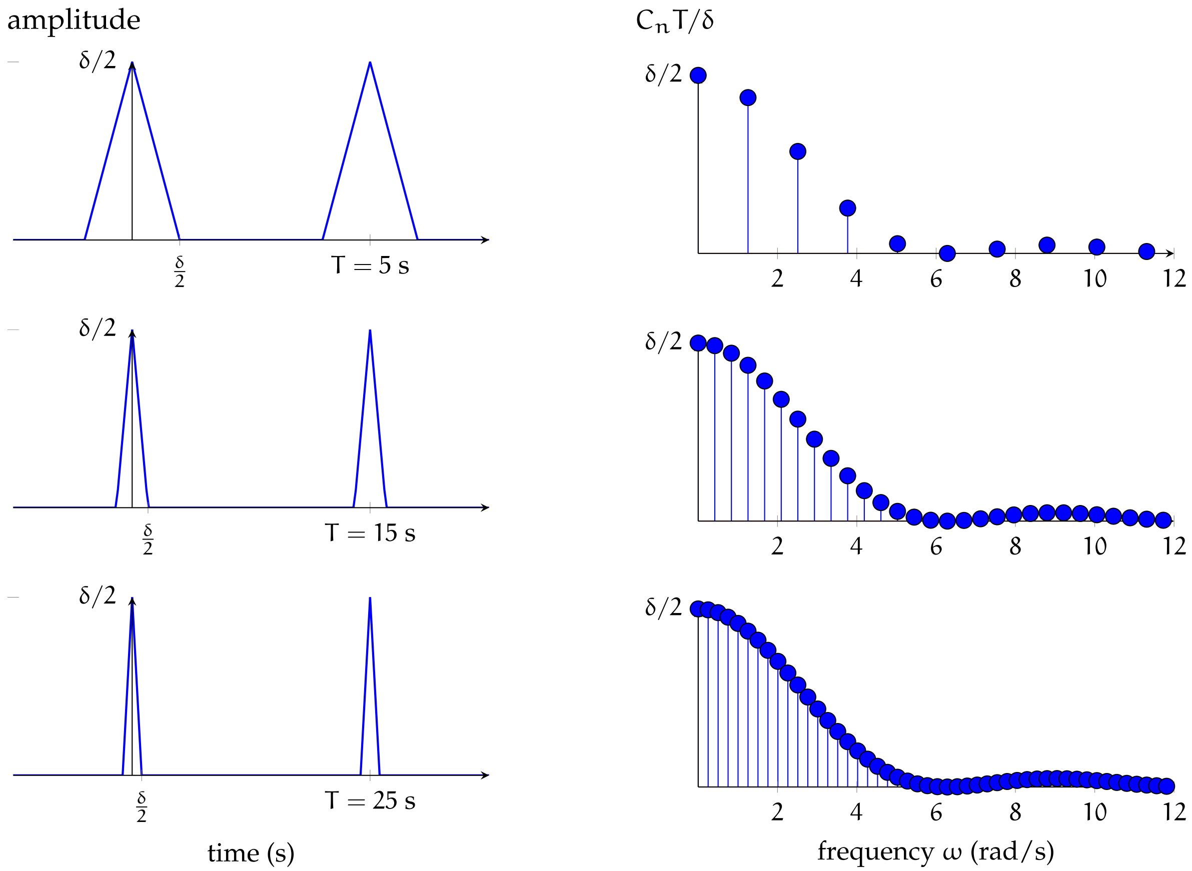


Figure trans.1: triangle pulse trains (left column) with longer periods, descending, and their corresponding line spectra (right column), scaled for convenient comparison.

```
F = lambda w: pulse_width/2* \
    np.sin(w*pulse_width/(2*2))**2/ \
    (w*pulse_width/(2*2))**2
N = 201 # number of points to plot
wpp = np.linspace(0.0001,omega_max,N)
Fpp = []
for i in range(0,N):
    Fpp.append(F(wpp[i])) # build array of function value
axes = plt.figure(3)
plt.plot(wpp,Fpp,'b-',linewidth=2) # plot
plt.xlim([0,omega_max])
plt.yticks([0,pulse_width/2],['0','$\delta/2$'])
plt.xlabel('frequency $\omega$ (rad/s)')
plt.ylabel('$F(\omega)$')
plt.show()
```

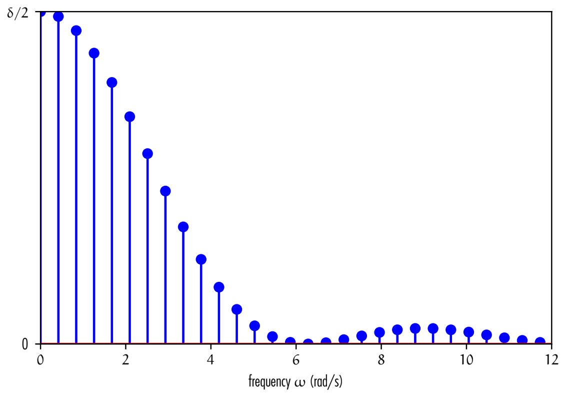


Figure trans.2: $F(\omega)$, our mysterious Fourier series amplitude analog.

Let's consider the plot in Fig. trans.2 of F. It's obviously the function emerging in Fig. trans.1

from increasing the period of our pulse train. Now we are ready to define the Fourier transform and its inverse.

Definition four.4: Fourier transforms: trigonometric form

Fourier transform (analysis):

$$A(\omega) = \int_{-\infty}^{\infty} y(t)\cos(\omega t)dt \tag{3}$$

$$B(\omega) = \int_{-\infty}^{\infty} y(t)\sin(\omega t)dt. \tag{4}$$

Inverse Fourier transform (synthesis):

$$y(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} A(\omega)\cos(\omega t)d\omega + \frac{1}{2\pi}\int_{-\infty}^{\infty} B(\omega)\sin(\omega t)d\omega. \tag{5}$$

Definition four.5: Fourier transforms: complex form

Fourier transform $\mathcal{F}$ (analysis):

$$\mathcal{F}(y(t)) = Y(\omega) = \int_{-\infty}^{\infty} y(t)e^{-j\omega t}dt. \tag{6}$$

Inverse Fourier transform $\mathcal{F}^{-1}$ (synthesis):

$$\mathcal{F}^{-1}(Y(\omega)) = y(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} Y(\omega)e^{j\omega t}d\omega. \tag{7}$$

So now we have defined the Fourier transform. There are many applications, including solving differential equations and frequency domain representations—called spectra—of time domain functions.

There is a striking similarity between the Fourier transform and the Laplace transform, with which you are already acquainted. In fact, the Fourier transform is a special case of a Laplace transform with Laplace transform variable $s = j\omega$ instead of having some real component. Both transforms convert differential equations to algebraic equations, which can be solved and inversely transformed to find time-domain solutions. The Laplace transform is especially important to use when an input function to a differential equation is not absolutely integrable and the Fourier transform is undefined (for example, our definition will yield a transform for neither the unit step nor the unit ramp functions). However, the Laplace transform is also preferred for initial value problems due to its convenient way of handling them. The two transforms are equally useful for solving steady state problems. Although the Laplace transform has many advantages, for spectral considerations, the Fourier transform is the only game in town.

A table of Fourier transforms and their properties can be found in Appendix B.02.

Example four.trans−1 re: a Fourier transform

Consider the aperiodic signal $y(t) = u_s(t)e^{-at}$ with u_s the unit step function and $a > 0$. The signal is plotted below. Derive the complex frequency spectrum and plot its magnitude and phase.

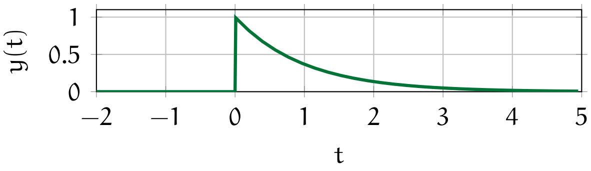


Figure trans.3: an aperiodic signal.

The signal is aperiodic, so the Fourier transform can be computed from Eq. 6:

$$
\begin{aligned}
Y(\omega) &= \int_{-\infty}^{\infty} y(t)e^{j\omega t}dt \\
&= \int_{-\infty}^{\infty} u_s(t)e^{-at}e^{j\omega t}dt && \text{(def. of y)} \\
&= \int_{0}^{\infty} e^{-at}e^{j\omega t}dt && \text{(u_s effect)} \\
&= \int_{0}^{\infty} e^{(-a+j\omega)t}dt && \text{(multiply)} \\
&= \frac{1}{-a+j\omega}e^{(-a+j\omega)t}\Big|_0^{\infty} dt \\
& && \text{(antiderivative)} \\
&= \frac{1}{-a+j\omega}\left(\lim_{t\to\infty} e^{(-a+j\omega)t} - e^0\right) \\
& && \text{(evaluate)}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{-a+j\omega}\left(\lim_{t\to\infty} e^{-at}e^{j\omega t} - 1\right) && \text{(arrange)}\\
&= \frac{1}{-a+j\omega}\left((0)(\text{complex with mag} \leqslant 1) - 1\right) && \text{(limit)}\\
&= \frac{-1}{-a+j\omega} && \text{(consequence)}\\
&= \frac{1}{a-j\omega}\\
&= \frac{a+j\omega}{a+j\omega}\cdot\frac{1}{a-j\omega} && \text{(rationalize)}\\
&= \frac{a+j\omega}{a^2+\omega^2}.
\end{aligned}
$$

The magnitude and phase of this complex function are straightforward to compute:

$$
\begin{aligned}
|Y(\omega)| &= \sqrt{\mathrm{Re}(Y(\omega))^2 + \mathrm{Im}(Y(\omega))^2}\\
&= \frac{1}{a^2+\omega^2}\sqrt{a^2+\omega^2}\\
&= \frac{1}{\sqrt{a^2+\omega^2}}\\
\angle Y(\omega) &= \arctan(\omega/a).
\end{aligned}
$$

Now we can plot these functions of ω. Setting $a = 1$ (arbitrarily), we obtain the plots of Fig. trans.4.

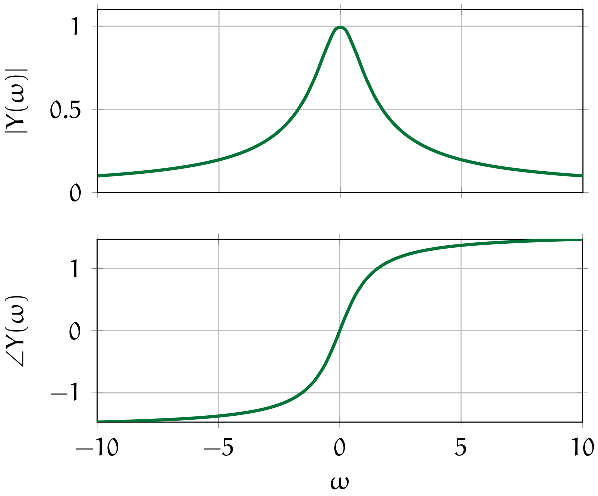


Figure trans.4: the magnitude and phase of the Fourier transform.

four.general Generalized fourier series and orthogonality

Let $f : \mathbb{R} \to \mathbb{C}$, $g : \mathbb{R} \to \mathbb{C}$, and $w : \mathbb{R} \to \mathbb{C}$ be complex functions. For square-integrable[2] f, g, and w, the inner product of f and g with weight function w over the interval $[a, b] \subseteq \mathbb{R}$ is[3]

inner product

weight function

$$\langle f, g \rangle_w = \int_a^b f(x)\overline{g}(x)w(x)\,dx \tag{1}$$

where $\overline{g}$ denotes the complex conjugate of g. The inner product of functions can be considered analogous to the inner (or dot) product of vectors.

The fourier series components can be found by a special property of the sin and cos functions called orthogonality. In general, functions f and g from above are orthogonal over the interval $[a, b]$ iff

orthogonality

$$\langle f, g \rangle_w = 0 \tag{2}$$

for weight function w. Similar to how a set of orthogonal vectors can be a basis for a vector space, a set of orthogonal functions can be a basis for a function space: a vector space of functions from one set to another (with certain caveats).

basis

function space

In addition to some sets of sinusoids, there are several other important sets of functions that are orthogonal. For instance, sets of legendre polynomials (Erwin Kreyszig. Advanced Engineering Mathematics. 10th. John Wiley & Sons, Limited, 2011. ISBN: 9781119571094. The authoritative resource for engineering mathematics. It includes detailed accounts of probability, statistics, vector calculus, linear algebra, fourier analysis, ordinary and partial differential equations, and complex analysis. It also includes several other topics with varying degrees of depth. Overall, it is the best place to start when seeking mathematical guidance.

legendre polynomials

[2] A function f is square-integrable if $\int_{-\infty}^{\infty} |f(x)|^2\,dx < \infty$.

[3] This definition of the inner product can be extended to functions on $\mathbb{R}^2$ and $\mathbb{R}^3$ domains using double- and triple-integration. See (Schey, Div, Grad, Curl, and All that: An Informal Text on Vector Calculus, p. 261).

§ 5.2) and bessel functions (Kreyszig, Advanced Engineering Mathematics, § 5.4) are orthogonal. As with sinusoids, the orthogonality of some sets of functions allows us to compute their series components. Let functions $f_0, f_1, \cdots$ be orthogonal with respect to weight function w on interval $[a, b]$ and let $\alpha_0, \alpha_1, \cdots$ be real constants. A generalized fourier series is (ibidem, § 11.6)

bessel functions

generalized fourier series

$$f(x) = \sum_{m=0}^{\infty} \alpha_m f_m(x) \tag{3}$$

and represents a function f as a convergent series. It can be shown that the Fourier components α_m can be computed from

Fourier components

$$\alpha_m = \frac{\langle f, f_m \rangle_w}{\langle f_m, f_m \rangle_w}. \tag{4}$$

In keeping with our previous terminology for fourier series, Eq. 3 and Eq. 4 are called general fourier synthesis and analysis, respectively. For the aforementioned legendre and bessel functions, the generalized fourier series are called fourier-legendre and fourier-bessel series (ibidem, § 11.6). These and the standard fourier series (Lec. four.series) are of particular interest for the solution of partial differential equations (Chapter pde).

synthesis

analysis

fourier−legendre series

fourier−bessel series

four.exe Exercises for Chapter four

Exercise four.stanislaw

Explain, in your own words (supplementary drawings are ok), what the frequency domain is, how we derive models in it, and why it is useful.

Exercise four.pug

Consider the function

$$f(t) = 8\cos(t) + 6\sin(2t) + \sqrt{5}\cos(4t) + 2\sin(4t) + \cos(6t - \pi/2).$$

(a) Find the (harmonic) magnitude and (harmonic) phase of its Fourier series components. (b) Sketch its magnitude and phase spectra. Hint: no Fourier integrals are necessary to solve this problem.

Exercise four.ponyo

______/25 p.

Consider the function with $a > 0$

$$f(t) = e^{-a|t|}.$$

From the transform definition, derive the Fourier transform $F(\omega)$ of $f(t)$. Simplify the result such that it is clear the expression is real (no imaginary component).

Exercise four.seesaw

______/20 p.

Consider the periodic function $f : \mathbb{R} \to \mathbb{R}$ with period T defined for one period as

______/25 p.

$$f(t) = at \quad \text{for } t \in (-T/2, T/2] \tag{1}$$

where $a, T \in \mathbb{R}$. Perform a fourier series analysis on f. Letting $a = 5$ and $T = 1$, plot f along with the partial sum of the fourier series synthesis, the first 50 nonzero components, over $t \in [-T, T]$.

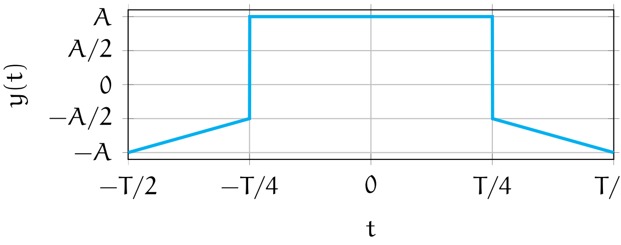


Figure exe.1: one period T of the function $y(t)$. Every line that appears straight is so.

Exercise four.totoro

Consider a periodic function $y(t)$ with some period $T \in \mathbb{R}$ and some parameter $A \in \mathbb{R}$ for which one period is shown in Fig. exe.1.

1. Perform a trigonometric Fourier series analysis of $y(t)$ and write the Fourier series $Y(\omega)$.
2. Plot the harmonic amplitude spectrum of $Y(\omega)$ for $A = T = 1$. Consider using computing software.
3. Plot the phase spectrum of $Y(\omega)$ for $A = T = 1$. Consider using computing software.

Exercise four.mall

______/20 p.

Consider the function $f : \mathbb{R} \to \mathbb{R}$ defined as

$$f(t) = \begin{cases} a - a|t|/T & \text{for } t \in [-T, T] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $a, T \in \mathbb{R}$. Perform a fourier series analysis on f, resulting in $F(\omega)$. Plot F for various a and T.

Exercise four.miyazaki

Consider the function $f : \mathbb{R} \to \mathbb{R}$ defined as

$$f(t) = a e^{-b|t-T|} \tag{3}$$

where $a, b, T \in \mathbb{R}$. Perform a fourier transform analysis on f, resulting in $F(\omega)$. Plot F for various a, b, and T.

Exercise four.haku

Consider the function $f : \mathbb{R} \to \mathbb{R}$ defined as

$$f(t) = a \cos \omega_0 t + b \sin \omega_0 t \tag{4}$$

where $a, b, \omega_0 \in \mathbb{R}$ constants. Perform a fourier transform analysis on f, resulting in $F(\omega)$.[4]

[4] It may be alarming to see a Fourier transform of a periodic function! Strictly speaking, it does not exist; however, if we extend the transform to include the distribution (not actually a function) Dirac $\delta(\omega)$, the modified-transform does exist and is given in Table four.1.

[4] Python code in this section was generated from a Jupyter notebook named `random_signal_fft.ipynb` with a `python3` kernel.

Exercise four.secrets

This exercise encodes a “secret word” into a sampled waveform for decoding via a discrete fourier transform (DFT). The nominal goal of the exercise is to decode the secret word. Along the way, plotting and interpreting the DFT will be important.
First, load relevant packages.

```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
```

We define two functions: `letter_to_number` to convert a letter into an `integer` index of the alphabet (a becomes 1, b becomes 2, etc.) and `string_to_number_list` to convert a string to a `list` of `int`s, as shown in the example at the end.

```
def letter_to_number(letter):
    return ord(letter) - 96

def string_to_number_list(string):
    out = [] # list
    for i in range(0,len(string)):
        out.append(letter_to_number(string[i]))
    return out # list

print(f"aces = { string_to_number_list('aces') }")
```

```
aces = [1, 3, 5, 19]
```

Now, we encode a code string `code` into a signal by beginning with “white noise,” which is broadband (appears throughout the spectrum) and adding to it `sin` functions with amplitudes corresponding to the letter assignments of the code and harmonic corresponding to the position of the letter in the string. For instance, the string `'bad'` would be represented by noise plus the signal

$$2 \sin 2\pi t + 1 \sin 4\pi t + 4 \sin 6\pi t. \tag{5}$$

Let’s set this up for secret word `'chupcabra'`.

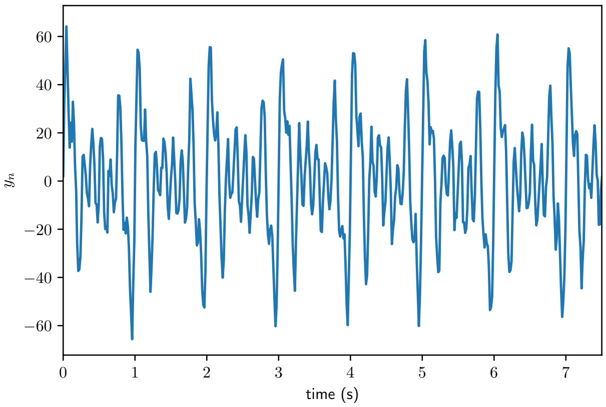


Figure exe.2: the chupacabra signal.

```
N = 2000
Tm = 30
T = float(Tm)/float(N)
fs = 1/T
x = np.linspace(0, Tm, N)
noise = 4*np.random.normal(0, 1, N)
code = 'chupcabra' # the secret word
code_number_array = np.array(string_to_number_list(code))
y = np.array(noise)
for i in range(0,len(code)):
    y = y + code_number_array[i]*np.sin(2.*np.pi*(i+1.)*x)
```

For proper decoding, later, it is important to know the fundamental frequency of the generated data.

```
print(f"fundamental frequency = {fs} Hz")
```

```
fundamental frequency = 66.66666666666667 Hz
```

Now, we plot.

```
fig, ax = plt.subplots()
plt.plot(x,y)
plt.xlim([0,Tm/4])
plt.xlabel('time (s)')
plt.ylabel('$y_n$')
plt.show()
```

Finally, we can save our data to a `numpy` file `secrets.npy` to distribute our message.

```
np.save('secrets',y)
```

Now, I have done this (for a different secret word!) and saved the data; download it here: `ricopic.one/mathematical_foundations/source/secrets.npy`. In order to load the `.npy` file into Python, we can use the following command.

```
secret_array = np.load('secrets.npy')
```

Your job is to (a) perform a DFT, (b) plot the spectrum, and (c) decode the message! Here are a few hints.

1. Use `from scipy import fft` to do the DFT.
2. Use a `hanning` window to minimize the end-effects. See `numpy.hanning` for instance. The `fft` call might then look like

   ```
   2*fft(np.hanning(N)*secret_array)/N
   ```

   where `N = len(secret_array)`.
3. Use only the positive spectrum; you can lop off the negative side and double the positive side.

Exercise four.society

Derive a fourier transform property for expressions including function $f : \mathbb{R} \to \mathbb{R}$ for

$$f(t)\cos(\omega_0 t + \psi)$$

where $\omega_0, \psi \in \mathbb{R}$.

Exercise four.flapper

Consider the function $f : \mathbb{R} \to \mathbb{R}$ defined as

$$f(t) = a u_s(t) e^{-bt} \cos(\omega_0 t + \psi) \tag{6}$$

where $a, b, \omega_0, \psi \in \mathbb{R}$ and $u_s(t)$ is the unit step function. Perform a fourier transform analysis on f, resulting in $F(\omega)$. Plot F for various a, b, ω_0, ψ and T.

Exercise four.eastegg

Consider the function $f : \mathbb{R} \to \mathbb{R}$ defined as

$$f(t) = g(t)\cos(\omega_0 t) \tag{7}$$

where $\omega_0 \in \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ will be defined in each part below. Perform a fourier transform analysis on f for each g below for $\omega_1 \in \mathbb{R}$ a constant and consider how things change if $\omega_1 \to \omega_0$.

a. $g(t) = \cos(\omega_1 t)$
b. $g(t) = \sin(\omega_1 t)$

Exercise four.savage

______/20 p.

An instrument called a "lock-in amplifier" can measure a sinusoidal signal $A\cos(\omega_0 t + \psi) = a\cos(\omega_0 t) + b\sin(\omega_0 t)$ at a known frequency ω_0 with exceptional accuracy even in the presence of significant noise $N(t)$. The workings of these devices can be described in two operations: first, the following operations on the signal with its noise, $f_1(t) = a\cos(\omega_0 t) + b\sin(\omega_0 t) + N(t)$,

$$f_2(t) = f_1(t)\cos(\omega_1 t) \quad \text{and} \quad f_3(t) = f_1(t)\sin(\omega_1 t). \tag{8}$$

where $\omega_0, \omega_1, a, b \in \mathbb{R}$. Note the relation of this operation to the Fourier transform analysis of Exercise four.. The key is to know with some accuracty ω_0 such that the instrument can set $\omega_1 \approx \omega_0$. The second operation on the signal is an aggressive low-pass filter. The filtered f_2 and f_3 are called the in-phase and quadrature components of the signal and are typically given a complex representation

$$(\text{in-phase}) + j\,(\text{quadrature}).$$

Explain with fourier transform analyses on f_2 and f_3

a. what $F_2 = \mathcal{F}(f_2)$ looks like,
b. what $F_3 = \mathcal{F}(f_3)$ looks like,
c. why we want $\omega_1 \approx \omega_0$,
d. why a low-pass filter is desirable, and
e. what the time-domain signal will look like.

Exercise four.strawman

Consider again the lock-in amplifier explored in Exercise four.. Investigate the lock-in amplifier numerically with the following steps.

a. Generate a noisy sinusoidal signal at some frequency ω_0. Include enough broadband white noise that the signal is invisible in a time-domain plot.
b. Generate f_2 and f_3, as described in Exercise four..
c. Apply a time-domain discrete low-pass filter to each $f_2 \mapsto \phi_2$ and $f_3 \mapsto \phi_3$, such as scipy's `scipy.signal.sosfiltfilt`, to complete the lock-in amplifier operation. Plot the results in time and as a complex (polar) plot.
d. Perform a discrete fourier transform on each $f_2 \mapsto F_2$ and $f_3 \mapsto F_3$. Plot the spectra.
e. Construct a frequency domain low-pass filter F and apply it (multiply!) to each $F_2 \mapsto F_2'$ and $F_3 \mapsto F_3'$. Plot the filtered spectra.
f. Perform an inverse discrete fourier transform to each $F_2' \mapsto f_2'$ and $F_3' \mapsto f_3'$. Plot the results in time and as a complex (polar) plot.

g. Compare the two methods used, i.e. time-domain filtering versus frequency-domain filtering.

pde

Partial differential equations

ordinary differential equations

lumped−parameter modeling

An ordinary differential equation is one with (ordinary) derivatives of functions a single variable each—time, in many applications. These typically describe quantities in some sort of lumped-parameter way: mass as a "point particle," a spring's force as a function of time-varying displacement across it, a resistor's current as a function of time-varying voltage across it. Given the simplicity of such models in comparison to the wildness of nature, it is quite surprising how well they work for a great many phenomena. For instance, electronics, rigid body mechanics, population dynamics, bulk fluid mechanics, and bulk heat transfer can be lumped-parameter modeled.

However, as we saw in ??, there are many phenomena of which we require more detailed models. These include:

- detailed fluid mechanics,
- detailed heat transfer,
- solid mechanics,
- electromagnetism, and
- quantum mechanics.

time−varying spatial distribution

In many cases, what is required to account for is the time-varying spatial distribution of a quantity. In fluid mechanics, we treat a fluid as having quantities such as density and velocity that vary continuous over space and time. Deriving the governing equations for such phenomena typically involves vector calculus;

we observed in ?? that statements about quantities like the divergence (e.g. continuity) can be made about certain scalar and vector fields. Such statements are governing equations (e.g. the continuity equation) and they are partial differential equations (PDEs) because the quantities of interest called dependent variables (e.g. density and velocity) are both temporally and spatially varying (temporal and spatial variables are therefore called independent variables).

partial differential equations

dependent variables

independent variables

In this chapter, we explore the analytic solution of PDEs. This is related to but distinct from the numeric solution (i.e. simulation) of PDEs, which is another important topic. Many PDEs have no known analytic solution, so numeric solution is the best available option.[1] However, it is important to note that the insight one can gain from an analytic solution is often much greater than that from a numeric solution. This is easily understood when one considers that a numeric solution is an approximation for a specific set of initial and boundary conditions. Typically, very little can be said of what would happen in general, although this is often what we seek to know. So, despite the importance of numeric solution, one should always prefer an analytic solution.

analytic solution

numeric solution

Three good texts on PDEs for further study are Kreyszig,[2] Strauss,[3] and Haberman.[4]

[1] There are some analytic techniques for gaining insight into PDEs for which there are no known solutions, such as considering the phase space. This is an active area of research; for more, see Bove, Colombini and Santo. (Antonio Bove, F. (Ferruccio) Colombini and Daniele Del Santo. Phase space analysis of partial differential equations. eng. Progress in nonlinear differential equations and their applications ; v. 69. Boston ; Berlin: Birkhäuser, 2006. ISBN: 9780817645212)

[2] Kreyszig, Advanced Engineering Mathematics, Ch. 12.

[3] W.A. Strauss. Partial Differential Equations: An Introduction. Wiley, 2007. ISBN: 9780470054567. A thorough and yet relatively compact introduction.

[4] R. Haberman. Applied Partial Differential Equations with Fourier Series and Boundary Value Problems (Classic Version). Pearson Modern Classics for Advanced Mathematics. Pearson Education Canada, 2018. ISBN: 9780134995434.

pde.class Classifying PDEs

PDEs often have an infinite number of solutions; however, when applying them to physical systems, we usually assume there is a deterministic or at least a probabilistic sequence of events will occur. Therefore, we impose additonal constraints on a PDE usually in the form of

initial conditions

1. initial conditions, values of independent variables over all space at an initial time and
2. boundary conditions, values of independent variables (or their derivatives) over all time.

boundary conditions

Ideally, imposing such conditions leaves us with a well-posed problem, which has three aspects. (Antonio Bove, F. (Ferruccio) Colombini and Daniele Del Santo. Phase space analysis of partial differential equations. eng. Progress in nonlinear differential equations and their applications ; v. 69. Boston ; Berlin: Birkhäuser, 2006. ISBN: 9780817645212, § 1.5)

well–posed problem

existence There exists at least one solution.
uniqueness There exists at most one solution.
stability If the PDE, boundary conditons, or initial conditions are changed slightly, the solution changes only slightly.

linear
nonlinear

As with ODEs, PDEs can be linear or nonlinear; that is, the independent variables and their derivatives can appear in only linear combinations (linear PDE) or in one or more nonlinear combination (nonlinear PDE). As with ODEs, there are more known analytic solutions to linear PDEs than nonlinear PDEs.
The order of a PDE is the order of its highest partial derivative. A great many physical models can be described by second-order PDEs or systems thereof. Let u be an independent

order

second–order PDEs

scalar variable, a function of m temporal and spatial variables $x_i \in \mathbb{R}^n$. A second-order linear PDE has the form, for coefficients $\alpha, \beta, \gamma, \delta$, real functions of x_i, (W.A. Strauss. Partial Differential Equations: An Introduction. Wiley, 2007. ISBN: 9780470054567. A thorough and yet relatively compact introduction. § 1.6)

$$\underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij}\partial^2_{x_i x_j} u}_{\text{second-order terms}} + \underbrace{\sum_{k=1}^{m} (\gamma_k \partial_{x_k} u + \delta_k u)}_{\text{first- and zeroth-order terms}} = \underbrace{f(x_1, \cdots, x_n)}_{\text{forcing}} \tag{1}$$

where f is called a forcing function. When f is zero, Eq. 1 is called homogeneous. We can consider the coefficients α_{ij} to be components of a matrix A with rows indexed by i and columns indexed by j. There are four prominent classes defined by the eigenvalues of A:

forcing function

homogeneous

elliptic the eigenvalues all have the same sign,

parabolic the eigenvalues have the same sign except one that is zero,

hyperbolic exactly one eigenvalue has the opposite sign of the others, and

ultrahyperbolic at least two eigenvalues of each signs.

The first three of these have received extensive treatment. They are named after conic sections due to the similarity the equations have with polynomials when derivatives are considered analogous to powers of polynomial variables. For instance, here is a case of each of the first three classes,

$$\partial^2_{xx} u + \partial^2_{yy} u = 0 \qquad \text{(elliptic)}$$
$$\partial^2_{xx} u - \partial^2_{yy} u = 0 \qquad \text{(hyperbolic)}$$
$$\partial^2_{xx} u - \partial_t u = 0. \qquad \text{(parabolic)}$$

When A depends on x_i, it may have multiple classes across its domain. In general, this equation and its associated initial and boundary conditions do not comprise a well-posed problem; however several special cases have been shown to be well-posed. Thus far, the most general statement of existence and uniqueness is the cauchy-kowalevski theorem for cauchy problems.

cauchy–kowalevski theorem
cauchy problems

pde.sturm Sturm–liouville problems

Before we introduce an important solution method for PDEs in Lec. pde.separation, we consider an ordinary differential equation that will arise in that method when dealing with a single spatial dimension x: the sturm-liouville (S-L) differential equation. Let p, q, σ be functions of x on open interval (a, b). Let X be the dependent variable and λ constant. The regular S-L problem is the S-L ODE[5]

sturm–liouville (S–L) differential equation

regular S–L problem

$$\frac{d}{dx}\left(pX'\right) + qX + \lambda\sigma X = 0 \tag{1a}$$

with boundary conditions

$$\beta_1 X(a) + \beta_2 X'(a) = 0 \tag{2a}$$
$$\beta_3 X(b) + \beta_4 X'(b) = 0 \tag{2b}$$

with coefficients $\beta_i \in \mathbb{R}$. This is a type of boundary value problem.

boundary value problems

This problem has nontrivial solutions, called eigenfunctions $X_n(x)$ with $n \in \mathbb{Z}_+$, corresponding to specific values of $\lambda = \lambda_n$ called eigenvalues.[6] There are several important theorems proven about this (see Haberman[7]). Of greatest interest to us are that

eigenfunctions

eigenvalues

1. there exist an infinite number of eigenfunctions X_n (unique within a multiplicative constant),
2. there exists a unique corresponding real eigenvalue λ_n for each eigenfunction X_n,
3. the eigenvalues can be ordered as $\lambda_1 < \lambda_2 < \cdots$,
4. eigenfunction X_n has $n - 1$ zeros on open interval (a, b),
5. the eigenfunctions X_n form an orthogonal basis with respect to weighting function σ such that any piecewise continuous function $f : [a, b] \to \mathbb{R}$ can be represented by a generalized fourier series on $[a, b]$.

[5] For the S-L problem to be regular, it has the additional constraints that p, q, σ are continuous and $p, \sigma > 0$ on $[a, b]$. This is also sometimes called the sturm-liouville eigenvalue problem. See Haberman (Haberman, Applied Partial Differential Equations with Fourier Series and Boundary Value Problems (Classic Version), § 5.3) for the more general (non-regular) S-L problem and Haberman (ibidem, § 7.4) for the multi-dimensional analog.

[6] These eigenvalues are closely related to, but distinct from, the "eigenvalues" that arise in systems of linear ODEs.

[7] ibidem, § 5.3.

This last theorem will be of particular interest in Lec. pde.separation.

Types of boundary conditions

Boundary conditions of the sturm-liouville kind (2) have four sub-types:

dirichlet for just $\beta_2, \beta_4 = 0$,
neumann for just $\beta_1, \beta_3 = 0$,
robin for all $\beta_i \neq 0$, and
mixed if $\beta_1 = 0$, $\beta_3 \neq 0$; if $\beta_2 = 0$, $\beta_4 \neq 0$.

There are many problems that are not regular sturm-liouville problems. For instance, the right-hand sides of Eq. 2 are zero, making them homogeneous boundary conditions; however, these can also be nonzero. Another case is periodic boundary conditions:

homogeneous boundary conditions

periodic boundary conditions

$$X(a) = X(b) \tag{3a}$$
$$X'(a) = X'(b). \tag{3b}$$

Example pde.sturm−1

re: a sturm−liouville problem with dirichlet boundary conditions

Consider the differential equation

$$X'' + \lambda X = 0 \tag{4}$$

with dirichlet boundary conditions on the boundary of the interval $[0, L]$

$$X(0) = 0 \quad \text{and} \quad X(L) = 0. \tag{5}$$

Solve for the eigenvalues and eigenfunctions.

This is a sturm-liouville problem, so we know the eigenvalues are real. The well-known general solutions to the ODE is

$$X(x) = \begin{cases} k_1 + k_2 x & \lambda = 0 \\ k_1 e^{j\sqrt{\lambda}x} + k_2 e^{-j\sqrt{\lambda}x} & \text{otherwise} \end{cases} \tag{6}$$

with real constants k_1, k_2. The solution must also satisfy the boundary conditions. Let's apply them to the case of $\lambda = 0$ first:

$$X(0) = 0 \Rightarrow k_1 + k_2(0) = 0 \Rightarrow k_1 = 0 \tag{7}$$
$$X(L) = 0 \Rightarrow k_1 + k_2(L) = 0 \Rightarrow k_2 = -k_1/L. \tag{8}$$

Together, these imply $k_1 = k_2 = 0$, which gives the trivial solution $X(x) = 0$, in which we aren't interested. We say, then, for nontrivial solutions $\lambda \neq 0$. Now let's check $\lambda < 0$. The solution becomes

$$\begin{aligned} X(x) &= k_1 e^{-\sqrt{|\lambda|}x} + k_2 e^{\sqrt{|\lambda|}x} && (9) \\ &= k_3 \cosh(\sqrt{|\lambda|}x) + k_4 \sinh(\sqrt{|\lambda|}x) && (10) \end{aligned}$$

where k_3 and k_4 are real constants. Again applying the boundary conditions:

$$X(0) = 0 \Rightarrow k_3 \cosh(0) + k_4 \sinh(0) = 0 \Rightarrow k_3 + 0 = 0 \Rightarrow k_3 = 0$$
$$X(L) = 0 \Rightarrow 0 \cosh(\sqrt{|\lambda|}L) + k_4 \sinh(\sqrt{|\lambda|}L) = 0 \Rightarrow k_4 \sinh(\sqrt{|\lambda|}L) = 0.$$

However, $\sinh(\sqrt{|\lambda|}L) \neq 0$ for $L > 0$, so $k_4 = k_3 = 0$—again, the trivial solution. Now let's try $\lambda > 0$. The solution can be written

$$X(x) = k_5 \cos(\sqrt{\lambda}x) + k_6 \sin(\sqrt{\lambda}x). \tag{11}$$

Applying the boundary conditions for this case:

$$X(0) = 0 \Rightarrow k_5 \cos(0) + k_6 \sin(0) = 0 \Rightarrow k_5 + 0 = 0 \Rightarrow k_5 = 0$$
$$X(L) = 0 \Rightarrow 0 \cos(\sqrt{\lambda}L) + k_6 \sin(\sqrt{\lambda}L) = 0 \Rightarrow k_6 \sin(\sqrt{\lambda}L) = 0.$$

Now, $\sin(\sqrt{\lambda}L) = 0$ for

$$\begin{aligned} \sqrt{\lambda}L &= n\pi \Rightarrow \\ \lambda &= \left(\frac{n\pi}{L}\right)^2. && (n \in \mathbb{Z}_+) \end{aligned}$$

Therefore, the only nontrivial solutions that satisfy both the ODE and the boundary conditions are the eigenfunctions

$$\begin{aligned} X_n(x) &= \sin\left(\sqrt{\lambda_n}x\right) && (12a) \\ &= \sin\left(\frac{n\pi}{L}x\right) && (12b) \end{aligned}$$

with corresponding eigenvalues

$$\lambda_n = \left(\frac{n\pi}{L}\right)^2. \tag{13}$$

Note that because $\lambda > 0$, λ_1 is the lowest eigenvalue.

Plotting the eigenfunctions

The following was generated from a Jupyter notebook with the following filename and kernel.

```
notebook filename: eigenfunctions_example_plot.ipynb
notebook kernel: python3
```

First, load some Python packages.

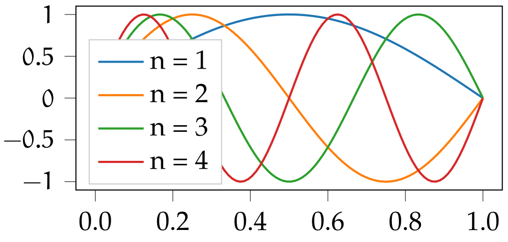
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
```

Set $L = 1$ and compute values for the first four eigenvalues `lambda_n` and eigenfunctions `X_n`.

```
L = 1
x = np.linspace(0,L,100)
n = np.linspace(1,4,4,dtype=int)
lambda_n = (n*np.pi/L)**2
X_n = np.zeros([len(n),len(x)])
for i,n_i in enumerate(n):
  X_n[i,:] = np.sin(np.sqrt(lambda_n[i])*x)
```

Plot the eigenfunctions.

```
for i,n_i in enumerate(n):
  plt.plot(
    x,X_n[i,:],
    linewidth=2,label='n = '+str(n_i)
  )
plt.legend()
plt.show() # display the plot
```



We see that the fourth of the S-L theorems appears true: $n - 1$ zeros of X_n exist on the open interval $(0, 1)$.

pde.separation PDE solution by separation of variables

We are now ready to learn one of the most important techniques for solving PDEs: separation of variables. It applies only to linear PDEs since it will require the principle of superposition. Not all linear PDEs yield to this solution technique, but several that are important do.

separation of variables

linear

The technique includes the following steps.

assume a product solution Assume the solution can be written as a product solution u_p: the product of functions of each independent variable.

product solution

separate PDE Substitute u_p into the PDE and rearrange such that at least one side of the equation has functions of a single independent variabe. If this is possible, the PDE is called separable.

separable PDEs

set equal to a constant Each side of the equation depends on different independent variables; therefore, they must each equal the same constant, often called $-\lambda$.

repeat separation, as needed If there are more than two independent variables, there will be an ODE in the separated variable and a PDE (with one fewer variables) in the other independent variables. Attempt to separate the PDE until only ODEs remain.

solve each boundary value problem Solve each boundary value problem ODE, ignoring the initial conditions for now.

solve the time variable ODE Solve for the general solution of the time variable ODE, sans initial conditions.

construct the product solution Multiply the solution in each variable to construct the product solution u_p. If the boundary value problems were sturm-liouville, the product solution is a family of eigenfunctions from which any function can be constructed via a generalized fourier series.

eigenfunctions

apply the initial condition The product solutions individually usually do not meet the initial condition. However, a generalized fourier series of them nearly always does. Superposition tells us a linear combination of solutions to the PDE and boundary conditions is also a solution; the unique series that also satisfies the initial condition is the unique solution to the entire problem.

superposition

Example pde.separation−1

re: 1D diffusion equation

Consider the one-dimensional diffusion equation PDE[a]

$$\partial_t u(t, x) = k \partial_{xx}^2 u(t, x) \tag{1}$$

with real constant k, with dirichlet boundary conditions on inverval $x \in [0, L]$

$$u(t, 0) = 0 \tag{2a}$$

$$u(t, L) = 0, \tag{2b}$$

and with initial condition

$$u(0, x) = f(x), \tag{3}$$

where f is some piecewise continuous function on $[0, L]$.

[a] For more on the diffusion or heat equation, see Haberman, (Haberman, Applied Partial Differential Equations with Fourier Series and Boundary Value Problems (Classic Version), § 2.3) Kreyszig, (Kreyszig, Advanced Engineering Mathematics, § 12.5) and Strauss. (Strauss, Partial Differential Equations: An Introduction, § 2.3)

Assume a product solution

First, we assume a product solution of the form $u_p(t, x) = T(t)X(x)$ where T and X are unknown functions on $t > 0$ and $x \in [0, L]$.

Separate PDE

Second, we substitute the product solution into Eq. 1 and separate variables:

$$T'X = kTX'' \Rightarrow \tag{4}$$

$$\frac{T'}{kT} = \frac{X''}{X}. \tag{5}$$

So it is separable! Note that we chose to group k with T, which was arbitrary but conventional.

Set equal to a constant

Since these two sides depend on different independent variables (t and x), they must equal the same constant we call $-\lambda$, so we have two ODEs:

$$\frac{T'}{kT} = -\lambda \quad \Rightarrow T' + \lambda k T = 0 \tag{6}$$

$$\frac{X''}{X} = -\lambda \quad \Rightarrow X'' + \lambda X = 0. \tag{7}$$

Solve the boundary value problem

The latter of these equations with the boundary conditions (2) is precisely the same sturm-liouville boundary value problem from Example pde.sturm-1, which had eigenfunctions

$$X_n(x) = \sin\left(\sqrt{\lambda_n}\, x\right) \tag{8a}$$

$$= \sin\left(\frac{n\pi}{L} x\right) \tag{8b}$$

with corresponding (positive) eigenvalues

$$\lambda_n = \left(\frac{n\pi}{L}\right)^2. \tag{9}$$

Solve the time variable ODE

The time variable ODE is homogeneous and has the familiar general solution

$$T(t) = ce^{-k\lambda t} \tag{10}$$

with real constant c. However, the boundary value problem restricted values of λ to λ_n, so

$$T_n(t) = ce^{-k(n\pi/L)^2 t}. \tag{11}$$

Construct the product solution

The product solution is

$$\begin{aligned} u_p(t,x) &= T_n(t)X_n(x) \\ &= ce^{-k(n\pi/L)^2 t} \sin\left(\frac{n\pi}{L}x\right). \end{aligned} \tag{12}$$

This is a family of solutions that each satisfy only exotically specific initial conditions.

Apply the initial condition

The initial condition is $u(0,x) = f(x)$. The eigenfunctions of the boundary value problem form a fourier series that satisfies the initial condition on the interval $[0,L]$ if we extend f to be periodic and odd over x (Kreyszig, Advanced Engineering Mathematics, p. 550); we call the extension f^*. The odd series synthesis can be written

$$f^*(x) = \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi}{L}x\right) \tag{13}$$

where the fourier analysis gives

$$b_n = \frac{2}{L}\int_0^L f^*(\chi)\sin\left(\frac{n\pi}{L}\chi\right). \tag{14}$$

So the complete solution is

$$u(t,x) = \sum_{n=1}^{\infty} b_n e^{-k(n\pi/L)^2 t} \sin\left(\frac{n\pi}{L}x\right). \tag{15}$$

Notice this satisfies the PDE, the boundary conditions, and the initial condition!

Plotting solutions

If we want to plot solutions, we need to specify an initial condition $u(0,x) = f^*(x)$ over $[0,L]$. We can choose anything piecewise continuous, but for simplicity let's let

$$f(x) = 1. \qquad (x \in [0, L])$$

The odd periodic extension is an odd square wave. The integral (14) gives

$$\begin{aligned} b_n &= \frac{4}{n\pi}(1 - \cos(n\pi)) \\ &= \begin{cases} 0 & n \text{ even} \\ \frac{4}{n\pi} & n \text{ odd.} \end{cases} \end{aligned} \tag{16}$$

Now we can write the solution as

$$u(t, x) = \sum_{n=1,\, n \text{ odd}}^{\infty} \frac{4}{n\pi} e^{-k(n\pi/L)^2 t} \sin\left(\frac{n\pi}{L}x\right). \tag{17}$$

Plotting in Python

First, load some Python packages.

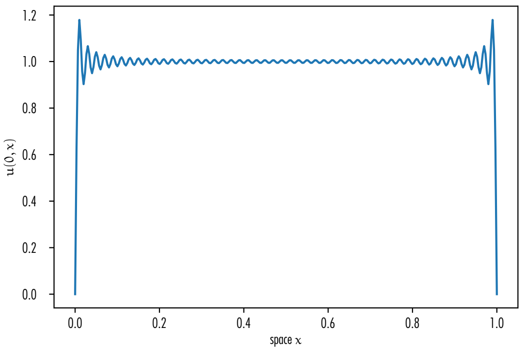
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
```

Set $k = L = 1$ and sum values for the first N terms of the solution.

```
L = 1
k = 1
N = 100
x = np.linspace(0,L,300)
t = np.linspace(0,2*(L/np.pi)**2,100)
u_n = np.zeros([len(t),len(x)])
for n in range(N):
  n = n+1 # because index starts at 0
  if n % 2 == 0: # even
    pass # already initialized to zeros
  else: # odd
    u_n += 4/(n*np.pi)*np.outer(
      np.exp(-k*(n*np.pi/L)**2*t),
      np.sin(n*np.pi/L*x)
    )
```

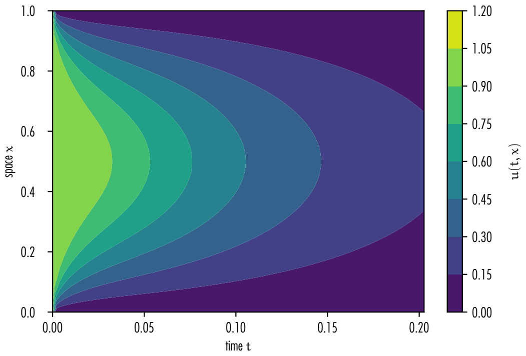
Let's first plot the initial condition.

```
p = plt.figure();
plt.plot(x,u_n[0,:]);
plt.xlabel('space $x$')
plt.ylabel('$u(0,x)$')
plt.show()
```



Now we plot the entire response.

```
p = plt.figure();
plt.contourf(t,x,u_n.T)
c = plt.colorbar()
c.set_label('$u(t,x)')
plt.xlabel('time $t$')
plt.ylabel('space $x$')
plt.show()
```



We see the diffusive action proceeds as we expected.

. Python code in this section was generated from a Jupyter notebook named `pde_separation_example_01.ipynb` with a `python3` kernel.

pde.wave The 1D wave equation

wave equation

The one-dimensional wave equation is the linear PDE

$$\partial_{tt}^2 u(t,x) = c^2 \partial_{xx}^2 u(t,x). \tag{1}$$

with real constant c. This equation models such phenomena as strings, fluids, sound, and light. It is subject to initial and boundary conditions and can be extended to multiple spatial dimensions. For 2D and 3D examples in rectangular and polar coordinates, see Kreyszig[8] and Haberman.[9]

Example pde.wave−1

re: vibrating string PDE solution by separation of variables

Consider the one-dimensional wave equation PDE

$$\partial_{tt}^2 u(t,x) = c^2 \partial_{xx}^2 u(t,x) \tag{2}$$

with real constant c and with dirichlet boundary conditions on inverval $x \in [0, L]$

$$u(t,0) = 0 \quad \text{and} \quad u(t,L) \qquad = 0, \tag{3a}$$

and with initial conditions (we need two because of the second time-derivative)

$$u(0,x) = f(x) \quad \text{and} \quad \partial_t u(0,x) = g(x), \tag{4}$$

where f and g are some piecewise continuous functions on $[0, L]$.

Assume a product solution

First, we assume a product solution of the form $u_p(t,x) = T(t)X(x)$ where T and X are unknown functions on $t > 0$ and $x \in [0, L]$.

[8] Kreyszig, Advanced Engineering Mathematics, § 12.9, 12.10.

[9] Haberman, Applied Partial Differential Equations with Fourier Series and Boundary Value Problems (Classic Version), § 4.5, 7.3.

Separate PDE

Second, we substitute the product solution into Eq. 2 and separate variables:

$$T''X = c^2TX'' \Rightarrow \tag{5}$$
$$\frac{T''}{c^2T} = \frac{X''}{X}. \tag{6}$$

So it is separable! Note that we chose to group c with T, which was arbitrary but conventional.

Set equal to a constant

Since these two sides depend on different independent variables (t and x), they must equal the same constant we call $-\lambda$, so we have two ODEs:

$$\frac{T''}{c^2T} = -\lambda \quad \Rightarrow T'' + \lambda c^2 T = 0 \tag{7}$$
$$\frac{X''}{X} = -\lambda \quad \Rightarrow X'' + \lambda X = 0. \tag{8}$$

Solve the boundary value problem

The latter of these equations with the boundary conditions (3) is precisely the same sturm-liouville boundary value problem from Example pde.sturm-1, which had eigenfunctions

$$X_n(x) = \sin\left(\sqrt{\lambda_n}x\right) \tag{9a}$$
$$= \sin\left(\frac{n\pi}{L}x\right) \tag{9b}$$

with corresponding (positive) eigenvalues

$$\lambda_n = \left(\frac{n\pi}{L}\right)^2. \tag{10}$$

Solve the time variable ODE

The time variable ODE is homogeneous and, with λ restricted by the reals by the boundary value problem, has the familiar general solution

$$T(t) = k_1 \cos(c\sqrt{\lambda}t) + k_2 \sin(c\sqrt{\lambda}t) \tag{11}$$

with real constants k_1 and k_2. However, the boundary value problem restricted values of λ to λ_n, so

$$T_n(t) = k_1 \cos\left(\frac{cn\pi}{L}t\right) + k_2 \sin\left(\frac{cn\pi}{L}t\right). \tag{12}$$

Construct the product solution

The product solution is

$$\begin{aligned} u_p(t,x) &= T_n(t)X_n(x) \\ &= k_1 \sin\left(\frac{n\pi}{L}x\right)\cos\left(\frac{cn\pi}{L}t\right) + k_2 \sin\left(\frac{n\pi}{L}x\right)\sin\left(\frac{cn\pi}{L}t\right). \end{aligned}$$

This is a family of solutions that each satisfy only exotically specific initial conditions.

Apply the initial conditions

Recall that superposition tells us that any linear combination of the product solution is also a solution. Therefore,

$$u(t,x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi}{L}x\right)\cos\left(\frac{cn\pi}{L}t\right) + b_n \sin\left(\frac{n\pi}{L}x\right)\sin\left(\frac{cn\pi}{L}t\right) \tag{13}$$

is a solution. If a_n and b_n are properly selected to satisfy the initial conditions, Eq. 13 will be the solution to the entire problem. Substituting $t = 0$ into our potential solution gives

$$u(0,x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi}{L}x\right) \tag{14a}$$

$$\partial_t u(t,x)|_{t=0} = \sum_{n=1}^{\infty} b_n \frac{cn\pi}{L} \sin\left(\frac{n\pi}{L}x\right). \tag{14b}$$

Let us extend f and g to be periodic and odd over x; we call the extensions f^* and g^*. From Eq. 14, the intial conditions are satsified if

$$f^*(x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi}{L}x\right) \tag{15a}$$

$$g^*(x) = \sum_{n=1}^{\infty} b_n \frac{cn\pi}{L} \sin\left(\frac{n\pi}{L}x\right). \tag{15b}$$

We identify these as two odd fourier syntheses. The corresponding fourier analyses are

$$a_n = \frac{2}{L}\int_0^L f^*(\chi)\sin\left(\frac{n\pi}{L}\chi\right) \tag{16a}$$

$$b_n\frac{cn\pi}{L} = \frac{2}{L}\int_0^L g^*(\chi)\sin\left(\frac{n\pi}{L}\chi\right) \tag{16b}$$

So the complete solution is Eq. 13 with components given by Eq. 16. Notice this satisfies the PDE, the boundary conditions, and the initial condition!

Discussion

It can be shown that this series solution is equivalent to two traveling waves that are interfering (see Haberman[a] and Kreyszig[b]). This is convenient because computing the series solution exactly requires an infinite summation. We show in the following section that the approximation by partial summation is still quite good.

Choosing specific initial conditions

If we want to plot solutions, we need to specify initial conditions over $[0, L]$. Let's model a string being suddenly struck from rest as

$$f(x) = 0 \tag{17}$$

$$g(x) = \delta(x - \Delta L) \tag{18}$$

where δ is the dirac delta distribution and $\Delta \in [0, L]$ is a fraction of L representing the location of the string being struck. The odd periodic extension is an odd pulse train. The integrals of (16) give

$$a_n = 0 \tag{19a}$$

$$\begin{aligned} b_n &= \frac{2}{cn\pi}\int_0^L \delta(\chi - \Delta L)\sin\left(\frac{n\pi}{L}\chi\right)dx \\ &= \frac{2}{cn\pi}\sin(n\pi\Delta). \qquad \text{(sifting property)} \end{aligned}$$

Now we can write the solution as

$$u(t,x) = \sum_{n=1}^{\infty} \frac{2}{cn\pi} \sin(n\pi\Delta) \sin\left(\frac{n\pi}{L}x\right) \sin\left(\frac{cn\pi}{L}t\right). \tag{20}$$

Plotting in Python

First, load some Python packages.

```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
```

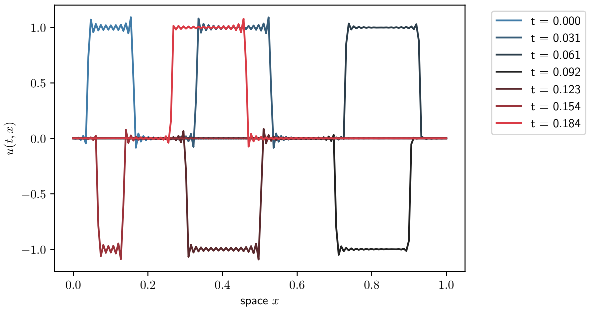
Set $c = L = 1$ and sum values for the first `N` terms of the solution for some striking location Δ.

```
Delta = 0.1 # 0 <= Delta <= L
L = 1
c = 1
N = 150
t = np.linspace(0,30*(L/np.pi)**2,100)
x = np.linspace(0,L,150)
t_b, x_b = np.meshgrid(t,x)
u_n = np.zeros([len(x),len(t)])
for n in range(N):
  n = n+1 # because index starts at 0
  u_n += 4/(c*n*np.pi)* \
    np.sin(n*np.pi*Delta)* \
    np.sin(c*n*np.pi/L*t_b)* \
    np.sin(n*np.pi/L*x_b)
```

Let's first plot some early snapshots of the response.

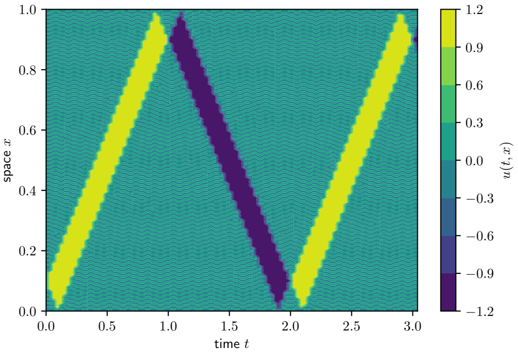
```
import seaborn as sns
n_snaps = 7
sns.set_palette(
  sns.diverging_palette(
    240, 10, n=n_snaps, center="dark"
  )
)

fig, ax = plt.subplots()
it = np.linspace(2,77,n_snaps,dtype=int)
for i in range(len(it)):
  ax.plot(x,u_n[:,it[i]],label=f"t = {t[i]:.3f}");
lgd = ax.legend(
  bbox_to_anchor=(1.05, 1),
  loc='upper left'
)
plt.xlabel('space $x$')
plt.ylabel('$u(t,x)$')
plt.show()
```



Now we plot the entire response.

```
fig, ax = plt.subplots()
p = ax.contourf(t,x,u_n)
c = fig.colorbar(p,ax=ax)
c.set_label('$u(t,x)$')
plt.xlabel('time $t$')
plt.ylabel('space $x$')
plt.show()
```



We see a wave develop and travel, reflecting and inverting off each boundary.

a. Haberman, Applied Partial Differential Equations with Fourier Series and Boundary Value Problems (Classic Version), § 4.4.
b. Kreyszig, Advanced Engineering Mathematics, § 12.2.
b. Python code in this section was generated from a Jupyter notebook named `pde_separation_example_02.ipynb` with a `python3` kernel.

pde.exe Exercises for Chapter pde

Exercise pde.horticulture

_____/20 p.

The PDE of Example pde.separation-1 can be used to describe the conduction of heat along a long, thin rod, insulated along its length, where $u(t, x)$ represents temperature. The initial and dirichlet boundary conditions in that example would be interpreted as an initial temperature distribution along the bar and fixed temperatures of the ends. Now consider the same PDE

$$\partial_t u(t, x) = k\partial^2_{xx} u(t, x) \tag{1}$$

with real constant k, with mixed boundary conditions on inverval $x \in [0, L]$

$$u(t, 0) = 0 \tag{2a}$$

$$\partial_x u(t, x)|_{x=L} = 0, \tag{2b}$$

and with initial condition

$$u(0, x) = f(x), \tag{3}$$

where f is some piecewise continuous function on $[0, L]$. This represents the insulation of one end (L) of the rod and the other end (0) is held at a fixed temperature.

a. Assume a product solution, separate variables into $X(x)$ and $T(t)$, and set the separation constant to $-\lambda$.
b. Solve the boundary value problem for its eigenfunctions X_n and eigenvalues λ_n.
c. Solve for the general solution of the time variable ODE.
d. Write the product solution and apply the initial condition $f(x)$ by constructing it from a generalized fourier series of the product solution.
e. Let $L = k = 1$ and

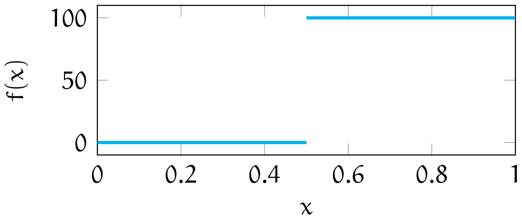


Figure exe.1: initial condition for Exercise pde..

$$f(x) = \begin{cases} 0 & \text{for } x \in [0, L/2) \\ 100 & \text{for } x \in [L/2, L] \end{cases} \tag{4}$$

as shown in Fig. exe.1. Compute the solution series components. Plot the sum of the first 50 terms over x and t.

Exercise pde.poltergeist

______/20 p.

The PDE of Example pde.separation-1 can be used to describe the conduction of heat along a long, thin rod, insulated along its length, where $u(t, x)$ represents temperature. The initial and dirichlet boundary conditions in that example would be interpreted as an initial temperature distribution along the bar and fixed temperatures of the ends. Now consider the same PDE

$$\partial_t u(t, x) = k\partial^2_{xx} u(t, x) \tag{5}$$

with real constant k, now with neumann boundary conditions on inverval $x \in [0, L]$

$$\partial_x u|_{x=0} = 0 \quad \text{and} \quad \partial_x u|_{x=L} = 0, \tag{6a}$$

and with initial condition

$$u(0, x) = f(x), \tag{7}$$

where f is some piecewise continuous function on $[0, L]$. This represents the complete insulation of the ends of the rod, such that no heat flows from the ends (or from anywhere else).

a. Assume a product solution, separate variables into $X(x)$ and $T(t)$, and set the separation constant to $-\lambda$.
b. Solve the boundary value problem for its eigenfunctions X_n and eigenvalues λ_n.
c. Solve for the general solution of the time variable ODE.

d. Write the product solution and apply the initial condition $f(x)$ by constructing it from a generalized fourier series of the product solution.
e. Let $L = k = 1$ and $f(x) = 100 - 200/L\,|x - L/2|$ as shown in Fig. exe.2. Compute the solution series components. Plot the sum of the first 50 terms over x and t.

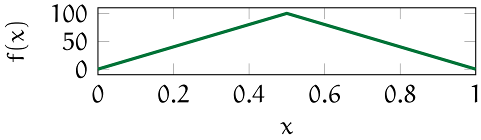


Figure exe.2: initial condition for ??.

Exercise pde.kathmandu

Consider the free vibration of a uniform and relatively thin beam—with modulus of elasticity E, second moment of cross-sectional area I, and mass-per-length μ—pinned at each end. The PDE describing this is a version of the euler-bernoulli beam equation for vertical motion u:

$$\partial^2_{tt} u(t, x) = -\alpha^2 \partial^4_{xxxx} u(t, x) \tag{8}$$

with real constant α defined as

$$\alpha^2 = \frac{EI}{\mu}. \tag{9}$$

Pinned supports fix vertical motion such that we have boundary conditions on interval $x \in [0, L]$

$$u(t, 0) = 0 \quad \text{and} \quad u(t, L) = 0. \tag{10a}$$

Additionally, pinned supports cannot provide a moment, so

$$\partial^2_{xx} u|_{x=0} = 0 \quad \text{and} \quad \partial^2_{xx} u|_{x=L} = 0. \tag{10b}$$

Furthermore, consider the initial conditions

$$u(0,x) = f(x), \quad \text{and} \quad \partial_t u|_{t=0} = 0. \tag{11a}$$

where f is some piecewise continuous function on $[0, L]$.

a. Assume a product solution, separate variables into $X(x)$ and $T(t)$, and set the separation constant to $-\lambda$.
b. Solve the boundary value problem for its eigenfunctions X_n and eigenvalues λ_n. Assume real $\lambda > 0$ (it's true but tedious to show).
c. Solve for the general solution of the time variable ODE.
d. Write the product solution and apply the initial conditions by constructing it from a generalized fourier series of the product solution.
e. Let $L = \alpha = 1$ and $f(x) = \sin(10\pi x/L)$ as shown in Fig. exe.3. Compute the solution series components. Plot the sum of the first 50 terms over x and t.

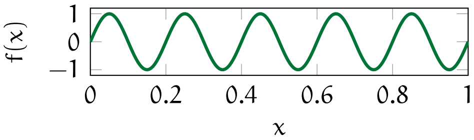


Figure exe.3: initial condition for Exercise pde..

opt

Optimization

opt.grad Gradient descent

Consider a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$ that represents some cost or value. This is called an objective function, and we often want to find an $\mathbf{X} \in \mathbb{R}^n$ that yields f's extremum: minimum or maximum, depending on whichever is desirable.

objective function

extremum

It is important to note however that some functions have no finite extremum. Other functions have multiple. Finding a global extremum is generally difficult; however, many good methods exist for finding a local extremum: an extremum for some region $R \subset \mathbb{R}^n$.

global extremum

local extremum

The method explored here is called gradient descent. It will soon become apparent why it has this name.

gradient descent

Stationary points

Recall from basic calculus that a function f of a single variable had potential local extrema where $df(x)/\,dx = 0$. The multivariate version of this, for multivariate function f, is

$$\operatorname{grad} f = \mathbf{0}. \tag{1}$$

A value $\mathbf{X}$ for which Eq. 1 holds is called a stationary point. However, as in the univariate case, a stationary point may not be a local extremum; in these cases, it called a saddle point.

stationary point

saddle point

Consider the hessian matrix H with values, for independent variables x_i,

hessian matrix

$$H_{ij} = \partial^2_{x_i x_j} f. \tag{2}$$

For a stationary point $\mathbf{X}$, the second partial derivative test tells us if it is a local maximum, local minimum, or saddle point:

second partial derivative test

minimum If $H(\mathbf{X})$ is positive definite (all its eigenvalues are positive),

positive definite

$\mathbf{X}$ is a local minimum.

maximum If $H(\mathbf{X})$ is negative definite (all its eigenvalues are negative),
$\mathbf{X}$ is a local maximum.

negative definite

saddle If $H(\mathbf{X})$ is indefinite (it has both positive and negative eigenvalues),
$\mathbf{X}$ is a saddle point.

indefinite

These are sometimes called tests for concavity: minima occur where f is convex and maxima where f is concave (i.e. where $-f$ is convex).
It turns out, however, that solving Eq. 1 directly for stationary points is generally hard.
Therefore, we will typically use an iterative technique for estimating them.

convex

concave

The gradient points the way

Although Eq. 1 isn't usually directly useful for computing stationary points, it suggests iterative techniques that are. Several techniques rely on the insight that the gradient points toward stationary points. Recall from Lec. vecs.grad that $\operatorname{grad} f$ is a vector field that points in the direction of greatest increase in f.
Consider starting at some point x_0 and wanting to move iteratively closer to a stationary point.
So, if one is seeking a maximum of f, then choose x_1 to be in the direction of $\operatorname{grad} f$. If one is seeking a minimum of f, then choose x_1 to be opposite the direction of $\operatorname{grad} f$.

the gradient points toward stationary points

The question becomes: how far α should we go in (or opposite) the direction of the gradient?
Surely too-small α will require more iteration and too-large α will lead to poor convergence or missing minima altogether. This framing of the problem is called line search. There are a few common methods for choosing α, called the step size, some more computationally efficient than others.

line search

step size α

Two methods for choosing the step size are described below. Both are framed as minimization methods, but changing the sign of the step turns them into maximization methods.

The classical method

Let

$$\mathbf{g}_k = \operatorname{grad} f(\mathbf{x}_k), \tag{3}$$

the gradient at the algorithm's current estimate $\mathbf{x}_k$ of the minimum. The classical method of choosing α is to attempt to solve analytically for

$$\alpha_k = \operatorname*{argmin}_{\alpha} f(\mathbf{x}_k - \alpha \mathbf{g}_k). \tag{4}$$

This solution approximates the function f as one varies α. It is approximate because as α varies, so should $\mathbf{x}$. But even with α as the only variable, Eq. 4 may be difficult or impossible to solve. However, this is sometimes called the "optimal" choice for α. Here "optimality" refers not to practicality but to ideality. This method is rarely used to solve practical problems.
The algorithm of the classical gradient descent method can be summarized in the pseudocode of Algorithm grad.1. It is described further in Kreyszig.[1]

Algorithm grad.1 Classical gradient descent

```
procedure classical_minimizer($f$,$\mathbf{x}_0$,$T$)
    while $\delta\mathbf{x} > T$ do ▷ until threshold $T$ is met
        $\mathbf{g}_k \leftarrow \operatorname{grad} f(\mathbf{x}_k)$
        $\alpha_k \leftarrow \operatorname{argmin}_\alpha f(\mathbf{x}_k - \alpha\mathbf{g}_k)$
        $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \alpha_k\mathbf{g}_k$
        $\delta\mathbf{x} \leftarrow \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$
        $k \leftarrow k + 1$
    end while
    return $\mathbf{x}_k$ ▷ the threshold was reached
end procedure
```

The Barzilai and Borwein method

In practice, several non-classical methods are used for choosing step size α. Most of these

[1] Kreyszig, Advanced Engineering Mathematics, § 22.1.

construct criteria for step sizes that are too small and too large and prescribe choosing some α that (at least in certain cases) must be in the sweet-spot in between. Barzilai and Borwein[2] developed such a prescription, which we now present.

Let $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\Delta\boldsymbol{g}_k = \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$. This method minimizes $\|\Delta x - \alpha\Delta g\|^2$ by choosing

$$\alpha_k = \frac{\Delta\boldsymbol{x}_k \cdot \Delta\boldsymbol{g}_k}{\Delta\boldsymbol{g}_k \cdot \Delta\boldsymbol{g}_k}. \tag{5}$$

The algorithm of this gradient descent method can be summarized in the pseudocode of Algorithm grad.2. It is described further in Barzilai and Borwein.[3]

Algorithm grad.2 Barzilai and Borwein gradient descent

```
procedure barzilai_minimizer($f$,$\boldsymbol{x}_0$,$T$)
    while $\delta\boldsymbol{x} > T$ do    ▷ until threshold $T$ is met
        $\boldsymbol{g}_k \leftarrow \operatorname{grad} f(\boldsymbol{x}_k)$
        $\Delta\boldsymbol{g}_k \leftarrow \boldsymbol{g}_k - \boldsymbol{g}_{k-1}$
        $\Delta\boldsymbol{x}_k \leftarrow \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$
        $\alpha_k \leftarrow \frac{\Delta\boldsymbol{x}_k \cdot \Delta\boldsymbol{g}_k}{\Delta\boldsymbol{g}_k \cdot \Delta\boldsymbol{g}_k}$
        $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k$
        $\delta\boldsymbol{x} \leftarrow \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|$
        $k \leftarrow k + 1$
    end while
    return $\boldsymbol{x}_k$    ▷ the threshold was reached
end procedure
```

Example opt.grad–1 re: Barzilai and Borwein gradient descent

Consider the functions (a) $f_1 : \mathbb{R}^2 \to \mathbb{R}$ and (b) $f_2 : \mathbb{R}^2 \to \mathbb{R}$ defined as

$$f_1(\boldsymbol{x}) = (x_1 - 25)^2 + 13(x_2 + 10)^2 \tag{6}$$

$$f_2(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x} \cdot A\boldsymbol{x} - \boldsymbol{b} \cdot \boldsymbol{x} \tag{7}$$

where

$$A = \begin{bmatrix} 20 & 0 \\ 0 & 10 \end{bmatrix} \quad \text{and} \tag{8a}$$

$$b = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top . \tag{8b}$$

[2] Jonathan Barzilai and Jonathan M. Borwein. ?Two-Point Step Size Gradient Methods? inIMA Journal of Numerical Analysis: 8.1 (january 1988), pages 141–148. issn: 0272-4979. doi: 10.1093/imanum/8.1.141. This includes an innovative line search method.

[3] ibidem.

Use the method of Barzilai and Borwein[a] starting at some x_0 to find a minimum of each function.

[a] Barzilai and Borwein, ?Two-Point Step Size Gradient Methods?

First, load some Python packages.

```
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
from tabulate import tabulate
```

```
---------------------------------------------------------------------------

ModuleNotFoundError                       Traceback
↪  (most recent call last)

/tmp/ipykernel_1136564/820890629.py in <module>
      3 import matplotlib.pyplot as plt
      4 from IPython.display import display,
      ↪  Markdown, Latex
----> 5 from tabulate import tabulate


ModuleNotFoundError: No module named 'tabulate'
```

We begin by writing a class gradient_descent_min to perform the gradient descent. This is not optimized for speed.

```
class gradient_descent_min():
  """ A Barzilai and Borwein gradient descent class.

  Inputs:
    * f:  Python function of x variables
    * x:  list of symbolic variables (eg [x1, x2])
    * x0: list of numeric initial guess of a min of f
    * T: step size threshold for stopping the descent

  To execute the gradient descent call descend method.

  nb: This is only for gradients in cartesian
      coordinates! Further work would be to implement
      this in multiple or generalized coordinates.
      See the grad method below for implementation.
  """

  def __init__(self,f,x,x0,T):
    self.f = f
    self.x = Array(x)
    self.x0 = np.array(x0)
    self.T = T
    self.n = len(x0) # size of x
    self.g = lambdify(x,self.grad(f,x),'numpy')
    self.xk = np.array(x0)
    self.table = {}

  def descend(self):
    # unpack variables
    f = self.f
    x = self.x
    x0 = self.x0
    T = self.T
    g = self.g
    # initialize variables
    N = 0
    x_k = x0
    dx = 2*T # can't be zero
    x_km1 = .9*x0-.1 # can't equal x0
    g_km1 = np.array(g(*x_km1))
    N_max = 100 # max iterations
    table_data = [[N,x0,np.array(g(*x0)),0]]
    while (dx > T and N < N_max) or N < 1:
      N += 1 # increment index
      g_k = np.array(g(*x_k))
      dg_k = g_k - g_km1
      dx_k = x_k - x_km1
      alpha_k = abs(dx_k.dot(dg_k)/dg_k.dot(dg_k))
      x_km1 = x_k # store
      x_k = x_k - alpha_k*g_k
      # save
      table_data.append([N,x_k,g_k,alpha_k])
      self.xk = np.vstack((self.xk,x_k))
      # store other variables
      g_km1 = g_k
      dx = np.linalg.norm(x_k - x_km1) # check
    self.tabulater(table_data)

  def tabulater(self,table_data):
    np.set_printoptions(precision=2)
    tabulate.LATEX_ESCAPE_RULES={}
    self.table['python'] = tabulate(
      table_data,
      headers=["N","x_k","g_k","alpha"],
    )
    self.table['latex'] = tabulate(
      table_data,
      headers=[
        "$N$","$\\bm{x}_k$","$\\bm{g}_k$","$\\alpha$"
      ],
      tablefmt="latex_raw",
    )

  def grad(self,f,x): # cartesian coord's gradient
```

```
    return derive_by_array(f(x),x)
```

First, consider f_1.

```
var('x1 x2')
x = Array([x1,x2])
f1 = lambda x: (x[0]-25)**2 + 13*(x[1]+10)**2
gd = gradient_descent_min(f=f1,x=x,x0=[-50,40],T=1e-8)
```

Perform the gradient descent.

```
gd.descend()
```

```
---------------------------------------------------------------------------


NameError                                 Traceback
↪  (most recent call last)

/tmp/ipykernel_1136564/2845911865.py in <module>
----> 1 gd.descend()


/tmp/ipykernel_1136564/2142203784.py in
↪  descend(self)
     55       g_km1 = g_k
     56       dx = np.linalg.norm(x_k - x_km1) #
     ↪  check
---> 57     self.tabulater(table_data)
     58
     59   def tabulater(self,table_data):


/tmp/ipykernel_1136564/2142203784.py in
↪  tabulater(self, table_data)
     59   def tabulater(self,table_data):
     60     np.set_printoptions(precision=2)
---> 61     tabulate.LATEX_ESCAPE_RULES={}
     62     self.table['python'] = tabulate(
     63       table_data,


NameError: name 'tabulate' is not defined
```

Print the interesting variables.

```
print(gd.table['python'])
```

```
---------------------------------------------------------------------------


KeyError                                  Traceback
↪  (most recent call last)
```

```
/tmp/ipykernel_1136564/1459049274.py in <module>
----> 1 Latex(gd.table['latex'])


KeyError: 'latex'
```

Now let's `lambdify` the function `f1` so we can plot.

```
f1_lambda = lambdify((x1,x2),f1(x),'numpy')
```

Now let's plot a contour plot with the gradient descent overlaid.

```
fig, ax = plt.subplots()
# contour plot
X1 = np.linspace(-100,100,100)
X2 = np.linspace(-50,50,100)
X1, X2 = np.meshgrid(X1,X2)
F1 = f1_lambda(X1,X2)
plt.contourf(X1,X2,F1)
plt.colorbar()
# gradient descent plot
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.collections import LineCollection
xX1 = gd.xk[:,0]
xX2 = gd.xk[:,1]
points = np.array([xX1, xX2]).T.reshape(-1, 1, 2)
segments = np.concatenate(
  [points[:-1], points[1:]], axis=1
)
lc = LineCollection(
  segments,
  cmap=plt.get_cmap('Reds')
)
lc.set_array(np.linspace(0,1,len(xX1))) # color segs
lc.set_linewidth(3)
ax.autoscale(False) # avoid the scatter changing lims
ax.add_collection(lc)
ax.scatter(
  xX1,xX2,
  zorder=1,
  marker="o",
  color=plt.cm.Reds(np.linspace(0,1,len(xX1))),
  edgecolor='none'
)
plt.show()
```

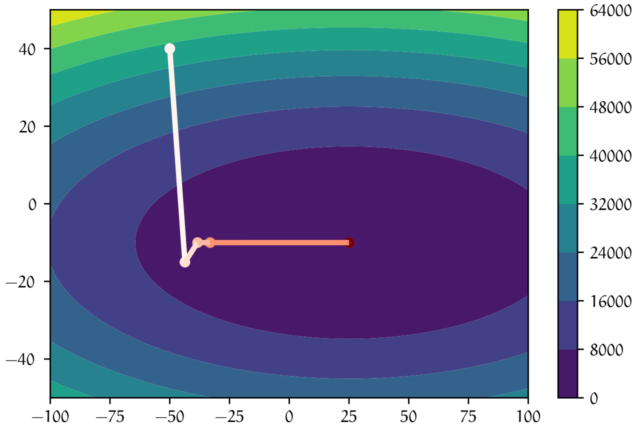


Figure grad.1:

Now consider f_2.

```
A = Matrix([[10,0],[0,20]])
b = Matrix([[1,1]])
def f2(x):
  X = Array([x]).tomatrix().T
  return 1/2*X.dot(A*X) - b.dot(X)
gd = gradient_descent_min(f=f2,x=x,x0=[50,-40],T=1e-8)
```

Perform the gradient descent.

```
gd.descend()
```

```
---------------------------------------------------------------------------

NameError                                 Traceback
↪  (most recent call last)

/tmp/ipykernel_1136564/2845911865.py in <module>
----> 1 gd.descend()


/tmp/ipykernel_1136564/2142203784.py in
↪  descend(self)
     55       g_km1 = g_k
     56       dx = np.linalg.norm(x_k - x_km1) #
     ↪  check
---> 57     self.tabulater(table_data)
     58
     59   def tabulater(self,table_data):


/tmp/ipykernel_1136564/2142203784.py in
↪  tabulater(self, table_data)
     59   def tabulater(self,table_data):
     60     np.set_printoptions(precision=2)
---> 61     tabulate.LATEX_ESCAPE_RULES={}
     62     self.table['python'] = tabulate(
     63       table_data,


NameError: name 'tabulate' is not defined
```

Print the interesting variables.

```
print(gd.table['python'])
```

```
---------------------------------------------------------------------------

KeyError                                  Traceback
↪  (most recent call last)

/tmp/ipykernel_1136564/1459049274.py in <module>
----> 1 Latex(gd.table['latex'])
```

```
KeyError: 'latex'
```

Now let's `lambdify` the function `f2` so we can plot.

```
f2_lambda = lambdify((x1,x2),f2(x),'numpy')
```

Now let's plot a contour plot with the gradient descent overlaid.

```
fig, ax = plt.subplots()
# contour plot
X1 = np.linspace(-100,100,100)
X2 = np.linspace(-50,50,100)
X1, X2 = np.meshgrid(X1,X2)
F2 = f2_lambda(X1,X2)
plt.contourf(X2,X1,F2)
plt.colorbar()
# gradient descent plot
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.collections import LineCollection
xX1 = gd.xk[:,0]
xX2 = gd.xk[:,1]
points = np.array([xX1, xX2]).T.reshape(-1, 1, 2)
segments = np.concatenate(
  [points[:-1], points[1:]], axis=1
)
lc = LineCollection(
  segments,
  cmap=plt.get_cmap('Reds')
)
lc.set_array(np.linspace(0,1,len(xX1))) # color segs
lc.set_linewidth(3)
ax.autoscale(False) # avoid the scatter changing lims
ax.add_collection(lc)
ax.scatter(
  xX1,xX2,
  zorder=1,
  marker="o",
  color=plt.cm.Reds(np.linspace(0,1,len(xX1))),
  edgecolor='none'
)
plt.show()
```

. Python code in this section was generated from a Jupyter notebook named `gradient_descent.ipynb` with a `python3` kernel.

opt.lin Constrained linear optimization

Consider a linear objective function $f : \mathbb{R}^n \to \mathbb{R}$ with variables x_i in vector $\mathbf{x}$ and coefficients c_i in vector $\mathbf{c}$:

$$f(\mathbf{x}) = \mathbf{c} \cdot \mathbf{x} \tag{1}$$

subject to the linear constraints—restrictions on x_i—

constraints

$$\begin{aligned} A\mathbf{x} &\leqslant \mathbf{a}, && \text{(2a)}\\ B\mathbf{x} &= \mathbf{b}, \text{ and} && \text{(2b)}\\ \mathbf{l} \leqslant \mathbf{x} &\leqslant \mathbf{u} && \text{(2c)}\end{aligned}$$

where A and B are constant matrices and $\mathbf{a}, \mathbf{b}, \mathbf{l}, \mathbf{u}$ are n-vectors. This is one formulation of what is called a linear programming problem. Usually we want to maximize f over the constraints. Such problems frequently arise throughout engineering, for instance in manufacturing, transportation, operations, etc. They are called constrained because there are constraints on $\mathbf{x}$; they are called linear because the objective function and the constraints are linear.

linear programming problem

maximize

constrained

linear

We call a pair $(\mathbf{x}, f(\mathbf{x}))$ for which $\mathbf{x}$ satisfies Eq. 2 a feasible solution. Of course, not every feasible solution is optimal: a feasible solution is optimal iff there exists no other feasible solution for which f is greater (assuming we're maximizing). We call the vector subspace of feasible solutions $S \subset \mathbb{R}^n$.

feasible solution

optimal

Feasible solutions form a polytope

Consider the effect of the constraints. Each of the equalities and inequalities defines a linear hyperplane in $\mathbb{R}^n$ (i.e. a linear subspace of dimension $n - 1$): either as a boundary of S (inequality) or as a restriction of S to the hyperplane. When joined, these hyperplanes are the boundary of S (equalities restrict S to

hyperplane

lower dimension). So we see that each of the boundaries of S is flat, which makes S a polytope (in $\mathbb{R}^2$, a polygon). What makes this especially interesting is that polytopes have vertices where the hyperplanes intersect. Solutions at the vertices are called basic feasible solutions.

flat
polytope
vertices
basic feasible solutions

Only the vertices matter

Our objective function f is linear, so for some constant h, $f(\boldsymbol{x}) = h$ defines a level set that is itself a hyperplane H in $\mathbb{R}^n$. If this hyperplane intersects S at a point $\boldsymbol{x}$, $(\boldsymbol{x}, f(\boldsymbol{x}) = h)$ is the corresponding solution. There are three possibilities when H intersects S:

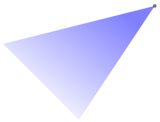
level set

1. $H \cap S$ is a vertex of S,
2. $H \cap S$ is a boundary hyperplane of S, or
3. $H \cap S$ slices through the interior of S.

However, this third option implies that there exists a level set G corresponding to $f(\boldsymbol{x}) = g$ such that G intersects S and $g > h$, so solutions on $H \cap S$ are not optimal. (We have not proven this, but it may be clear from our progression.) We conclude that either the first or second case must be true for optimal solutions. And notice that in both cases, a (potentially optimal) solution occurs at at least one vertex. The key insight, then, is that an optimal solution occurs at a vertex of S.

an optimal solution occurs at a vertex of S

This means we don't need to search all of S, or even its boundary: we need only search the vertices. Helpful as this is, it restricts us down to $\binom{n}{\#\text{ constraints}}$ potentially optimal solutions—usually still too many to search in a naïve way. In Lec. opt.simplex, this is mitigated by introducing a powerful searching method.



opt.simplex The simplex algorithm

simplex algorithm

The simplex algorithm (or “method”) is an iterative technique for finding an optimal solution of the linear programming problem of Eqs. 1 and 2. The details of the algorithm are somewhat involved, but the basic idea is to start at a vertex of the feasible solution space S and traverse an edge of the polytope that leads to another vertex with a greater value of f. Then, repeat this process until there is no neighboring vertex with a greater value of f, at which point the solution is guaranteed to be optimal. Rather than present the details of the algorithm, we choose to show an example using Python. There have been some improvements on the original algorithm that have been implemented into many standard software packages, including Python’s `scipy` package (Pauli Virtanen andothers. ?SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python? inarXiv e-prints: arXiv:1907.10121 [july 2019], arXiv:1907.10121) module `scipy.optimize`.[4]

Example opt.simplex–1

re: simplex method using `scipy.optimize`

Maximize the objective function

$$f(\mathbf{x}) = \mathbf{c} \cdot \mathbf{x} \tag{1a}$$

for $\mathbf{x} \in \mathbb{R}^2$ and

$$\mathbf{c} = \begin{bmatrix} 5 & 2 \end{bmatrix}^\top \tag{1b}$$

subject to constraints

$$0 \leqslant x_1 \leqslant 10 \tag{2a}$$

$$-5 \leqslant x_2 \leqslant 15 \tag{2b}$$

$$4x_1 + x_2 \leqslant 40 \tag{2c}$$

$$x_1 + 3x_2 \leqslant 35 \tag{2d}$$

$$-8x_1 - x_2 \geqslant -75. \tag{2e}$$

[4] Another Python package `pulp` (PuLP) is probably more popular for linear programming; however, we choose `scipy.optimize` because it has applications beyond linear programming.

First, load some Python packages.

```
from scipy.optimize import linprog
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
```

Encoding the problem

Before we can use `linprog`, we must first encode Eqs. 1 and 2 into a form `linprog` will recognize. We begin with f, which we can write as $\mathbf{c} \cdot \mathbf{x}$ with the coefficients of $\mathbf{c}$ as follows.

```
c = [-5, -2] # negative to find max
```

We've negated each constant because `linprog` minimizes f and we want to maximize f. Now let's encode the inequality constraints. We will write the left-hand side coefficients in the matrix A and the right-hand-side values in vector $\mathbf{a}$ such that

$$A\mathbf{x} \leqslant \mathbf{a}. \tag{3}$$

Notice that one of our constraint inequalities is $\geqslant$ instead of $\leqslant$. We can flip this by multiplying the inequality by -1. We use simple lists to encode A and $\mathbf{a}$.

```
A = [
  [4, 1],
  [1, 3],
  [8, 1]
]
a = [40, 35, 75]
```

Now we need to define the lower $\mathbf{l}$ and upper $\mathbf{u}$ bounds of $\mathbf{x}$. The function `linprog` expects these to be in a single list of lower- and upper-bounds of each x_i.

```
lu = [
  (0, 10),
  (-5,15),
]
```

We want to keep track of each step linprog takes. We can access these by defining a function callback, to be passed to linprog.

```python
x = [] # for storing the steps
def callback(res): # called at each step
  global x
  print(f"nit = {res.nit}, x_k = {res.x}")
  x.append(res.x.copy()) # store
```

Now we need to call linprog. We don't have any equality constraints, so we need only use the keyword arguments A_ub=A and b_ub=a. For demonstration purposes, we tell it to use the 'simplex' method, which is not as good as its other methods, which use better algorithms based on the simplex.

```python
res = linprog(
  c,
  A_ub=A,
  b_ub=a,
  bounds=lu,
  method='simplex',
  callback=callback
)

x = np.array(x)
```

```
nit = 0, x_k = [ 0. -5.]
nit = 1, x_k = [10. -5.]
nit = 2, x_k = [8.75 5.  ]
nit = 3, x_k = [7.72727273 9.09090909]
nit = 4, x_k = [7.72727273 9.09090909]
nit = 5, x_k = [7.72727273 9.09090909]
nit = 5, x_k = [7.72727273 9.09090909]
```

So the optimal solution $(\boldsymbol{x}, f(\boldsymbol{x}))$ is as follows.

```python
print(f"optimum x: {res.x}")
print(f"optimum f(x): {-res.fun}")
```

```
optimum x: [7.72727273 9.09090909]
optimum f(x): 56.81818181818182
```

The last point was repeated

1. once because there was no adjacent vertex with greater $f(\boldsymbol{x})$ and

2. twice because the algorithm calls ‘callback‘ twice on the last step.

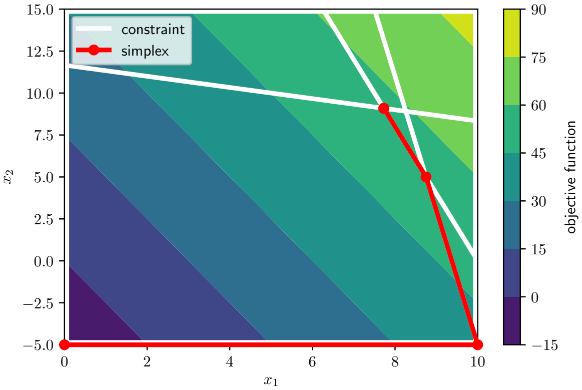
Plotting

When the solution space is in $\mathbb{R}^2$, it is helpful to graphically represent the solution space, constraints, and the progression of the algorithm. We begin by defining the inequality lines from A and $\boldsymbol{a}$ over the bounds of x_1.

```
n = len(c) # number of variables x
m = np.shape(A)[0] # number of inequality constraints
x2 = np.empty([m,2])
for i in range(0,m):
  x2[i,:] = -A[i][0]/A[i][1]*np.array(lu[0]) + a[i]/A[i][1]
```

Now we plot a contour plot of f over the bounds of x_1 and x_2 and overlay the inequality constraints and the steps of the algorithm stored in x.

```
lu_array = np.array(lu)
fig, ax = plt.subplots()
mpl.rcParams['lines.linewidth'] = 3
# contour plot
X1 = np.linspace(*lu_array[0],100)
X2 = np.linspace(*lu_array[1],100)
X1, X2 = np.meshgrid(X1,X2)
F2 = -c[0]*X1 + -c[1]*X2 # negative because max hack
con = ax.contourf(X1,X2,F2)
cbar = fig.colorbar(con,ax=ax)
cbar.ax.set_ylabel('objective function')
# bounds on x
un = np.array([1,1])
opts = {'c':'w','label':None,'linewidth':6}
plt.plot(lu_array[0],lu_array[1,0]*un,**opts)
plt.plot(lu_array[0],lu_array[1,1]*un,**opts)
plt.plot(lu_array[0,0]*un,lu_array[1],**opts)
plt.plot(lu_array[0,1]*un,lu_array[1],**opts)
# inequality constraints
for i in range(0,m):
  p, = plt.plot(lu[0],x2[i,:],c='w')
p.set_label('constraint')
# steps
plt.plot(
  x[:,0],x[:,1],
  '-o',c='r',
  clip_on=False,zorder=20,
  label='simplex'
)
plt.ylim(lu_array[1])
plt.xlabel('$x_1$')
plt.ylabel('$x_2$')
plt.legend()
plt.show()
```



. Python code in this section was generated from a Jupyter notebook named `simplex_linear_programming.ipynb` with a `python3` kernel.

opt.exe Exercises for Chapter opt

Exercise opt.chortle

______/20 p.

Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$, defined as

$$f(\boldsymbol{x}) = \cos(x_1 - e^{x_2} + 2)\sin(x_1^2/4 - x_2^2/3 + 4) \quad (1)$$

Use the method of Barzilai and Borwein[5] starting at $\boldsymbol{x}_0 = (1, 1)$ to find a minimum of the function.

Exercise opt.cummerbund

Consider the functions (a) $f_1 : \mathbb{R}^2 \to \mathbb{R}$ and (b) $f_2 : \mathbb{R}^2 \to \mathbb{R}$ defined as

$$f_1(\boldsymbol{x}) = 4(x_1 - 16)^2 + (x_2 + 64)^2 + x_1 \sin^2 x_1 \quad (2)$$

$$f_2(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x} \cdot A\boldsymbol{x} - \boldsymbol{b} \cdot \boldsymbol{x} \quad (3)$$

where

$$A = \begin{bmatrix} 5 & 0 \\ 0 & 15 \end{bmatrix} \quad \text{and} \quad (4a)$$

$$b = \begin{bmatrix} -2 & 1 \end{bmatrix}^\top . \quad (4b)$$

Use the method of Barzilai and Borwein[6] starting at some $\boldsymbol{x}_0$ to find a minimum of each function.

Exercise opt.melty

Maximize the objective function

$$f(\boldsymbol{x}) = \boldsymbol{c} \cdot \boldsymbol{x} \quad (5a)$$

for $\boldsymbol{x} \in \mathbb{R}^3$ and

$$\boldsymbol{c} = \begin{bmatrix} 3 & -8 & 1 \end{bmatrix}^\top \quad (5b)$$

subject to constraints

$$0 \leqslant x_1 \leqslant 20 \quad (6a)$$

$$-5 \leqslant x_2 \leqslant 0 \quad (6b)$$

$$5 \leqslant x_3 \leqslant 17 \quad (6c)$$

$$x_1 + 4x_2 \leqslant 50 \quad (6d)$$

$$2x_1 + x_3 \leqslant 43 \quad (6e)$$

$$-4x_1 + x_2 - 5x_3 \geqslant -99. \quad (6f)$$

[5] Barzilai and Borwein, ?Two-Point Step Size Gradient Methods?

[6] ibidem.

Exercise opt.lateness

Find the minimum of the function,

$$f(x) = x_1^2 + x_2^2 - \frac{x_1}{10} + \cos(2x_1),$$

starting at the location $x = [-0.5,\ 0.75]^\top$, and with a constant value $\alpha = 0.01$.

1. What is the location of the minimum you found?
2. Is this location the global minimum?

nlin

Nonlinear analysis

1 The ubiquity of near-linear systems and the tools we have for analyses thereof can sometimes give the impression that nonlinear systems are exotic or even downright flamboyant. However, a great many systems[1] important for a mechanical engineer are frequently hopelessly nonlinear. Here are a some examples of such systems.

- A robot arm.
- Viscous fluid flow (usually modelled by the navier-stokes equations).
- ____________________
  ____
- Anything that “fills up” or “saturates.”
- Nonlinear optics.
- Einstein’s field equations (gravitation in general relativity).
- Heat radiation and nonlinear heat conduction.
- Fracture mechanics.
- ______________________________

2 Lest we think this is merely an inconvenience, we should keep in mind that it is actually the nonlinearity that makes many phenomena useful. For instance, the ______________ depends on the nonlinearity of its optics. Similarly, transistors and the digital circuits made thereby (including the microprocessor) wouldn’t function if their physics were linear.

[1] As is customary, we frequently say “system” when we mean “mathematical system model.” Recall that multiple models may be used for any given physical system, depending on what one wants to know.

3 In this chapter, we will see some ways to formulate, characterize, and simulate nonlinear systems. Purely ____________________________ are few for nonlinear systems. Most are beyond the scope of this text, but we describe a few, mostly in Lec. nlin.char. Simulation via numerical integration of nonlinear dynamical equations is the most accessible technique, so it is introduced.

4 We skip a discussion of linearization; of course, if this is an option, it is preferable. Instead, we focus on the

_________________________.

5 For a good introduction to nonlinear dynamics, see Strogatz and Dichter.[2] A more engineer-oriented introduction is Kolk and Lerman.[3]

[2] S.H. Strogatz and M. Dichter. Nonlinear Dynamics and Chaos. Second. Studies in Nonlinearity. Avalon Publishing, 2016. isbn: 9780813350844.

[3] W. Richard Kolk and Robert A. Lerman. Nonlinear System Dynamics. 1 edition. Springer US, 1993. isbn: 978-1-4684-6496-2.

nlin.ss Nonlinear state–space models

1 A state-space model has the general form

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},t) \tag{1a}$$

$$\mathbf{y} = \underline{\qquad\qquad\qquad} \tag{1b}$$

where $\mathbf{f}$ and $\mathbf{g}$ are vector-valued functions that depend on the system. Nonlinear state-space models are those for which $\mathbf{f}$ is a ________________ functional of either $\mathbf{x}$ or $\mathbf{u}$. For instance, a state variable x_1 might appear as x_1^2 or two state variables might combine as $x_1 x_2$ or an input u_1 might enter the equations as $\log u_1$.

nonlinear state–space models

Autonomous and nonautonomous systems

2 An autonomous system is one for which $\mathbf{f}(\mathbf{x})$, with neither time nor input appearing explicitly. A nonautonomous system is one for which either t or $\mathbf{u}$ do appear explicitly in $\mathbf{f}$. It turns out that we can always write nonautonomous systems as autonomous by substituting in $\mathbf{u}(t)$ and introducing an extra ____________________ for t[4].

autonomous system

nonautonomous system

3 Therefore, without loss of generality, we will focus on ways of analyzing autonomous systems.

Equilibrium

4 An equilibrium state (also called a ______________________) $\overline{\mathbf{x}}$ is one for which $d\mathbf{x}/dt = \mathbf{0}$. In most cases, this occurs only when the input $\mathbf{u}$ is a constant $\overline{\mathbf{u}}$ and, for time-varying systems, at a given time $\bar{t}$. For autonomous systems, equilibrium occurs when the following holds:

equilibrium state

stationary point

$$\underline{\qquad\qquad\qquad} \tag{2}$$

This is a system of nonlinear algebraic equations, which can be challenging to solve for

[4] Strogatz and Dichter, Nonlinear Dynamics and Chaos.

$\bar{x}$. However, frequently, several solutions—that is, equilibrium states—do exist.

nlin.char Nonlinear system characteristics

1 Characterizing nonlinear systems can be challenging without the tools developed for ______________ system characterization. However, there are ways of characterizing nonlinear systems, and we'll here explore a few.

Those in-common with linear systems

system order

2 As with linear systems, the system order is either the number of state-variables required to describe the system or, equivalently, the highest-order __________________ in a single scalar differential equation describing the system.

3 Similarly, nonlinear systems can have state variables that depend on _____________ alone or those that also depend on ______________ (or some other independent variable). The former lead to ordinary differential equations (ODEs) and the latter to partial differential equations (PDEs).

4 Equilibrium was already considered in Lec. nlin.ss.

Stability

5 In terms of system performance, perhaps no other criterion is as important as ________________.

Definition nlin.1: Stability

If x is perturbed from an equilibrium state $\bar{x}$, the response $x(t)$ can:

1. asymptotically return to $\bar{x}$ (asymptotically ______________),
2. diverge from $\bar{x}$ (________________), or
3. remain perturned or oscillate about $\bar{x}$ with a constant amplitude (___________________ stable).

Notice that this definition is actually local: stability in the neighborhood of one equilibrium may not be the same as in the neighborhood of another.

6 Other than nonlinear systems' lack of linear systems' eigenvalues, poles, and roots of the characteristic equation from which to compute it, the primary difference between the stability of linear and nonlinear systems is that nonlinear system stability is often difficult to establish ________________. Using a linear system's eigenvalues, it is straightforward to establish stable, unstable, and marginally stable subspaces of state-space (via transforming to an eigenvector basis). For nonlinear systems, no such method exists. However, we are not without tools to explore nonlinear system stability. One mathematical tool to consider is _________________________________, which is beyond the scope of this course, but has good treatments in[5] and[6].

Lyapunov stability theory

Qualities of equilibria

7 Equilibria (i.e. stationary points) come in a variety of qualities. It is instructive to consider the first-order differential equation in state variable __________ with real constant __________:

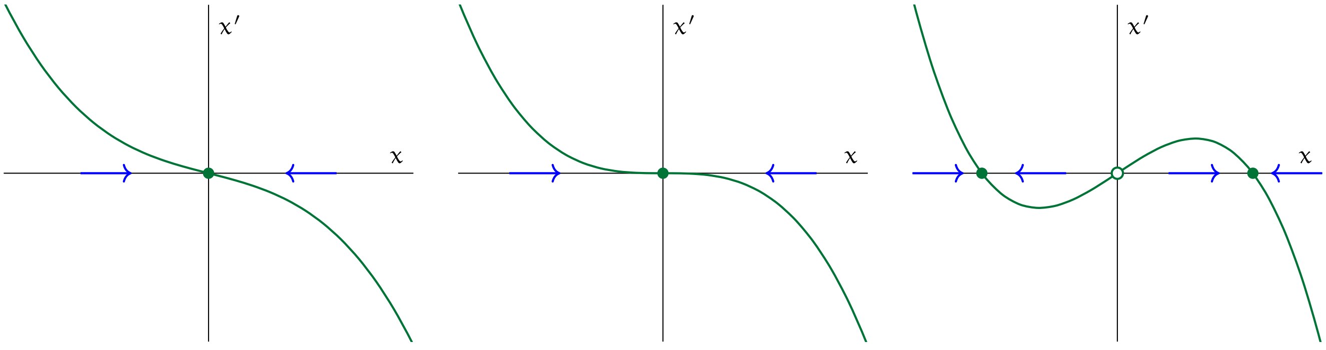
$$x' = rx - x^3. \tag{1}$$

If we plot x' versus x for different values of r, we obtain the plots of Fig. char.1.

8 By definition, equilibria occur when $x' = 0$, so the x-axis crossings of Fig. char.1 are equilibria. The blue arrows on the x-axis show the _______________________ of state change x', quantified by the plots. For both (a) and (b), only one equilibrium exists: $x = 0$. Note that the blue arrows in both plots point toward the equilibrium. In such cases—that is, when a ______________________ exists around an

[5] William L Brogan. Modern Control Theory. Third. Prentice Hall, 1991, Ch. 10.

[6] A. Choukchou-Braham andothers. Analysis and Control of Underactuated Mechanical Systems. SpringerLink : Bücher. Springer International Publishing, 2013. isbn: 9783319026367, App. A.



(a) $r < 0$ (b) $r = 0$ (c) $r > 0$

Figure char.1: plots of x' versus x for Eq. 1.

equilibrium for which state changes point toward the equilibrium—the equilibrium is called an ______________ or ___________. Note that attractors are ____________.

attractor
sink
stable

9 Now consider (c) of Fig. char.1. When $r > 0$, three equilibria emerge. This change of the number of equilibria with the changing of a parameter is called a ________________. A plot of bifurcations versus the parameter is called a bifurcation diagram. The $x = 0$ equilibrium now has arrows that point ____________ from it. Such an equilibrium is called a ______________ or _____________ and is ______________. The other two equilibria here are (stable) attractors. Consider a very small initial condition $x(0) = \epsilon$. If $\epsilon > 0$, the repeller pushes away x and the positive attractor pulls x to itself. Conversely, if $\epsilon < 0$, the repeller again pushes away x and the negative attractor pulls x to itself.

bifurcation

bifurcation diagram

repeller
source
unstable

10 Another type of equilibrium is called the _____________: one which acts as an attractor along some lines and as a repeller along others. We will see this type in the following example.

saddle

Example nlin.char−1 re: Saddle bifurcation

Consider the dynamical equation

$$x' = x^2 + r \qquad (2)$$

with r a real constant. Sketch x' vs x for negative, zero, and positive r. Identify and classify each of the equilibria.

nlin.sim Nonlinear system simulation

nlin.pysim Simulating nonlinear systems in Python

Example nlin.pysim−1 re: a nonlinear unicycle

asdf

First, load some Python packages.

```
import numpy as np
import sympy as sp
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, Latex
```

The state equation can be encoded via the following function f.

```
def f(t, x, u, c):
  dxdt = [
    x[3]*np.cos(x[2]),
    x[3]*np.sin(x[2]),
    x[4],
    1/c[0] * u(t)[0],
    1/c[1] * u(t)[1]
  ]
  return dxdt
```

The input function u must also be defined.

```
def u(t):
  return [
    15*(1+np.cos(t)),
    25*np.sin(3*t)
  ]
```

```
# %% Define time spans, initial values, and constants
tspan = np.linspace(0, 50, 300)
xinit = [0,0,0,0,0]
mass = 10
inertia = 10
c = [mass,inertia]

# %% Solve differential equation
sol = solve_ivp(
  lambda t, x: f(t, x, u, c),
  [tspan[0], tspan[-1]],
  xinit,
  t_eval=tspan
)
```

Let’s first plot the trajectory and instantaneous velocity.

```
xp = sol.y[3]*np.cos(sol.y[2])
yp = sol.y[3]*np.sin(sol.y[2])
p = plt.figure();
plt.plot(sol.y[0],sol.y[1])
plt.quiver(sol.y[0],sol.y[1],xp,yp)
plt.xlabel('$x$')
plt.ylabel('$y$')
plt.show()
```

. Python code in this section was generated from a Jupyter notebook named `horcrux.ipynb` with a `python3` kernel.

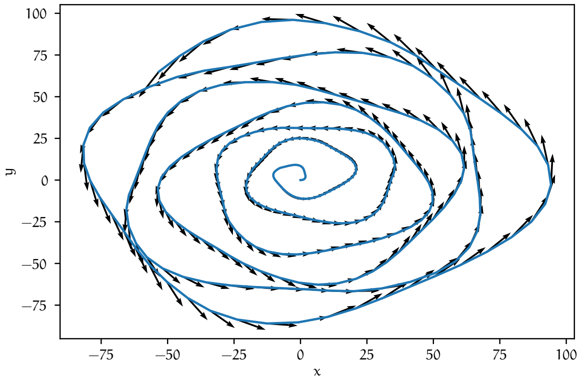


Figure pysim.1:

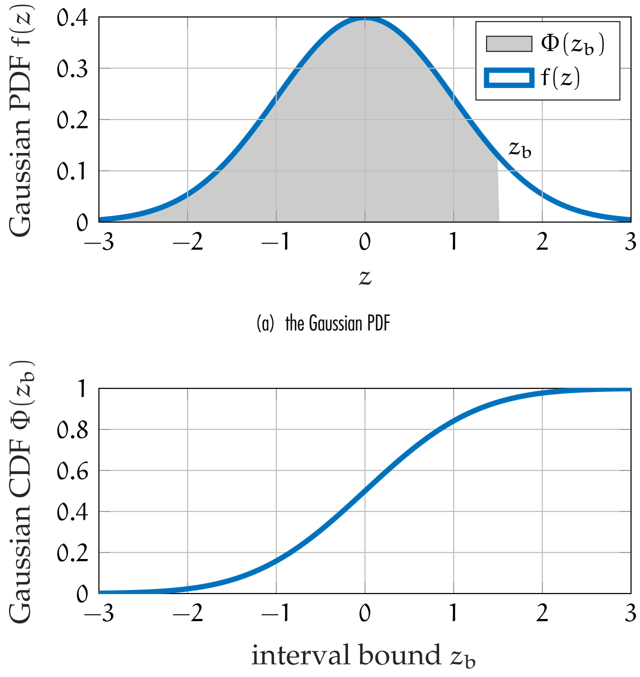
nlin.exe Exercises for Chapter nlin

tblstat

Distribution tables

A.01 Gaussian distribution table

Below are plots of the Gaussian probability density function f and cumulative distribution function Φ. Below them is Table guass.1 of CDF values.



(a) the Gaussian PDF

(b) the Gaussian CDF

Figure guass.1: the Gaussian PDF and CDF for z-scores.

Table guass.1: z-score table $\Phi(z_b) = P(z \in (-\infty, z_b])$.

z_b	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9
.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0002
.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
.1	0.0010	0.0009	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143

Table guass.1: z-score table $\Phi(z_b) = P(z \in (-\infty, z_b])$.

z_b	. 0	. 1	. 2	. 3	. 4	. 5	. 6	. 7	. 8	. 9
.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993

Table guass.1: z-score table $\Phi(z_b) = P(z \in (-\infty, z_b])$.

z_b	. 0	. 1	. 2	. 3	. 4	. 5	. 6	. 7	. 8	. 9
.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998

A.02 Student's t−distribution table

Table t.1: two-tail inverse student's t-distribution table.

	percent probability										
ν	60.0	66.7	75.0	80.0	87.5	90.0	95.0	97.5	99.0	99.5	99.9
1	0.325	0.577	1.000	1.376	2.414	3.078	6.314	12.706	31.821	63.657	318.31
2	0.289	0.500	0.816	1.061	1.604	1.886	2.920	4.303	6.965	9.925	22.327
3	0.277	0.476	0.765	0.978	1.423	1.638	2.353	3.182	4.541	5.841	10.215
4	0.271	0.464	0.741	0.941	1.344	1.533	2.132	2.776	3.747	4.604	7.173
5	0.267	0.457	0.727	0.920	1.301	1.476	2.015	2.571	3.365	4.032	5.893
6	0.265	0.453	0.718	0.906	1.273	1.440	1.943	2.447	3.143	3.707	5.208
7	0.263	0.449	0.711	0.896	1.254	1.415	1.895	2.365	2.998	3.499	4.785
8	0.262	0.447	0.706	0.889	1.240	1.397	1.860	2.306	2.896	3.355	4.501
9	0.261	0.445	0.703	0.883	1.230	1.383	1.833	2.262	2.821	3.250	4.297
10	0.260	0.444	0.700	0.879	1.221	1.372	1.812	2.228	2.764	3.169	4.144
11	0.260	0.443	0.697	0.876	1.214	1.363	1.796	2.201	2.718	3.106	4.025
12	0.259	0.442	0.695	0.873	1.209	1.356	1.782	2.179	2.681	3.055	3.930
13	0.259	0.441	0.694	0.870	1.204	1.350	1.771	2.160	2.650	3.012	3.852
14	0.258	0.440	0.692	0.868	1.200	1.345	1.761	2.145	2.624	2.977	3.787
15	0.258	0.439	0.691	0.866	1.197	1.341	1.753	2.131	2.602	2.947	3.733
16	0.258	0.439	0.690	0.865	1.194	1.337	1.746	2.120	2.583	2.921	3.686
17	0.257	0.438	0.689	0.863	1.191	1.333	1.740	2.110	2.567	2.898	3.646
18	0.257	0.438	0.688	0.862	1.189	1.330	1.734	2.101	2.552	2.878	3.610
19	0.257	0.438	0.688	0.861	1.187	1.328	1.729	2.093	2.539	2.861	3.579
20	0.257	0.437	0.687	0.860	1.185	1.325	1.725	2.086	2.528	2.845	3.552
21	0.257	0.437	0.686	0.859	1.183	1.323	1.721	2.080	2.518	2.831	3.527
22	0.256	0.437	0.686	0.858	1.182	1.321	1.717	2.074	2.508	2.819	3.505
23	0.256	0.436	0.685	0.858	1.180	1.319	1.714	2.069	2.500	2.807	3.485
24	0.256	0.436	0.685	0.857	1.179	1.318	1.711	2.064	2.492	2.797	3.467
25	0.256	0.436	0.684	0.856	1.178	1.316	1.708	2.060	2.485	2.787	3.450
26	0.256	0.436	0.684	0.856	1.177	1.315	1.706	2.056	2.479	2.779	3.435
27	0.256	0.435	0.684	0.855	1.176	1.314	1.703	2.052	2.473	2.771	3.421
28	0.256	0.435	0.683	0.855	1.175	1.313	1.701	2.048	2.467	2.763	3.408
29	0.256	0.435	0.683	0.854	1.174	1.311	1.699	2.045	2.462	2.756	3.396
30	0.256	0.435	0.683	0.854	1.173	1.310	1.697	2.042	2.457	2.750	3.385
35	0.255	0.434	0.682	0.852	1.170	1.306	1.690	2.030	2.438	2.724	3.340
40	0.255	0.434	0.681	0.851	1.167	1.303	1.684	2.021	2.423	2.704	3.307
45	0.255	0.434	0.680	0.850	1.165	1.301	1.679	2.014	2.412	2.690	3.281
50	0.255	0.433	0.679	0.849	1.164	1.299	1.676	2.009	2.403	2.678	3.261
55	0.255	0.433	0.679	0.848	1.163	1.297	1.673	2.004	2.396	2.668	3.245
60	0.254	0.433	0.679	0.848	1.162	1.296	1.671	2.000	2.390	2.660	3.232
∞	0.253	0.431	0.674	0.842	1.150	1.282	1.645	1.960	2.326	2.576	3.090

tbltx

Fourier and Laplace tables

B.01 Laplace transforms

Table lap.1 is a table with functions of time $f(t)$ on the left and corresponding Laplace transforms $L(s)$ on the right. Where applicable, $s = \sigma + j\omega$ is the Laplace transform variable, T is the time-domain period, $\omega_0 2\pi/T$ is the corresponding angular frequency, $j = \sqrt{-1}$, $a \in \mathbb{R}^+$, and $b, t_0 \in \mathbb{R}$ are constants.

Table lap.1: Laplace transform identities.

function of time t	function of Laplace s
$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(s) + a_2 F_2(s)$
$f(t - t_0)$	$F(s)e^{-t_0 s}$
$f'(t)$	$sF(s) - f(0)$
$\dfrac{d^n f(t)}{dt^n}$	$s^n F(s) + s^{(n-1)} f(0) + s^{(n-2)} f'(0) + \cdots + f^{(n-1)}(0)$
$\displaystyle\int_0^t f(\tau) d\tau$	$\dfrac{1}{s} F(s)$
$tf(t)$	$-F'(s)$
$f_1(t) * f_2(t) = \displaystyle\int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$	$F_1(s) F_2(s)$
$\delta(t)$	1
$u_s(t)$	$1/s$
$u_r(t)$	$1/s^2$
$t^{n-1}/(n-1)!$	$1/s^n$
e^{-at}	$\dfrac{1}{s+a}$
te^{-at}	$\dfrac{1}{(s+a)^2}$

$\frac{1}{(n-1)!}t^{n-1}e^{-at}$	$\frac{1}{(s+a)^n}$	
$\frac{1}{a-b}(e^{at}-e^{bt})$	$\frac{1}{(s-a)(s-b)}$	$(a \neq b)$
$\frac{1}{a-b}(ae^{at}-be^{bt})$	$\frac{s}{(s-a)(s-b)}$	$(a \neq b)$
$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$	
$\cos \omega t$	$\frac{s}{s^2+\omega^2}$	
$e^{at}\sin \omega t$	$\frac{\omega}{(s-a)^2+\omega^2}$	
$e^{at}\cos \omega t$	$\frac{s-a}{(s-a)^2+\omega^2}$	

B.02 Fourier transforms

Table four.1 is a table with functions of time $f(t)$ on the left and corresponding Fourier transforms $F(\omega)$ on the right. Where applicable, T is the time-domain period, $\omega_0 2\pi/T$ is the corresponding angular frequency, $j = \sqrt{-1}$, $a \in \mathbb{R}^+$, and $b, t_0 \in \mathbb{R}$ are constants. Furthermore, f_e and f_0 are even and odd functions of time, respectively, and it can be shown that any function f can be written as the sum $f(t) = f_e(t) + f_0(t)$. (Hsu1967)

Table four.1: Fourier transform identities.

function of time t	function of frequency ω
$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(\omega) + a_2 F_2(\omega)$
$f(at)$	$\dfrac{1}{\lvert a\rvert} F(\omega/a)$
$f(-t)$	$F(-\omega)$
$f(t - t_0)$	$F(\omega) e^{-j\omega t_0}$
$f(t) \cos \omega_0 t$	$\dfrac{1}{2} F(\omega - \omega_0) + \dfrac{1}{2} F(\omega + \omega_0)$
$f(t) \sin \omega_0 t$	$\dfrac{1}{j2} F(\omega - \omega_0) - \dfrac{1}{j2} F(\omega + \omega_0)$
$f_e(t)$	$\operatorname{Re} F(\omega)$
$f_0(t)$	$j \operatorname{Im} F(\omega)$
$F(t)$	$2\pi f(-\omega)$
$f'(t)$	$j\omega F(\omega)$
$\dfrac{d^n f(t)}{dt^n}$	$(j\omega)^n F(\omega)$

$\int_{-\infty}^{t} f(\tau)d\tau$	$\frac{1}{j\omega}F(\omega) + \pi F(0)\delta(\omega)$
$-jtf(t)$	$F'(\omega)$
$(-jt)^n f(t)$	$\frac{d^n F(\omega)}{d\omega^n}$
$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau)f_2(t-\tau)d\tau$	$F_1(\omega)F_2(\omega)$
$f_1(t)f_2(t)$	$\frac{1}{2\pi}F_1(\omega) * F_2(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(\alpha)F_2(\omega-\alpha)d\alpha$
$e^{-at}u_s(t)$	$\frac{1}{j\omega + a}$
$e^{-a\|t\|}$	$\frac{2a}{a^2+\omega^2}$
e^{-at^2}	$\sqrt{\pi/a}\, e^{-\omega^2/(4a)}$
1 for $\|t\| < a/2$, else 0	$\frac{a\sin(a\omega/2)}{a\omega/2}$
$te^{-at}u_s(t)$	$\frac{1}{(a+j\omega)^2}$
$\frac{t^{n-1}}{(n-1)!}e^{-at})^n u_s(t)$	$\frac{1}{(a+j\omega)^n}$
$\frac{1}{a^2+t^2}$	$\frac{\pi}{a}e^{-a\|\omega\|}$
$\delta(t)$	1
$\delta(t-t_0)$	$e^{-j\omega t_0}$
$u_s(t)$	$\pi\delta(\omega) + \frac{1}{j\omega}$
$u_s(t-t_0)$	$\pi\delta(\omega) + \frac{1}{j\omega}e^{-j\omega t_0}$
1	$2\pi\delta(\omega)$
t	$2\pi j\delta'(\omega)$
t^n	$2\pi j^n \frac{d^n\delta(\omega)}{d\omega^n}$
$e^{j\omega_0 t}$	$2\pi\delta(\omega-\omega_0)$

$\cos \omega_0 t$	$\pi\delta(\omega - \omega_0) + \pi\delta(\omega + \omega_0)$
$\sin \omega_0 t$	$-j\pi\delta(\omega - \omega_0) + j\pi\delta(\omega + \omega_0)$
$u_s(t) \cos \omega_0 t$	$\dfrac{j\omega}{\omega_0^2 - \omega^2} + \dfrac{\pi}{2}\delta(\omega - \omega_0) + \dfrac{\pi}{2}\delta(\omega + \omega_0)$
$u_s(t) \sin \omega_0 t$	$\dfrac{\omega_0}{\omega_0^2 - \omega^2} + \dfrac{\pi}{2j}\delta(\omega - \omega_0) - \dfrac{\pi}{2j}\delta(\omega + \omega_0)$
$tu_s(t)$	$j\pi\delta'(\omega) - 1/\omega^2$
$1/t$	$\pi j - 2\pi j u_s(\omega)$
$1/t^n$	$\dfrac{(-j\omega)^{n-1}}{(n-1)!}\left(\pi j - 2\pi j u_s(\omega)\right)$
$\operatorname{sgn} t$	$\dfrac{2}{j\omega}$
$\displaystyle\sum_{n=-\infty}^{\infty} \delta(t - nT)$	$\displaystyle\omega_0 \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_0)$

mref

Mathematics reference

C.01 Quadratic forms

The solution to equations of the form $ax^2 + bx + c = 0$ is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \qquad (1)$$

Completing the square

This is accomplished by re-writing the quadratic formula in the form of the left-hand-side (LHS) of this equality, which describes factorization

$$x^2 + 2xh + h^2 = (x + h)^2. \qquad (2)$$

C.02 Trigonometry

Triangle identities

With reference to the below figure, the law of sines is

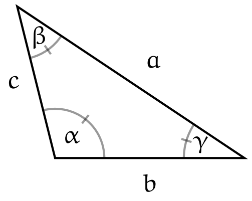
$$\frac{\sin\alpha}{a} = \frac{\sin\beta}{b} = \frac{\sin\gamma}{c} \tag{1}$$

and the law of cosines is

$$c^2 = a^2 + b^2 - 2ab\cos\gamma \tag{2a}$$

$$b^2 = a^2 + c^2 - 2ac\cos\beta \tag{2b}$$

$$a^2 = c^2 + b^2 - 2cb\cos\alpha \tag{2c}$$



Reciprocal identities

$$\csc u = \frac{1}{\sin u} \tag{3a}$$

$$\sec u = \frac{1}{\cos u} \tag{3b}$$

$$\cot u = \frac{1}{\tan u} \tag{3c}$$

Pythagorean identities

$$1 = \sin^2 u + \cos^2 u \tag{4a}$$

$$\sec^2 u = 1 + \tan^2 u \tag{4b}$$

$$\csc^2 u = 1 + \cot^2 u \tag{4c}$$

Co-function identities

$$\sin\left(\frac{\pi}{2}-u\right)=\cos u \tag{5a}$$

$$\cos\left(\frac{\pi}{2}-u\right)=\sin u \tag{5b}$$

$$\tan\left(\frac{\pi}{2}-u\right)=\cot u \tag{5c}$$

$$\csc\left(\frac{\pi}{2}-u\right)=\sec u \tag{5d}$$

$$\sec\left(\frac{\pi}{2}-u\right)=\csc u \tag{5e}$$

$$\cot\left(\frac{\pi}{2}-u\right)=\tan u \tag{5f}$$

Even-odd identities

$$\sin(-u)=-\sin u \tag{6a}$$

$$\cos(-u)=\cos u \tag{6b}$$

$$\tan(-u)=-\tan u \tag{6c}$$

Sum-difference formulas (AM or lock-in)

$$\sin(u\pm v)=\sin u\cos v\pm\cos u\sin v \tag{7a}$$

$$\cos(u\pm v)=\cos u\cos v\mp\sin u\sin v \tag{7b}$$

$$\tan(u\pm v)=\frac{\tan u\pm\tan v}{1\mp\tan u\tan v} \tag{7c}$$

Double angle formulas

$$\sin(2u)=2\sin u\cos u \tag{8a}$$

$$\cos(2u)=\cos^2 u-\sin^2 u \tag{8b}$$

$$=2\cos^2 u-1 \tag{8c}$$

$$=1-2\sin^2 u \tag{8d}$$

$$\tan(2u)=\frac{2\tan u}{1-\tan^2 u} \tag{8e}$$

Power-reducing or half-angle formulas

$$\sin^2 u=\frac{1-\cos(2u)}{2} \tag{9a}$$

$$\cos^2 u=\frac{1+\cos(2u)}{2} \tag{9b}$$

$$\tan^2 u=\frac{1-\cos(2u)}{1+\cos(2u)} \tag{9c}$$

Sum-to-product formulas

$$\sin u + \sin v = 2 \sin \frac{u+v}{2} \cos \frac{u-v}{2} \tag{10a}$$

$$\sin u - \sin v = 2 \cos \frac{u+v}{2} \sin \frac{u-v}{2} \tag{10b}$$

$$\cos u + \cos v = 2 \cos \frac{u+v}{2} \cos \frac{u-v}{2} \tag{10c}$$

$$\cos u - \cos v = -2 \sin \frac{u+v}{2} \sin \frac{u-v}{2} \tag{10d}$$

Product-to-sum formulas

$$\sin u \sin v = \frac{1}{2} [\cos(u-v) - \cos(u+v)] \tag{11a}$$

$$\cos u \cos v = \frac{1}{2} [\cos(u-v) + \cos(u+v)] \tag{11b}$$

$$\sin u \cos v = \frac{1}{2} [\sin(u+v) + \sin(u-v)] \tag{11c}$$

$$\cos u \sin v = \frac{1}{2} [\sin(u+v) - \sin(u-v)] \tag{11d}$$

Two-to-one formulas

$$A \sin u + B \cos u = C \sin(u+\phi) \tag{12a}$$

$$= C \cos(u+\psi) \text{ where} \tag{12b}$$

$$C = \sqrt{A^2 + B^2} \tag{12c}$$

$$\phi = \arctan \frac{B}{A} \tag{12d}$$

$$\psi = -\arctan \frac{A}{B} \tag{12e}$$

C.03 Matrix inverses

This is a guide to inverting 1×1, 2×2, and $n \times n$ matrices.

Let A be the 1×1 matrix

$$A = \begin{bmatrix} a \end{bmatrix}.$$

The inverse is simply the reciprocal:

$$A^{-1} = \begin{bmatrix} 1/a \end{bmatrix}.$$

Let B be the 2×2 matrix

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

It can be shown that the inverse follows a simple pattern:

$$\begin{aligned} B^{-1} &= \frac{1}{\det B} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix} \\ &= \frac{1}{b_{11}b_{22} - b_{12}b_{21}} \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix}. \end{aligned}$$

Let C be an $n \times n$ matrix. It can be shown that its inverse is

$$C^{-1} = \frac{1}{\det C} \operatorname{adj} C,$$

where adj is the adjoint of C.

adjoint

C.04 Laplace transforms

The definition of the one-side Laplace and inverse Laplace transforms follow.

Definition C.1: Laplace transforms (one-sided)

Laplace transform $\mathcal{L}$:

$$\mathcal{L}(y(t)) = Y(s) = \int_0^\infty y(t)e^{-st}dt. \tag{1}$$

Inverse Laplace transform $\mathcal{L}^{-1}$:

$$\mathcal{L}^{-1}(Y(s)) = y(t) = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} Y(s)e^{st}ds. \tag{2}$$

See Table lap.1 for a list of properties and common transforms.

can

Complex analysis

D.01 Euler's formulas

Euler's formula

Euler's formula is our bridge back-and forth between trigonomentric forms ($\cos\theta$ and $\sin\theta$) and complex exponential form ($e^{j\theta}$):

$$e^{j\theta} = \cos\theta + j\sin\theta. \tag{1}$$

Here are a few useful identities implied by Euler's formula.

$$\begin{aligned}
e^{-j\theta} &= \cos\theta - j\sin\theta & \text{(2a)}\\
\cos\theta &= \operatorname{Re}\left(e^{j\theta}\right) & \text{(2b)}\\
&= \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right) & \text{(2c)}\\
\sin\theta &= \operatorname{Im}\left(e^{j\theta}\right) & \text{(2d)}\\
&= \frac{1}{j2}\left(e^{j\theta} - e^{-j\theta}\right). & \text{(2e)}
\end{aligned}$$

Bibliography

[1] Robert B. Ash. Basic Probability Theory. Dover Publications, Inc., 2008.

[2] Joan Bagaria. ?Set Theory? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2019. Metaphysics Research Lab, Stanford University, 2019.

[3] Maria Baghramian and J. Adam Carter. ?Relativism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2019. Metaphysics Research Lab, Stanford University, 2019.

[4] Stephen Barker and Mark Jago. ?Being Positive About Negative Facts? inPhilosophy and Phenomenological Research: 85.1 (2012), pages 117–138. doi: 10.1111/j.1933-1592.2010.00479.x.

[5] Jonathan Barzilai and Jonathan M. Borwein. ?Two-Point Step Size Gradient Methods? inIMA Journal of Numerical Analysis: 8.1 (january 1988), pages 141–148. issn: 0272-4979. doi: 10.1093/imanum/8.1.141. This includes an innovative line search method.

[6] Anat Biletzki and Anat Matar. ?Ludwig Wittgenstein? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford

University, 2018. An introduction to Wittgenstein and his thought.

[7] Antonio Bove, F. (Ferruccio) Colombini and Daniele Del Santo. Phase space analysis of partial differential equations. eng. Progress in nonlinear differential equations and their applications ; v. 69. Boston ; Berlin: Birkhäuser, 2006. isbn: 9780817645212.

[8] William L Brogan. Modern Control Theory. Third. Prentice Hall, 1991.

[9] Francesco Bullo and Andrew D. Lewis. Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems. byeditorJ.E. Marsden, L. Sirovich and M. Golubitsky. Springer, 2005.

[10] Francesco Bullo and Andrew D. Lewis. Supplementary Chapters for Geometric Control of Mechanical Systems[1]. january 2005.

[11] A. Choukchou-Braham andothers. Analysis and Control of Underactuated Mechanical Systems. SpringerLink : Bücher. Springer International Publishing, 2013. isbn: 9783319026367.

[12] K. Ciesielski. Set Theory for the Working Mathematician. London Mathematical Society Student Texts. Cambridge University Press, 1997. isbn: 9780521594653. A readable introduction to set theory.

[13] Marian David. ?The Correspondence Theory of Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2016. Metaphysics Research Lab, Stanford University, 2016. A detailed overview of the correspondence theory of truth.

[1] FB/ADL:04.

[14] David Dolby. ?Wittgenstein on Truth? inA Companion to Wittgenstein: John Wiley & Sons, Ltd, 2016. chapter 27, pages 433–442. isbn: 9781118884607. doi: 10.1002/9781118884607.ch27.

[15] Peter J. Eccles. An Introduction to Mathematical Reasoning: Numbers, Sets and Functions. Cambridge University Press, 1997. doi: 10.1017/CBO9780511801136. A gentle introduction to mathematical reasoning. It includes introductory treatments of set theory and number systems.

[16] H.B. Enderton. Elements of Set Theory. Elsevier Science, 1977. isbn: 9780080570426. A gentle introduction to set theory and mathematical reasoning—a great place to start.

[17] Richard P. Feynman, Robert B. Leighton and Matthew Sands. The Feynman Lectures on Physics. New Millennium. Perseus Basic Books, 2010.

[18] Michael Glanzberg. ?Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018.

[19] Hans Johann Glock. ?Truth in the Tractatus? inSynthese: 148.2 (january 2006), pages 345–368. issn: 1573-0964. doi: 10.1007/s11229-004-6226-2.

[20] Mario Gómez-Torrente. ?Alfred Tarski? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019.

[21] Paul Guyer and Rolf-Peter Horstmann. ?Idealism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta.

Winter 2018. Metaphysics Research Lab, Stanford University, 2018.

[22] R. Haberman. Applied Partial Differential Equations with Fourier Series and Boundary Value Problems (Classic Version). Pearson Modern Classics for Advanced Mathematics. Pearson Education Canada, 2018. isbn: 9780134995434.

[23] G.W.F. Hegel and A.V. Miller. Phenomenology of Spirit. Motilal Banarsidass, 1998. isbn: 9788120814738.

[24] Wilfrid Hodges. ?Model Theory? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018.

[25] Wilfrid Hodges. ?Tarski's Truth Definitions? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018.

[26] Peter Hylton and Gary Kemp. ?Willard Van Orman Quine? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019.

[27] E.T. Jaynes andothers. Probability Theory: The Logic of Science. Cambridge University Press, 2003. isbn: 9780521592710. An excellent and comprehensive introduction to probability theory.

[28] I. Kant, P. Guyer and A.W. Wood. Critique of Pure Reason. The Cambridge Edition of the Works of Immanuel Kant. Cambridge University Press, 1999. isbn: 9781107268333.

[29] Juliette Kennedy. ?Kurt Gödel? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2018. Metaphysics Research Lab, Stanford University, 2018.

[30] Drew Khlentzos. ?Challenges to Metaphysical Realism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University, 2016.

[31] Peter Klein. ?Skepticism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Summer 2015. Metaphysics Research Lab, Stanford University, 2015.

[32] M. Kline. Mathematics: The Loss of Certainty. A Galaxy book. Oxford University Press, 1982. isbn: 9780195030853. A detailed account of the "illogical" development of mathematics and an exposition of its therefore remarkable utility in describing the world.

[33] W. Richard Kolk and Robert A. Lerman. Nonlinear System Dynamics. 1 edition. Springer US, 1993. isbn: 978-1-4684-6496-2.

[34] Erwin Kreyszig. Advanced Engineering Mathematics. 10th. John Wiley & Sons, Limited, 2011. isbn: 9781119571094. The authoritative resource for engineering mathematics. It includes detailed accounts of probability, statistics, vector calculus, linear algebra, fourier analysis, ordinary and partial differential equations, and complex analysis. It also includes several other topics with varying degrees of depth. Overall, it is the best place to start when seeking mathematical guidance.

[35] John M. Lee. Introduction to Smooth Manifolds. second. volume 218. Graduate Texts in Mathematics. Springer, 2012.

[36] Catherine Legg and Christopher Hookway. ?Pragmatism? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Spring 2019. Metaphysics Research Lab, Stanford University, 2019. An introductory article on the philsophical movement "pragmatism." It includes an important clarification of the pragmatic slogan, "truth is the end of inquiry."

[37] Daniel Liberzon. Calculus of Variations and Optimal Control Theory: A Concise Introduction. Princeton University Press, 2012. isbn: 9780691151878.

[38] Panu Raatikainen. ?Gödel's Incompleteness Theorems? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018. A through and contemporary description of Gödel's incompleteness theorems, which have significant implications for the foundations and function of mathematics and mathematical truth.

[39] Paul Redding. ?Georg Wilhelm Friedrich Hegel? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Summer 2018. Metaphysics Research Lab, Stanford University, 2018.

[40] H.M. Schey. Div, Grad, Curl, and All that: An Informal Text on Vector Calculus. W.W. Norton, 2005. isbn: 9780393925166.

[41] Christopher Shields. ?Aristotle? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2016.

Metaphysics Research Lab, Stanford University, 2016.

[42] Steven S. Skiena. Calculated Bets: Computers, Gambling, and Mathematical Modeling to Win. Outlooks. Cambridge University Press, 2001. doi: 10.1017/CBO9780511547089. This includes a lucid section on probability versus statistics, also available here: https://www3.cs.stonybrook.edu/~skiena/jaialai/excerpts/node12.html.

[43] George Smith. ?Isaac Newton? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2008. Metaphysics Research Lab, Stanford University, 2008.

[44] Daniel Stoljar and Nic Damnjanovic. ?The Deflationary Theory of Truth? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Fall 2014. Metaphysics Research Lab, Stanford University, 2014.

[45] W.A. Strauss. Partial Differential Equations: An Introduction. Wiley, 2007. isbn: 9780470054567. A thorough and yet relatively compact introduction.

[46] S.H. Strogatz and M. Dichter. Nonlinear Dynamics and Chaos. Second. Studies in Nonlinearity. Avalon Publishing, 2016. isbn: 9780813350844.

[47] Mark Textor. ?States of Affairs? inThe Stanford Encyclopedia of Philosophy: byeditorEdward N. Zalta. Winter 2016. Metaphysics Research Lab, Stanford University, 2016.

[48] Pauli Virtanen andothers. ?SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python? inarXiv

e-prints: arXiv:1907.10121 (july 2019), arXiv:1907.10121.

[49] Wikipedia. Algebra — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Algebra&oldid=920573802. [Online; accessed 26-October-2019]. 2019.

[50] Wikipedia. Carl Friedrich Gauss — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Carl%20Friedrich%20Gauss&oldid=922692291. [Online; accessed 26-October-2019]. 2019.

[51] Wikipedia. Euclid — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Euclid&oldid=923031048. [Online; accessed 26-October-2019]. 2019.

[52] Wikipedia. First-order logic — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=First-order%20logic&oldid=921437906. [Online; accessed 29-October-2019]. 2019.

[53] Wikipedia. Fundamental interaction — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Fundamental%20interaction&oldid=925884124. [Online; accessed 16-November-2019]. 2019.

[54] Wikipedia. Leonhard Euler — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Leonhard%20Euler&oldid=921824700. [Online; accessed 26-October-2019]. 2019.

[55] Wikipedia. Linguistic turn—Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Linguistic%20turn&oldid=922305269.

[Online; accessed 23-October-2019]. 2019. Hey, we all do it.

[56] Wikipedia. Probability space — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Probability%20space&oldid=914939789. [Online; accessed 31-October-2019]. 2019.

[57] Wikipedia. Propositional calculus — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Propositional%20calculus&oldid=914757384. [Online; accessed 29-October-2019]. 2019.

[58] Wikipedia. Quaternion — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Quaternion&oldid=920710557. [Online; accessed 26-October-2019]. 2019.

[59] Wikipedia. Set-builder notation — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Set-builder%20notation&oldid=917328223. [Online; accessed 29-October-2019]. 2019.

[60] Wikipedia. William Rowan Hamilton — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=William%20Rowan%20Hamilton&oldid=923163451. [Online; accessed 26-October-2019]. 2019.

[61] L. Wittgenstein, P.M.S. Hacker and J. Schulte. Philosophical Investigations. Wiley, 2010. isbn: 9781444307979.

[62] Ludwig Wittgenstein. Tractatus Logico-Philosophicus. byeditorC.}, familyi=C., given=K. Ogden, giveni=. O. Project Gutenberg. International Library of Psychology Philosophy and Scientific Method. Kegan Paul, Trench, Trubner & Co., Ltd., 1922. A brilliant work on what is

possible to express in language—and what is not. As Wittgenstein puts it, “What can be said at all can be said clearly; and whereof one cannot speak thereof one must be silent.”

[63] Slavoj i ek. Less Than Nothing: Hegel and the Shadow of Dialectical Materialism. Verso, 2012. isbn: 9781844678976. This is one of the most interesting presentations of Hegel and Lacan by one of the most exciting contemporary philosophers.