The USC Primer for EE 539

W. V. Unglaub and A. F. J. Levi

October 4, 2024

Preface

Before studying applied quantum mechanics, reviewing elements of mathematics that support the ideas and concepts can be helpful. Quantum mechanics is a vast topic with many specialized sub-fields. *Applied* quantum mechanics is about practical things of value that can be put to work and contribute to economic activity. Naturally, such applications are a subset of what quantum mechanics offers. However, some mathematics, such as the linear algebra of non-commuting operators, is common to both applied quantum mechanics and the broader field of quantum mechanics. Other mathematical techniques, such as optimization, linear regression, and machine learning, are numerical tools often best suited for applications and quantum engineering.

The following is a brief review of mathematical techniques that you should already be familiar with and will find helpful if you attend the USC ECE class *Applied Quantum Mechanics*. [1] The material is organized in such a way that if you wish to explore a topic in greater depth, you can do so easily via the "Explore more" sections and, of course, by solving the homework problems.

Please note, this primer is not a substitute for the EE 539 experience!

[1] A. F. J. Levi, *Applied Quantum Mechanics*, 3rd ed. Cambridge: Cambridge University Press, 2023.

Contents

1 Numbers

Learning objectives:

- Understand the importance of abstraction and various number sets.
- Understand the algebra of imaginary and complex numbers.

A practical application of positive integer numbers described by the set of natural numbers $\mathbb{N}$ is counting classical physical objects. Experiments can be performed in which real objects are added or subtracted, and the result is observed as a measurement. If all the objects are taken away, nothing will be observed. Clearly, the idea of zero and negative integer numbers is not directly related to physical objects whose presence can be measured. These useful concepts are an *abstraction* - different from a physical object's measured presence.[2]

1.1 Real numbers

More than 2,400 years ago the Babylonians counted in base 60 and made use of a zero symbol when writing numbers, but it did not represent a zero in the same way it is today. Similarly, the Ancient Greeks and Romans did not use zero as an actual number. The first use of the number zero in the modern sense is documented in India in 628 AD. This is important because it requires a level of abstraction that allows the representation of a concept that cannot be observed in an experiment that, for example, counts physical objects. Negative numbers, such as the solution to the algebraic expression $4x + 32 = 0$, are likewise an abstraction that allows calculations of practical importance and development of predictive models. Positive and negative integers belong to the set of numbers labelled $\mathbb{Z}$. Division and multiplication, including division and multiplication by zero, add to the ability to develop predictive models. Rational ($\mathbb{Q}$) and irrational numbers ($\mathbb{I}$) follow. These various categories of numbers make up the set of *real* numbers ($\mathbb{R}$).

Explore More: Sets of real numbers

1.2 Complex numbers

In Euclidean geometry, calculating the area of a square of side a, where a is a positive number, gives $A = a \cdot a = a^2$ in which A is a positive number and $a = A^{1/2} = \sqrt{A}$. However, if the area is a negative number, $-A$, then $a = \mathrm{i}\sqrt{A}$ where $\mathrm{i} = \sqrt{-1}$ so that $-A = \mathrm{i}^2\sqrt{A}\sqrt{A} = -1 \cdot A = -A$.[3]

Notice, for two imaginary numbers a and b the order in which they are operated on matters so that, for example, $\sqrt{a}\sqrt{b} \neq \sqrt{ab}$ since $\sqrt{-1}\sqrt{-1} = \mathrm{i}^2 = -1$ which is not the same as $\sqrt{(-1)\cdot(-1)} = 1$. Imaginary numbers such as $\mathrm{i}y$ may be combined with real numbers such as x to form *complex* numbers ($\mathbb{C}$) through addition:

$$z = x + \mathrm{i}y = \mathrm{Re}(z) + \mathrm{i} \cdot \mathrm{Im}(z) \tag{1}$$

where the real part of the complex number z is $\mathrm{Re}(z) = x$ and the imaginary part is $\mathrm{Im}(z) = y$, where x and y are purely-real numbers. A schema of the various types of numbers is visually depicted in Fig. 1, in which the set of complex numbers $\mathbb{C}$ includes the sets of purely-real numbers $\mathbb{R}$ and purely-imaginary numbers $\mathrm{i}\mathbb{R}$. Various mathematical objects can be constructed from the set of complex numbers, allowing higher-level and often more helpful forms of abstraction. We explore these and their corresponding algebra in the following sections.

Complex numbers play a key role in the mathematical structure supporting quantum mechanics. However, it is worth noting that in contemporary quantum mechanics, the results of any physical measurement can *only* be real numbers. This means that the number i, and hence the set of pure *imaginary* numbers ($\mathrm{i}\mathbb{R}$), cannot be the result of a physically measured observable.

Explore More: Other generalized number systems

[2] For example, in semiconductor devices, the concept of positively charged holes is the *absence* of physical electrons with negative charge in the valence band.

[3] Carl Friedrich Gauss introduced the symbol i for the square root of minus one in 1831.

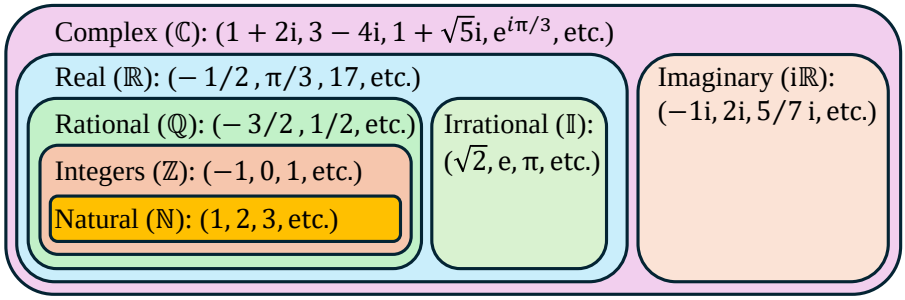


Figure 1: Classification schema of numbers with respect to their properties.

A helpful and intuitive tool for visualizing and understanding complex numbers is the *complex plane*, which is a 2D representation of the space of all possible complex numbers. The complex plane is similar to a standard graph or coordinate system but splits the real and imaginary numbers into two *orthogonal*[4] axes, as shown in Fig. 2. The horizontal axis is called the real axis, and for a complex number $z = x + \mathrm{i}y$, it represents the real part x. Likewise, the vertical axis is called the imaginary axis and represents the imaginary part y. Thus, a point in the complex plane can represent every complex number with a unique set of coordinates (x, y).

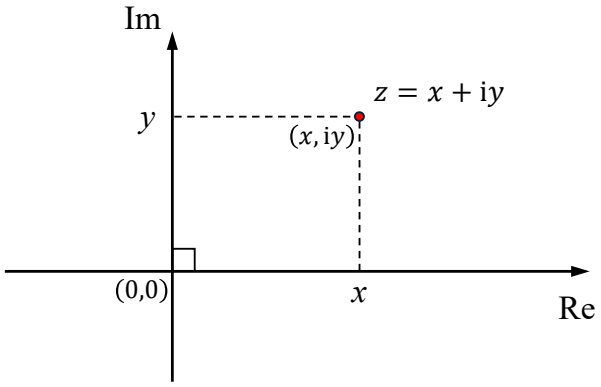


Figure 2: Visualization of a complex number $z = x + \mathrm{i}y$ in the complex plane.

With complex numbers comes a rich extension of algebra and, therefore, mathematical capabilities. An important concept associated with complex numbers is the *complex conjugate*. This plays a role in simplifying the division of complex numbers and computing the magnitude of a complex number. The complex conjugate operation has widespread applications in physics and engineering, including signal processing, control theory, quantum mechanics, optics, and electrical engineering. In alternating current (AC) circuits, for example, impedance can be represented as a complex number. The complex conjugate can determine power flow and voltage drop across different circuit elements in this context. In addition, the concept of *time reversal* may be represented by taking the complex conjugate of a complex signal.[5]

The complex conjugate of a complex number z, denoted as z^*, is defined as

$$z^* = (x + \mathrm{i}y)^* = x - \mathrm{i}y \tag{2}$$

and may thus be viewed as the mirror image of Fig. 2 about the real axis. Several important properties are associated with the algebra of complex numbers and their complex conjugates. The sum and product of a complex number and its conjugate yield real numbers:

[4]The concept of orthogonality plays an important role in linear algebra and quantum mechanics, and is explored further in section 5.1

[5]In signal processing, particularly with systems described by linear time-invariant (LTI) theory, the time reversal of a signal $x(t)$ is represented by $x(-t)$. If $x(t)$ includes complex components, its time reversal can also involve taking the complex conjugate, especially when dealing with signals represented in the *frequency* domain.

$$\begin{aligned} z + z^* &= (x + \mathrm{i}y) + (x - \mathrm{i}y) = 2x \\ zz^* &= (x + \mathrm{i}y)(x - \mathrm{i}y) = x^2 - \mathrm{i}xy + \mathrm{i}xy + y^2 = x^2 + y^2 = |z|^2 \end{aligned} \tag{3}$$

These properties can be used to divide complex numbers, as shown below. Assume we have two complex numbers $w = u + \mathrm{i}v$ and $z = x + \mathrm{i}y$. We can straightforwardly divide w and z by multiplying the numerator and denominator by z^*, resulting in a real number in the denominator since zz^* is guaranteed to be real:

$$\begin{aligned} \frac{w}{z} = \frac{w}{z} \cdot \frac{z^*}{z^*} &= \frac{(u + \mathrm{i}v)(x - \mathrm{i}y)}{(x + \mathrm{i}y)(x - \mathrm{i}y)} = \frac{ux - \mathrm{i}uy + \mathrm{i}vx + vy}{x^2 - \mathrm{i}xy + \mathrm{i}xy + y^2} \\ &= \frac{(ux + vy) + \mathrm{i}(vx - uy)}{x^2 + y^2} = \left(\frac{ux + vy}{x^2 + y^2}\right) + \mathrm{i}\left(\frac{vx - uy}{x^2 + y^2}\right) \end{aligned} \tag{4}$$

Furthermore, the complex conjugate has the distributive property such that the conjugate of a sum or product of a set of complex numbers equals the sum or product of their conjugates. That is,

$$\begin{aligned} (z_1 + z_2)^* &= z_1^* + z_2^* \\ (z_1 z_2)^* &= z_1^* z_2^* \end{aligned} \tag{5}$$

Finally, the complex conjugate is *involutive*, meaning that the conjugate of a conjugate returns the original complex number. That is, $(z^*)^* = z$.

2 Combinatorics

Learning objectives:

- Understand factorials and how permutations represent a generalization of factorials.
- Understand the difference between counting distinguishable and indistinguishable objects by respectively using the permutation and combination formulas.

Combinatorics is associated with counting and the properties of finite structures such as binomials, which are polynomials of the form $ax^m + bx^n$. It plays an important role in computing probabilities and building different configurations out of a defined set of parameters.

2.1 Factorials

The factorial of a non-negative integer n, denoted by $n!$, is the product of all consecutive positive integers (whole numbers greater than zero) less than or equal to n.[6] This corresponds to counting the number of ways to arrange a total of n *distinguishable* objects in which the order of the objects is not important. It is defined as

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1 \tag{6}$$

For example, $4! = 4 \cdot 3 \cdot 2 \cdot 1 = 24$. As will be shown later, a mathematically consistent definition for $0!$ is necessary. Since the factorial definition can be written as $n! = n \cdot (n-1)!$, we may consider the case where $n = 1$. Thus,

$$1! = 1 \cdot (1-1)! = 1 \cdot 0! = 1 \Rightarrow 0! \equiv 1 \tag{7}$$

which agrees with the logical notion that there is only a single way in which zero objects can be arranged: arrange nothing. This convention ensures that expressions and equations involving factorials are consistent

[6] The earliest known mention of factorial-like calculations can be traced to ancient Indian mathematics in the work of scholars like Pingala around 300-200 BC, where counting permutations was involved in studying poetic meters.

and meaningful even when no objects are present, a necessary feature of combinatorics. Finally, the factorial of negative integers is *undefined*, requiring that $n \in \mathbb{Z}^+$ where "$\in$" means "is an element of." [7] In the next section, we explore the concept of *permutations*, which generalizes factorials for counting arrangements of a subset of k distinguishable objects from the total set of n.

Explore More: Rising and falling factorials

2.2 Permutations (k distinguishable objects selected from a total of n distinguishable objects)

As a means to quantify the ways in which *distinguishable* objects from a set can be *uniquely* arranged or selected, factorials are a necessary ingredient in permutations and combinations. Permutations involve the arrangement of all or part of a set of objects, with the order of arrangement being important. The number of permutations $P(n,k)$ of n distinct objects taken k at a time is given by

$$P(n,k) = {}_nP_k = \frac{n!}{(n-k)!} \tag{8}$$

which corresponds to the number of unique ways k distinguishable objects can be ordered when chosen from a set of n objects. If we select all n distinguishable objects such that $k = n$, then $P(n,n) = n!/0! = n!$ and we recover the factorial definition. Since negative factorials are undefined, this requires that $k \leq n$ for $(k,n) \in \mathbb{Z}^+$.

Example 1: Suppose you have three available sensors for respectively measuring *temperature*, *pressure*, and *humidity*, and you are designing a prototype device for optimizing performance. If the device can only measure the output of two sensors, what are the unique configurations of selected sensors if the order in which the data is extracted makes a difference? What is the complete set of possible configurations?

Here, we identify $n = 3$ as the total set of sensors available and $k = 2$ as the set of selected sensors. Since the order in which the sensor readout matters, we calculate $P(n = 3, k = 2)$:

$$P(3,2) = \frac{3!}{(3-2)!} = \frac{3 \cdot 2 \cdot 1}{1} = 6 \tag{9}$$

Using the corresponding sensor labels, the unique set of possible configurations are {*temperature*, *pressure*}, {*temperature*, *humidity*}, {*pressure*, *temperature*}, {*pressure*, *humidity*}, {*humidity*, *temperature*}, and {*humidity*, *pressure*}.

2.3 Combinations (k indistinguishable objects selected from a total of n indistinguishable objects)

Combinations involve selecting items from a group without regard to the order of the items. The number of combinations $C(n,k)$ of n objects taken k at a time, stated as "*n-choose-k*," is given by the binomial coefficient,

$$C(n,k) = {}_nC_k = \binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{10}$$

We can use this to determine how many unique groupings of k objects can be formed from a larger set of n objects, in which the selected k objects are *indistinguishable* and therefore their ordering *does not matter*. Permutations and combinations can, therefore, be related by

$$P(n,k) = k!C(n,k) \tag{11}$$

Since the type of counting associated with permutations involves *distinguishable* arrangements of a set of chosen objects, imposing order introduces additional information when arranging objects, thereby increasing

[7] Daniel Bernoulli developed the complex Gamma function $\Gamma(z)$ in 1729, which extends the domain of the factorial function to include non-integer and negative values; however, negative *integers* remain excluded from the extended domain of the factorial function.

the number of countable possibilities. Thus, $P(n,k) \geq C(n,k)$, and $P(n,k) = C(n,k)$ only when $k = 0$ or $k = 1$.

Example 2: Suppose you need to analyze the pattern distribution in an 8-bit binary signal. Specifically, you want to determine how many different signals can be formed if you have exactly three bits set to '1' and the remaining five bits set to '0'. Calculate the number of different 8-bit binary signals that can be formed with exactly three bits set to '1'.

This problem requires determining the number of combinations of 8 bits taken 3 at a time where order does not matter since any '1' bit is indistinguishable from any other '1' bit. Since we are choosing 3 bits to be '1' out of 8 possible positions, we straightforwardly use the combination formula, where $n = 8$ and $k = 3$:

$$C(n=8,k=3) = \binom{8}{3} = \frac{8!}{3!(8-3)!} = \frac{8 \cdot 7 \cdot 6 \cdot \cancel{5!}}{(3 \cdot 2 \cdot 1) \cdot \cancel{5!}} = 56 \tag{12}$$

Thus, there are 56 different 8-bit binary signals that can be formed with exactly three bits set to '1'. This means that if you are dealing with a system that signals special modes or errors using three specific 'on' bits in an 8-bit code, there are 56 unique ways those error or mode signals can be configured. This information could help design error-checking algorithms and coding schemes or even help understand potential configurations in networked devices where limited signal variations are used for efficiency.

Explore More: The binomial theorem

Homework Problems: Combinatorics

3 Real functions

Learning objectives:

- Understand the definition of a function, including different types.

3.1 Definitions

Functions are mathematical expressions that describe the relationship between a set of inputs and outputs. In engineering, they are, for example, used to model physical phenomena, signal processing algorithms, and control systems. Functions can be linear or nonlinear, as well as discrete or continuous. Linear functions have the form

$$f(x) = a_0 + a_1 x \tag{13}$$

where the constant coefficients a_0 and a_1 respectively represent the y-intercept and slope. Nonlinear functions, such as the Nth-order polynomial

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_N x^N \tag{14}$$

generally exhibits more complex properties such as local extrema and complex roots. The broad utility of polynomial functions is ubiquitous in electrical engineering. Their applications include filter design, control systems, analog circuit design, signal processing, antenna design, power electronics, motor control, digital circuit timing, transmission lines, and optimization. Common examples of *non-polynomial* functions include exponential, logarithmic, trigonometric, and power functions. More details on these functions, including their general form and common applications, may be found in the supplementary information section below.

Explore More: Common functions with applications

In general, a function may be viewed as a mapping between a domain of numbers X corresponding to a set of independent variables $\{x_i\}$ and a codomain Y corresponding to the dependent variable $y(\{x_i\})$, as

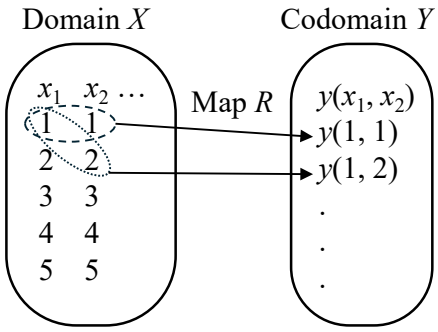


Figure 3: Diagram of a function $y(x_i)$ with domain X, codomain Y, and graph relation (mapping) R. Generally speaking, functions can represent one-to-one or many-to-one mappings.

shown schematically in Fig 3. This mapping R is a relation between X and Y that satisfies two conditions, *by definition*:

(i) for every independent variable x_i in X there exists a single value y in Y such that $(x_i, y) \in R$,

(ii) if $(x_i, y) \in R$ and $(x_i, z) \in R$ then this directly implies $y = z$.

Therefore, *one-to-one* and *many-to-one* mappings may be described by functions, however a *one-to-many* function, strictly speaking, does not exist within the framework of standard mathematics as it would violate the second condition. There are circumstances in both mathematics and physics, however, where concepts resembling one-to-many relationships may be useful.

Explore More: Examples of *one-to-many* functions

Explore More: Coordinate systems

4 Calculus

4.1 Differentiation

Differentiation in calculus involves computing the derivative of a function. The derivative measures how a function changes as its input changes and thus represents the rate of change or the tangential slope of the function at any given point.

4.1.1 Differentiation of continuous functions

The derivative of a continuous function f at x is the slope of the line tangent to the curve of $f(x)$. It is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}x} \equiv f'(x) = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{15}$$

where the limit $\Delta x \to 0$ is taken.

Explore More: Limits and asymptotic behavior

More generally, the derivative of a function $f(x)$ with respect to variable x is another function $f'(x)$ that provides the slope (degree of *change*) of f at every point x. Crucially, in order for $f(x)$ to be differentiable at a point $x = a$, $f'(x)$ must be well-defined in the neighborhood of $x = a$. That is to say, f must be *continuous* and *smooth*. Continuity requires that a small variation of the function argument x must induce a small variation of the function value f without any abrupt changes, known as *discontinuities*. Smoothness

requires that the derivative of a function $f(x)$ must be continuous in the neighborhood of $x = a$, such that for variation δ,

$$\lim_{\delta \to 0} f'(a + \delta) - f'(a - \delta) = 0 \tag{16}$$

As some examples of differentiation, we can consider linear, polynomial, and exponential functions. For a linear function $f(x) = a_0 + a_1 x$, the definition of the derivative can be applied straightforwardly:

$$f'(x) = \lim_{\Delta x \to 0} \frac{(a_0 + a_1(x + \Delta x)) - (a_0 + a_1 x)}{\Delta x} = \lim_{\Delta x \to 0} \frac{a_1 \Delta x}{\Delta x} = a_1 \tag{17}$$

Thus, the constant a_1 represents the slope, or rate of change, of the linear function $f(x)$. For a higher-order polynomial such as the quadratic function $f(x) = a_0 + a_1 x + a_2 x^2$, the derivative is $f'(x) = a_1 + 2a_2 x$ which represents the function of the line tangent to the quadratic curve at any point x.

In describing certain systems, some models might involve functions composed of two or more other functions. Such function composition may involve multiplication, division, or nesting of two functions. For each scenario, there exist differentiation rules:

- **The Product Rule**
  If a function $f = f(x)$ is formed from the multiplication of two differentiable functions $g = g(x)$ and $h = h(x)$, the derivative of $f = g \cdot h$ may be taken by using the *product rule*:

  $$f' = (g \cdot h)' = g' \cdot h + g \cdot h' \tag{18}$$

  Example: Given $f(x) = x^2 \sin(x)$, find $f'(x)$. We identify the first function as $g(x) = x^2$ and the second function as $h(x) = \sin(x)$. Thus, $g'(x) = 2x$ and $h'(x) = \cos(x)$, resulting in $f'(x) = 2x \sin(x) + x^2 \cos(x)$.

- **The Quotient Rule**
  In the case of rational functions where a function $f = f(x)$ is written as a function $g = g(x)$ divided by another function $h = h(x)$, the derivative of $f = g/h$ may be taken using the *quotient rule*:

  $$f' = \left(\frac{g}{h}\right)' = \frac{g' \cdot h - g \cdot h'}{h^2} \tag{19}$$

  Example: Given $f(x) = x^3 / \cos(x)$, find $f'(x)$. We identify $g(x) = x^3$ and $h(x) = \cos(x)$. Therefore, $g'(x) = 3x^2$ and $h'(x) = -\sin(x)$ so that

  $$f'(x) = \frac{((3x^2)(\cos(x)) - (x^3)(-\sin(x)))}{(\cos(x))^2} = \frac{3x^2 \cos(x) + x^3 \sin(x)}{\cos^2(x)} \tag{20}$$

- **The Chain Rule**
  In describing certain systems, some models might involve functions *of* functions. Consider the case in which the argument, or input, g of the function f is itself a function of a variable x so that $g = g(x)$. The derivative of the *composite* function $f(g(x))$, *alternatively* denoted $(f \circ g)(x)$, with respect to x is performed using the *chain rule*:

  $$f' = (f(g(x)))' = (f \circ g)'(x) = f'(g(x)) \cdot g'(x) \tag{21}$$

  This rule can be repeatedly used for multiple function nestings and in conjunction with the product rule, it can also be used to derive the quotient rule, an exercise left to the reader.

  Example: Given $f(x) = A\mathrm{e}^{-x^2}$, find $f'(x)$. We begin by identifying $h(x) = A\mathrm{e}^{x}$ as the outermost function and $g(x) = -x^2$ as the nested, or inner, function. Thus, we can compute the derivative as

  $$f'(x) = h'(g(x)) \cdot g'(x) = A\mathrm{e}^{-x^2} \cdot (-2x) = -2Ax\mathrm{e}^{-x^2} \tag{22}$$

  This illustrates that the rate of change of the exponential function is proportional to its current value at any point x.

4.1.2 Partial vs. total differentiation

In many cases, functions can have multiple variables, requiring a distinction between *total* and *partial* derivatives. A total derivative of a function $f(x)$, denoted as $\mathrm{d}f(x)/\mathrm{d}x$, refers to the derivative of a function with respect to one variable (x, in this case) when the function depends on that variable *either* directly or indirectly through other variables. Total derivatives are used when dealing with functions where variables are interdependent or when the function is implicitly defined. It provides a measure of how a function changes as all variables change, considering their interdependence.

A *partial* derivative refers to the derivative of a function with respect to one variable while holding all other variables constant. Partial derivatives are used in the context of multivariable functions to examine the rate of change in one direction (with respect to one variable) while ignoring the others. It is denoted using the partial derivative symbol ∂, such as $\partial f/\partial x$ which represents the partial derivative of function f with respect to variable x while keeping any and all other variables constant.[8] The partial derivative operator can be efficiently expressed in terms of a variable by using the variable as a subscript, and the partial derivative of a function can also be efficiently written by using the explicit independent variable as a subscript. These conventions can be equivalently used to denote the partial derivative of a function $f(x)$ with respect to the variable x:

$$\frac{\partial f}{\partial x} \equiv \partial_x f \equiv f_x \tag{23}$$

Example 1: Consider the function $f(x, y) = x^2 + 3xy - y^2$, where x and y are independent variables. To find the partial derivative of f with respect to x and thus show how f changes as x varies for a fixed value of y, we treat y as a constant:

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x}\left(x^2 + 3xy - y^2\right) = 2x + 3y \tag{24}$$

Example 2: We now consider the case where y is not independent of x, but is itself a function $y(x) = 4x$. To compute the total derivative of $f(x, y) = x^2 + 3xy - y^2$, we can substitute $y = 4x$ into f, resulting in $f(x) = x^2 + 3x(4x) - (4x)^2 = x^2 + 12x^2 - 16x^2 = -3x^2$. Thus, the total derivative of f with respect to x is now simply the derivative of $-3x^2$ with respect to x:

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\mathrm{d}}{\mathrm{d}x}\left(-3x^2\right) = -6x \tag{25}$$

Thus, the total derivative reflects the cumulative effect on f as x changes, which includes changes induced in y. By rewriting the function purely in terms of x, this is also an example of *dimensionality reduction*.

Explore More: The gradient

Explore More: Taylor series expansion

Explore More: First-order differentiation of discrete functions

Explore More: Higher-order differentiation of discrete functions

Homework Problems: Differentiation

[8] For this reason, the momentum operator in quantum theory is formally defined in terms of a partial derivative for the sake of consistency, even in one dimension ($\hat{p} = -\mathrm{i}\hbar\,\partial/\partial x$), since the wavefunction can generally depend on time t. Generalizing to three dimensions results in defining the momentum operator in terms of the *gradient* ($\hat{\mathbf{p}} = -\mathrm{i}\hbar\nabla$). More information on the gradient may be found in the corresponding *Explore More* appendix.

4.2 Integration

Integration is the sum of infinitesimal elements to calculate areas, volumes, and other quantities. Essentially, it may be viewed as the reverse process of differentiation, and in 1D, the integral of a function may be referred to as an *anti-derivative*. Calculating the integral of a function $f(x)$ within an interval of x corresponds to determining the total value of the function over this interval. Such a function may either be continuous or discrete with respect to the integration variable x. Overviews of each are provided in the following *Explore More* appendices along with examples and exercises left for the reader.

Explore More: Integration of continuous functions

Explore More: Integration of discrete functions

4.3 Complex functions

Complex functions are mathematical expressions where the independent and dependent variables are generally both complex numbers. These functions contribute to various fields of science and engineering, where they are used to describe phenomena such as electromagnetic waves and alternating current (AC) in circuits.

Both the input (domain) and output (codomain) of a complex function are complex numbers, typically expressed as $f : \mathbb{C} \to \mathbb{C}$, where $f(z) = u(x, y) + \mathrm{i}v(x, y)$ for a complex variable $z = x + \mathrm{i}y$. As with purely real functions, a complex function is considered analytic at a point if it is differentiable at every point in some neighborhood of that point. Analytic functions have derivatives everywhere in their domain, allowing them to be expanded using the Taylor series.

By plotting complex numbers in the complex plane, it is possible to visualize how the real and imaginary components of a complex function are related through Euler's formula[9], and how they can be represented by respectively projecting onto the real and imaginary axes. Euler's formula states that for any real number θ, the complex exponential function $z(\theta) = x(\theta) + \mathrm{i}y(\theta)$ can be generally expressed as

$$z(\theta) = A\mathrm{e}^{\mathrm{i}\theta} = A\left(\cos(\theta) + \mathrm{i}\sin(\theta)\right) = A\cos(\theta) + \mathrm{i}A\sin(\theta) \tag{26}$$

where i is the imaginary unit and A is the amplitude (a real positive number), or magnitude, of the complex number $z(\theta)$. This formula shows that the exponential function with an imaginary exponent results in a complex number whose real and imaginary parts are respectively proportional to the cosine and sine components of the original exponent, scaled by the amplitude A. Thus, for $z = x + \mathrm{i}y = A\mathrm{e}^{\mathrm{i}\theta}$, $x = x(\theta) = A\cos(\theta)$ and $y = y(\theta) = A\sin(\theta)$. This is visually depicted in Fig. 4.

Explore More: Introduction to plane waves

Explore More: Introduction to the Fourier transform

Explore More: The Fast Fourier Transform (FFT)

Explore More: Complex differentiation

Homework Problems: Complex differentiation

[9]Euler's formula, established by the Swiss mathematician Leonhard Euler in 1748, is an equation in complex analysis that establishes a relationship between trigonometric functions and exponential functions. It finds use in fields that involve complex numbers, such as electrical engineering, physics, and applied mathematics.

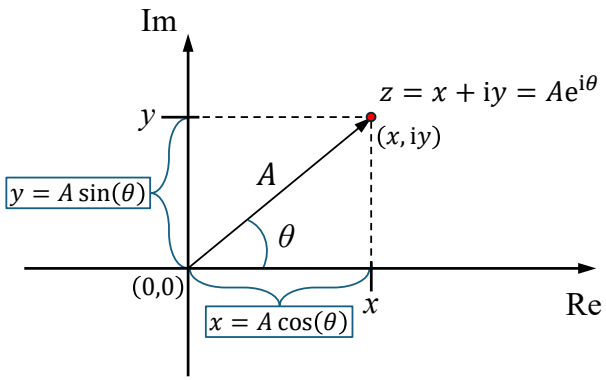


Figure 4: Visualization of a complex number $z = x + \mathrm{i}y$ in terms of the complex phase angle θ. The amplitude, or magnitude, of the complex number is given by A. This magnitude is the radius of the circle traced out by varying $0 \leq \theta \leq 2\pi$.

5 Linear algebra

Linear algebra is a field of mathematics that focuses on vector spaces and linear *mappings* between them. It plays a central role in both pure and applied mathematics, representing linear systems through the use of matrices, efficient methods for solving problems in such systems, and the ability to quantify the behavior and stability of systems. Linear algebra has applications in many fields including circuit analysis, signal processing, control systems and control theory, communication, and power systems. Generally speaking, the utility of linear algebra in physics and electrical engineering cannot be overstated, as it enables the design, analysis, and optimization of a variety of complex systems. In this section, we will explore several elementary mathematical concepts and operations necessary to work with and understand linear systems.

Familiarization with software designed for numerical linear algebra and fast, numerically robust, matrix calculations will be indispensable moving forward. In particular, downloading and installing MATLAB is recommended, as well as completing the onramp tutorial course. GNU Octave is a compatible open-source alternative that can execute any MATLAB script which uses base MATLAB function libraries.

As an optional alternative to MATLAB or GNU Octave, standardized software libraries for numerical linear algebra can be used for popular high-level scientific programming languages such as Python or Julia. In the case of Python, the NumPy, SciPy, and Matplotlib libraries are very helpful for scientific programming and data visualization, for which installation and syntax documentation is provided through their respective hyperlinks. Through the linear algebra library package LAPACK, Julia provides native implementations of basic and common linear algebra operations with high-level syntax.

5.1 Vectors

Learning objectives:

- Add, subtract, scale, and translate vectors.
- Calculate the inner product of two vectors and understand the concept of projection.
- Relate the concept of vectors and complex numbers to the complex plane, including projecting out real and imaginary components.
- Understand and apply the concept of unit vectors.
- Understand linear superposition, including several examples and applications in signal processing.

In an N-dimensional Euclidean $\mathbb{R}^N$ space, vectors consist of a magnitude and direction in which each dimension is orthogonal to all other dimensions. As such, they are useful for representing a variety of physical quantities such as electrical fields, currents, or forces.

5.1.1 Vector notation and algebra in $\mathbb{R}^N$ space

Using Cartesian coordinates in the $\mathbb{R}^N$ space, a vector in a $N = 2$ dimensional space can be represented as $\mathbf{v} = [x\ y]$, and in a $N = 3$ dimensional space as $\mathbf{v} = [x\ y\ z]$, where x, y, and z are the components of the vector in each dimension. As with numbers, the concept of operations carries over to vectors though with some modifications. Key operations include but are not limited to transposition, addition, subtraction, scalar multiplication, and the inner product.

Vectors can either be an $1 \times N$ row vector in which there is a single row of N elements (N columns),

$$\mathbf{v} = [v_1 \quad v_2 \quad \cdots \quad v_N] \tag{27}$$

or an $N \times 1$ column vector in which there are a single column of N elements (N rows),

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \tag{28}$$

By convention, we will default to describing vectors as column vectors.[10]

Transposition in $\mathbb{R}^N$ space

The transpose operation on vectors involves converting a row vector into a column vector, or vice versa, denoted by the superscript $^\intercal$ on the vector symbol. If $\mathbf{v}$ is an $N \times 1$ column vector, its transpose $\mathbf{v}^\intercal$ is then a $1 \times N$ row vector. Conversely, if $\mathbf{v}$ is a $1 \times N$ row vector, its transpose $\mathbf{v}^\intercal$ is an $N \times 1$ column vector.[11] This is visually depicted in Fig. 5. Sequentially applying the transpose twice gives the original vector, $(\mathbf{v}^\intercal)^\intercal = \mathbf{v}$.

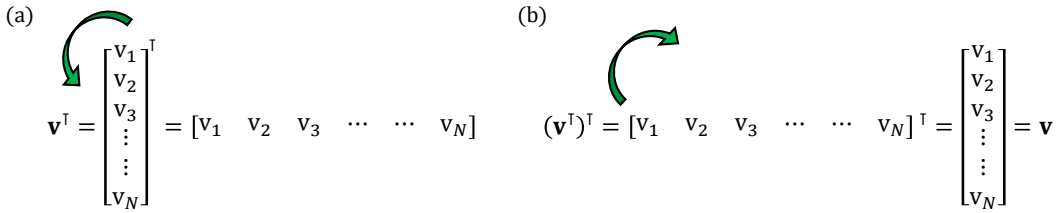


Figure 5: (a) The transpose of an $N \times 1$ column vector may be viewed as rotating the array of numbers anticlockwise by 90° such that it becomes an $1 \times N$ row vector. (b) The transpose of a $1 \times N$ row vector may be viewed as rotating the array of numbers clockwise by 90° such that it becomes an $N \times 1$ column vector. If the transpose operation is performed twice as done in this example, the original vector is recovered.

Addition and subtraction in $\mathbb{R}^N$ space

The addition or subtraction of vectors requires that the dimensionality of all vectors being summed is equal. For example, adding or subtracting vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ require they all either be row or column vectors with the same number of N elements:

$$\mathbf{a} \pm \mathbf{b} \pm \mathbf{c} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \pm \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \pm \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} a_1 \pm b_1 \pm c_1 \\ a_2 \pm b_2 \pm c_2 \\ \vdots \\ a_N \pm b_N \pm c_N \end{bmatrix} \tag{29}$$

Multiplication in $\mathbb{R}^N$ space

[10] By default, vectors are represented as $1 \times N$ row arrays in MATLAB.

[11] To convert a row array into a column array in MATLAB, the function transpose() is used. Thus, $\mathbf{v}^\intercal$ = transpose($\mathbf{v}$), where $\mathbf{v}$ is by default an $1 \times N$ row array and therefore $\mathbf{v}^\intercal$ is an $N \times 1$ column array.

Vectors may also be multiplied by numbers, an operation known as *scalar* multiplication since the number *scales* the vector by stretching or shrinking its magnitude without changing its orientation, or direction, in the N-dimensional space. Thus, multiplying a scalar a with a vector $\mathbf{v}$ results in each of the vector elements scaled by a:

$$a\mathbf{v} = a\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} = \begin{bmatrix} av_1 \\ av_2 \\ \vdots \\ av_N \end{bmatrix} \tag{30}$$

Finally, element-wise or *Hadamard* multiplication of two vectors results in a vector in which each element is a scalar product of the respective elements of the original two vectors:

$$\mathbf{u} \odot \mathbf{v} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \odot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} = \begin{bmatrix} u_1 v_1 \\ u_2 v_2 \\ \vdots \\ u_N v_N \end{bmatrix} \tag{31}$$

In MATLAB, this is expressed as `u.*v` in which `.*` takes the role of the symbol $\odot$ in Eqn. (31).

Magnitude in $\mathbb{R}^N$ space

The magnitude, or length, of an N-dimensional vector $\mathbf{v}$ is described by its *Euclidean* norm $\|\mathbf{v}\|_2$, also referred to as its 2-norm. In $\mathbb{R}^N$ space, it is calculated as

$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + v_3^2 + \cdots + v_N^2} \tag{32}$$

The formula may essentially be viewed as the Pythagorean theorem extended to N dimensions. The 2-norm gives a direct measure of the distance of the vector from the origin of the coordinate system, and has the properties of being non-negative as well as satisfying the *triangle inequality*, $\|\mathbf{u} + \mathbf{v}\|_2 \leq \|\mathbf{u}\|_2 + \|\mathbf{v}\|_2$, which asserts that the magnitude of the sum of two vectors is less than or equal to the sum of their individual magnitudes.

Unit vectors in $\mathbb{R}^N$ space

A unit vector is a vector that has a magnitude of one. This makes unit vectors very useful for specifying space directions and normalizing vectors. As such, unit vectors are often used in linear algebra to define the axes of coordinate systems and can, therefore, be used as a geometrical basis. Each component is divided by the vector magnitude to calculate the unit vector of any given vector. The unit (magnitude) vector $\hat{\mathbf{v}}$ in the direction of $\mathbf{v}$ is then given by

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{33}$$

resulting in $\|\hat{\mathbf{v}}\| = 1$, in which we suspend use of the 2-norm subscript for notational convenience.

Unit vectors are widely used in vector calculations, including vector projections, cross and dot product operations, and when defining coordinate systems in both two and three dimensions.

Inner product in $\mathbb{R}^N$ space

If θ is the angle between two vectors, then the inner product may be defined as

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos(\theta) = ab\cos(\theta) \tag{34}$$

where $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ are the respective magnitudes of vectors $\mathbf{a}$ and $\mathbf{b}$. This procedure is visualized in Fig. 6, where the inner product between vectors $\mathbf{a}$ and $\mathbf{b}$ may be viewed as a projection of one vector onto the other. The inner product also satisfies the *Cauchy-Schwarz inequality*[12],

[12]The Cauchy-Schwarz inequality, also known as the Cauchy-Bunyakovsky-Schwarz inequality, is named after Augustin-Louis Cauchy, Viktor Bunyakovsky, and Hermann Schwarz. Augustin-Louis Cauchy first published the

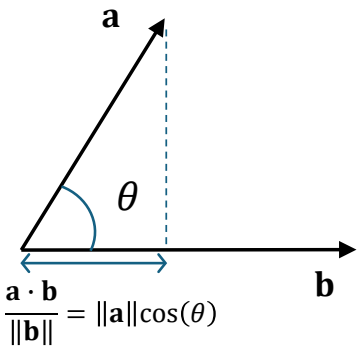


Figure 6: The inner product between vectors **a** and **b** involves the concept of projecting one vector onto another.

In terms of the set of (purely-real) vector elements a_i and b_i with $i \in \{1, 2, \ldots, N\}$ for real column vectors **a** and **b** respective, the inner product can be computed by multiplying the transpose of **a** with **b** and is given as

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^\intercal \mathbf{b} = \begin{bmatrix} a_1 & a_2 & \cdots & a_N \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} = \sum_{i=1}^{N} a_i b_i \tag{35}$$

Thus, we see that the inner product may be regarded as a *multiply-accumulate*, or MAC, operation. In addition, the inner product can be used to determine whether two vectors are orthogonal.

Orthogonality and completeness in $\mathbb{R}^N$ space

Orthogonality refers to the idea that two vectors are perpendicular to each other in a vector *space*. By definition, two vectors **a** and **b** are orthogonal if their inner product is exactly zero so that $\mathbf{a}^\intercal \mathbf{b} = \mathbf{b}^\intercal \mathbf{a} = 0$.

Example 1: Consider two vectors in $\mathbb{R}^2$:

$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

The inner product of **u** and **v** is

$$\mathbf{u} \cdot \mathbf{v} = 1 \cdot 2 + 2 \cdot (-1) = 2 - 2 = 0$$

proving **u** and **v** are orthogonal in $\mathbb{R}^2$ space. Geometrically, this means there is an angle of exactly 90°, or $\pi/2$ radians, between them. The concept of orthogonality generalizes the notion of perpendicularity from 2D or 3D Euclidean space to N-dimensional spaces.

An orthogonal *basis* for a vector space is a set of vectors that are mutually orthogonal and *span* the vector space. If these vectors are also unit vectors (i.e., have a norm of 1), the basis is called an *orthonormal* basis.

Example 2: Consider the standard basis vectors in $\mathbb{R}^3$ space:

$$\hat{\mathbf{i}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \hat{\mathbf{j}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \hat{\mathbf{k}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{36}$$

These vectors can easily be seen to be mutually orthogonal since $\hat{\mathbf{i}} \cdot \hat{\mathbf{j}} = \hat{\mathbf{i}} \cdot \hat{\mathbf{k}} = \hat{\mathbf{j}} \cdot \hat{\mathbf{k}} = 0$, and furthermore they each have norm $\left\|\hat{\mathbf{i}}\right\| = \left\|\hat{\mathbf{j}}\right\| = \left\|\hat{\mathbf{k}}\right\| = 1$. Thus, the set $\{\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}\}$ forms an orthonormal basis in $\mathbb{R}^3$.

inequality for sums in 1821, while the corresponding inequality for integrals was published by Viktor Bunyakovsky in 1859 and Hermann Schwarz in 1888.

Completeness refers to a set of vectors being able to represent any vector in the space through a linear combination. In other words, a set of vectors is *complete* if it spans the entire vector space. If a set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$ spans the vector space V, then any vector $\mathbf{w} \in V$ can be expressed as a linear superposition of this basis:

$$\mathbf{w} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_N\mathbf{v}_N$$

where $c_1, c_2, \ldots, c_N$ are real scalar numbers.

Example 3: Consider the orthonormal basis vectors $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ in $\mathbb{R}^2$ space. Since any vector $\mathbf{w}$ in this space can be written as a linear combination $\mathbf{w} = a\hat{\mathbf{i}} + b\hat{\mathbf{j}}$ where $a, b \in \mathbb{R}$, the set $\{\hat{\mathbf{i}}, \hat{\mathbf{j}}\}$ is complete in $\mathbb{R}^2$ by definition.

When a set of vectors is both orthonormal and complete, it is considered an orthonormal basis. This provides a convenient way to express *any* vector in the space with straightforward computation of coefficients using inner products.

Example 4: Given the orthnormal basis $\{\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}\}$ in $\mathbb{R}^3$ space from Example 2, any vector $\mathbf{w} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ in $\mathbb{R}^3$ can be written as

$$\mathbf{w} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix} = a\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + b\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + c\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = a\hat{\mathbf{i}} + b\hat{\mathbf{j}} + c\hat{\mathbf{k}}.$$

The coefficients a, b, and c can then simply be extracted using the inner product:

$$a = \mathbf{w} \cdot \hat{\mathbf{i}} = [a \quad b \quad c]\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = a$$

$$b = \mathbf{w} \cdot \hat{\mathbf{j}} = [a \quad b \quad c]\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = b$$

$$c = \mathbf{w} \cdot \hat{\mathbf{k}} = [a \quad b \quad c]\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = c$$

These properties carry over to complex vector spaces and are useful in quantum mechanics, where states are often expressed in terms of orthonormal basis functions.

5.1.2 Vector notation and algebra in $\mathbb{C}^N$ space

Additional algebra must be introduced if the vector elements are complex. By convention, a complex vector $\mathbf{u}$ is written as a column vector of complex values $u_i \in \mathbb{C}$ (in the $\mathbb{C}^N$ space):

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \tag{37}$$

Hermitian adjoint in $\mathbb{C}^N$ space

We define the *Hermitian adjoint* of the complex vector $\mathbf{u}$ by taking the complex conjugate *and* transpose of this vector:

$$\mathbf{u}^\dagger = (\mathbf{u}^*)^\intercal = (\mathbf{u}^\intercal)^* = [u_1^* \quad u_2^* \quad \ldots \quad u_N^*] \tag{38}$$

Note that for a purely-real vector $\mathbf{v} \in \mathbb{R}^N$, the Hermitian adjoint is equivalent to just taking the transpose since $\mathbf{v}^* = \mathbf{v}$ and therefore $\mathbf{v}^\dagger = \mathbf{v}^\intercal$.

Binary vector operations in $\mathbb{C}^N$ space

The same $\mathbb{R}^N$ space algebra of Eqns. (29)-(31) carries over to the $\mathbb{C}^N$ space, in which the binary operations of addition, subtraction, and multiplication are performed on the respective real and imaginary scalar components of each vector element. Corresponding examples are provided below in Eqns. (39)-(41), for (generally complex) scalar α and complex vectors $\mathbf{u} = \mathbf{a} + \mathrm{i}\mathbf{b}$ and $\mathbf{v} = \mathbf{c} + \mathrm{i}\mathbf{d}$ where vectors $\mathbf{a}$ and $\mathbf{c}$ are the real parts of $\mathbf{u}$ and $\mathbf{v}$ respectively, and vectors $\mathbf{b}$ and $\mathbf{d}$ are the imaginary parts of $\mathbf{u}$ and $\mathbf{v}$ respectively.

Addition and subtraction may be performed by separately adding the real and imaginary components of each complex vector:

$$\begin{aligned}\mathbf{u} \pm \mathbf{v} &= (\mathbf{a} + \mathrm{i}\mathbf{b}) \pm (\mathbf{c} + \mathrm{i}\mathbf{d}) = (\mathbf{a} \pm \mathbf{c}) + \mathrm{i}\,(\mathbf{b} \pm \mathbf{d}) \\ &= \begin{bmatrix} a_1 \pm c_1 \\ a_2 \pm c_2 \\ \vdots \\ a_N \pm c_N \end{bmatrix} + \mathrm{i} \begin{bmatrix} b_1 \pm d_1 \\ b_2 \pm d_2 \\ \vdots \\ b_N \pm d_N \end{bmatrix} = \begin{bmatrix} (a_1 \pm c_1) + \mathrm{i}(b_1 \pm d_1) \\ (a_2 \pm c_2) + \mathrm{i}(b_2 \pm d_2) \\ \vdots \\ (a_N \pm c_N) + \mathrm{i}(b_N \pm d_N) \end{bmatrix}\end{aligned} \tag{39}$$

Scalar multiplication:

$$\alpha\mathbf{u} = \alpha \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha(a_1 + \mathrm{i}b_1) \\ \alpha(a_2 + \mathrm{i}b_2) \\ \vdots \\ \alpha(a_N + \mathrm{i}b_N) \end{bmatrix} = \begin{bmatrix} \alpha a_1 + \mathrm{i}\alpha b_1 \\ \alpha a_2 + \mathrm{i}\alpha b_2 \\ \vdots \\ \alpha a_N + \mathrm{i}\alpha b_N \end{bmatrix} \tag{40}$$

Hadamard multiplication:

$$\mathbf{u} \odot \mathbf{v} = \begin{bmatrix} u_1 v_1 \\ u_2 v_2 \\ \vdots \\ u_N v_N \end{bmatrix} = \begin{bmatrix} (a_1 + \mathrm{i}b_1)(c_1 + \mathrm{i}d_1) \\ (a_2 + \mathrm{i}b_2)(c_2 + \mathrm{i}d_2) \\ \vdots \\ (a_N + \mathrm{i}b_N)(c_N + \mathrm{i}d_N) \end{bmatrix} = \begin{bmatrix} (a_1 c_1 - b_1 d_1) + \mathrm{i}(a_1 d_1 + b_1 d_1) \\ (a_1 c_1 - b_1 d_1) + \mathrm{i}(a_2 d_2 + b_2 d_2) \\ \vdots \\ (a_N c_N - b_N d_N) + \mathrm{i}(a_N d_N + b_N d_N) \end{bmatrix} \tag{41}$$

Inner product in $\mathbb{C}^N$ space

Using the concept of Hermitian adjoint, the *complex* inner product of two complex vectors $\mathbf{u}$ and $\mathbf{v}$ is computed as

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^\dagger \mathbf{v} = \begin{bmatrix} u_1^* & u_2^* & \cdots & u_N^* \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} = \sum_{i=1}^{N} u_i^* v_i \tag{42}$$

Magnitude and unit vectors in $\mathbb{C}^N$ space

Whether a vector is purely real or complex, its magnitude is strictly a real number. Similarly to the case of $\mathbb{R}^N$ space, we can compute the 2-norm for a complex vector by first taking the complex inner product of the vector with itself. This is done by taking Eqn. (42) and setting $\mathbf{v} = \mathbf{u}$:

$$\mathbf{u} \cdot \mathbf{u} = \mathbf{u}^\dagger \mathbf{u} = \begin{bmatrix} u_1^* & u_2^* & \cdots & u_N^* \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \sum_{i=1}^{N} u_i^* u_i = \sum_{i=1}^{N} |u_i|^2 \tag{43}$$

where the definition of complex number (scalar) magnitude in Eqn. (3) is used. Thus, the 2-norm formula given in Eqn. (32) is modified accordingly for complex vectors:

$$\|\mathbf{u}\| = \sqrt{\mathbf{u}^\dagger \mathbf{u}} = \sqrt{u_1^* u_1 + u_2^* u_2 + \cdots + u_N^* u_N} = \sqrt{|u_1|^2 + |u_2|^2 + \cdots + |u_N|^2} \tag{44}$$

Finally, normalization of a complex vector $\mathbf{u}$ requires that $\|\mathbf{u}\| = 1$. Normalization is performed in the same manner as Eqn. (33), which involves dividing the complex vector $\mathbf{u}$ by its magnitude,

$$\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|} \tag{45}$$

resulting in a unit vector $\hat{\mathbf{u}}$ in the direction of $\mathbf{u}$ within the $\mathbb{C}^N$ space.

Orthogonality and completeness in $\mathbb{C}^N$ space

As with $\mathbb{R}^N$ space, orthogonality and completeness are incredibly useful properties when working with complex vector spaces. Just like purely-real vectors, two complex vectors are considered orthogonal if their (complex) inner product is zero so that $\mathbf{u}^\dagger \mathbf{v} = \mathbf{v}^\dagger \mathbf{u} = 0$. A set of complex vectors is considered complete if any vector in the space can be expressed as a linear combination of that set. This can manifest, for instance, as a set of purely real vectors simply multiplied by complex coefficients. The orthonormal (real) vectors $\{\hat{\mathbf{i}}, \hat{\mathbf{j}}\}$ form a basis for $\mathbb{C}^2$, for example, since any complex vector $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$ can be expressed as

$$\mathbf{z} = z_1 \hat{\mathbf{i}} + z_2 \hat{\mathbf{j}}$$

where z_1 and z_2 are complex numbers. In general, any orthonormal basis in $\mathbb{C}^N$ consists of vectors that are orthogonal, normalized, and complete.

Example 5: Consider the complex vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ and determine whether they form an orthonormal basis:

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ \mathrm{i} \end{bmatrix}, \quad \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -\mathrm{i} \end{bmatrix}$$

We verify orthonormality by checking normality ($\mathbf{v}_1^\dagger \mathbf{v}_1$ and $\mathbf{v}_2^\dagger \mathbf{v}_2$) and orthogonality ($\mathbf{v}_1^\dagger \mathbf{v}_2$):

$$\mathbf{v}_1^\dagger \mathbf{v}_1 = \frac{1}{2} \begin{bmatrix} 1 & -\mathrm{i} \end{bmatrix} \begin{bmatrix} 1 \\ \mathrm{i} \end{bmatrix} = \frac{1}{2}(1+1) = 1 \checkmark$$

$$\mathbf{v}_2^\dagger \mathbf{v}_2 = \frac{1}{2} \begin{bmatrix} 1 & \mathrm{i} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathrm{i} \end{bmatrix} = \frac{1}{2}(1+1) = 1 \checkmark$$

$$\mathbf{v}_1^\dagger \mathbf{v}_2 = \frac{1}{2} \begin{bmatrix} 1 & -\mathrm{i} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathrm{i} \end{bmatrix} = \frac{1}{2}(1-1) = 0 \checkmark$$

Thus, complex vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ span the $\mathbb{C}^2$ space and the set $\{\mathbf{v}_1, \mathbf{v}_2\}$ forms an orthonormal basis for $\mathbb{C}^2$.

Homework Problems: Vectors

Homework Problems: The Poynting vector

5.2 Matrices

Learning objectives:

- Become familiar with matrix notation.
- Add, subtract, and multiply matrices.

- Compute the determinant of a matrix.
- Calculate the cross product of vectors.
- Take the inverse of a matrix.
- Understand the properties and algebra of linear operators, including commutativity, complex conjugate, transpose, and Hermitian adjoint.

5.2.1 Matrix notation and algebra

Matrices are rectangular arrays of numbers, symbols, or expressions arranged in rows and columns, widely used to solve systems of linear equations and to perform linear transformations. A matrix is typically denoted by a bolded capital letter and can be represented as $\mathbf{A} = [a_{ij}]$, where i and j indicate the row and column positions of an element in the matrix. Common matrix operations include addition, subtraction, multiplication, and finding the determinant and inverse in the case of square matrices. Matrix multiplication is particularly useful for performing linear transformations and solving linear systems. As with numbers and vectors, matrices may be added, subtracted, and scaled. However, as with vectors, the multiplication of matrices involves special rules. In this section, we provide a brief review of matrix arithmetic along with examples.

As in the case of vectors, we begin with the concept of transposing a matrix, which is a basic operation in linear algebra. Transposing a matrix involves rearranging the rows of a matrix into columns (or vice versa), which can be best visualized by flipping the matrix over its diagonal in the case of a square ($m = n$) matrix as depicted in Fig. 7(a) or in general, swapping the row and column indices such that each row becomes a column and vice versa as depicted in Fig. 7(b) for a non-square ($m \neq n$) matrix.

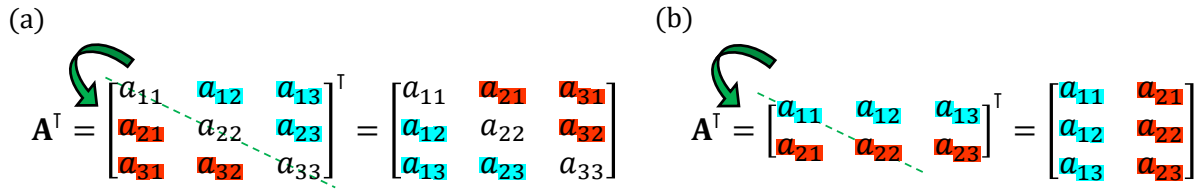


Figure 7: (a) The transpose of a square matrix (in this case, a 3×3 matrix) may be viewed as swapping the upper (highlighted in blue) and lower (highlighted in red) triangular submatrix elements by rotating the matrix about its diagonal, resulting in the swapping of off-diagonal matrix elements. (b) Example of the transpose operation on a non-square 2×3 matrix, resulting in a 3×2 matrix. The transpose operation may be performed on matrices of arbitrary dimension and essentially swaps the matrix element indices. In this example, the first-row elements (highlighted in blue) become the first column, and the second-row elements (highlighted in red) become the second-column elements.

Given a matrix $\mathbf{A}$ of size $m \times n$, where m is the number of rows and n is the number of columns, the transpose of $\mathbf{A}$, denoted as $\mathbf{A}^\intercal$, is a new matrix of size $n \times m$. In particular, the elements of $\mathbf{A}^\intercal$ are defined such that the element at row i and column j in $\mathbf{A}^\intercal$ is the element at row j and column i in the original matrix $\mathbf{A}$. Thus, for $\mathbf{A} = [a_{ij}]$, we can define the transpose as

$$\mathbf{A}^\intercal = [a_{ji}] \tag{46}$$

There are several important and useful properties of transposed matrices. If $\mathbf{A}$ is a symmetric square matrix, then $\mathbf{A}^\intercal = \mathbf{A}$. In addition, the transpose of a transposed matrix is equal to the original matrix; that is, $(\mathbf{A}^\intercal)^\intercal = \mathbf{A}$. Furthermore, the transpose operation is distributive, meaning that the transpose of the sum of two matrices is equal to the sum of their transposes so that

$$(\mathbf{A} + \mathbf{B})^\intercal = \mathbf{A}^\intercal + \mathbf{B}^\intercal \tag{47}$$

Finally, the transpose of a product of matrices is equal to the product of their transposes in reverse order; that is,

$$(\mathbf{AB})^\intercal = \mathbf{B}^\intercal \mathbf{A}^\intercal \tag{48}$$

The addition of matrices is commutative, that is, for two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$. However, whether two matrices are added or subtracted, they must have the same dimensions. That is, the number of rows m and columns n must match. Thus as an example, for a set of 2×2 matrices $\mathbf{A}$ and $\mathbf{B}$,

$$\mathbf{A} \pm \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \pm \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} \end{bmatrix} \tag{49}$$

One type of multiplication of matrices is known as the Hadamard, or element-wise, product, which is also defined for two matrices of identical dimensions. As with multiplication of numbers, the Hadamard product is commutative. Using the 2×2 matrices above, the Hadamard product would be computed as follows,

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix} \tag{50}$$

This element-wise operation can be quite useful when performing numerical operations on large arrays of numbers.

The more common type of matrix product is a multiply-accumulate (MAC) operation which is *not* commutative and requires that the number of columns of the left matrix match the number of rows of the right matrix. In general, for an $m \times n$ dimensional matrix $\mathbf{A}$ and an $n \times p$ dimensional matrix $\mathbf{B}$, the matrix $\mathbf{C}$ resulting from the product $\mathbf{C} = \mathbf{AB}$ will have dimension $m \times p$. As an example, consider a 2×2 matrix $\mathbf{A}$ and a 2×2 matrix $\mathbf{B}$. The matrix product $\mathbf{AB}$ is then calculated by taking the first row of $\mathbf{A}$ and multiplying the various column elements with the respective set of row elements of the first column of $\mathbf{B}$. These products are then added to obtain the first row, first column element value of the resulting product matrix. Thus, we have

$$\mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} (a_{11}b_{11} + a_{12}b_{21}) & (a_{11}b_{12} + a_{12}b_{22}) \\ (a_{21}b_{11} + a_{22}b_{21}) & (a_{21}b_{12} + a_{22}b_{22}) \end{bmatrix} \tag{51}$$

5.2.2 Outer product

A matrix can also be formed from two vectors by computing the *outer* product. As opposed to an inner product in which a row vector is multiplied with a column vector, the outer product is performed by multiplying a column vector with a row vector, resulting in a matrix. For an $m \times 1$ complex vector $\mathbf{u}$ and an $n \times 1$ complex vector $\mathbf{v}$ where

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \text{ and } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \tag{52}$$

the outer product is defined as:

$$\mathbf{u}\mathbf{v}^\dagger = \mathbf{u}\left(\mathbf{v}^*\right)^\intercal = \mathbf{u}\left(\mathbf{v}^\intercal\right)^* = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \begin{bmatrix} v_1^* & v_2^* & \dots & v_n^* \end{bmatrix} = \begin{bmatrix} u_1v_1^* & u_1v_2^* & \cdots & u_1v_n^* \\ u_2v_1^* & u_2v_2^* & \cdots & u_2v_n^* \\ \vdots & \vdots & \ddots & \vdots \\ u_mv_1^* & u_mv_2^* & \dots & u_mv_n^* \end{bmatrix} \tag{53}$$

Thus, we see that the resulting matrix has dimensions $m \times n$.

5.2.3 Tensor product

Tensor products, also known as Kronecker products, are mathematical operations that extend the concept of an outer product of vectors to matrices. Such an operation is widely used in the description of quantum information processing as well as classical signal processing. The tensor product combines the dimensionality of two vectors to create a higher dimensional space.

If column vector $\mathbf{a}$ has N_1 elements $(a_1, \dots, a_{N_1})$ and column vector $\mathbf{b}$ has N_2 elements $(b_1, \dots, b_{N_2})$, the *product state* $\mathbf{a} \otimes \mathbf{b}$ is *also* a column vector of length $N_1 \times N_2$,

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1\mathbf{b} \\ a_2\mathbf{b} \\ \vdots \\ a_{N_1}\mathbf{b} \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_2 b_2 \\ \vdots \\ a_1 b_{N_2} \\ a_2 b_1 \\ a_2 b_2 \\ \vdots \\ a_{N_1} b_{N_2} \end{bmatrix} \tag{54}$$

For example,

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{bmatrix}.$$

If a matrix $\mathbf{A} = [a_{ij}]$ with dimensions $N_1 \times N_1$ and matrix $\mathbf{B} = [b_{ij}]$ with dimensions $N_1 \times N_2$, then the tensor product is

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \dots & a_{1N_1}\mathbf{B} \\ \vdots & & \vdots \\ a_{N_1 1}\mathbf{B} & \dots & a_{N_1 N_1}\mathbf{B} \end{bmatrix} \tag{55}$$

For example,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \otimes \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}\mathbf{B} & a_{11}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{bmatrix}$$

5.2.4 Matrix determinants

For a 2×2 matrix $\mathbf{A}$, the determinant is given by

$$\det(\mathbf{A}) \equiv |\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \tag{56}$$

For a 3×3 matrix $\mathbf{A}$, the determinant is given by

$$\begin{aligned} \det(\mathbf{A}) \equiv |\mathbf{A}| &= \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ &= a_{11}M_{11} - a_{12}M_{12} + a_{13}M_{13} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned} \tag{57}$$

where M_{ij} is the *minor* of the matrix element a_{ij}. These three minors (determinants) are computed from 2×2 submatrices which do not share the row and column indices of each element in the first row. This procedure can be generalized to larger $N \times N$ matrices, in which a negative sign is used for every other column. Thus in general, the determinant $|\mathbf{A}|$ of a square $N \times N$ matrix $\mathbf{A} = [a_{ij}]$ may be computed in terms of a sum involving the minors:

$$|\mathbf{A}| = \det([a_{ij}]) = \sum_{j=1}^{N} (-1)^{1+j} M_{1j} \tag{58}$$

where, for $\mathbf{A} = [a_{mn}]$ with $m, n \in \{1, 2, \ldots, N\}$, the minor $M_{ij} = \det([a_{m\neq i, n\neq j}])$ is the determinant of the submatrix in which $i \neq m$ and $j \neq n$.

Explore More: Cross product

5.2.5 Inverse of a matrix

Understanding how to find the inverse of a matrix is an important concept in linear algebra, with extensive applications in engineering. While the inverse of a matrix is not guaranteed to exist, it can be a powerful tool used for solving systems of linear equations, analyzing electrical circuits, optimization of a variety of systems, and many other applications. The inverse of a square matrix $\mathbf{A}$ is another square matrix, denoted as $\mathbf{A}^{-1}$, such that when $\mathbf{A}$ is multiplied by $\mathbf{A}^{-1}$, the result is the identity matrix $\mathbf{1}$ (sometimes written as $\mathbf{I}$ or $\hat{1}$), such that

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{1} \tag{59}$$

The identity matrix is defined for any size $n \times n$ and consists of ones on the diagonal and zeros elsewhere. Its general form is given as

$$\mathbf{1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{60}$$

Importantly, the identity matrix serves as the multiplicative identity in matrix multiplication. For any $n \times n$ matrix $\mathbf{A}$,

$$\mathbf{1}\mathbf{A} = \mathbf{A}\mathbf{1} = \mathbf{A} \tag{61}$$

Thus, multiplying any matrix by the identity matrix leaves the original matrix unchanged. Two conditions required for a matrix to have an inverse are that it must be square and the determinant of the matrix must be non-zero. If the determinant is zero, the matrix is referred to as *singular* with no defined inverse. There are several methods that may be used to compute the inverse of a matrix, including Gaussian elimination, analytic, and decomposition methods. Additional details regarding these methods may be found in the following *Explore More* appendices.

Explore More: Gaussian elimination

Explore More: Analytic solution to matrix inversion

Since manually computing the inverse of an invertible matrix quickly becomes impractical with respect to the matrix size, it is best to either automate the procedure with a computer program or use an optimized library function in a language such as MATLAB or Python. In MATLAB, for example, the inverse of a matrix may be computed using the inverse function, `inv()`, such that $\mathbf{A}^{-1} = \text{inv}(\mathbf{A})$. If the inputted matrix $\mathbf{A}$ is Hermitian such that $\mathbf{A}^{\dagger} = \mathbf{A}$, this function performs an LDL decomposition and otherwise it performs an LU decomposition; these inversion methods are described in greater detail in their respective *Explore More* appendices below. The results are then used to form a linear system whose solution is the matrix inverse `inv(A)`.

Explore More: LU decomposition

Explore More: LDL decomposition

It is worth noting that forming the explicit inverse of a matrix is rarely necessary. A common mistake is using the `inv()` function in MATLAB to solve a linear system of equations given by $\mathbf{Ax} = \mathbf{b}$, in which $\mathbf{A}$ is the matrix of linear coefficients, $\mathbf{x}$ is the vector of unknowns, and $\mathbf{b}$ is the vector of constants. In MATLAB it is recommended to use the matrix backslash operator `x=A\b` which is more efficient and accurate than `x=inv(A)*b` since other methods, such as Gaussian elimination, may be used to produce the solution without the need to explicitly generate the inverse.

5.2.6 Eigensystem of a matrix

A matrix eigensystem is a concept in linear algebra involving the relationship between a matrix and its eigenvalues and eigenvectors. Understanding eigensystems is crucial for solving many problems in engineering, physics, and applied mathematics.

Explore More: Eigenvalues and eigenvectors

Explore More: Norm of a matrix

Explore More: Singular value decomposition

Homework Problems: Matrices

6 Regression analysis

Learning objectives:

- Explicitly perform linear regression.
- Understand the concept of coefficient of determination as a measure for fit quality.
- Use prior information about a physical system to build good models.
- Understand the limitations of polynomial regression.

Regression analysis is a powerful statistical method used to examine the relationship between a dependent variable and one or more independent variables. The primary goal is to model the dependent variable based on the independent variables, allowing for predictions, inference about the relationships, and adjustment of effects based on the data.

Curve fitting is considered a particularly important and ubiquitous application for extracting knowledge from experimental data. In curve fitting, a set of n points $\{x_i, y_i\}$ where $i \in \{1, 2, ..., N\}$ is given and we wish to determine a function $f(x)$ such that $f(x_1) \approx y_1, ..., f(x_N) \approx y_N$. This process can be performed using the *least squares* method[13] to estimate the parameters of a regression model, which are found by minimizing the sum of the squared differences (known as *residuals*) between observed values and those predicted by the model. Thus, least squares can be used to find the line or curve that best fits the data according to the criterion of minimizing the sum of the squared residuals.

While a polynomial model of a sufficiently high degree may be useful for characterizing the behavior of data in many cases, ultimately the choice of fit function which is most appropriate for modeling the data is largely dependent on the underlying mechanisms which generate the data. Automating the selection of the most appropriate basis function is left for future discussion; however, we will explore the impact made by selecting different types of functions in this section.

[13] Kreyszig, E., *Advanced Engineering Mathematics 10th Ed.*, John Wiley & Sons, Inc. (2011).

6.1 Least squares

The least squares method determines the best-fitting curve or line to a given set of data points by minimizing the sum of the squares of the differences (termed *residuals*) between the observed values and those predicted by the model.

Formally, we begin by considering a series of observations strictly described by real numbers (x_i, y_i) for $i \in \{1, 2, \ldots, N\}$, and a function f which could be linear, polynomial, or any other type parameterized by a parameter vector $\mathbf{a}$ such that $f(\mathbf{x}, \mathbf{a})$ predicts the value of $\mathbf{y}$. The goal of least squares fitting is to find the parameter vector $\mathbf{a}$ with elements a_j where $j \in \{1, 2, \ldots, M\}$ that minimizes the cost function C. Using the L^2 norm (least *squares*), the cost function to be minimized can be written as

$$C(\mathbf{a}) = \sum_{i=1}^{N} r_i^2(\mathbf{a}) = \sum_{i=1}^{N} [f(x_i, \mathbf{a}) - y_i]^2 \tag{62}$$

where r_i is the ith residual, y_i is the ith observed value, $f(x_i, \mathbf{a})$ is the predicted value from the model, and the residuals are squared to ensure they are positive as well as to emphasize larger differences. Note that the residual exponent of 2 in Eq.(62) refers to the L^2 norm, which is commonly used since it *guarantees differentiability* while measuring the shortest distance from the absolute smallest value the cost function $C(\mathbf{a})$ can take: zero. Thus, the function $C(\mathbf{a})$ quantifies the discrepancy between the observed data and the model. By minimizing C, we find the values of $\mathbf{a}$ that make $f(\mathbf{x}, \mathbf{a})$ as close as possible to y_i for all i. The resulting estimate for the parameter vector $\mathbf{a}$ is known as the least squares estimate, which is the solution to the least squares problem.

By taking the gradient of the argument in Eq.(62) and setting this gradient to zero, we can construct gradient equations for the set of model parameters a_j. Thus,

$$\frac{\partial C}{\partial a_j} = 2 \sum_{i=1}^{N} r_i \frac{\partial r_i}{\partial a_j} = 0 \tag{63}$$

Since $r_i = f(x_i, \mathbf{a}) - y_i$, this becomes

$$\frac{\partial C}{\partial a_j} = 2 \sum_{i=1}^{N} r_i \frac{\partial f(x_i, \mathbf{a})}{\partial a_j} = 0 \tag{64}$$

where $j \in \{1, 2, \ldots, M\}$. Although any given problem might require particular expressions for the model selected and its partial derivatives, these gradient equations apply to *all* least squares problems.

6.2 Linear least squares

To solve the least squares problem efficiently, different mathematical models are employed depending on the nature of the function f that models the relationship between the dependent and independent variables. While numerical methods are generally used due to the complexity and dimensionality of the problem, it is possible to construct analytical solutions for models where f is a linear function of the parameters like $f(\mathbf{x}, \mathbf{a}) = \mathbf{X}\mathbf{a}$, in which $\mathbf{X}$ is a matrix of input data and $\mathbf{a}$ is a vector of coefficients. For some applications in signal processing, such as digital image processing, there may exist many features (high *dimensionality*) but relatively few data points. Such scenarios can make the problem more challenging and prone to overfitting.

As a straightforward example, we can construct a quadratic polynomial model $f(x, \mathbf{a}) = a_0 + a_1 x + a_2 x^2$ to fit a set of observed data $\mathbf{x}$ with parameter vector $\mathbf{a} = [a_0 \; a_1 \; a_2]$. This model is *linear* with respect to the model parameters a_i. A simple example of a *nonlinear* model - that is, a model that is nonlinear with respect to the parameters that we seek to optimize - would be

$$f(x, \mathbf{a}) = \frac{a_0 + a_1 x + a_2 x^2}{a_3} \tag{65}$$

However, in such a case, we can simply construct a model that is linear with respect to a new set of coefficients built from the original parameters:

$$f(x,\mathbf{a}) = \left(\frac{a_0}{a_3}\right) + \left(\frac{a_1}{a_3}\right)x + \left(\frac{a_2}{a_3}\right)x^2 = a_0' + a_1'x + a_2'x^2 = f(x,\mathbf{a}') \tag{66}$$

This is an example of dimensionality reduction by simplifying the model without loss of generality. For models that are linear with respect to $\mathbf{a}$, then, we can solve the minimization problem given by Eq.(64) and derive an analytic solution for the model parameters $\mathbf{a}$ that does not require differentiation.

Starting with a matrix of input data $\mathbf{X}$, the optimal parameter vector $\mathbf{a}$ minimizes the sum of squared differences between observed outcomes $\mathbf{y}$ and model predictions $\mathbf{Xa}$, leading to the problem formulation

$$\min_{\mathbf{a}} |\mathbf{Xa} - \mathbf{y}|^2 \tag{67}$$

This is known as the *normal equation of least squares*, and the parameter vector $\mathbf{a}$ may be found analytically for the linear case, assuming the matrix $\mathbf{X}$ is non-singular, that is, it has a non-zero determinant and therefore a defined inverse.

We start by expanding the argument of the minimization problem in Eq.(67):

$$\begin{aligned} |\mathbf{Xa} - \mathbf{y}|^2 &= (\mathbf{Xa} - \mathbf{y})^\intercal (\mathbf{Xa} - \mathbf{y}) \\ &= (\mathbf{Xa})^\intercal \mathbf{Xa} - (\mathbf{Xa})^\intercal \mathbf{y} - \mathbf{y}^\intercal \mathbf{Xa} + \mathbf{y}^\intercal \mathbf{y} \\ &= \mathbf{a}^\intercal \mathbf{X}^\intercal \mathbf{Xa} - \mathbf{a}^\intercal \mathbf{X}^\intercal \mathbf{y} - \mathbf{y}^\intercal \mathbf{Xa} + \mathbf{y}^\intercal \mathbf{y} \end{aligned} \tag{68}$$

By taking the gradient (partial derivative) of Eq.(68) with respect to the parameter vector $\mathbf{a}$ and setting the result to zero, we can solve for the normal equations:

$$\begin{aligned} &\frac{\partial}{\partial \mathbf{a}} \left(\mathbf{a}^\intercal \mathbf{X}^\intercal \mathbf{Xa} - \mathbf{a}^\intercal \mathbf{X}^\intercal \mathbf{y} - \mathbf{y}^\intercal \mathbf{Xa} + \mathbf{y}^\intercal \mathbf{y}\right) = -2\mathbf{X}^\intercal \mathbf{y} + 2\mathbf{X}^\intercal \mathbf{Xa} = 0 \\ &\Rightarrow 2\mathbf{X}^\intercal \mathbf{Xa} = 2\mathbf{X}^\intercal \mathbf{y} \Rightarrow \mathbf{a} = \left(\mathbf{X}^\intercal \mathbf{X}\right)^{-1} \mathbf{X}^\intercal \mathbf{y} \end{aligned} \tag{69}$$

6.2.1 Explicit analytic solutions for regression coefficients

For a set of n data points $\{x_i, y_i\}$ where $i \in \{1, 2, ..., n\}$, we can utilize the least squares method to analytically fit a polynomial of degree N to the set of data and extract explicit solutions for the set of regression coefficients. To fit a polynomial of degree N to data, the condition $n \geq N$ must be satisfied. The regression coefficients (polynomial coefficients) may be found analytically using Gaussian elimination with partial pivoting on the following linear equation. In general, we have

$$\mathbf{X} \cdot \mathbf{a} = \mathbf{Y} \tag{70}$$

where $\mathbf{a}$ is a vector with the regression coefficients as its elements; that is, for a polynomial of degree N, we have the following fit function:

$$y_{\text{fit}} = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N \tag{71}$$

with the coefficient vector given by

$$\mathbf{a}^\intercal = (a_0, a_1, a_2, \cdots, a_N) \tag{72}$$

The elements of matrix $\mathbf{X}$ are given by

$$\mathbf{X}_{kl} = \begin{cases} n, & \text{if } k = l = 1. \\ \sum_{i=1}^{n} x_i^{k+l-2}, & \text{otherwise.} \end{cases} \tag{73}$$

where $\{k, l\} \in \{1, 2, ..., N+1\}$, and the elements of vector $\mathbf{Y}$ are given by

$$\mathbf{Y}_k = \sum_{i=1}^{n} y_i x_i^{k-1} \tag{74}$$

We can solve for $\mathbf{a}$ by computing the inverse of $\mathbf{X}$:

$$\mathbf{a} = \mathbf{X}^{-1} \cdot \mathbf{Y} \tag{75}$$

Example: As an example, we can write out the various matrices and vectors to fit a 2nd-order polynomial (y_N with $N = 2$). Thus, the fit function to be used for polynomial regression is given by

$$y_2 = a_0 + a_1 x + a_2 x^2 \tag{76}$$

The regression coefficients a_0, a_1, and a_2 can be computed by solving Eq. (75) for $\mathbf{a}$; using Eqs. (72)-(74), we can explicitly write out the following set of equations:

$$\mathbf{X} \cdot \mathbf{a} = \begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{bmatrix} = \mathbf{Y}, \tag{77}$$

where the sums are over the set of n data points. Thus, the coefficients can be explicitly computed by inverting $\mathbf{X}$:

$$\mathbf{a} = \mathbf{X}^{-1} \cdot \mathbf{Y} \Rightarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum y_i \\ \sum y_i x_i \\ \sum y_i x_i^2 \end{bmatrix} \tag{78}$$

Finally, in regression analysis, a useful measure of fit is the *coefficient of determination*, R^2. This measure represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s). Additional details may be found below.

Explore More: Coefficient of determination

6.2.2 Noiseless vs. noiseless data

6.2.3 Leveraging prior knowledge

Incorporating prior knowledge about a physical system into regression and statistical analysis enhances the robustness, interpretability, and predictive power of the models developed. Depending on the system being studied, prior knowledge may encompass known sources of noise, historical data trends, and established physical laws. As such, one can use prior knowledge to guide the selection of appropriate variables, inform the functional form of the model, and constrain the parameter space. This approach can not only capture the empirical relationships present in the observed data, but it can also align with the underlying physical principles governing the system.

By embedding well-established physical laws or constraints into the model, one can reduce the complexity of the model or introduce regularization terms that prioritize solutions consistent with prior knowledge. This not only enhances the model's performance on unseen data but also facilitates the interpretation of the results, ensuring that the outcomes are physically plausible and actionable. In engineering and the natural sciences, where decisions and predictions must often be made under uncertainty and the stakes for accuracy can be high, the integration of prior knowledge into statistical models is not only beneficial — it is essential for advancing understanding and making reliable predictions.

A straightforward example of using prior knowledge with linear regression considers the current-voltage characteristics of a Shockley diode. The *ideal* Shockley diode equation that models the current-voltage relationship of a diode under a forward or reverse voltage bias is

$$I = I_0 \left(\mathrm{e}^{\frac{eV_{\mathrm{bias}}}{k_{\mathrm{B}}T}} - 1 \right) \tag{79}$$

where I is the current flowing through the diode, V_{bias} is the voltage across the diode, e is the electron charge, and $k_{\mathrm{B}}T$ is the thermal energy where k_{B} is Boltzmann's constant and T is the absolute temperature of the diode, and I_0 is the reverse-bias saturation current at temperature T.

For this example, we assume ignorance of the exponential nature of the diode equation and collect six measurements of the current $I(V_{\mathrm{bias}})$ with three positive and three negative bias voltage values across the diode, as shown in Fig. 8.

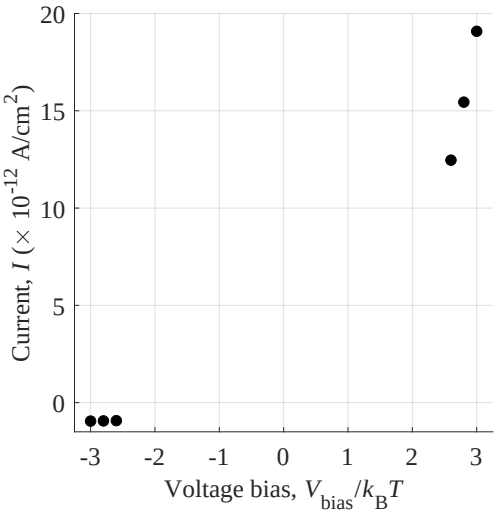


Figure 8: Data points are shown as black dots. For voltage values $V_{\mathrm{bias}}/k_{\mathrm{B}}T \in \{-3.0, -2.8, -2.6, 2.6, 2.8, 3.0\}$, the respective measured current I values are $\{-0.9502, -0.9392, -0.9257, 12.4637, 15.4446, 19.0855\}$ in units of pA/cm^2. Parameters used are $T = 300$ K, $I_0 = 10^{-12}$ A/cm^2. MATLAB code used is `diode_fitting.m`.

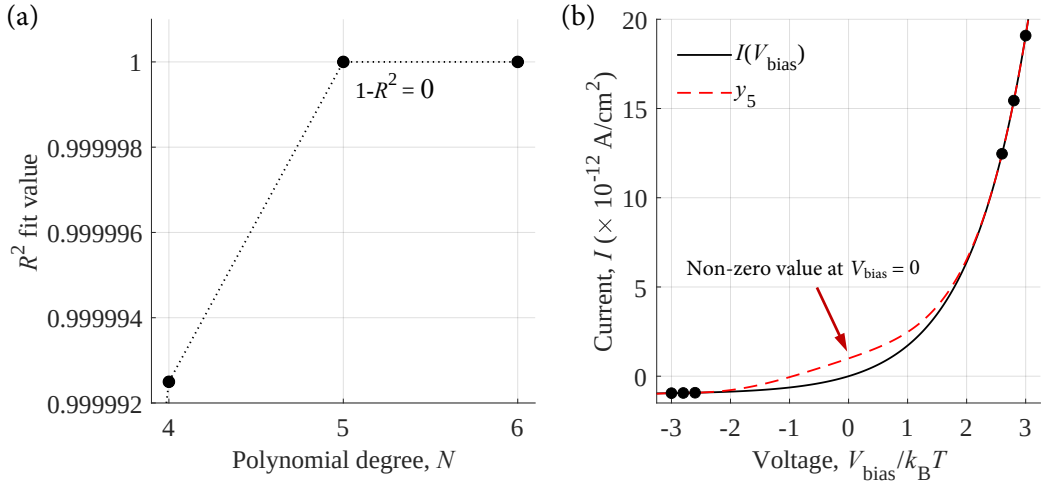


Figure 9: (a) R^2 values resulting from fitting the measured $V_{\mathrm{bias}}/V_{\mathrm{T}} = \{-3.0, -2.8, -2.6, 2.6, 2.8, 3.0\}$. For $N \geq n - 1$, the maximal $R^2 = 1$ value is achieved with $N = 5$, resulting in overfitting. (b) The best fit for $N < (n = 6)$ is found with a 5th-order model (dashed red curve) with coefficient of determination $R^2 = 1$ and coefficients $a_0 = 0.9959$, $a_1 = 1.1305$, $a_2 = 0.1303$, $a_3 = 0.1178$, $a_4 = 0.0852$, $a_5 = 0.0142$, with the (noiseless) Shockley diode curve $I(V_{\mathrm{bias}})$ shown in black for comparison. Parameters are as in Fig. 8. MATLAB code used is `diode_fitting.m` with zero-padding parameter `nzero` (n_{zero}) = 0 on line 21.

For the sake of simplicity, we also assume the system to be *noiseless* such that there is no uncertainty in the measurement data. Because the data trend appears to follow a smooth curve, we can use linear regression to try fitting a polynomial model and capture the general behavior of the diode. To extract the coefficient of determination R^2 and coefficient vector $\mathbf{a}$ for a hypothesized Nth-order polynomial model, we can numerically implement equations (75) and (146) in MATLAB or in any other scientific computing language. These are equivalent to using MATLAB's `polyfit()` function.

The numerical implementation of these equations is straightforward, and MATLAB may be used to demonstrate polynomial regression. In general, for n points, you can fit a polynomial of degree $n-1$ to *exactly* pass through the points. This is observed for the case of the $N=5$ polynomial fit y_5 in Fig. 9(b), which has the maximal coefficient of determination value $R^2=1$.

It is important to note that *overfitting* will occur for polynomial models of order $N \geq (n-1)$ data points, resulting in spurious high values of R^2. If we plot the first polynomial model which results in $R^2=1$ as a function of voltage bias (as shown in Figs. 9(b) and 11(a)), non-zero current is predicted when the voltage bias is zero, implying the non-physical existence of free energy in the system, despite the fit measure R^2 indicating an optimal fit to all available data points.

We can use our knowledge of energy conservation as prior information and *force* the polynomial model to pass through $(V_{\text{bias}}, I) = (0,0)$ by *zero-padding*, that is, including additional, experimentally *unmeasured* data points at $(0,0)$. Doing so *weights* the regression coefficients such that $a_0 \to 0$ as the number of zero points n_{zero} are increased while the remaining coefficients are modified to minimize the difference between the Nth-order polynomial model and the remaining data points. Using this prior knowledge about the system results in a significantly better fit as far as approximating Eqn. (79), as shown in Fig. 10(b) and Fig. 11(b). Finally, it is important to note that arbitrarily adding zero values to a dataset for linear regression without a clear, contextually relevant justification can distort the analysis, leading to incorrect interpretations.

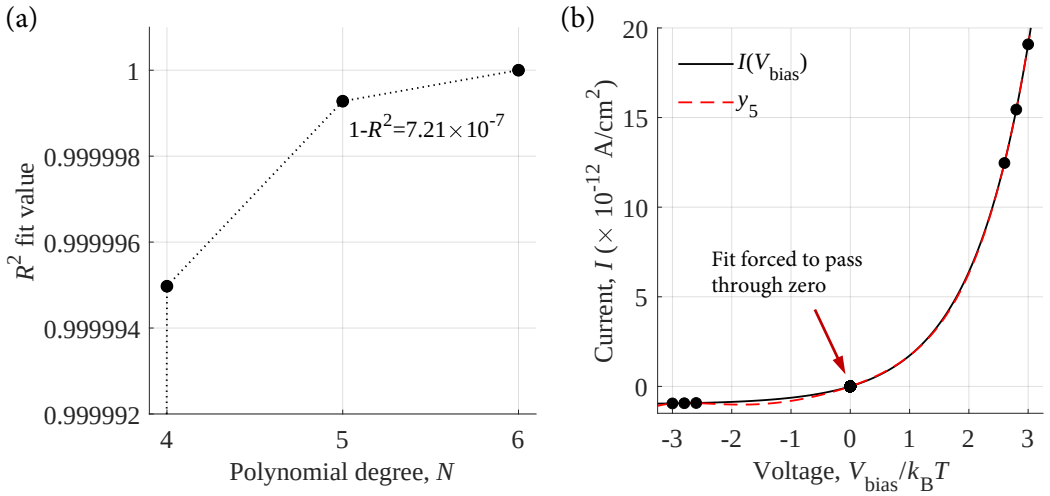


Figure 10: (a) R^2 values resulting from fitting the data set from Fig. 8 but now with zero-padding in which $n_{\text{zero}} = 10^3$ data points at $(V_{\text{bias}}, I) = (0,0)$ are included to ensure that the fitted curve goes through zero. An overall better fit is achieved despite $R^2 < 1$ for $N=5$ as there are now 7 total unique data points. (b) With zero-padding, the 5th-order model y_5 (dashed red curve) results in a better fit with fit measure $1-R^2 = 7.21\times10^{-7}$ and coefficients $a_0 = 5.4943\times10^{-7}$, $a_1 = 1.1305$, $a_2 = 0.3854$, $a_3 = 0.1178$, $a_4 = 0.0691$. $a_5 = 0.0142$, and the noiseless Shockley diode curve $I(V_{\text{bias}})$ is shown in black for comparison. Parameters are as in Fig. 8. MATLAB code used is `diode_fitting.m` with zero-padding parameter `nzero` (n_{zero}) $= 10^3$ on line 21.

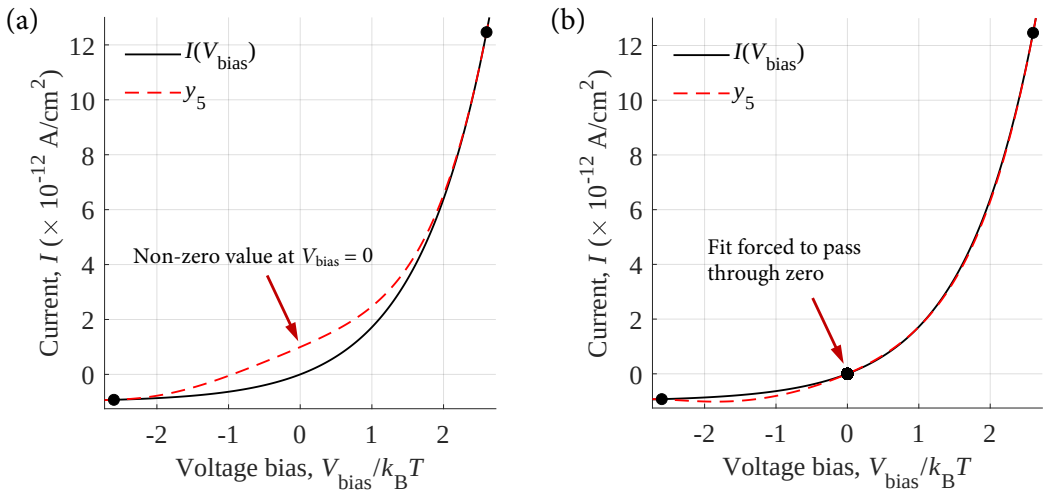


Figure 11: (a) Zoomed-in view of Fig. 9(b) comparing polynomial fit with analytic diode function. Note that for $V_{\text{bias}} = 0$, the polynomial model predict non-zero current, implying energy non-conservation. Parameters are as in Fig. 8 with zero-padding parameter $n_{\text{zero}} = 0$. (b) Zoomed-in view of Fig. 10(b) showing how zero-padding forces the fitted model to go through zero, resulting in significantly better fits when compared to the idealized diode model. Parameters are as in Fig. 8 with zero-padding parameter $n_{\text{zero}} = 10^3$. MATLAB code used is `diode_fitting.m`.

7 Introduction to photons

Photons are elementary particles of light, which are crucial to understanding both classical optics and quantum mechanics. Unlike everyday objects, photons do not have mass and always travel at the speed of light in a vacuum. They are the carriers of the electromagnetic force, meaning that whenever electromagnetic energy is transferred, photons are involved.

Photons can exhibit key features that distinguish them from everyday classical objects. One such feature is wave-particle duality since photons can exhibit particle-like and wave-like behavior. As particles, they can be counted and interact with matter in a discrete manner. However, they also exhibit wave-like behavior, such as self-interference and diffraction, which are characteristics of waves. Wave-particle duality is central to understanding how light behaves in different contexts, including optics and quantum technologies. Unlike classical particles, photons can be identical and indistinguishable. They can have identical *state* characteristics, including polarization, wavelength, and phase. This state can be mathematically described using a vector. *Linearity* allows for the linear combination, or *superposition*, of photon state vectors. For example, if a photon can exhibit two sets of physical characteristics respectively described by vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, then the photon can *also* exist as a linear superposition, represented by a linear combination of the two vectors so that $\mathbf{v}_3 = a\mathbf{v}_1 + b\mathbf{v}_2$, where, in general, a and b are complex scalar coefficients.

A quantized particle with wave properties moving from point a to b in free space might take a path s_n and arrive at b with an associated amplitude magnitude A_n and phase ϕ_n, where n labels the path. If the particle leaves point a at time $t = 0$ then the total quantum field amplitude at point b at time t is the *sum* of amplitudes $\mathbf{A}_{\text{tot}}(t) = A_1 e^{ik s_1 - i\omega t} + A_2 e^{ik s_2 - i\omega t} + \ldots$. In quantum mechanics the probability of detecting the particle at position b is $P_b(t) = \mathbf{A}^*_{\text{tot}}\mathbf{A}_{\text{tot}} = |\mathbf{A}_{\text{tot}}|^2$, which is the magnitude of the total probability amplitude $\mathbf{A}_{\text{tot}}$ squared.

As shown in Fig. 12, the shortest path between a and b for a particle of energy $E = \hbar\omega$ moving in free space is $s(n_0)$. A path such as $s(n_1)$ might have a small difference in length compared to $s(n_0)$ and so almost the same phase at b as the particle that took path $s(n_0)$. A different path such as $s(n_2)$ is much longer compared to $s(n_0)$ and hence can have a very different phase at b.

The paths between a and b are $s(n)$, where n labels each path in the ordered sequence of increasing path length. For the case being considered, the particle is free to propagate at energy $\hbar\omega$ using any of the infinite

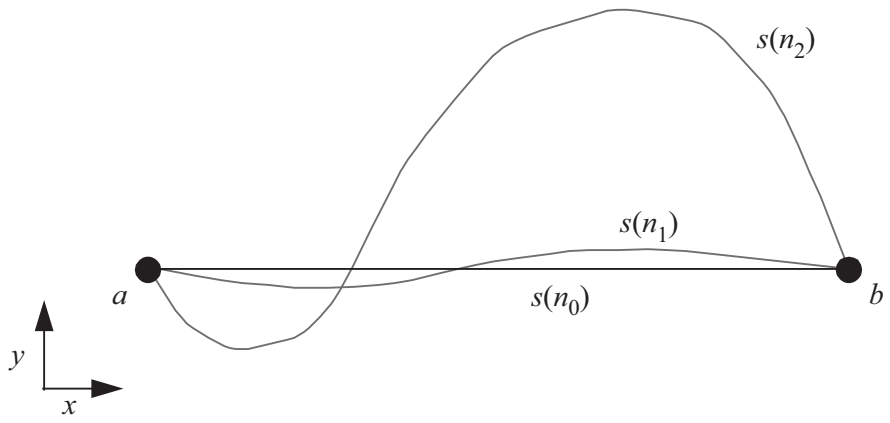


Figure 12: Illustration of paths s a particle can take in free space between position a and b. The shortest path is a straight line, $s(n_0)$. Other paths such as $s(n_1)$ can make small deviations from $s(n_0)$ and so have almost the same phase at b as $s(n_0)$. Paths such as $s(n_2)$ are much longer compared to $s(n_0)$ and hence can have a very different phase at b.

number of allowed paths between a and b. In free space, each path has the same amplitude magnitude A_0 and so the total probability amplitude at position b is an integral over all paths. The resulting path integral[14] is

$$\mathbf{A}_{\text{tot}} = A_0 \mathrm{e}^{-\mathrm{i}\omega t} \int \mathrm{e}^{\mathrm{i}ks(n)} \mathrm{d}n \tag{80}$$

Because $s(n_0)$ has the minimum length corresponding to a straight-line path, then

$$\frac{\mathrm{d}s(n_0)}{\mathrm{d}n} = 0 \tag{81}$$

Expanding $s(n)$ in a Taylor series for small variations in path length about this minimum results in

$$s(n) = s(n_0) + \frac{\mathrm{d}s(n_0)}{\mathrm{d}n}(n - n_0) + \mathcal{O}((n - n_0)^2) \tag{82}$$

Hence, to first order, all paths near $s(n_0)$ have the same length and the phases directly add in the integral to make a large contribution to $\mathbf{A}_{\text{tot}}$. Other paths can result in phases that vary considerably and that, when added, tend to cancel each other out. It is in this way, and in contrast to the classical case, quantum mechanics allows a particle moving between point a and b in free space to explore *all* paths.

While this path integral description shows how the classical straight-line path of a particle in free space moving between point a and b *emerges* from the sum of all possible quantum paths, in general, the method is not an efficient way to solve practical problems and so a different approach is needed.

7.1 An experiment to prove the photon exists

Many years after the initial suggestion that light is quantized the first laboratory experiments were performed that proved the existence of the photon. Famously, Kimble, Dagenais, and Mandel published a paper in 1977 showing that light is made up of discrete photons, each of which can create a single "click" in a detector.[15] In the 1980's Grangier, Roger, and Aspect were able to refine these experiments[16] and also show the interference of a single photon, thereby demonstrating in complementary experiments the particle and wave nature of the photon.

[14] R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals*, New York, McGraw-Hill, 1965 (ISBN 978-0-07-020650-2).

[15] H. J. Kimble, M. Dagenais and L. Mandel, *Phys. Rev. Lett.* **39**, 691 (1977).

[16] P. Grangier, G. Roger and A. Aspect, *Europhys. Lett.* **1**, 173 (1986).

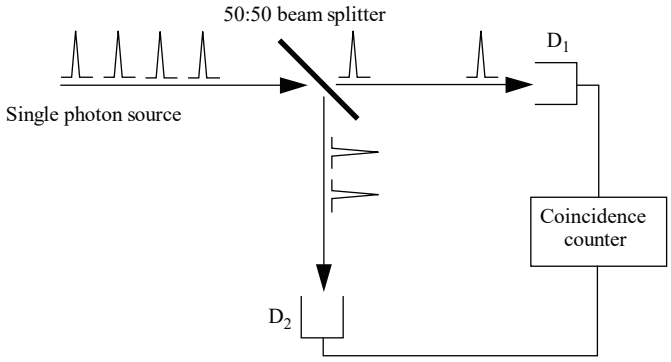


Figure 13: An optical source emits single photons that are incident on an ideal, lossless, symmetric, 50:50 beam splitter. Single-photon detectors D_1 and D_2 are placed at the two output ports of the beam splitter. Because the photon is an indivisible elementary particle it must either be detected by D_1 or D_2, but not both.

It is possible to perform experiments similar to those of Kimble, Dagenais, and Mandel using laser diode based single-photon sources, fiber-optic components, and single-photon detectors as illustrated in Fig. 13. An optical source emits single photons that are, on average, spaced apart in time by $\langle \tau_{\rm ph} \rangle$. The photon flux is incident on an ideal, lossless, symmetric, 50:50 beam splitter with linear response. Single-photon detectors D_1 and D_2 are placed at the two output ports of the beam splitter. In the simplest configuration the optical path length between the beam splitter output port and the associated detector is the same and each detector response time is very much smaller than $\langle \tau_{\rm ph} \rangle$. The photon path taken through the beam splitter can only be inferred *after* it is detected by D_1 or D_2. The inferred photon path selection is purely *random* and hence non-causal. The photon is either detected by D_1 or D_2, but not both, and it is fundamentally not known beforehand at which output port the photon will be detected. When properly implemented, the experiment reveals that there are no coincidence counts between the two detectors thereby proving that the photon is an indivisible quantized particle and therefore elementary.

The absence of single-photon coincidence counts in the experiment is something that cannot be explained by a classical wave model of light. Maxwell's electromagnetic waves appear at both detectors at the same time and so give rise to coincidence detection - something that is not observed experimentally. A classical description using Maxwell's equations is accurate when there are a large number of incoherent photons associated with a particular electromagnetic field. If there are very few photons, special conditions involving identical indistinguishable photons, or a coherent superposition of photons, then a quantum description is appropriate.

7.2 Random number generation and stochastic computing

A single photon incident on an ideal, lossless, symmetric, 50:50 beam splitter with linear response has an exactly 50% chance of being detected at one of the two output ports. This pure random behavior is *guaranteed* by quantum mechanics and can be used as a mechanism to generate random numbers for applications that include computation.

In *stochastic computing*, numbers are represented as probabilities p of a binary 1 or 0 signal in a clocked bit-stream of length $n_{\rm bits}$. As $n_{\rm bits} \to \infty$ the average value of the signal is p distributed in the interval $[0, 1]$. A circuit that can perform multiplication followed by addition on stochastic data is illustrated in Fig. 14. Since basic linear algebra operations involve multiplication of matrix $\mathbf{A}$ with vector $\mathbf{s}$, stochastic computing might be tasked with computing both $\mathbf{x} = \mathbf{A}\mathbf{s}$ and $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$. For the simplest 2×2 matrix, $\mathbf{x} = \mathbf{A}\mathbf{s}$ may be written as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11}\, a_{12} \\ a_{21}\, a_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

so that evaluation of $x_1 = a_{11} \times s_1 + a_{12} \times s_2$ requires multiplication and addition (also known as multiply-accumulate, or MAC). In the limit $n_{\text{bits}} \to \infty$, the output of the AND function on stochastic input streams a_{11} and s_1 is the product of probabilities $p(a_{11})$ and $p(s_1)$. The sum of the two AND outputs is found by multiplexing using a select (SEL) that has a random value of binary 1 or 0 each clock cycle. In this way, the average value of the MAC output is scaled to fall in the interval [0,1].

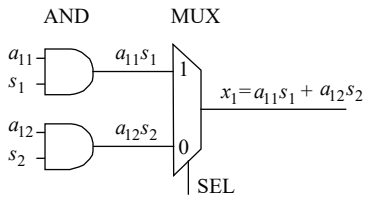


Figure 14: Illustration of the AND and MUX functions to perform matrix element multiplication followed by addition in stochastic computing. Numbers are represented as probabilities of a binary 1 or 0 signal in a clocked bit-stream of length n_{bits}. Select (SEL) has a random value of binary 1 or 0 for each clock cycle, and the output is scaled to fall in the interval [0,1].

Using random bit streams to represent numbers has the advantage that the multiplication and addition circuits are very simple to implement. There is also some inherent robustness to random errors in the bit stream. However, the accuracy of the calculation is sensitive to correlations between random number generators, the finite number of bits used to represent a number, and the condition of the matrix. While the use of a single photon source can, in principle, be used to guarantee random number generation physically, the number of bits, n_{bits}, and matrix condition number are also important considerations when solving $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$ for which the determinant of matrix $\mathbf{A}$ must be calculated.

Explore More: Logic gates

Homework Problems: Logic gate design

Explore More: Condition number of a matrix

8 Photon detection after a beam splitter

Controlling the interaction of photons with matter is of fundamental and practical interest. A basic component in photonics that requires a model of photon-matter interaction is the beam splitter. This is considered next.

As illustrated schematically in Fig. (15), a beam splitter has two input ports and two output ports. The reflection and transmission amplitudes experienced by a linearly polarized photon at port 1 are $r_{\text{ph},1}$ and $t_{\text{ph},1}$, respectively. Similarly, at port 2 they are $r_{\text{ph},2}$ and $t_{\text{ph},2}$. *If* there are integer n_1 photons incident at port 1 and integer n_2 photons incident at port 2 then the input Fock-state is $|n_1, n_2\rangle_{\text{in}}$.[17] Zero photons at an input port are described by the vacuum state. The output of the beam splitter has photons with quantum field amplitude a_3 at port 3 and amplitude a_4 at port 4.

It is a consequence of unitarity that an ideal, lossless, symmetric, 50:50 dielectric beam splitter has $r_{\text{ph},1} = r_{\text{ph},2} = r_{\text{ph}}$ and this can be chosen such that

$$r_{\text{ph}} = \frac{-1}{\sqrt{2}} \tag{83}$$

and $t_{\text{ph},1} = t_{\text{ph},2} = t_{\text{ph}}$ is

[17] Ignore, for the moment, that the input state could be a superposition involving quantum field amplitude at both port 1 and 2.

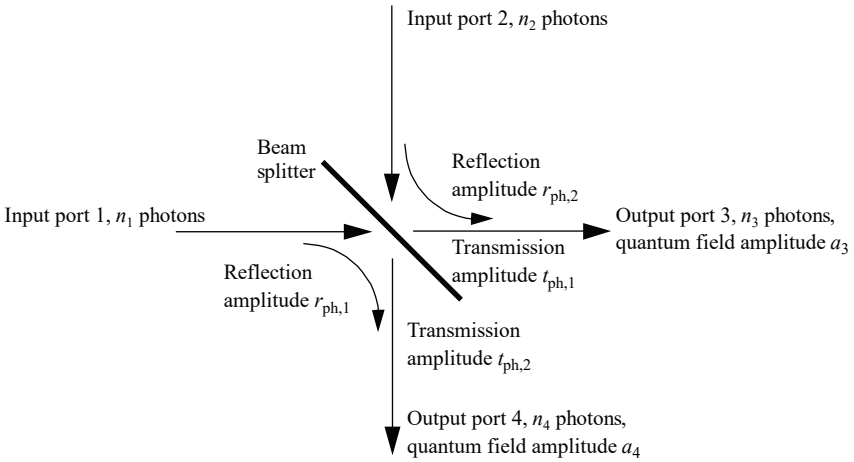


Figure 15: Sketch of a beam splitter showing input ports 1 and 2 and output ports 3 and 4. In the case considered, there are integer n_1 photons incident at port 1 and integer n_2 photons incident at port 2. Single-photon quantum field transmission amplitude is $t_{\mathrm{ph},1,2}$ and reflection amplitude is $r_{\mathrm{ph},1,2}$.

$$t_{\mathrm{ph}} = \frac{\mathrm{i}}{\sqrt{2}} \tag{84}$$

The origin of the phase difference between transmission and reflection amplitude is described in the *Explore More* appendix linked below.

Explore More: Origin of beam splitter amplitudes

8.1 An integer number of photons at each input port of a beam splitter

A photon number input state $|n_1, n_2\rangle_{\mathrm{in}}$ corresponds to *integer* n_1 photons incident at port 1 and *integer* n_2 photons incident at port 2. The output of the beam splitter has photons with quantum field amplitude a_3 at port 3 and amplitude a_4 at port 4. The input state of the beam splitter $|n_1, n_2\rangle_{\mathrm{in}}$ is connected to the output state of the beam splitter $|a_3, a_4\rangle_{\mathrm{out}}$ by a linear transformation $|a_3, a_4\rangle_{\mathrm{out}} = \mathbf{B}|n_1, n_2\rangle_{\mathrm{in}}$ where $\mathbf{B}$ is a 2×2 matrix

$$\mathbf{B} = \begin{bmatrix} t_{\mathrm{ph},1} & r_{\mathrm{ph},2} \\ r_{\mathrm{ph},1} & t_{\mathrm{ph},2} \end{bmatrix} \tag{85}$$

so that

$$\begin{bmatrix} a_3 \\ a_4 \end{bmatrix} = \mathbf{B} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} = \begin{bmatrix} t_{\mathrm{ph},1} & r_{\mathrm{ph},2} \\ r_{\mathrm{ph},1} & t_{\mathrm{ph},2} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \tag{86}$$

If the photons incident on the beam splitter are *indistinguishable* then the probability of transmission or reflection must take into account the number of ways of arranging the photons among themselves. For example with $n_2 = 0$ the probability that a single beam of n_1 indistinguishable photons is transmitted as n_3 photons and reflected as $n_1 - n_3 = n_4$ photons at an ideal, lossless, symmetric, 50:50 beam splitter is given by

$$P\left(n_1, n_2 = 0, n_3, n_4 = n_1 - n_3\right) = \frac{n_1!}{n_3!(n_1 - n_3)!} \left(\frac{1}{2}\right)^{n_1} = \binom{n_1}{n_3} \left(\frac{1}{2}\right)^{n_1} \tag{87}$$

where the binomial coefficient represents number of ways of choosing n_3 indistinguishable photons from a set of n_1 indistinguishable photons.

If the photons are *distinguishable* the probability that a single beam of n_1 photons is transmitted as n_3 photons at the beam splitter would have the smaller value

$$P\left(n_1, 0, n_3, n_4\right) = \left(\frac{1}{2}\right)^{n_1} \tag{88}$$

Photons are indistinguishable if they have the same polarization, the same frequency, the same phase, and arrive at the detector at the same time. Photon polarization, frequency, and phase are internal degrees of freedom. One way to continuously tune the system from quantum (indistinguishable) to classical (distinguishable) behavior is to introduce a delay between the detected time of arrival of photons[18].

8.2 Transmission of a single photon at a beam splitter

If the total number of Fock-state photons incident on the 50:50 beam splitter is $n_{\rm tot} = n_1 + n_2 = 1$, then there is either one photon present at port 1 or one photon present at port 2. This means input state $|n_1 = 1, n_2 = 0\rangle_{\rm in}$ has only one path to output state $|n_3 = 1, n_4 = 0\rangle_{\rm out}$ and the quantum field amplitude at port 3 is $t_{\rm ph}$:

$$|n_1 = 1, n_2 = 0\rangle_{\rm in} \rightarrow |n_3 = 1, n_4 = 0\rangle_{\rm out} \; : \; t_{\rm ph} = \frac{\rm i}{\sqrt{2}} \tag{89}$$

Similarly, the input state $|1, 0\rangle_{\rm in}$ has only one path to output state $|0, 1\rangle_{\rm out}$ and the quantum field amplitude at port 4 is $r_{\rm ph}$:

$$|n_1 = 1, n_2 = 0\rangle_{\rm in} \rightarrow |n_3 = 0, n_4 = 1\rangle_{\rm out} \; : \; r_{\rm ph} = \frac{-1}{\sqrt{2}} \tag{90}$$

The photon number detection probability is just the magnitude squared of the quantum field amplitude so that at the output port 3 detector

$$P_{\rm out}(n_1 = 1, n_2 = 0, n_3 = 1, n_4 = 0) = |t_{\rm ph}|^2 = \left|\frac{\rm i}{\sqrt{2}}\right|^2 = \frac{1}{2} \tag{91}$$

and at the output port 4 detector

$$P_{\rm out}(n_1 = 1, n_2 = 0, n_3 = 0, n_4 = 1) = |r_{\rm ph}|^2 = \left|\frac{-1}{\sqrt{2}}\right|^2 = \frac{1}{2} \tag{92}$$

Placing a single photon at input port 2, so that the input state $|n_1 = 0, n_2 = 1\rangle_{\rm in}$, produces similar results.

Table 1: Single photon detection probability after a 50:50 beam splitter.

$n_{\rm tot} = 1$	$\|1, 0\rangle_{\rm out}$	$\|0, 1\rangle_{\rm out}$
$\|1, 0\rangle_{\rm in}$	$\frac{1}{2}$	$\frac{1}{2}$
$\|0, 1\rangle_{\rm in}$	$\frac{1}{2}$	$\frac{1}{2}$

As indicated in Table 1, the probabilities are the same as the flux ratios predicted for a classical electromagnetic wave interacting with the same beam splitter. However, the situation changes dramatically if there are two or more identical indistinguishable photons interacting with the beam splitter and subsequently detected. This is considered next.

[18] Y.-S. Ra, M. C. Tichy, H.-T. Lim, O. Kwon, F. Mintert, A. Buchleitner, and Y.-H. Kim, *Proc. Nat. Academy Sci.* **110**, 1227 (2013).

8.3 The Mandel effect: transmission of two indistinguishable photons at a beam splitter

In general, if there is an integer number of photons at the inputs of a beam splitter, then the Fock state is $|n_1, n_2\rangle_{\rm in}$ with integer n_1 photons at input port 1 and integer n_2 photons at input port 2. Likewise a Fock output state of the beam splitter $|n_3, n_4\rangle_{\rm out}$ has n_3 photons at output port 3 and n_4 photons at output port 4.

A biphoton source can be used to create two indistinguishable photons. These can be input to the two input ports of an ideal, lossless, symmetric, 50:50 beam splitter. Every possible combination of inferred photon paths through the beam splitter that is consistent with the exchange-symmetric product states for the boson two-particle system must be accounted for. In contrast to the description of classical *particles*, indistinguishable quantum particles may be viewed as *simultaneously* experiencing *every possible path* through the system.

If one photon is introduced at input port 1 and the other at input port 2, then the input state is $|n_1 = 1, n_2 = 1\rangle_{\rm in}$. This input state is transformed to output states by passing through the beam splitter so that $|n_1 = 1, n_2 = 1\rangle_{\rm in} \rightarrow |n_3, n_4\rangle_{\rm out}$. When $n_1 = n_2$, there are just three exchange-symmetric output product states. Setting reflection coefficient $r_{\rm ph} = -1$ and transmission coefficient $t_{\rm ph} = {\rm i}$ results in non-normalized output states

$$|n_1 = 1, n_2 = 1\rangle_{\rm in} \rightarrow |n_3 = 2, n_4 = 0\rangle_{\rm out} : t_{\rm ph} r_{\rm ph} = -{\rm i} \tag{93}$$

$$|n_1 = 1, n_2 = 1\rangle_{\rm in} \rightarrow |n_3 = 1, n_4 = 1\rangle_{\rm out} : \frac{r_{\rm ph} r_{\rm ph} + t_{\rm ph} t_{\rm ph}}{\sqrt{2}} = \frac{(1-1)}{\sqrt{2}} = 0 \tag{94}$$

$$|n_1 = 1, n_2 = 1\rangle_{\rm in} \rightarrow |n_3 = 0, n_4 = 2\rangle_{\rm out} : r_{\rm ph} t_{\rm ph} = -{\rm i} \tag{95}$$

For the case $|n_1 = 1, n_2 = 1\rangle_{\rm in} \rightarrow |n_3 = 1, n_4 = 1\rangle_{\rm out}$ there are two indistinguishable paths the photons can take from input to output that can be inferred after detection. They are either both reflected or both transmitted through the beam splitter. Each inferred path after detection is equally likely in the 50:50 beam splitter so the photons may be considered to be simultaneously experiencing both processes with the same weight. The superposition of product amplitudes $(r_{\rm ph} r_{\rm ph} + t_{\rm ph} t_{\rm ph})/2$ describes this. The photon field amplitude interference that results in $(r_{\rm ph} r_{\rm ph} + t_{\rm ph} t_{\rm ph})/2 = 0$ occurs because the detectors are unable to distinguish between the two two-photon paths.

The photon-number detection probabilities at the output ports are *proportional* to the absolute value squared of the non-normalized quantum amplitudes

$$P_{\rm out}^{\rm non}\left(n_1 = 1, n_2 = 1, n_3 = 2, n_4 = 0\right) = |t_{\rm ph} r_{\rm ph}|^2 = 1 \tag{96}$$

$$P_{\rm out}^{\rm non}\left(n_1 = 1, n_2 = 1, n_3 = 1, n_4 = 1\right) = \left|\frac{r_{\rm ph} r_{\rm ph} + t_{\rm ph} t_{\rm ph}}{\sqrt{2}}\right|^2 = \left|\frac{1-1}{\sqrt{2}}\right|^2 = 0 \tag{97}$$

$$P_{\rm out}^{\rm non}\left(n_1 = 1, n_2 = 1, n_3 = 0, n_4 = 2\right) = |r_{\rm ph} t_{\rm ph}|^2 = 1 \tag{98}$$

Normalization of $P_{out,non}$ output values in Eqns. (96) – (98) enables interpretation as probability $P_{\rm out}$. Normalization may be achieved via division by the sum

$$P_{\rm sum} = \sum_j P_{j,{\rm out}}^{\rm non} \tag{99}$$

In this case $P_{\rm sum} = 2$ and so the normalized probability values are

$$P_{\rm out}\left(n_1 = 1, n_2 = 1, n_3 = 2, n_4 = 0\right) = \frac{|t_{\rm ph} r_{\rm ph}|^2}{P_{\rm sum}} = \frac{1}{2} \tag{100}$$

$$P_{\rm out}\left(n_1 = 1, n_2 = 1, n_3 = 1, n_4 = 1\right) = 0 \tag{101}$$

$$P_{\rm out}\left(n_1 = 1, n_2 = 1, n_3 = 0, n_4 = 2\right) = \frac{|r_{\rm ph} t_{\rm ph}|^2}{P_{\rm sum}} = \frac{1}{2} \tag{102}$$

If there is one indistinguishable photon at each input port then *the detected quantum amplitudes interfere and cancel exactly* so there is precisely *zero* probability of detecting one photon at each output port. The zero probability of detecting a $|n_3 = 1, n_4 = 1\rangle_{\rm out}$ output state when there is a $|n_1 = 1, n_2 = 1\rangle_{\rm in}$ input state is a strong quantum correlation effect first measured by Hong Ou and Mandel [19]. In addition, one indistinguishable photon at each input port can *only* result in *two* photons detected at an output port. This effect, also driven by the symmetry of identical indistinguishable boson particles in quantum mechanics, is an example of *photon bunching*.

If two photons are introduced at input port 1 and zero at input port 2 then the input state is $|n_1 = 2, n_2 = 0\rangle_{\rm in}$ and the possible output state amplitudes are proportional to

$$|n_1 = 2, n_2 = 0\rangle_{\rm in} \rightarrow |n_3 = 2, n_4 = 0\rangle_{\rm out} : t_{\rm ph} t_{\rm ph} = -1 \tag{103}$$

$$|n_1 = 2, n_2 = 0\rangle_{\rm in} \rightarrow |n_3 = 1, n_4 = 1\rangle_{\rm out} : t_{\rm ph} r_{\rm ph} + r_{\rm ph} t_{\rm ph} = -\frac{2{\rm i}}{\sqrt{2}} = -\sqrt{2}{\rm i} \tag{104}$$

$$|n_1 = 2, n_2 = 0\rangle_{\rm in} \rightarrow |n_3 = 0, n_4 = 2\rangle_{\rm out} : r_{\rm ph} r_{\rm ph} = 1 \tag{105}$$

In this case $P_{\rm sum} = 4$ and the corresponding photon-number detection probabilities at the output ports are

$$P_{\rm out}\left(n_1 = 2, n_2 = 0, n_3 = 2, n_4 = 0\right) = \frac{|t_{\rm ph} t_{\rm ph}|^2}{P_{\rm sum}} = \frac{1}{4} \tag{106}$$

$$P_{\rm out}\left(n_1 = 2, n_2 = 0, n_3 = 1, n_4 = 1\right) = \frac{|t_{\rm ph} r_{\rm ph} + r_{\rm ph} t_{\rm ph}|^2}{P_{\rm sum}} = \frac{1}{2} \tag{107}$$

$$P_{\rm out}\left(n_1 = 2, n_2 = 0, n_3 = 0, n_4 = 2\right) = \frac{|r_{\rm ph} r_{\rm ph}|^2}{P_{\rm sum}} = \frac{1}{4} \tag{108}$$

When the total number of indistinguishable photons $n_{\rm tot} = 2$ there are three possible input states and three possible output states with detected output probabilities as indicated in Table 2 and represented graphically in Figure 16. If there is one indistinguishable photon at each input port then the detected output is two photons at one of the output ports. In general, if there is an equal number of indistinguishable photons at each input port ($n_1 = n_2$), then it is not possible to have an odd number of photons at an output port.

Table 2: Output probabilities of two identical indistinguishable photons interacting with an ideal, lossless, symmetric, 50:50 beam splitter.

Input state $\|n_1, n_2\rangle$	Output state $\|n_3, n_4\rangle$		
	$\|n_3 = 2, n_4 = 0\rangle$	$\|n_3 = 1, n_4 = 1\rangle$	$\|n_3 = 0, n_4 = 2\rangle$
$\|2, 0\rangle$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$\|1, 1\rangle$	$\frac{1}{2}$	0	$\frac{1}{2}$
$\|0, 2\rangle$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$

Two-photon quantum interference (the $n_{\rm tot} = 2$ case) is not interference of two separate photons at the beam splitter but rather it is interference of the two two-photon amplitudes at the detectors. The photon paths can only be inferred *after* detection at the detectors. This is a consequence of the standard Copenhagen interpretation of quantum mechanics in which the properties of a system are only obtained *after* interaction between the quantum system and the measurement instrument – in this case the detectors. In fact photons do not have to arrive simultaneously at the beam splitter to have their quantum field amplitudes interfere at the detectors; rather it is only that the inferred two two-photon paths must be indistinguishable [20]. In practice, it is the *inferred* interpretation of indistinguishable two-photon paths that may be used as a convenient way to predict quantum field amplitude interference.

[19] C. K. Hong, Z. Y. Ou, and L. Mandel, *Phys. Rev. Lett.* **59**, 2044 (1987).

[20] T. B. Pittman, D. V. Strekalov, A. Migdall, M. H. Rubin, A. V. Sergienko, and Y. H. Shih, *Phys. Rev. Lett.* **77**, 1917 (1996).

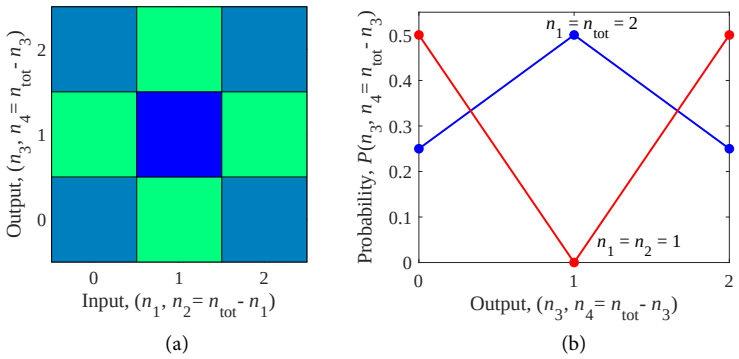


Figure 16: (a) Probability of photon number detection as a function of input and output states of an ideal, lossless, symmetric, 50:50 beam splitter. There are n_1 photons at input port 1 and n_2 photons at input port 2 when there is a total of $n_{\text{tot}} = 2$ indistinguishable photons in the system. For the case when $n_1 = 1$ and $n_2 = 1$ there is zero probability that the value of $n_3 = 1$ and that the value of $n_4 = 1$. (b) Dots show output probability into output port 3 for the case when the input port 1 contains $n_1 = n_{\text{tot}} = 2$ and when $n_1 = n_2 = n_{\text{tot}}/2 = 1$. Lines connecting dots are to guide the eye.

8.3.1 Experimental demonstration of the Mandel effect

A Mandel effect quantum interference experiment results in a reduction in coincidence counts below that expected from a classical light source. The reduction is called the *Mandel dip*. Observing the Mandel dip requires an experimental setup that typically includes the following components:

- **Laser source**: This is a stabilized high-power laser diode with single-mode, spectrally narrow line-width, emission peaked at photon energy E.
- **Biphoton source**: This a source that emits indistinguishable biphotons via the interaction of laser light with a nonlinear medium. A typical mechanism for biphoton generation is *spontaneous parametric down-conversion* (SPDC) in which a single photon of energy E is converted into two indistinguishable photons of energy $E/2$.
- **Optical fibers and connectors**: These are used to guide photons between optical elements and to minimize the introduction of noise such as light from other sources.
- **Beam splitters**: Used to direct photons into different paths or to mix different photon streams coherently. The manipulation of photon states at a beam splitter is modeled as a unitary transformation.
- **Variable delay line**: Used to vary photon path length and control photon distinguishability.
- **Photon detectors**: Detectors such as avalanche photodiodes (APDs) or superconducting nanowire single-photon detectors (SNSPDs) are used. These are capable of detecting individual photons with high efficiency and timing resolution.
- **Coincidence counter (CC)**: This measures the time interval between the arrival of detected photons.
- **Data acquisition system**: To record the counts and analyze the statistics of the detected photons.

To demonstrate the Mandel effect, the components are configured as shown in Fig. 17. The experimental setup is optimized to minimize losses and maximize the indistinguishability of the photons, which is needed to observe the Mandel dip.

In this experiment, SPDC is used to generate indistinguishable photon pairs. In SPDC, single wavelength lasing light directed into a crystal is absorbed and re-emitted as two photons with double the wavelength of the original pump beam. For example, a blue pump photon at 405 nm wavelength can spontaneously convert into two red photons (ideally, each photon has exactly double the lasing wavelength) in a crystal, demonstrating energy down-conversion.

The efficiency of SPDC is low; typically, out of 100 billion pump photons, only one or two are down-converted. This inefficiency is advantageous for creating controlled, low numbers of photons. The photons are produced

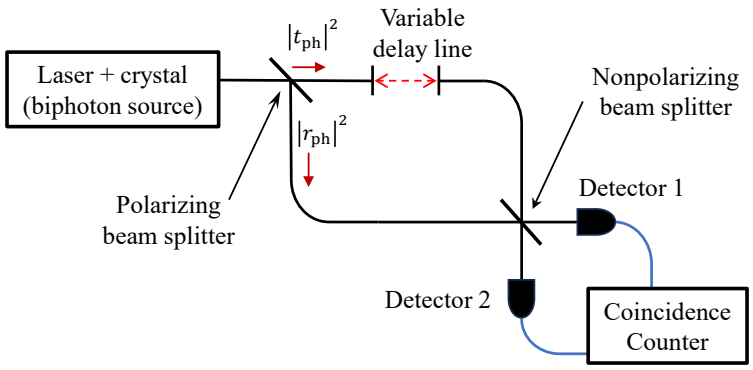


Figure 17: Block diagram of the components used to demonstrate the Mandel effect. Labelled experimental setup shown in Fig. 18

in pairs, allowing, for example, one photon to be detected to confirm the presence of its pair, a process known as *heralding*.

Several factors affect the efficiency of SPDC, including the pump beam's intensity, wavelength, and polarization, as well as the crystal's cut. Our setup uses a Type-II SPDC configuration in a periodically-poled Potassium Titanyl Phosphate (PPKTP) crystal, which converts a blue pump photon into an H-polarized red signal photon and a V-polarized red idler photon. A dichroic mirror ensures only the red photons are collected.

The system also includes a heater to adjust the crystal's temperature T_{crystal}, altering the wavelength of each emitted photon. We aim to produce at least 10,000 photon pairs per second per milliwatt of pump power. A fiber-optic polarizing beam splitter separates the photons based on their polarization, facilitating further manipulation and experimentation. If the crystal temperature is set correctly such that the down-converted photons have precisely the same wavelength, and so are degenerate, then the biphotons are indistinguishable.

As shown in Figs. 17 and 18, one output of the polarizing beam splitter is attached to a fiber-optic wave-guide that connects directly to a 50:50 non-polarizing beam splitter. The other output of the polarizing beam splitter leads to a variable delay line. This variable delay line can be used to change the photon path length between output and input. The variable delay line is controlled using a stepper motor, which is coded to move between respectively specified start and stop positions x_{start} and x_{stop} with increment Δx.

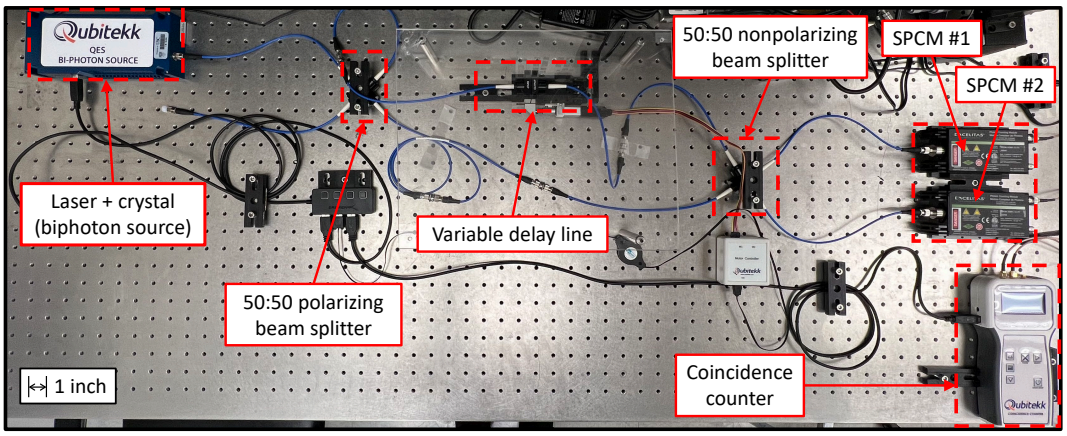


Figure 18: Experimental setup used to demonstrate the Mandel effect.

Each photon may either transmit through or reflect off the second 50:50 beam splitter, which is *non-polarizing*, and exit from the output ports as described in Fig. 15 with detection probabilities listed in Table 2. Two single-photon counting detector modules (SPCMs) are individually connected to the output ports of the non-polarizing beam splitter. These detectors produce an output signal when a photon is detected at its input port. The SPCMs are silicon-based avalanche photodiodes that detect the photon. Both SPCMs are connected to a coincidence counter. When both SPCMs detect a photon within a coincidence time window

t_{window}, it is considered a *coincidence event*. When this occurs, the coincidence counter will add a single coincidence count to the register. The counter will collect coincidences within a defined time t_{dwell} which, along with t_{window}, is a parameter that can be set when running the experiment.

As the stepper motor varies the length of one path to the detector, there will be a position for which the difference in the time it takes the photon to travel each path to the detector is minimized. In this scenario, the difference in arrival time of each photon path to the detectors will be minimized, and the photons are maximally indistinguishable. It is this indistinguishability that results in the detection probabilities, as described by Eqns. (100)-(102). There is a reduction in coincidence counts since, after accounting for noise, the biphotons are both *only* detected at a single detector. This is demonstrated through the experimental results shown in Fig. 19, which are the average of running 100 experiments for the same set of parameters. The visibility of the Mandel dip may be quantified by either computing the peak-to-valley ratio (PVR) or a normalized visibility measure $V_{\text{HOM}} = (\langle C \rangle_{\text{classical}} - \min(C))/\langle C \rangle_{\text{classical}}$, where $\langle C \rangle_{\text{classical}}$ is the average classical baseline count when the photons are distinguishable and $\min(C)$ is the dip valley.

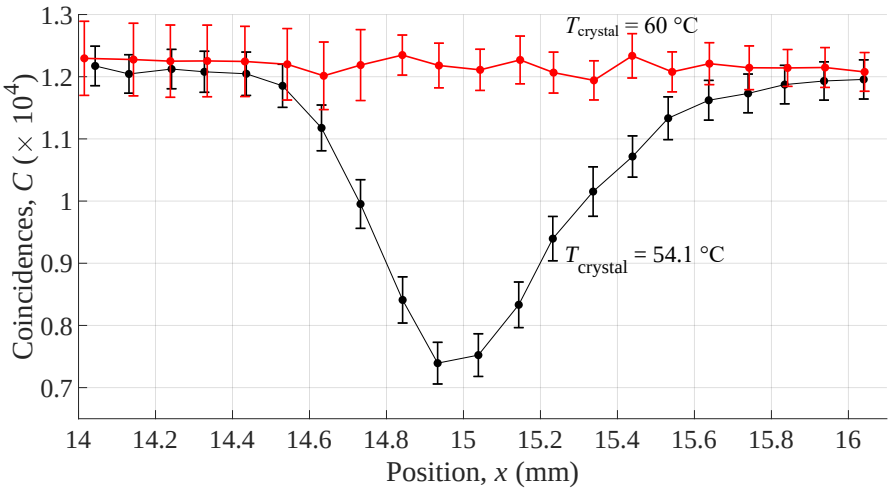


Figure 19: Measured baseline coincidence counts resulting from *distinguishable* photons when $T_{\text{crystal}} = 60$ °C (red curve) and measured Mandel dip resulting from identical, *indistinguishable* photons when $T_{\text{crystal}} = 54.1$ °C (black curve). Red dots are the mean values of 100 experimental runs using distinguishable photons ($T_{\text{crystal}} = 60$ °C), while black dots are the mean values of 100 experimental runs using identical, indistinguishable photons ($T_{\text{crystal}} = 54.1$ °C), with the corresponding standard deviations respectively shown as red and black vertical error bars. Parameters used in this experiment are laser current $I_{\text{laser}} = 109$ mA, $t_{\text{window}} = 5$ ns, $t_{\text{dwell}} = 1$ s, $x_{\text{start}} = 14$ mm, $x_{\text{stop}} = 16$ mm, $\Delta x = 0.1$ mm, resulting in a Mandel dip with PVR = 1.64 and $V_{\text{HOM}} = 0.39$.

The temperature of the crystal involved in the SPDC process plays an important role in the production of identical, indistinguishable photons. SPDC requires a specific phase-matching condition where the refractive indices of the pump photon, signal photon, and idler photon are matched. Since the crystal's refractive index is generally temperature-dependent, changes in crystal temperature can shift the phase-matching requirement, resulting in non-identical, and therefore distinguishable, photons.

In addition, temperature affects the crystal's dispersion properties, and temperature variation can cause changes in the signal and idler photon spectra. Producing indistinguishable photons requires high spectral overlap with narrow spectral bandwidth. Finally, the sensitivity of photon group velocity to temperature in the crystal impacts the temporal overlap of photon pairs, which is critical for ensuring simultaneous arrival at the detectors when the paths are identical.

A change in temperature can introduce timing mismatches between the down-converted photons, making the photon pairs distinguishable. Generating identical, indistinguishable photons requires tight crystal temperature control with precision temperature stabilization to within fractions of a degree Celsius. As shown in Fig. 19, when the crystal temperature T_{crystal} is increased to 60 °C, the photon pairs become distinguishable, the Mandel dip disappears, and the classical result is measured.

8.4 Transmission of n indistinguishable photons at a beam splitter

The quantum amplitude of integer n_3 and n_4 indistinguishable photons appearing at the output ports of the beam splitter is

$$|n_1, n_2, n_3, n_4\rangle = (-1)^{n_1} \left(\frac{1}{2}\right)^{\frac{n_{\text{tot}}}{2}} \sum_k (-1)^k \sqrt{\binom{n_1}{k}\binom{n_{\text{tot}}-n_1}{n_3-k}\binom{n_3}{k}\binom{n_{\text{tot}}-n_3}{n_1-k}} \tag{109}$$

where because the total number of particles $n_{\text{tot}} = n_1 + n_2$ is conserved $n_4 = n_1 + n_2 - n_3$. In this expression the number of ways of choosing k indistinguishable photons from a set of n_{tot} indistinguishable photons is given by the binomial coefficient

$$\binom{n_{\text{tot}}}{k} = \frac{n_{\text{tot}}!}{k!(n_{\text{tot}}-k)!} \tag{110}$$

If k is negative or greater than n_{tot} the binomial coefficient is set to zero. The terms $(-1)^{n_1}$ and $(-1)^k$ are due to the relative phase difference between a transmitted or reflected photon and it is this that gives rise to strong quantum interference effects. When using Eqn. (109) the probability of detecting photons at the output ports of the beam splitter is

$$P_{\text{out}} = ||n_1, n_2, n_3, n_4\rangle|^2 \tag{111}$$

8.4.1 Transmission of $n_{\text{tot}} = 8$ indistinguishable photons at a beam splitter

Figure 20 shows the calculated probability of photon output from an ideal, lossless, symmetric, 50:50 beam splitter with the input of integer n_1 photons at input port 1 and n_2 photons at input port 2 for a total of $n_{\text{tot}} = 8$ indistinguishable photons in the system, and Table 3 gives probability of n_3 detected photons at output port 3 for the cases when $n_1 = 8$ and $n_1 = 4$ at input port 1. Note the zero values for odd-integer n_3 and the rational numbers with denominator $2^{n_{\text{tot}}}$ for the non-zero probability values.

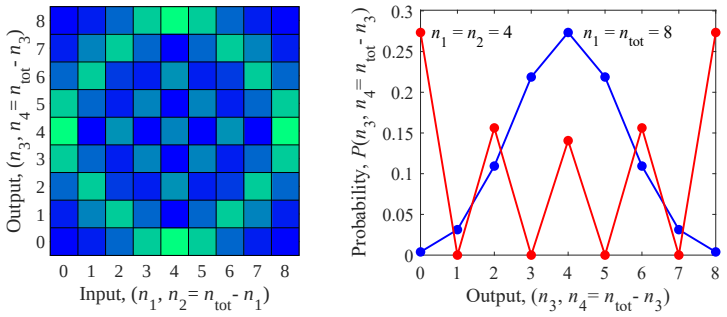


Figure 20: (a) Probability of photon output from a lossless symmetric 50:50 beam splitter showing input of n_1 photons at input port 1 and n_2 photons at input port 2 when there is a total of $n_{\text{tot}} = 8$ indistinguishable photons in the system. (b) Dots shows output probability into output port 3 for the case when the input port 1 contains $n_1 = n_{\text{tot}} = 8$ and when $n_1 = n_2 = n_{\text{tot}}/2 = 4$. Lines connecting dots are to guide the eye.

The blue dots in Figure 20(b) are port 3 output probability for the case when the input port 1 contains $n_1 = n_{\text{tot}} = 8$ indistinguishable photons. The probability has an approximately normal distribution centered at $n_{\text{tot}}/2 = 4$. The red dots show port 3 output probability for the case when the input port 1 contains $n_1 = n_2 = n_{\text{tot}}/2 = 4$ indistinguishable photons. In this situation, symmetry dictates that the probability of an odd number of indistinguishable photons at an output port is zero. The system behavior is closest to expectations of a continuous unmodulated classical electromagnetic wave when photons are only present at one input port. The system behavior is most non-classical when there are equal numbers of photons at each input port of the beam splitter. Increasing the value of n_{tot} can be used to explore the transition between the most non-classical behavior and behavior that seems closest to classical expectations. As will be illustrated

Table 3: Photon detection probability after a 50:50 beam splitter with $n_{\rm tot} = 8$ when $n_1 = 8$ and $n_1 = 4$.

$n_{\rm tot} = 8$	$n_1 = 8$	$n_1 = 4$
$n_3 = 0$	$\frac{1}{256}$	$\frac{70}{256}$
$n_3 = 1$	$\frac{8}{256}$	0
$n_3 = 2$	$\frac{28}{256}$	$\frac{40}{256}$
$n_3 = 3$	$\frac{56}{256}$	0
$n_3 = 4$	$\frac{70}{256}$	$\frac{36}{256}$
$n_3 = 5$	$\frac{56}{256}$	0
$n_3 = 6$	$\frac{28}{256}$	$\frac{40}{256}$
$n_3 = 7$	$\frac{8}{256}$	0
$n_3 = 8$	$\frac{1}{256}$	$\frac{70}{256}$

next near-classical results for a continuous unmodulated electromagnetic wave are retrieved when either $n_1 = n_{\rm tot}$ or $n_2 = n_{\rm tot}$ in the limit $n_{\rm tot} \to \infty$.

8.4.2 Transmission of $n_{\rm tot} = 64$ indistinguishable photons at a beam splitter

Figure 21 shows the results of calculating transmission of $n_{\rm tot} = 64$ identical indistinguishable photons at an ideal, lossless, symmetric, 50:50 beam splitter. The blue curve in Figure 21(b) is port 3 detected output probability for the case when the input port 1 contains $n_1 = n_{\rm tot}$ indistinguishable photons. The probability has an approximately normal distribution centered at $n_{\rm tot}/2$ and full-width-half-maximum (FWHM) slightly greater than $8 = \sqrt{64}$. In the limit when $n_{\rm tot} \to \infty$ (the large particle number thermodynamic limit) and $n_1 = n_{\rm tot}$ the detected photon number output probability distribution exhibits the normal (classical) result $FWHM \to \sqrt{n_{\rm tot}}$. Extrema of the red curve in Figure 21(b) is port 3 detected output probability showing the most non-classical behavior. This occurs when $n_1 = n_2 = \frac{n_{\rm tot}}{2} = 32$.

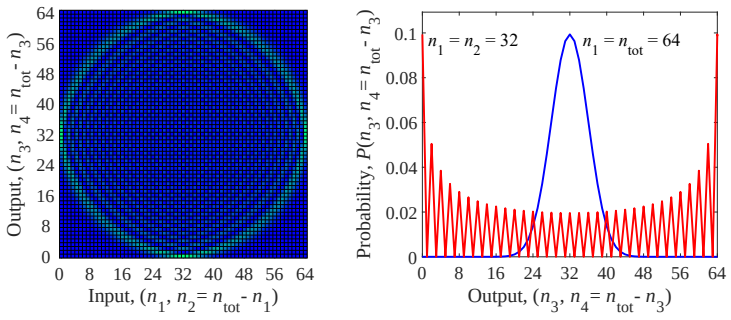


Figure 21: (a) Probability of detected photon output from an ideal, lossless, symmetric, 50:50 beam splitter showing input of n_1 photons at input port 1 and n_2 photons at input port 2 when there is a total of $n_{\rm tot} = 64$ indistinguishable photons in the system. (b) Port 3 detected output probability for the case when the input port 1 contains $n_1 = n_{\rm tot}$ and when $n_1 = n_2 = n_{\rm tot}/2 = 32$. Lines connecting probability values on the vertical axis for integer values of n_3 on the horizontal axis are to guide the eye.

Figure 22 is a three-dimensional plot of the probability of photon output from an ideal, lossless, symmetric, 50:50 beam splitter when there is a total of $n_{\rm tot} = 64$ indistinguishable photons in the system. The appearance of detected quantum interference effects is limited to a "quantum cauldron" of radius $\sqrt{n_{\rm tot}}/2$ [21]. The closest to classical behavior (an approximately normal probability distribution) occurs at the boundary of the domain

[21] F. Lalöe and W. J. Mullin, *Found. Phys.* **42**, 53 (2012).

when $n_1 = n_{\rm tot}$ and $n_2 = 0$ or $n_1 = 0$ and $n_2 = n_{\rm tot}$. The quantum cauldron illustrated in Figure 22 is one way to depict the transition from closest to classical behavior to most non-classical behavior in the system.

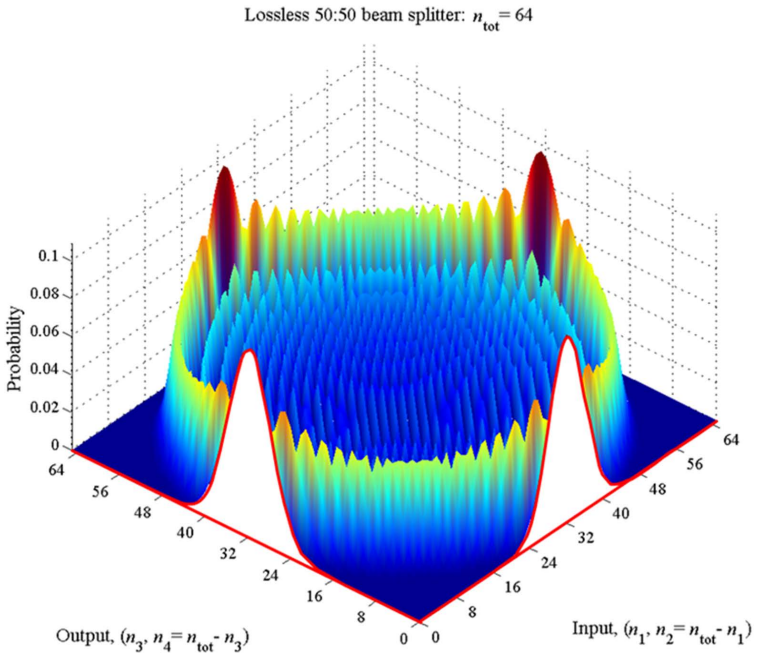


Figure 22: Three-dimensional plot of the probability of detected photon output from an ideal, lossless, symmetric, 50:50 beam splitter when there is a total of $n_{\rm tot} = 64$ indistinguishable photons in the system. There is a "quantum cauldron" of interference inside a radius $\sqrt{n_{\rm tot}}/2$. The closest to classical behavior occurs at the boundary of the domain when $n_1 = n_{\rm tot}$ and $n_2 = 0$ or when $n_1 = 0$ and $n_2 = n_{\rm tot}$.

In the preceding, photon number is preserved and the interaction of the optical field with the beam splitter is ideal. The symmetry associated with indistinguishable particles results in detected quantum amplitude interference between different inferred paths through the system. It should be noted that fundamental to the Copenhagen interpretation of quantum mechanics, the photon paths taken can only be inferred *after* detection. As illustrated by the Mandel effect, because of strong quantum correlations, the probability of detecting a fixed number of discrete photons at an output port can be dramatically different from the expectations of a continuous unmodulated classical electromagnetic wave interacting with the system.

Homework Problems: Beam splitter numerical error

8.5 Quantum interference and distinguishability

The Mandel effect is an example of quantum interference between two indistinguishable particles. *Distinguishability* between two photon particles is achieved if they have different polarization, different spectral frequency content, different phase or a time delay between pulses. To quantify how quantum interference is suppressed as the distinguishability of the photons increases, it is convenient to describe interaction with a beam splitter using boson creation and annihilation operators.

If there is a single indistinguishable photon at each input port 1 and 2 of the beam splitter then $|1\rangle_1|1\rangle_2 = b_1^\dagger b_2^\dagger |0\rangle_1|0\rangle_2$ and as illustrated in Figure 23 there are four different paths the two photons can take to the output ports 3 and 4. The unitary transform that connects input port states to output port states is given by the unitary matrix

$$\hat{U}_B = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathrm{i} & -1 \\ -1 & \mathrm{i} \end{bmatrix} = \begin{bmatrix} t_{\rm ph} & r_{\rm ph} \\ r_{\rm ph} & t_{\rm ph} \end{bmatrix} \tag{112}$$

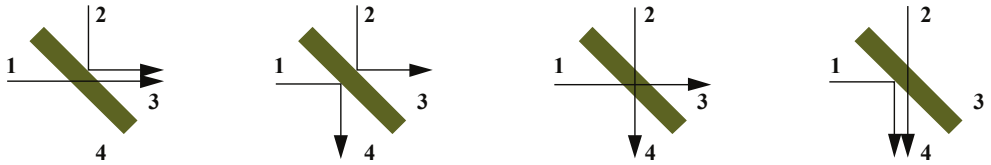
which has an inverse such that $\hat{U}_B^\dagger = \hat{U}_B^{-1}$ and the creation operator of the superposition output states are such that $b_1^\dagger \rightarrow -\left(\mathrm{i}b_3^\dagger + b_4^\dagger\right)/\sqrt{2}$ and $b_2^\dagger \rightarrow -\left(b_3^\dagger + \mathrm{i}b_4^\dagger\right)/\sqrt{2}$. The output state is

$$\frac{1}{2}(\mathrm{i}b_3^\dagger + b_4^\dagger)(b_3^\dagger + \mathrm{i}b_4^\dagger)|0\rangle_3|0\rangle_4 = \frac{1}{2}(\mathrm{i}b_3^\dagger b_3^\dagger - b_3^\dagger b_4^\dagger + b_3^\dagger b_4^\dagger + \mathrm{i}b_4^\dagger b_4^\dagger)|0\rangle_3|0\rangle_4 = \frac{\mathrm{i}}{2}(b_3^\dagger b_3^\dagger + b_4^\dagger b_4^\dagger)|0\rangle_3|0\rangle_4 \tag{113}$$

in which the terms $-b_3^\dagger b_4^\dagger + b_3^\dagger b_4^\dagger$ creating a single photon in each output port exactly cancel. Hence when a single indistinguishable photon is present at each input port 1 and 2 of the beam splitter then the only possible output is either two photons at output port 3 or two photons at output port 4. The output state is

$$\frac{\mathrm{i}}{2}(b_3^\dagger b_3^\dagger + b_4^\dagger b_4^\dagger)|0\rangle_3|0\rangle_4 = \frac{\mathrm{i}}{2}(|2\rangle_3|0\rangle_4 + |0\rangle_3|2\rangle_4) \tag{114}$$

where use is made of the fact that $b_3^\dagger b_3^\dagger|0\rangle_3|0\rangle_4 = b_3^\dagger|1\rangle_3|0\rangle_4 = \sqrt{2}|2\rangle_3|0\rangle_4$ and $b_4^\dagger b_4^\dagger|0\rangle_3|0\rangle_4 = b_4^\dagger|0\rangle_3|1\rangle_4 = \sqrt{2}|0\rangle_3|2\rangle_4$. The probability of detecting two photons at either output port is exactly one-half and the output port at which the two photons are detected is a fundamentally random quantum mechanical process.



$\frac{\mathrm{i}}{2}(\hat{b}_3^\dagger \hat{b}_3^\dagger)|0,0\rangle_{34}$ $\quad$ $\frac{1}{2}(\hat{b}_3^\dagger \hat{b}_4^\dagger)|0,0\rangle_{34}$ $\quad$ $-\frac{1}{2}(\hat{b}_3^\dagger \hat{b}_4^\dagger)|0,0\rangle_{34}$ $\quad$ $\frac{\mathrm{i}}{2}(\hat{b}_4^\dagger \hat{b}_4^\dagger)|0,0\rangle_{34}$

Figure 23: Illustration showing four possible outputs from an ideal, lossless, 50:50 beam splitter.

It is possible to tune the distinguishability of identical photons by creating a time delay τ in arrival time at the detector. Introducing the time delay at port 2 changes the creation operator to $b_2^\dagger \mathrm{e}^{\mathrm{i}\omega\tau}$ and if the spectral amplitude of each photon pulse is a Gaussian centered at frequency ω_0 with standard deviation σ_0 such that $\phi_0(\omega) = \mathrm{e}^{-(\omega-\omega_0)^2/(2\sigma_0^2)}/\sqrt{2\pi\sigma_0}$ it can be shown that the coincidence probability measured by the detectors is [22]

$$\frac{1}{2} - \frac{1}{2}\mathrm{e}^{-\sigma_0^2\tau^2/2} \tag{115}$$

Hence, in this case the "Mandel dip" in detected photon coincidence counts measured as a function of delay is described using a Gaussian. A typical value of delay that characterizes the extent of the dip in coincidence counts is $\tau = 1$ ps corresponding to a photon propagating 300 μm in free space. The *wave-particle duality* of *indistinguishable* photons manifesting as wave-like interference and a measured particle-like "click" output from a detector is a purely quantum phenomenon with no classical counterpart.

Explore More: Classical analog of the "Mandel dip"

[22] C. Drago and A. M. Brańczyk, *Can. J. Phys.* **102**, 411 (2024).

9 Explore More

9.1 Sets of real numbers [return to section]

Different types of real numbers are classified into categories referred to as *sets*.

The set of natural numbers ($\mathbb{N}$) consists of all positive numbers starting from 1, extending indefinitely in an increasing sequence. This set includes numbers like 1, 2, 3, etc. typically used for counting and ordering, and under some contexts in mathematics and computer science, can also include 0 as a basis for arithmetic operations. In such cases, the set of natural numbers may conventionally be labelled either $\mathbb{N}_0$ or $\mathbb{W}$ which represents whole positive semi-definite numbers. There is no general consensus regarding whether zero should be included in the set of natural numbers, however.

The set of integers ($\mathbb{Z}$)[a]comprises all whole numbers, including positive numbers, negative numbers, and zero. Integers are used for counting, ordering, and various arithmetic operations that include subtraction and addition across positive and negative values. Examples of integers include -2, -1, 0, 1, 2, etc.

The set of rational numbers ($\mathbb{Q}$) consists of all numbers that can be expressed as the ratio of two integers, where the numerator is an integer and the denominator is a non-zero integer, such as integers, fractions and numbers that have a repeating or terminating decimal representation. Examples of rational numbers include $22/7$, $-1/12$, and $1/137$.

The set of irrational numbers ($\mathbb{I}$) includes all real numbers that cannot be expressed as a ratio of two integers. Irrational numbers have non-repeating, non-terminating decimal expansions. Common examples of irrational numbers include $\sqrt{2}$, π, and e.

In summary, the sets of real numbers and their corresponding symbols is given below:

- $\mathbb{N}$: Natural set (1, 2, 3, etc.); denoted as $\mathbb{N}_0$ if 0 is included.
- $\mathbb{Z}$: Integer set (-1, 0, 1, etc.); includes $\mathbb{N}$.
- $\mathbb{Q}$: Rational set (1/2, 1/3, 2/3, etc.); includes $\mathbb{Z}$.
- $\mathbb{I}$: Irrational set (e, π, $\sqrt{2}$, etc.).
- $\mathbb{R}$: Real set; includes all above sets ($\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{I}$).

[a]The symbol representing the set of integers, $\mathbb{Z}$, has been attributed to German mathematician David Hilbert and stands for "Zahlen", the German word for numbers.

9.2 Other generalized number systems [return to section]

Beyond complex numbers, other generalized number systems include quaternions, octonions, and p-adic numbers.

- **Quaternions** ($\mathbb{H}$)[a]: Quaternions extend complex numbers to four dimensions. A quaternion is expressed as $a + b\mathrm{i} + c\mathrm{j} + d\mathrm{k}$, where a, b, c, and d are real numbers, and i, j, and k are imaginary unit numbers satisfying the following relationships:
  1. $\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1$
  2. $\mathrm{ij} = \mathrm{k},\ \mathrm{ji} = -\mathrm{k}$
  3. $\mathrm{jk} = \mathrm{i},\ \mathrm{kj} = -\mathrm{i}$
  4. $\mathrm{ki} = \mathrm{j},\ \mathrm{ik} = -\mathrm{j}$

  Quaternions are *non-commutative*, meaning the order of multiplication matters. They are particularly useful in 3D computer graphics and physics to represent rotations.
- **Octonions** ($\mathbb{O}$)[b]: Octonions are an extension of quaternions to eight dimensions. An octonion is expressed as a linear combination of eight basis elements: $\{e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$, where e_0 is the scalar element and typically identified with the real unit so that $e_0 = 1$. Octonions follow *non-associative* multiplication rules, which means that how operations are grouped affects the result. They are used for the construction of more advanced mathematical objects and are sometimes used in theoretical high-energy particle physics.
- **p-adic numbers** ($\mathbb{Q}_p$): p-adic numbers are a system of number representation used primarily in number theory. They are defined with respect to a given prime number p and differ from real or complex numbers in that they measure distances based on divisibility by p. p-adic numbers are useful for solving equations in modular arithmetic and also see application in theoretical physics as well as in cryptography.

  These as well as other specialized number systems explore different properties and applications, often extending concepts found in more familiar number systems to new contexts or addressing problems that are intractable using conventional numbers. Mathematics continues to expand with new number systems and structures to solve emerging problems, leading to creative approaches across many different disciplines.

[a] Quaternions were first introduced by Irish mathematician William Rowan Hamilton in 1843.
[b] Octonions were independently discovered by John Graves in 1843 and Arthur Cayley in 1845.

9.3 Common functions with applications

[**return to section**]

Common examples of non-polynomial functions along with common applications in, for example, electrical engineering include:

- **Exponential functions**: The general form of an exponential is $f(x) = ae^{bx}$, where a and b are constants. If b is purely real, such functions describe exponential growth or decay, depending on whether b is respectively positive or negative. Applications in electrical engineering include modeling capacitor charging and discharging, describing growth and decay processes in filters and amplifiers, and signal processing for analysis of exponential signals.
- **Logarithmic functions**: The general form of a logarithmic function is $f(x) = a\log_b(x)$, where a is a constant and b is the logarithmic base. Applications include signal processing for dB calculations in amplifiers and attenuation, and analysis of nonlinear distortion and compression effects.
- **Sinusoidal functions**: These cyclic functions include $f(x) = a\sin(bx+c)$ and $f(x) = a\cos(bx+c)$, where a, b, and c are constants. Applications include representing AC signals, oscillators, and waveforms in signal processing, and modeling electromagnetic waves and impedance in AC circuits.
- **Hyperbolic functions**: These typically include hyperbolic sine and cosine, with the general forms respectively given as $f(x) = a\sinh(bx)$ and $f(x) = a\cosh(bx)$, where a and b are constants. Applications include describing transient responses of transmission lines and representing charge distributions in semiconductor junctions.
- **Power functions**: The general form of a power function is $f(x) = ax^b$, where a and b are constants. Applications include modeling nonlinear amplifier gains and describing signal scaling as well as voltage-power relationships. These functions are useful for understanding scaling laws in physical systems, such as how resistance varies with conductor length and cross-sectional area, thereby aiding in the design of efficient electrical components.
- **Gaussian functions**: These are functions with the general form $f(x) = ae^{-(x-b)^2/(2c^2)}$ and are commonly used in statistical analysis, such as analyzing the distribution of various measurements of many real-world phenomena. Applications in electrical engineering include designing filters and analyzing signal noise, as well as image processing and pattern recognition algorithms.

9.4 Examples of *one-to-many* functions

[return to section]

In complex analysis, the complex logarithm $\log(z)$ can yield multiple values for a single input because it has an infinite number of possible angles in the complex plane. These distinct values exist within *branches* corresponding to a continuous and single-valued segment of the multi-valued complex logarithm. Thus, unless a particular branch is specified, the complex logarithm cannot be considered a function under the standard definition. By selecting a specific branch, we define a function in a way that it is continuous and single-valued, making it practical for analysis and computations and ensuring correct outputs with respect to complex differentiation and integration. In broader terms of set theory, one might consider a general relation that can map an input to multiple outputs. Although these are not functions, they are helpful in some mathematical contexts, including database theory and computer science, where an input might correspond to multiple outputs.

In physics, an initial state might lead to multiple states, particularly in quantum mechanics and statistical mechanics. For example, the outcome of a quantum measurement can be fundamentally probabilistic, suggesting a one-to-many relationship from initial state to potential outcomes. However, these are not functions in the mathematical sense but rather stochastic or probabilistic processes.

Another example is the complex band structure description of a bulk crystal that describes the energy eigenvalues E of electrons and their Bloch wave vectors (crystal momenta $\hbar k$) in a periodic lattice. Complex band structure has applications in electronic device design.[a] The *reduced* Brillouin zone is a convenient method for visualizing the band structure. In this case, all real wave vectors are folded back into the first Brillouin zone due to the periodicity of the reciprocal lattice. Hence, as shown in Fig. 9.4.1, a given wave vector k can correspond to multiple energy values E.

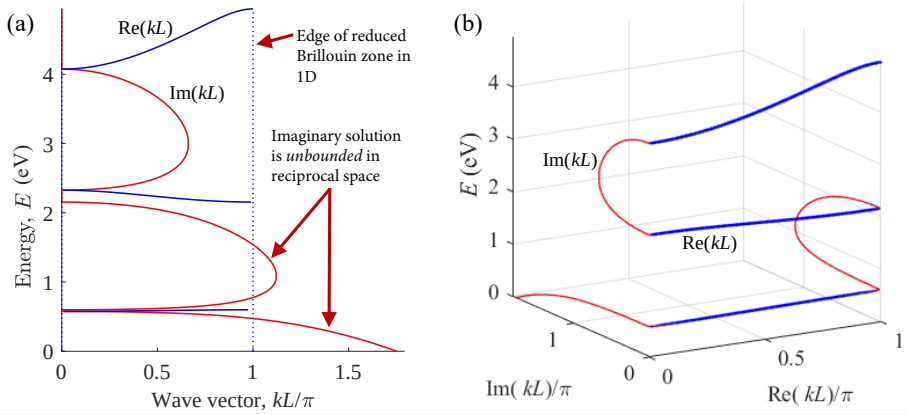


Fig. 9.4.1. (a) Complex band structure of the Kronig-Penney model[b] depicted in the reduced Brillouin zone scheme with total unit cell spacing $L = 1$ nm, potential barrier of thickness $L_B = 0.4$ nm and energy $V_B = 4$ eV, and potential well of thickness $L_W = 0.6$ nm and energy $V_W = 0$ eV. (b) Same as (a) but energy plotted vs. the complex wave vector plane to illustrate complex band structure in a 3D plot.

In some cases, what might appear as a one-to-many function can be resolved by considering more variables or parameters. For example, a parametric equation in physics might describe a trajectory or a set of states over time, with time as an implicit parameter that resolves the apparent one-to-many nature.

To summarize, while a one-to-many function does not exist in the strict mathematical sense, various constructs like multi-valued functions, relations, and parametric equations exhibit one-to-many characteristics under certain conditions or interpretations in the fields of mathematics and physics.

[a] W. Unglaub and A. F. J. Levi. *Physics Open* **17**, 100164 (2023) and W. Unglaub and A. F. J. Levi. *Physica E* **165**, 116067 (2025).

[b] For more details on the Kronig-Penney model and electron dispersion in periodic potentials, see section 3.6 in A. F. J. Levi, *Applied Quantum Mechanics*, 3rd ed. Cambridge: Cambridge University Press, 2023.

9.5 Limits and asymptotic behavior [return to section]

The concepts of asymptotes and limits are important for analyzing the behavior of functions as they approach specific points or infinity. An asymptote is a line that a function approaches but never reaches as the independent variable approaches a certan value or infinity. Asymptotic lines can be vertical, horizontal, or have a finite non-zero slope. Vertical asymptotes occur when the function approaches infinity or negative infinity as the independent variable x approaches a specific value. For example, the function $f(x) = 1/x$ has a vertical asymptote at $x = 0$. Horizontal asymptotes describe the behavior of a function as the independent variable x approaches $\pm\infty$. For example, the function $f(x) = 2x/(x+1)$ approaches the horizontal line $y = 2$ as x approaches $\pm\infty$. Finally, *oblique* asymptotes refer to the case when the function $f(x)$ approaches a line with finite non-zero slope as $x \to \pm\infty$. This typically occurs when the degree of the numerator is one more than the degree of the denominator. An example of this would be the function $f(x) = (x^2 - 1)/x$, which approaches the line $y = x$ as $x \to \pm\infty$.

Related to asymptotes is the concept of *limits*, which are critical for understanding derivatives and integrals of functions. The limit of a function describes the value that the function approaches as the input or independent variable approaches some value. Generally, there are three types of limits: finite limits, infinite limits, and limits at infinity. Finite limits are characterized by the limit of $f(x)$ equaling L as x approaches some value a. This is denoted as

$$\lim_{x \to a} f(x) = L, \tag{116}$$

provided $f(x)$ can be made arbitrarily close to L by making x sufficiently close to a. Infinite limits occur when $f(x)$ increases or decreases without bound as $x \to a$, resulting in $f(x) \to \pm\infty$, where the sign depends on which direction the function diverges. Finally, limits at infinity describe a third type of limit in which the behavior of $f(x)$ is characterized as x approaches $\pm\infty$. This can result in $f(x)$ taking on either a finite value or diverging to $\pm\infty$.

Example: If we consider the rational function $f(x)$ constructed by dividing two polynomials

$$f(x) = \frac{3x^2 + 7x + 5}{2x^2 + 2}$$

and take the limit as x approaches ∞, we have the limit

$$\lim_{x \to \infty} \frac{3x^2 + 7x + 5}{2x^2 + 2} = \frac{3}{2},$$

which demonstrates a horizontal asymptote at $y = 3/2$.

Problem 1: Evaluate the following limit:

$$\lim_{x \to \infty} \frac{2x^2 - 3x + 4}{x^3 + x + 1}.$$

Problem 2: Calculate the limit:

$$\lim_{x \to 3} \frac{x^2 - 9}{x - 3}.$$

Problem 3: Identify and describe the asymptotes for the function:

$$f(x) = \frac{2x}{x - 1}.$$

9.6 The gradient [return to section]

The *gradient* of a function represents the generalization of the derivative to functions of multiple variables, providing a *vector* of partial derivatives for each variable. While formally introduced in a later section, a vector may be viewed as an array of N numbers, each corresponding to some amount along that number's direction in an N-dimensional space, in which each direction is *orthogonal*. The gradient, therefore, points in the direction of the steepest ascent of the function at a given point and its magnitude gives the rate of increase in that direction.

For a function $f(\mathbf{x}) = f(x_1, x_1, \ldots, x_N)$ of N variables, the gradient of f is denoted as ∇f and defined as:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \cdots & \frac{\partial f}{\partial x_N} \end{bmatrix}, \tag{117}$$

where $\partial f / \partial x_i$ is the partial derivative of f with respect to the ith variable, indicating how f changes as x_i changes while keeping the other variables constant. At any point described by a set of values x_i, the gradient vector gives the direction of the steepest ascent from that point, and the magnitude, or length, or the gradient vector represents the rate of increase of the function in the direction of the steepest ascent.

Example: Using the same function from Example 1, we can compute the partial derivatives for each variable (x and y):

$$\begin{aligned} \frac{\partial f}{\partial x} &= \frac{\partial}{\partial x}\left(x^2 + 3xy - y^2\right) = 2x + 3y \\ \frac{\partial f}{\partial y} &= \frac{\partial}{\partial y}\left(x^2 + 3xy - y^2\right) = 3x - 2y \\ \therefore \nabla f &= \begin{bmatrix} 2x + 3y & 3x - 2y \end{bmatrix} \end{aligned}$$

At any point (x, y), the gradient $\nabla f(x, y)$ gives the direction and rate of fastest increase of the function f. For example, at the point $(1, 1)$, $\nabla f(1, 1) = (2 \cdot 1 + 3 \cdot 1, 3 \cdot 1 - 2 \cdot 1) = (5, 1)$. This indicates that starting from $(1, 1)$, the function f increases most in the direction of vector $(5, 1)$.

Generally speaking, the gradient is a powerful tool that provides useful insights into the behavior of functions across different dimensions and is extensively used in gradient-based optimization methods and for designing gradient descent algorithms in machine learning.

9.7 Taylor series expansion [return to section]

The Taylor series expansion is a mathematical method for approximating functions using factorials and the sum of its derivatives at a single point. It is particularly useful for approximating complex functions with a series of polynomial terms.[a]

The Taylor series of a function $f(x)$ around a point a is given by

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots \quad (118)$$

Here, $f'(a)$, $f''(a)$, and so on are the first, second, and higher derivatives of f evaluated at the point a, and $n!$ denotes the factorial of integer n. This expansion is particularly powerful in engineering for approximating functions where direct computation is complex or infeasible. A few common examples of expanding nonlinear functions (around $a = 0$) include the following:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots \quad (119)$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!}x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \cdots \quad (120)$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!}x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \cdots \quad (121)$$

Example: Using the definition of a Taylor series expansion provided by Eq. (118), derive a third-order approximation of $\ln(1+x)$ using a Taylor series expansion around $x = 0$. What are the first three non-zero terms of the series approximation?

We identify $a = 0$, so for $f(x) = \ln(1+x)$ we have:

$$\begin{aligned} f(x) &\approx \frac{f(0)}{0!}\cdot(x-0)^0 + \frac{f'(0)}{1!}\cdot(x-0)^1 + \frac{f''(0)}{2!}(x-0)^2 + \frac{f'''(0)}{3!}(x-0)^3 \\ &= \frac{\ln(1+x)|_{x=0}}{1}\cdot 1 + \frac{(1+x)^{-1}\big|_{x=0}}{1}\cdot x + \frac{(-1)\,(1+x)^{-2}\big|_{x=0}}{2}\cdot x^2 + \frac{(-2)(-1)\,(1+x)^{-3}\big|_{x=0}}{6}\cdot x^3 \\ &= \ln(1) + 1\cdot x - \frac{1}{2}\cdot x^2 + \frac{2}{6}\cdot x^3 = 0 + x - \frac{x^2}{2} + \frac{x^3}{3}. \end{aligned}$$

Thus, the first three non-zero terms of the Taylor series approximation to $\ln(1+x)$ are x, $-x^2/2$, and $x^3/3$.

Problem 1: Use the Taylor series expansion of $f(x) = \mathrm{e}^x$ to approximate e using the first four terms. How many terms are required to achieve a relative error less than 10^{-6}?
Problem 2: Find the first three non-zero terms of the Taylor series expansion of $f(x) = \mathrm{e}^x \ln(2x+1)$ around $x = 0$. Using this finite series as an approximation function for $f(x)$, what is the relative error between the two functions when $x = -0.4$?
Problem 3: Using the Taylor series expansions for $\sin(x)$ and e^x, determine the coefficients a_i for a 4th-order approximation of $f(x)$, where $i \in \{0, 1, \ldots, 4\}$:

$$f(x) = \frac{\sin(x)}{\mathrm{e}^x} \approx a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4. \quad (122)$$

[a] This method is named after the British mathematician Brook Taylor, who formalized the technique in 1715.

9.8 First-order differentiation of discrete functions [return to section]

Discrete differentiation appears frequently in digital signal processing and is necessary for approximating the derivative of signals with either uniform or non-uniform sampling. Unlike continuous differentiation which deals with functions that have an infinite number of points within any given range of the domain, discrete differentiation focuses on functions defined at discrete intervals or points. This is particularly relevant in digital systems where signals are sampled at specific times, for example. With respect to discrete systems, differentiation is not performed in the traditional calculus sense but is instead approximated through differences between successive samples.

In discrete differentiation, the derivative of a function at a point is approximated by the difference between its values at successive points. The simplest general form of discrete differentiation between the nth and $(n+1)$th samples is given by the forward difference $\Delta y[n]$ divided by the sampling interval $\Delta x[n]$, defined as:

$$y'[n] = \frac{\Delta y[n]}{\Delta x[n]} = \frac{y[n+1] - y[n]}{x[n+1] - x[n]}, \tag{123}$$

where $y[n]$ represents the value of the function or signal at the nth interval, and $x[n]$ represents the nth sample point. This approximation assumes that the change between consecutive samples is indicative of the slope at the point. If the signal is uniformly-sampled in x, then $\Delta x[1] = \Delta x[2] = \ldots = \Delta x[N] = \Delta x$ for N samples. Thus, Eq.(123) becomes:

$$y'[n] = \frac{\Delta y[n]}{\Delta x} = \frac{y[n+1] - y[n]}{\Delta x}, \tag{124}$$

Example: Consider the discrete sequence $y = [2, 3, 5, 7, 11]$ and compute the first derivative using the forward difference method, assuming uniform sampling $\Delta x = 2$.

We can straightforwardly compute the forward difference sequence as:

$$\begin{aligned}
\Delta y[1] &= y[2] - y[1] = 3 - 2 = 1 \\
\Delta y[2] &= y[3] - y[2] = 5 - 3 = 2 \\
\Delta y[3] &= y[4] - y[3] = 7 - 5 = 2 \\
\Delta y[4] &= y[5] - y[4] = 11 - 7 = 4.
\end{aligned}$$

Thus, the numerical derivative array is

$$\frac{\Delta y}{\Delta x} = \frac{1}{2}\,[1, 2, 2, 4] = [0.5, 1, 1, 2]\,.$$

Note that computing the first order forward difference sequence results in $N - 1$ terms, since each value requires at least *two* consecutive sample points.

Problem 1: Given the sequence $y = [3, 3, 6, 9, 12]$, compute the first derivative using the forward difference method.

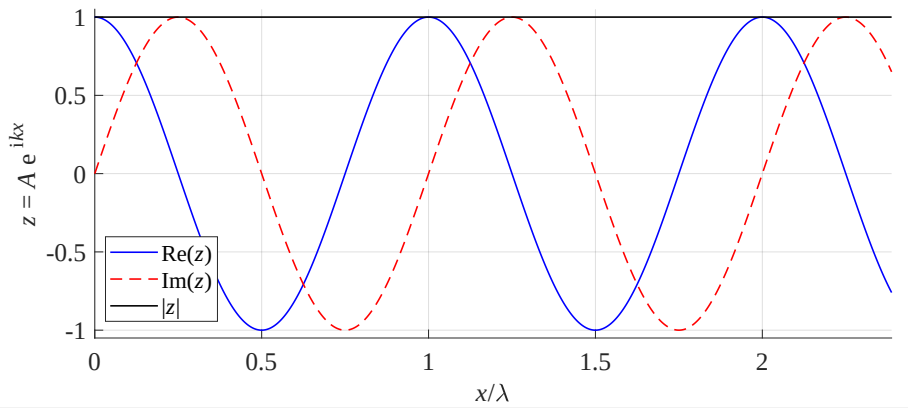
Problem 2: For the sequence $y = [2, 5, 10, 17, 26]$, compute the first derivative using the forward difference method. Assuming $\Delta x = 1$, what continuous function does the sequence trace out? Is the derivative of such a function consistent with the forward difference result?

Problem 3: Given the signal $y = [0, -1, -2, -1, 0, 1, 2, 1, 0]$, compute the discrete derivative and interpret the result.

9.9 Introduction to plane waves [return to section]

As an example of applying Euler's formula, we consider the concept of a plane wave. A plane wave in physics and engineering represents the propagation of a set of wavefronts (such as an electromagnetic or acoustic wave) which are infinite, parallel, and equidistant - meaning the wave has a well-defined wavelength λ.

This is visualized below for a plane wave $z(x) = Ae^{ikx}$ with amplitude $A = 1$ and wavelength $\lambda = 2\pi/k$, where k is the wave vector. The real part of the plane wave (solid blue curve) is given by $\mathrm{Re}(z) = A\cos(kx)$, and the imaginary part of the plane wave (dashed red curve) is given by $\mathrm{Im}(z) = A\sin(kx)$.



In the context of electromagnetism, a plane electromagnetic wave can be described using complex exponential functions and analyzed using Euler's formula. Consider a plane electromagnetic wave traveling in the $+x$ direction with an electric field represented as follows:

$$\mathbf{E}(x,t) = \mathbf{E}_0 e^{\mathrm{i}(kx-\omega t)}, \tag{125}$$

where $\mathbf{E}_0$ is the amplitude of the electric field (a constant vector), $k = 2\pi/\lambda$ is the wave number which is inversely proportional to the wavelength λ, $\omega = 2\pi/T$ is the angular frequency which is inversely proportional to the wave period T, x is the positon along the x-axis, and t is time.

Example: Calculate the real part of this field, assuming that the electric field oscillates in the y-direction only, and $\mathbf{E}_0 = E_0\hat{\mathbf{j}}$, where $\hat{\mathbf{j}}$ is the unit vector associated with the y-direction.

Given the wave

$$\mathbf{E}(x,t) = E_0 e^{\mathrm{i}(kx-\omega t)}\hat{\mathbf{j}}, \tag{126}$$

we use Euler's formula to expand it:

$$\mathbf{E}(x,t) = E_0\left(\cos(kx-\omega t) + \mathrm{i}\sin(kx-\omega t)\right)\hat{\mathbf{j}} \tag{127}$$

The real part of this electric field, which is physically meaningful in the context of measurable electric fields is

$$\mathrm{Re}\left(\mathbf{E}(x,t)\right) = E_0\cos(kx-\omega t)\hat{\mathbf{j}}. \tag{128}$$

This represents a sinusoidal wave in the y-direction, with a wavelength $\lambda = 2\pi/k$ and a period $T = 2\pi/\omega$, propagating along the x-direction.

In general, the use of complex functions to describe plane wave simplifies calculations and visualizations of wave phenomena, particularly by converting trigonometric problems into exponential ones, which are often easier to manipulate mathematically.

9.10 Complex differentiation [return to section]

In complex analysis, the Cauchy-Riemann equations are a set of partial differential equations that, along with certain continuity conditions, form a *criterion* for a function of a complex variable to be differentiable in the complex sense. Differentiability in this context is similar to the concept of differentiability for real functions but includes some additional constraints due to the nature of complex numbers.

Suppose $f(z)$ is a complex function expressed in therms of a complex variable $z = x + \mathrm{i}y$, where x and y are real numbers. If f can be decomposed into real and imaginary parts $f(z) = u(x, y) + \mathrm{i}v(x, y)$, where u and v are real-valued functions of two variables, then the function f is differentiable at a point in the complex plane if both the Cauchy-Riemann equations are satisfied *and* the partial derivatives comprising these equations are *continuous*. Using the shorthand notation for partial derivatives introduced in (23), the Cauchy-Riemann equations are

$$\begin{aligned} \partial_x u &= \partial_y v \\ \partial_y u &= -\partial_x v. \end{aligned} \tag{129}$$

We can combine these two equations and write a compact, *homogeneous* system of linear equations by defining a matrix $\mathbf{D}$ using the partial derivative operators, acting on the complex function vector $\mathbf{f}$:

$$\begin{bmatrix} \partial_x u - \partial_y v \\ \partial_y u + \partial_x v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \mathbf{Df} = \mathbf{0}. \tag{130}$$

Note the structural similarity between $\mathbf{D}$ and the rotation matrix $\mathbf{R}$ used to perform vector rotations in Euclidean space by an angle θ,

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{131}$$

In contrast, the Cauchy-Riemann equations convey that the local behavior of an analytic function in the complex plane corresponds to a transformation that can be understood as *both* a rotation and dilation, equivalent to multiplication by a complex number. For a complex function $f(z)$ to be differentiable about a complex value z, the derivative

$$f'(z) \equiv \lim_{\Delta z \to 0} \frac{f(z + \Delta z) - f(z)}{\Delta z} \tag{132}$$

must be independent of the direction from which the limit is taken as Δz approaches zero. Specifically, whether Δz approaches zero along the real axis, the imaginary axis, or any other direction, the resulting derivative must be the same for the derivative to exist.

Example: Verify whether the function $f(z) = x^2 - y^2 + \mathrm{i}2xy$ satisfies the Cauchy-Riemann equations and is therefore complex differentiable.

We begin by expressing f in terms of u and v:

$$\begin{aligned} u(x, y) &= x^2 - y^2 \\ v(x, y) &= 2xy. \end{aligned}$$

Next, we calculate the partial derivatives and check whether the Cauchy-Riemann equations are satisfied according to Eqn. (130):

$$\begin{bmatrix} \partial_x & -\partial_y \\ \partial_y & \partial_x \end{bmatrix} \cdot \begin{bmatrix} x^2 - y^2 \\ 2xy \end{bmatrix} = \begin{bmatrix} \partial_x(x^2 - y^2) - \partial_y(2xy) \\ \partial_y(x^2 - y^2) + \partial_x(2xy) \end{bmatrix} = \begin{bmatrix} 2x - 2x \\ -2y + 2y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \checkmark$$

Therefore, since the equations are satisfied *and* each partial derivative results in a continuous function, we can conclude that $f(z)$ is complex-differentiable.

9.11 Cross product [return to section]

With the concept of a matrix and the determinant, we can introduce another operation between vectors in three dimensions: the cross product. Also known as the vector product, it is a binary operation on two vectors resulting in a new vector that is orthogonal to both of the original vectors, making it particularly useful in physics and engineering for determining the direction of forces, torque, and rotation in three dimensions.

For two vectors $\mathbf{a}$ and $\mathbf{b}$, the magnitude of the resulting vector $\mathbf{c}$ from the cross product $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ is given as

$$|\mathbf{c}| = c = |\mathbf{a}||\mathbf{b}| \sin(\theta) = ab \sin(\theta), \tag{133}$$

where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$. The cross product is anti-commutative, meaning that $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$, and it is distributive over addition so that $\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c}$.

To determine the components of the cross product vector, however, we must calculate a determinant. Using Cartesian coordinates in three dimensions, we can use the unit vectors $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ to indicate direction along the $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ directions respectively. For two vectors $\mathbf{a}$ and $\mathbf{b}$, we can write

$$\mathbf{a} = a_x \hat{\mathbf{i}} + a_y \hat{\mathbf{j}} + a_z \hat{\mathbf{k}} \tag{134}$$

and

$$\mathbf{b} = b_x \hat{\mathbf{i}} + b_y \hat{\mathbf{j}} + b_z \hat{\mathbf{k}}. \tag{135}$$

The cross product $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ can then be found by computing the determinant,

$$\mathbf{c} = \mathbf{a} \times \mathbf{b} = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = (a_y b_z - a_z b_y)\hat{\mathbf{i}} - (a_x b_z - a_z b_x)\hat{\mathbf{j}} + (a_x b_y - a_y b_x)\hat{\mathbf{k}} = c_x \hat{\mathbf{i}} + c_y \hat{\mathbf{j}} + c_z \hat{\mathbf{k}}. \tag{136}$$

The magnitude $c = |\mathbf{c}|$ can then be computed as

$$c = \sqrt{c_x^2 + c_y^2 + c_z^2} = |\mathbf{a}||\mathbf{b}| \sin(\theta). \tag{137}$$

The resulting vector $\mathbf{c}$ is visualized below, where the cross product between vectors $\mathbf{a}$ and $\mathbf{b}$ results in vector $\mathbf{c}$ which is orthogonal to both of the original vectors. Orthogonality in this context means that the inner product between vectors $\mathbf{c}$ and $\mathbf{a}$, or $\mathbf{c}$ and $\mathbf{b}$, is equal to zero since the angle between each pair of vectors is $90° = \pi/2$ radians. That is,

$$\mathbf{a} \cdot \mathbf{c} = \mathbf{b} \cdot \mathbf{c} = ac \cos(\pi/2) = bc \cos(\pi/2) = 0. \tag{138}$$

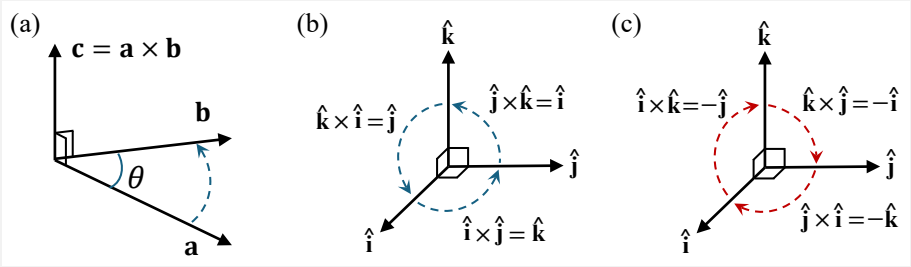


Fig. 9.11.1. (a) The cross product of two vectors $\mathbf{a}$ and $\mathbf{b}$ results in a third vector $\mathbf{c}$ which is orthogonal to *both* $\mathbf{a}$ and $\mathbf{b}$. (b) The unit vectors $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are all orthogonal to each other and taking the cross product of any ordered pair results in the third under the *right-hand rule* convention. (c) If the pair ordering is reversed, the resulting cross product is negative.

9.12 Gaussian elimination [return to section]

Gaussian elimination is a systematic method for solving systems of linear equations, and it can also be used to find the inverse of a matrix. The process involves augmenting the original matrix $\mathbf{A}$ with the identity matrix $\mathbf{1}$ and then performing row operations to transform $\mathbf{A}$ into $\mathbf{1}$. Allowed row operations include swapping rows, multiplying a row by a nonzero scalar, and adding a multiple of one row to another, with the goal of transforming the original matrix into an *upper triangular* form and then into *reduced row echelon* form. The resulting matrix on the right-hand side of the augmented matrix will then be $\mathbf{A}^{-1}$.

For Gaussian elimination to be used to find the inverse of a matrix, the matrix must be square (same number of rows and columns) and invertible, meaning that its determinant must be non-zero. If the determinant is zero, the matrix is *singular* and does not have an inverse. This singularity occurs if the matrix has linearly-dependent rows or columns, leading to at least one entire row of zeros in its row-reduced form.

Example: Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \end{bmatrix}.$$

We begin by augmenting it with the identity matrix on the right, then adding the first row to the third row, followed by subtracting the first row from the second row:

$$[\ \mathbf{A} \mid \mathbf{1}\ ] = \left[\begin{array}{ccc|ccc} 1 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 & 1 & 0 \\ -1 & 1 & 1 & 0 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & -2 & 2 & -1 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \end{array}\right].$$

For the next several linear operations towards transforming the left-hand side of the augmented matrix into the identity matrix, we multiply the second row by $-1/2$, then subtract the second row from the first row, followed by subtracting 2 times the second row from the third row:

$$\Rightarrow \left[\begin{array}{ccc|ccc} 1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 2 & 0 & 1 & 1 \end{array}\right].$$

Finally, we multiply the third row by $1/2$ and then add the third row to the second row, resulting in the identity matrix on the left-hand side of the augmented matrix and the inverse $\mathbf{A}^{-1}$ of the original matrix $\mathbf{A}$ on the right-hand side:

$$\Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{2} \end{array}\right] \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{2} \end{array}\right] \Rightarrow [\ \mathbf{1} \mid \mathbf{A}^{-1}\ ].$$

Thus, the matrix inverse $\mathbf{A}^{-1}$ is

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

9.13 Analytic solution to matrix inversion [return to section]

An explicit method for computing the inverse of a matrix is to do so analytically with the cofactor matrix method. Also known as the adjugate method, it is based on computing the cofactor matrix of a square matrix $\mathbf{A}$, transposing it to get the adjugate matrix, and finally dividing each element by the determinant of $\mathbf{A}$. This method can be expressed by first defining the *cofactor* matrix.

Given a matrix $\mathbf{A} = [a_{ij}]$, the cofactor c_{ij} is given by multiplying the *minor* determinant associated with element a_{ij} with $(-1)^{i+j}$, in which the matrix associated with the minor is the submatrix comprised of matrix elements which do not share the row *nor* column number of matrix element a_{ij}.

Example 1: Assuming we have the 3×3 matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix},$$

if we wish to compute the cofactor associated with, say, element a_{21}, we first identify the submatrix corresponding to this element and compute the minor $M_{2,1}$ by taking the determinant:

$$M_{2,1} = \begin{vmatrix} \cancel{a_{11}} & a_{12} & a_{13} \\ \cancel{a_{21}} & \cancel{a_{22}} & \cancel{a_{23}} \\ \cancel{a_{31}} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} = a_{12}a_{33} - a_{13}a_{32}.$$

The cofactor c_{21} is then found by multiplying this determinant with $(-1)^{i+j}$, where $i = 2$ and $j = 1$:

$$c_{21} = (-1)^{2+1} M_{2,1} = -(a_{12}a_{33} - a_{13}a_{32}) = a_{13}a_{32} - a_{12}a_{33}.$$

Thus, if a square $N \times N$ matrix $\mathbf{A}$ is invertible, the inverse $\mathbf{A}^{-1}$ is given in terms of the transpose of the cofactor matrix $\mathbf{C} = [c_{ij}]$ and the determinant $|\mathbf{A}|$ as

$$\mathbf{A}^{-1} = \frac{\mathbf{C}^{\mathsf{T}}}{|\mathbf{A}|} \therefore \left[\mathbf{A}^{-1}\right]_{ij} = \frac{[c_{ij}]^{\mathsf{T}}}{|\mathbf{A}|} = \frac{c_{ji}}{|\mathbf{A}|} = \frac{(-1)^{j+i} M_{ji}}{\sum_{j=1}^{N} (-1)^{1+j} M_{1j}}, \tag{139}$$

where, for $\mathbf{A} = [a_{mn}]$ with $m, n \in \{1, 2, \ldots, N\}$, the minor M_{ij} is the determinant of the submatrix in which $i \neq m$ and $j \neq n$:

$$M_{ij} = \det\left([a_{m \neq i, n \neq j}]\right). \tag{140}$$

Example 2: Compute the inverse of a general 2×2 matrix:

Given a 2×2 matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

the inverse is generally and straightforwardly computed as

$$\mathbf{A}^{-1} = \frac{1}{(a_{11}a_{22} - a_{12}a_{21})} \begin{bmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{bmatrix}^{\mathsf{T}} = \frac{1}{(a_{11}a_{22} - a_{12}a_{21})} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}. \tag{141}$$

9.14 Norm of a matrix [return to section]

The Euclidean norm, or 2-norm, of a matrix is a specific way to measure the size or length of the matrix, based on the Euclidean distances between points represented by the matrix's vectors. Specifically, for matrices, the 2-norm is defined as the *maximum singular* value of the matrix, which is also the largest eigenvalue of $\mathbf{A}^\intercal\mathbf{A}$ when $\mathbf{A}$ is a reaql matrix. In simpler terms, it can be thought of as the greatest stretch factor by which the matrix can increase the length of a vector.

For a matrix $\mathbf{A}$, the Euclidean norm (or *spectral* norm) is given by:

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}\left(\mathbf{A}^\intercal\mathbf{A}\right)}, \tag{142}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of $\mathbf{A}^\intercal\mathbf{A}$.

Example: Consider the matrix $\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix}$.

To find the Euclidean norm of $\mathbf{A}$, we first calculate $\mathbf{A}^\intercal\mathbf{A}$:

$$\mathbf{A}^\intercal\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix}^\intercal \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 4 & 0 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 12 \\ 12 & 16 \end{bmatrix}.$$

We then solve for the eigenvalues by solving the characteristic equation and select the largest eigenvalue:

$$|\mathbf{A}^\intercal\mathbf{A} - \lambda\mathbf{1}| = \begin{bmatrix} 9-\lambda & 12 \\ 12 & 16-\lambda \end{bmatrix} = 0$$

$$(9-\lambda)(16-\lambda) - (12)(12) = \lambda^2 - 25\lambda = \lambda(\lambda - 25) = 0$$

$$\therefore \lambda = \{0, 25\} \Rightarrow \lambda_{\max} = 25.$$

Thus, the 2-norm of $\mathbf{A}$ is $\|\mathbf{A}\|_2 = \sqrt{25} = 5$, suggesting that the maximum length by which this matrix can stretch a vector is by a factor of 5. This measure is particularly useful in applications where the distortion caused by a matrix to input signals or data vectors is critical, such as in signal processing and numerical simulations.

9.15 Condition number of a matrix [return to section]

The *condition number* of a matrix in the context of numerical linear algebra is a measure that describes how sensitive the solution of a system of linear equations is to changes in the input or errors in the calculations. It essentially quantifies the stability and accuracy of solutions computed using the matrix.

The condition number of a matrix, particularly with respect to solving linear systems, is defined in terms of its norm. For a square, invertible matrix $\mathbf{A}$, the condition number $\kappa(\mathbf{A})$ is defined as:

$$\kappa(\mathbf{A}) = \|\mathbf{A}\| \cdot \left\|\mathbf{A}^{-1}\right\|, \tag{143}$$

where $\|\mathbf{A}\|$ is *a* norm of $\mathbf{A}$, such as the 2-norm. While this measure depends on the choice of norm, the 2-norm is commonly used. A matrix is considered *well-conditioned* if its condition number is close to 1. This implies that errors in the input data or in the computational process result in only small errors in the final result. A matrix is *ill-conditioned* if its condition number is very large, implying that even small errors in the input data or during computation can lead to very large errors in the output, making the computational results potentially unreliable.

Example: Consider a 2×2 matrix $\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 5 \end{bmatrix}$.

Although we could manually compute the condition number by computing the inverse $\mathbf{A}^{-1}$, then the norm of $\mathbf{A}$ and $\mathbf{A}^{-1}$, and finally multiplying these two values together, the step-by-step manual computation is left as an exercise to the reader. Using MATLAB's `cond()` function to compute the condition number (or a combination of the `norm()` and `inv()` functions representing matrix norm and inverse, respectively), we have:

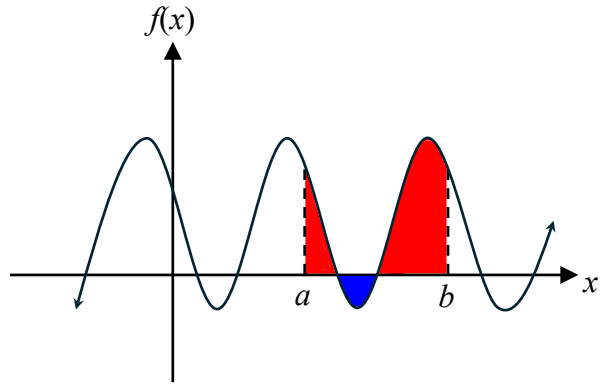
$$\kappa(\mathbf{A}) = \text{cond}(\mathbf{A}) = \text{norm}(\mathbf{A}) * \text{norm}(\text{inv}(\mathbf{A})) \approx 2.094.$$

Since this matrix has a condition number close to 1, it implies that solving a system using $\mathbf{A}$ would result in stable and accurate solutions. Thus, we would say matrix $\mathbf{A}$ is well-conditioned.

However, if the lower-right element is changed to $a_{22} = 0.333$, the determinant becomes a small number when computing the inverse, leading to a significantly larger condition number $\kappa(\mathbf{A}) \approx 11,110.89$ and thus $\mathbf{A}$ is now ill-conditioned. This implies that even small errors in the input data or during computation can lead to very large errors in the output, making the computational results potentially unreliable. As the determinant of a matrix approaches zero, the matrix becomes singular and the condition number tends towards infinity, resulting in numerical instability. Therefore, the condition number is a useful measure to ensure reliability in numerical precision in any calculation or simulation that requires it.

9.16 Integration of continuous functions [return to section]

With respect to function which are continuous in x, there are generally two types of integration: definite and indefinite. The former computes the integral of a function between two specific bounds a and b of an independent variable x, with the result being a number representing the area under the curve of the function from $x = a$ to $x = b$, as shown diagrammatically below, for which areas above the x-axis (shown in red) are positive while areas below the x-axis (shown in blue) are negative.



Indefinite integration finds the general form of the integral of $f(x)$ without specific bounds on x. The general result is a function plus a constant of integration, representing a family of solutions. This has widespread applications in engineering, such as calculating the total charge from a current vs. time graph or calculating average values of a signal when working with control systems and performing signal processing.

Example 1: To find the area under the curve $f(x) = x^2$ from $x = 0$ to $x = 2$, the integral is calculated as

$$\int_0^2 f(x)\mathrm{d}x = \int_0^2 x^2\mathrm{d}x = \frac{x^3}{3}\bigg|_0^2 = \frac{(2)^3}{3} - \frac{(0)^3}{3} = \frac{8}{3},$$

where $\mathrm{d}x$ is the *differential element* of the variable x for which the function f is being integrated over.

Example 2: Find the area under the curve of $f(x) = 4x - x^2$ in the interval $0 \leq x \leq 4$.

Since the interval is finite in extent, we can compute the following definite integral:

$$\int_0^4 f(x)\mathrm{d}x = \int_0^4 4x - x^2\mathrm{d}x = \left[\frac{4x^2}{2} - \frac{x^3}{3}\right]_0^4 = \left[2(16) - \frac{(64)}{3}\right] - [0 - 0] = \frac{32}{3}.$$

Problem 1: If the current I, measured in mA = mC/s, flowing through a device is $I(t) = 4\sin(\pi t)$, find the total charge Q, measured in mC, transferred from $0 \leq t \leq 1$ seconds.

Problem 2: In an RC circuit, the voltage across the capacitor decreases as the capacitor discharges without any external voltage source. If the initial voltage across the capacitor is V_0 and it discharges through a resistor, the voltage $V(t)$ at time t is given by

$$V(t) = V_0\mathrm{e}^{-t/RC}, \tag{144}$$

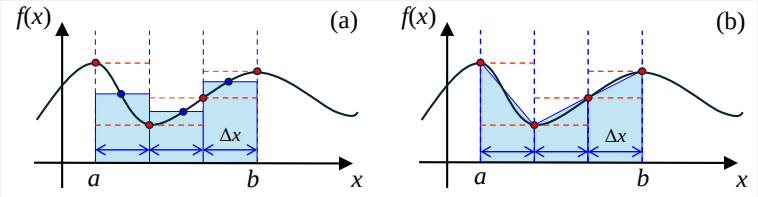
where R is resistance, C is capacitance, and t is time. Calculate the energy dissipated in the resistor during the first 5 seconds.

Problem 3: The power P in watts delivered by a power supply over time t in hours is modeled by the equation $P(t) = 5t^3 - 15t^2 + 20t$. Find the total energy delivered by the power supply from $t = 0$ to $t = 3$ hours.

9.17 Integration of discrete functions [return to section]

Discrete integration involves numerical techniques to estimate the integral of a function, and is a common example of a *multiply-accumulate*, or MAC, operation. This can be useful where analytic integration is difficult or impossible, or when working with data from experiments and measurements that are inherently discrete. For example, discrete integration can be used for calculating the total energy or power in a signal based on sampled data or analyzing data from sensors and systems which provide measurements at discrete intervals.

There are three common techniques used for discrete integration: Riemann sums, the Trapezoidal rule, and Simpson's rule. Riemann sums approximate integration by dividing the area under a curve into rectangles and summing their areas. This can be done with either uniform or non-uniform spacing between sample points, and there are three types of Riemann sums: left, right, and midpoint rules. The left and right rules respectively use the left and right subinterval endpoints, while the midpoint rule uses the average of the left and right endpoints in a given subinterval for the height. The trapezoidal rule improves upon Riemann sums by approximating the area under the curve with trapezoids instead of rectangles, which generally provides a better approximation. The midpoint and trapezoidal methods are visually depicted below for uniformly sampled intervals of width Δx in subfigures (a) and (b) respectively.



Simpson's rule utilizes *splines* instead of lines to approximate the curve, providing even more accuracy, especially when the function is smooth. Generally, a spline is a piecewise function composed of connected polynomials. In the case of Simpson's rule, the simplest nonlinear spline would be composed of parabolic arcs (second-order polynomials).

Example: Approximate the integral of $f(x) = x^2$ from $0 \leq x \leq 1$ using $N = 5$ subintervals and the midpoint rule and calculate the relative error. We begin by first computing the interval width,

$$\Delta x = \frac{x_{\max} - x_{\min}}{N} = \frac{1-0}{5} = 0.2.$$

The endpoints of each subinterval are thus $[0, 0.2]$, $[0.2, 0.4]$, $[0.4, 0.6]$, $[0.6, 0.8]$, and $[0.8, 1]$, and therefore the corresponding midpoints are 0.1, 0.3, 0.5, 0.7, and 0.9, respectively. Finally, we calculate the integral by multiplying the value of the function (the height) with the subinterval length, and adding these subinterval areas together. Since the subinterval spacing is uniform, we can factor it out and simply multiply it with the sum of function values at the midpoints:

$$\begin{aligned}\text{Area} = \sum_{n=1}^{N=5} f(x_n)\Delta x &= \left[f(x_1 + f(x_2) + f(x_3) + f(x_4) + f(x_5)\right]\Delta x \\ &= \left[0.1^2 + 0.3^2 + 0.5^2 + 0.7^2 + 0.9^2\right](0.2) = 0.33.\end{aligned}$$

Comparing this with the actual integral of $f(x)$, we get a relative error of only 1%:

$$\int_0^1 x^2 \mathrm{d}x = \left.\frac{x^3}{3}\right|_0^1 = 0.\overline{3} \Rightarrow \frac{|0.\overline{3} - 0.33|}{0.\overline{3}} \times 100\% = 1\%$$

Problem 1: Calculate the approximate value of the integral of $f(x) = \mathrm{e}^{-x^2}$ in the interval $0 \leq x \leq 1$ using the trapezoidal rule with $n = 4$ subintervals.
Problem 2: Estimate the integral of $f(x) = \ln(x)$ in the interval $1 \leq x \leq 2$ using the midpoint rule with $n = 2$ subintervals.

9.18 Coefficient of determination [return to section]

The coefficient of determination, often denoted as R^2, is a statistical measure used in the context of least squares fitting to evaluate how well a regression model fits the observed data. It is particularly useful in linear regression analysis, where it quantifies the proportion of the variance in the dependent variable that is predictable from the independent variables. In simpler terms, it describes how much of the variation in the data can be explained by the fitted model.

Using Eq.(62), the coefficient of determination for n sample points is calculated as:

$$R^2 = 1 - \frac{S(\mathbf{a})}{S_{\text{tot}}}, \tag{145}$$

where $S_{\text{tot}} = n\sigma_y^2$ is the total sum of squares, proportional to the variance of the data. In terms of Eq.(72) and (74), the coefficient of determination can be expressed as

$$R^2 = \frac{\mathbf{a}^\intercal \cdot \boldsymbol{Y} - n\langle y\rangle^2}{n\sigma_y^2}, \tag{146}$$

where $\langle y\rangle = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sigma_y^2 = \langle y^2\rangle - \langle y\rangle^2$ denote the mean and variance of y respectively, with $\langle y^2\rangle = \frac{1}{n}\sum_{i=1}^{n} y_i^2$.

Example: Once the coefficients $\mathbf{a}$ are calculated in the case of a quadratic ($N = 2$) model, we can then compute the coefficient of determination explicitly. Using Eq. (146), we have

$$R^2 = \frac{a_0 \sum y_i + a_1 \sum y_i x_i + a_2 \sum y_i x_i^2 - \frac{1}{n}\left(\sum y_i\right)^2}{\sum y_i^2 - \frac{1}{n}\left(\sum y_i\right)^2} \tag{147}$$

The coefficient of determination has a value $0 \leq R^2 \leq 1$, where $R^2 = 1$ implies a perfect fit and the regression model explains all of the variation in the data, while $R^2 = 0$ implies the model does not explain any variation in the data. Thus, intermediate values represent the partial explanatory power of the model, indicating that some but not all of the variation is captured.

However, a high value of R^2 does not always imply a better model due to potential *overfitting*, especially with many independent variables, or *predictors*. Thus, it should be used carefully when comparing models with different numbers of predictors.

9.19 The binomial theorem [return to section]

The binomial theorem provides a formula for expanding expressions that are raised to a power, and is given by

$$(a+b)^n = \sum_{k=0}^{n} C(n,k) a^{n-k} b^k = a^n + na^{n-1}b + \ldots + nab^{n-1} + b^n, \tag{148}$$

where $C(n,k)$ is the *n-choose-k* binomial coefficient, representing the number of ways to choose k elements from a set of n total elements.

Example: Use the binomial theorem to expand $(x+2)^3$.

We can directly apply (148) to get

$$\begin{aligned}(x+2)^3 = \sum_{k=0}^{3} C(3,k) x^{3-k} 2^k &= \binom{3}{0} x^{3-0}2^0 + \binom{3}{1} x^{3-1}2^1 + \binom{3}{2} x^{3-2}2^2 + \binom{3}{3} x^{3-3}2^3 \\ &= \frac{3!}{0!(3-0)!}x^3 + \frac{3!}{1!(3-1)!}2x^2 + \frac{3!}{2!(3-2)!}4x + \frac{3!}{3!(3-3)!}8 \\ &= x^3 + 6x^2 + 12x + 8\end{aligned}$$

9.20 *LU* decomposition [return to section]

LU decomposition is a technique that simplifies solving systems of linear equations, inverting matrices, and calculating determinants. It is a useful tool in numerical linear algebra due to its efficiency and effectiveness in handling large matrices.

The *LU* decomposition of a square matrix $\mathbf{A}$ involves factorizing it as the product of a lower triangular matrix ($\mathbf{L}$) with ones on the diagonal and an upper triangular matrix ($\mathbf{U}$) such that $\mathbf{A} = \mathbf{LU}$. This factorization can be used to solve the linear equation $\mathbf{Ax} = \mathbf{b}$ by first solving $\mathbf{Ly} = \mathbf{b}$ for vector $\mathbf{y}$ using *forward* substitution, and then solving $\mathbf{Ux} = \mathbf{y}$ for vector $\mathbf{x}$ using *back* substitution.

Example: Consider the following matrix $\mathbf{A}$ and decompose it into the two triangular matrices $\mathbf{L}$ and $\mathbf{U}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 1 \\ 4 & 7 & 2 \\ 6 & 12 & 5 \end{bmatrix}$$

To find the *LU* decomposition, we execute the following steps:

1. Initialize $\mathbf{L}$ and $\mathbf{U}$ as identity and zero matrices respectively:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

2. Fill the $\mathbf{U}$ matrix with the values from $\mathbf{A}$:

$$\mathbf{U} = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

3. Perform row operations to fill in the $\mathbf{L}$ and $\mathbf{U}$ matrices by initially eliminating the first column below the diagonal:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

4. Next, the second column is eliminated below the diagonal:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 3 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 2 & 3 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Problem 1: Solve the system of linear equations using *LU* decomposition:

$$\begin{aligned} 2x + 3y + z &= 5 \\ 4x + 7y + 2z &= 11 \\ 6x + 18y + 5z &= 25 \end{aligned}$$

9.21 *LDL* decomposition [return to section]

LDL decomposition is a type of matrix factorization particularly useful for symmetric and Hermitian matrices. It expresses a given symmetric matrix $\mathbf{A}$ as the product of a lower triangular matrix $\mathbf{L}$, a diagonal matrix $\mathbf{D}$, and the transpose (or conjugate transpose) of the lower triangular matrix $\mathbf{L}^\intercal$ so that $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^\intercal$. This decomposition is valuable in numerical analysis, especially for solving linear systems as well as for understanding the properties of the matrix. Furthermore, this decomposition is particularly efficient because it avoids the need for *pivoting* and is stable for symmetric positive definite matrices.

Example: Let's consider a symmetric matrix $\mathbf{A}$ and decompose it into $\mathbf{L}$, $\mathbf{D}$, and $\mathbf{L}^\intercal$:

$$\mathbf{A} = \begin{bmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{bmatrix}$$

To find the LDL decomposition, the following steps are taken:

1. Initialize $\mathbf{L}$ and $\mathbf{D}$ matrices:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

2. Compute the elements of $\mathbf{D}$ and $\mathbf{L}$:
   - For D_{11}:
   $$D_{11} = A_{11} = 4$$
   - For L_{21} and L_{31}:
   $$L_{21} = \frac{A_{21}}{D_{11}} = \frac{12}{4} = 3$$
   $$L_{31} = \frac{A_{31}}{D_{11}} = \frac{-16}{4} = -4$$
   - For D_{22}:
   $$D_{22} = A_{22} - L_{21}^2 D_{11} = 37 - 3^2 \cdot 4 = 37 - 36 = 1$$
   - For L_{32}:
   $$L_{32} = \frac{A_{32} - L_{31}L_{21}D_{11}}{D_{22}} = \frac{-43 - (-4) \cdot 3 \cdot 4}{1} = \frac{-43 + 48}{1} = 5$$
   - For D_{33}:
   $$D_{33} = A_{33} - (L_{31}^2 D_{11} + L_{32}^2 D_{22}) = 98 - ((-4)^2 \cdot 4 + 5^2 \cdot 1) = 98 - (16 \cdot 4 + 25) = 98 - 89 = 9$$

3. Thus, the final matrices $\mathbf{L}$, $\mathbf{D}$, and $\mathbf{L}^\intercal$ are:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -4 & 5 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 9 \end{bmatrix}, \quad \mathbf{L}^\intercal = \begin{bmatrix} 1 & 3 & -4 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

Problem 1: Solve the system of linear equations using LDL decomposition:

$$\begin{aligned} 4x + 12y - 16z &= 12 \\ 12x + 37y - 43z &= 30 \\ -16x - 43y + 98z &= -48 \end{aligned}$$

9.22 Eigenvalues and eigenvectors [return to section]

Eigenvectors and eigenvalues are important concepts in linear algebra with applications across physics, engineering, computer science, and other quantitative fields. They are especially useful in systems analysis, stability studies, quantum mechanics, and methods for solving differential equations.

9.23 Singular value decomposition [return to section]

Singular value decomposition (SVD) is a matrix factorization technique in linear algebra. It generalizes the eigendecomposition of a square matrix to any $m \times n$ matrix. SVD has applications in signal processing, statistics, and machine learning, particularly in data compression, noise reduction, and dimensionality reduction.

9.24 Rising and falling factorials [return to section]

The rising factorial, also known as the Pochhammer function or Pochhammer symbol, is best denoted[a] as $x^{\overline{n}}$,

$$x^{\overline{n}} = \prod_{k=1}^{n}(x+k-1) = \prod_{k=0}^{n-1}(x+k) = x(x+1)(x+2)\cdots(x+n-1), \tag{149}$$

where n is an integer. Similarly, the falling factorial is best denoted as $x^{\underline{n}}$, where

$$x^{\underline{n}} = \prod_{k=1}^{n}(x-k+1) = \prod_{k=0}^{n-1}(x-k) = x(x-1)(x-2)\cdots(x-n+1). \tag{150}$$

Rising and falling factorials are related by

$$x^{\underline{n}} = (-1)^n(-x)^{\overline{n}} \therefore x^{\overline{n}} = (-1)^n(-x)^{\underline{n}}. \tag{151}$$

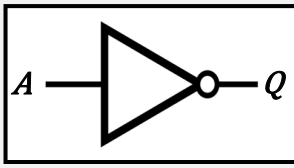
Example:

[a] In the interest of mitigating notational ambiguity, overline ($x^{\overline{n}}$) and underline ($x^{\underline{n}}$) notation was introduced to respectively represent rising and falling factorials in D. Knuth, *The art of computer programming*, (Pearson Education, 1997).

9.25 Logic gates [return to section]

In electrical engineering, logic gates are the building blocks of digital circuits. They perform Boolean operations on one or more binary inputs to produce a single output and are therefore useful for creating digital circuits that can perform complex computations, from simple arithmetic to intricate algorithms.

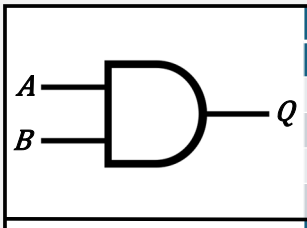
In digital electronics, the function of a given gate may be quantified using a truth table which lists all possible values of input variables along with the gate's output value. Along with their inverses, the basic set of logic gates and their corresponding truth tables are shown in the following figures.



NOT gate truth table

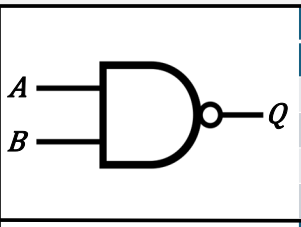
A	$Q = \overline{A}$
0	1
1	0

Fig. 1. The NOT logic gate, which takes a single Boolean input A and outputs the opposite value $Q = \text{NOT}(A) = \overline{A}$.



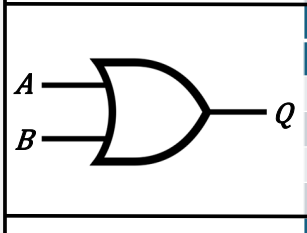
AND gate truth table

A	B	$Q = A \times B$
0	0	0
0	1	0
1	0	0
1	1	1



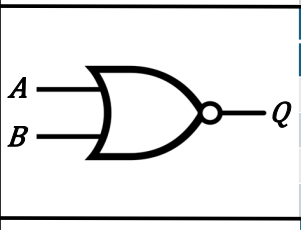
NAND gate truth table

A	B	$Q = \overline{A \times B}$
0	0	1
0	1	1
1	0	1
1	1	0



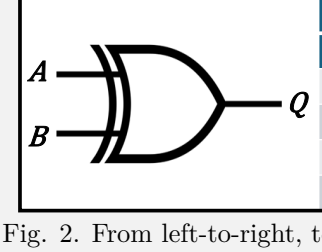
OR gate truth table

A	B	$Q = A + B$
0	0	0
0	1	1
1	0	1
1	1	1



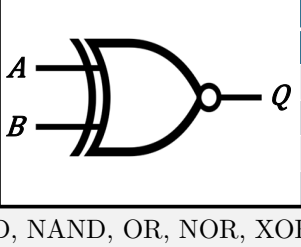
NOR gate truth table

A	B	$Q = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0



XOR gate truth table

A	B	$Q = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



XNOR gate truth table

A	B	$Q = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

Fig. 2. From left-to-right, top-to-bottom, the AND, NAND, OR, NOR, XOR, and XNOR logic gate symbols and corresponding truth tables are respectively given. Each gate has two inputs A and B and a single output Q which is either a logical 1 ("True") or logical 0 ("False") according to the truth table.

These set of logic gates can be combined in various ways to build circuits capable of performing any logical operation. Such a concept is known as *functional completeness*, which is independently satisfied by both the NAND and NOR logic gates. That is, with either of these logic gates, all other logical operations can be replicated.

A set of practice problems may be found in **Homework Problems: Logic gate design**.

9.26 Higher-order differentiation of discrete functions [return to section]

9.27 Origin of beam splitter amplitudes [return to section]

Consider the ideal, lossless, symmetric, 50:50 beam splitter configured with two input ports and two output ports and illustrated in Fig. 15. Quantum field reflection and transmission coefficients for a single photon entering port 1 or 2 are $r_{\rm ph,1}$, $r_{\rm ph,2}$, $t_{\rm ph,1}$ and $t_{\rm ph,2}$ respectively. If a single isolated photon enters port 1 then it is in input state $|n_1 = 1, n_2 = 0\rangle_{\rm in}$. For the case being considered, it is well known that the phase of the transmitted field leads the phase of the reflected field by $\pi/2$ [a]. The single-photon input state $|n_1 = 1, n_2 = 0\rangle_{\rm in}$ has quantum field amplitude at port 1 that can be set to $a_1 = 1$ and at port 2 it is set to $a_1 = 0$. In this case the input state is a product state so that $|1, 0\rangle_{\rm in} = |1\rangle_1 \otimes |0\rangle_2$ where $|0\rangle$ is the vacuum state. For input state $|n_1 = 0, n_2 = 1\rangle_{\rm in}$ the quantum field output is $a_3 = r_{\rm ph,2}$ at port 3 and $a_4 = t_{\rm ph,2}$ at port 4. The single-photon input and output *quantum field amplitudes* are related via

$$\begin{bmatrix} a_3 \\ a_4 \end{bmatrix}_{\rm out} = \begin{bmatrix} t_{\rm ph,1} & t_{\rm ph,2} \\ r_{\rm ph,1} & r_{\rm ph,2} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}_{\rm in} = \hat{U}_{\rm B} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}_{\rm in}, \tag{152}$$

where $\hat{U}_{\rm B}$ is a 2×2 matrix describing the ideal lossless beam splitter. Photon probability is conserved because an ideal lossless beam splitter is being considered. This means that the 2×2 matrix $\hat{U}_{\rm B}$ is unitary and consequently its Hermitian adjoint is its inverse $\hat{U}_{\rm B}^{\dagger} = \hat{U}_{\rm B}^{-1}$. Hence,

$$\hat{U}_{\rm B}^{\dagger} = \begin{bmatrix} t_{\rm ph,1}^* & r_{\rm ph,1}^* \\ r_{\rm ph,2}^* & t_{\rm ph,2}^* \end{bmatrix} = \frac{1}{t_{\rm ph,1}t_{\rm ph,2} - r_{\rm ph,1}r_{\rm ph,2}} \begin{bmatrix} t_{\rm ph,2} & -r_{\rm ph,2} \\ -r_{\rm ph,1} & t_{\rm ph,1} \end{bmatrix} = \hat{U}_{\rm B}^{-1}. \tag{153}$$

Because the determinant of a unitary matrix has unit magnitude, in general, $t_{\rm ph,1}t_{\rm ph,2} - r_{\rm ph,1}r_{\rm ph,2} = {\rm e}^{{\rm i}\phi}$ where ϕ is a global phase factor that has no impact on relative phase between matrix elements and may be set to $\phi = \pi$, so that $t_{\rm ph,1}t_{\rm ph,2} - r_{\rm ph,1}r_{\rm ph,2} = -1$ since ${\rm e}^{{\rm i}\pi} = -1$. Inserting this into the expression for $\hat{U}_B^{\dagger}$ gives $r_{\rm ph,1} = r_{\rm ph,2}^*$ and $t_{\rm ph,1} = -t_{\rm ph,2}^*$ from which it may be concluded that $|r_{\rm ph,1}| = |r_{\rm ph,2}|$ and $|t_{\rm ph,1}| = |t_{\rm ph,2}|$. Re-expressing the complex terms for $r_{\rm ph,1}$ and $t_{\rm ph,1}$ gives

$$|r_{\rm ph,1}|{\rm e}^{{\rm i}\theta_{r_{\rm ph,1}}} = |r_{\rm ph,2}|{\rm e}^{-{\rm i}\theta_{r_{\rm ph,2}}} \quad \text{and} \quad |t_{\rm ph,1}|{\rm e}^{{\rm i}\theta_{t_{\rm ph,1}}} = -|t_{\rm ph,2}|{\rm e}^{-{\rm i}\theta_{t_{\rm ph,2}}}. \tag{154}$$

Dividing these equations gives

$$\frac{|t_{\rm ph,1}|{\rm e}^{{\rm i}\theta_{t_{\rm ph,1}}}}{|r_{\rm ph,1}|{\rm e}^{{\rm i}\theta_{r_{\rm ph,1}}}} = \frac{-|t_{\rm ph,2}|{\rm e}^{-{\rm i}\theta_{t_{\rm ph,2}}}}{|r_{\rm ph,2}|{\rm e}^{-{\rm i}\theta_{r_{\rm ph,2}}}} \rightarrow {\rm e}^{{\rm i}\theta_{t_{\rm ph,1}} - {\rm i}\theta_{r_{\rm ph,1}}} = -{\rm e}^{-{\rm i}\theta_{t_{\rm ph,2}} + {\rm i}\theta_{r_{\rm ph,2}}} = {\rm e}^{-{\rm i}\theta_{t_{\rm ph,2}} + {\rm i}\theta_{r_{\rm ph,2}} + {\rm i}\pi} \tag{155}$$

where use is made of ${\rm e}^{{\rm i}\pi} = -1$. Hence,

$$\left(\theta_{t_{\rm ph,1}} - \theta_{r_{\rm ph,1}}\right) + \left(\theta_{t_{\rm ph,2}} - \theta_{r_{\rm ph,2}}\right) = \pi. \tag{156}$$

For the ideal, lossless, symmetric, 50:50 beam splitter $r_{\rm ph,1} = r_{\rm ph,2}$ and $t_{\rm ph,1} = t_{\rm ph,2}$. Therefore, the phase difference between transmission and reflection at each port is the same,

$$\left(\theta_{t_{\rm ph,1}} - \theta_{r_{\rm ph,1}}\right) + \left(\theta_{t_{\rm ph,2}} - \theta_{r_{\rm ph,2}}\right) = \frac{\pi}{2} \tag{157}$$

and it is clear that the phase of the transmitted field leads the phase of the reflected field by $\pi/2$. For the perfect, lossless, symmetric, 50:50 dielectric beam splitter $|r_{\rm ph,1}| = |r_{\rm ph,2}| = |t_{\rm ph,1}| = |t_{\rm ph,2}|$ which can only be satisfied if $|r_{\rm ph,1}| = |r_{\rm ph,2}| = |r_{\rm ph}|$ is pure real and $|t_{\rm ph,1}| = |t_{\rm ph,2}| = |t_{\rm ph}|$ is pure imaginary. Given the fact that the determinant of the unitary matrix requires $t_{\rm ph,1}t_{\rm ph,2} - r_{\rm ph,1}r_{\rm ph,2} = 1$, then

$$r_{\rm ph} = -\frac{1}{\sqrt{2}} \quad \text{and} \quad t_{\rm ph} = \frac{\rm i}{\sqrt{2}}, \tag{158}$$

so that the unitary 2×2 matrix $\hat{U}_{\rm B}$ describing a single photon interacting with an ideal, lossless, symmetric, 50:50 beam splitter and satisfying the unitary requirement $\hat{U}_{\rm B}^{\dagger} = \hat{U}_{\rm B}^{-1}$ is

$$\hat{U}_{\rm B} = \frac{1}{\sqrt{2}} \begin{bmatrix} {\rm i} & -1 \\ -1 & {\rm i} \end{bmatrix} = \begin{bmatrix} t_{\rm ph} & r_{\rm ph} \\ r_{\rm ph} & t_{\rm ph} \end{bmatrix}. \tag{159}$$

[a] A. Agnesi and V. Degiorio, *Opt. Laser Tech.* **95**, 72-73 (2017). V. Degiorio, *Am. J. Phys.* **48**, 81 (1980).

9.28 Introduction to the Fourier transform [return to section]

9.29 The Fast Fourier Transform (FFT) [return to section]

9.30 Classical analog of the "Mandel dip" **[return to section]**

In a somewhat contrived experiment it is possible to configure a classical (or a single photon) analog of the "Mandel dip" by applying external controls to randomly switch the quadrature phase difference of classical electromagnetic radiation (or a single photon field) entering the respective input ports of an ideal, lossless, 50:50 beam splitter such that the output field is multiplexed to appear at either one but not both output ports[a]. So, in this sense, any claim of measuring quantum interference and correlation associated with the Mandel dip requires specifying the absence of special external phase modulation of input fields or replacing single photon detectors with photon number-resolving detectors.

To see how interference can be used to multiplex two input signals into a single output port, recall that reflection amplitude at a perfect, lossless, symmetric, 50:50 beam splitter is $r_{\rm ph} = -1/\sqrt{2}$ and transmission is $t_{\rm ph} = {\rm i}/\sqrt{2}$ (Eqns. (83) and (84)). Flux conservation in the lossless system requires $|r_{\rm ph}|^2 + |t_{\rm ph}|^2 = 1$. If the field amplitude at input port 1 is 1 and at port 2 it is in phase quadrature with a value $-{\rm i} = t_{\rm ph}/r_{\rm ph}$ then the output field at port 3 is zero and the output field at port 4 is 2. The output field can be multiplexed by switching the phase of the port 1 and port 2 input fields.

[a] S. Sadana, D. Ghosh, K. Joarder, A. N. Lakshmi, B. C. Sanders, and U. Sinha, *Phys. Rev. A* **100**, 013839 (2019).

9.31 Coordinate systems [return to section]

10 Homework Problems

10.1 Complex differentiation [return to section]

Problem 1: Use the Cauchy-Riemann conditions to determine whether the function $f(z) = \mathrm{e}^x \cos(y) + \mathrm{i}\mathrm{e}^x \sin(y)$ is complex-differentiable at any point in the complex plane.

Problem 2: The standard numerical technique to compute the derivative of a function $f(x)$ is

$$f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}, \tag{160}$$

where Δx is the step size. However we have a subtraction which causes difference errors under numerical implementation.

Given the function f over the $\mathbb{R}$ line, we can construct an analytic function $f(z)$ (where x, a real variable, gets replaced by z, a complex variable). Then from the Taylor series

$$f(x + ih) = f(x) + ihf'(x) - h^2 f''(x)/2 - ... \tag{161}$$

where $h \in \mathbb{R}$. Using the above equation, show that

$$f'(x) \approx \frac{\mathrm{Im}(f(x + ih))}{h} \tag{162}$$

and compute the error in the approximation E_1.

Problem 3: Compute the appropriate numerical approximation of $f''(x)$ from equation *Eq.* 161 and find the error E_2. Observe and state your observation of the merit in this method vs the second-order derivative approximation

$$f''(x) \approx \frac{f(x + \Delta x) + f(x - \Delta x) - 2f(x)}{\Delta x^2}. \tag{163}$$

Problem 4: For the following function

$$f(x) = \frac{e^x}{(\cos^3(x) + \sin^3(x))^{1/2}},$$

compute $f'(x)$ at $x = \pi/4$ by both the complex method and finite difference method. Using a symbolic toolbox (using either MATLAB, Mathematica, SciPy, or some other tool of choice), find the actual derivative $f'(x)$ at $x = \pi/4$. Now compare the errors in both methods. Throughout the whole problem let h iterate over the order from $h = 10^{-1}$ to $h = 10^{-16}$. Plot the error.

10.2 Poynting vector [return to section]

The Poynting vector represents the directional energy flux (the rate of energy transfer per unit area) of an electromagnetic field.[a] Such a concept can be used to understand, for example, how electromagnetic energy is transmitted from a radio antenna to a receiver. In vacuum, the Poynting vector $\mathbf{S}$ is defined as the cross product of the electric field $\mathbf{E}$, measured in volts per meter (V/m), and the magnetic field $\mathbf{B}$, measured in teslas (T),

$$\mathbf{S} = \frac{1}{\mu_0}\mathbf{E} \times \mathbf{B} = \mathbf{E} \times \mathbf{H}, \tag{164}$$

where $\mu_0 = 4\pi \times 10^{-7}$ H/m (henries per meter) is the permeability of free space and the magnetic field intensity $\mathbf{H} = \mathbf{B}/\mu_0$. The direction of $\mathbf{S}$ indicates the direction of energy propagation of the electromagnetic wave, and its magnitude represents the rate of energy transfer per unit area.

Problem 1: If *complex* field $\mathbf{G} = \left(\mathbf{D}/\sqrt{\varepsilon_0} + \mathrm{i}\mathbf{B}/\sqrt{\mu_0}\right)/\sqrt{2}$ show that Maxwell's equations in free space and in the absence of free charges may be written as the complex equations

$$\nabla \cdot \mathbf{G} = 0 \tag{165}$$

and

$$\mathrm{i}\frac{\partial \mathbf{G}}{\partial t} = \frac{1}{\sqrt{\varepsilon_0}\mu_0}\nabla \times \mathbf{G}, \tag{166}$$

where $\mathbf{G} = \frac{1}{\sqrt{2}}\left(\frac{\mathbf{G}}{\sqrt{2}} + \mathrm{i}\frac{\mathbf{B}}{\sqrt{\mu_0}}\right)$.

Problem 2: Show that the energy flux density in the electromagnetic field given by the Poynting vector is

$$\mathbf{S} = \mathbf{E} \times \mathbf{H} = \frac{-\mathrm{i}}{\sqrt{\varepsilon_0}\mu_0}\left(\mathbf{G}^* \times \mathbf{G}\right). \tag{167}$$

Problem 3: If the field $\mathbf{G}$ is purely real, what is the value of $\mathbf{S}$?

Problem 4: Show that the electromagnetic energy density is $U = |\mathbf{G}|^2$.

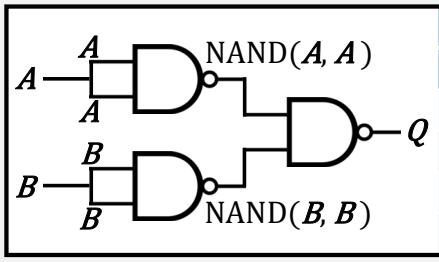
Problem 5: How would Maxwell's equations be modified if magnetic charge g exists (thereby implying the existence of magnetic monopoles)? Derive an expression for conservation of magnetic current and write down a generalized Lorentz force law that includes magnetic charge. Write Maxwell's equations with magnetic charge in termns of a field $\mathbf{G}$.

[a] The Poynting vector is named after the physicist John Henry Poynting who first derived it in 1884.

10.3 Logic gate design [return to section]

The set of logic gates described in **Explore More: Logic gates** can be combined in various ways to build circuits capable of performing any logical operation. Such a concept is known as *functional completeness*, which is independently satisfied by both the NAND and NOR logic gates. That is, with either of these logic gates, all other logical operations can be replicated.

Example: Construct an OR gate using only NAND gates. As shown in Fig. 1, we can use a set of three NAND gates to construct an OR gate by having the first value A go into both inputs of one NAND gate, the second value B go into both inputs of a second NAND gate, and finally the outputs of these two NAND gates are passed as inputs to a third NAND gate. Completing the truth table for this gate shows that it is functionally equivalent to a single OR gate.



NAND(NAND(A, A), NAND(B, B))		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 1. By combining three NAND gates, an OR logic gate can be constructed. That is, the output $Q = \text{NAND}(\text{NAND}(A, A), \text{NAND}(B, B)) = \text{OR}(A, B)$.

Problem 1: Design a circuit that outputs the opposite state of the input using NAND gates only.

Problem 2: Suppose you are designing a safety system where a machine should only operate if two sensors (A and B) are activated and there is manual confirmation (C). Construct a logic circuit using only AND and OR gates.

Problem 3: Implement an XOR gate using AND, OR, and NOT gates.

Problem 4: Design a simple binary multiplier circuit for two 2-bit numbers ($A1A0$ and $B1B0$) and then accumulate the result to a previous total using an AND and an XOR gate for addition.

10.4 Combinatorics **[return to section]**

Problem 1: Calculate $C(100, 99)$ by hand.

Problem 2: Use the binomial theorem to expand $(x - 2)^6$.

Problem 3: A student forgot the 3-digit code to their bicycle lock.

(a) If each digit can range from 0 to 9, how many unique codes could exist?

(b) The student remembers that none of the digits repeat in their code. How many unique codes can exist now?

(c) The student then remembers at least one of the three digits to their unique code. How many unique codes can exist now?

(d) By what fraction do each of these two pieces of information reduce the original number of possible codes (and therefore the guessing difficulty) by themselves as well as together?

Problem 4: An engineer must design a security system featuring a set of *unique* numbers comprising a code which must have strictly less than one in a million chance of being randomly guessed.

(a) Assuming each *unique* number in the code can be a value between 0 and 9, determine the minimum number of code digits required to satisfy the design requirement.

(b) If the constraint of uniqueness is removed such that a given digit could be used more than once and still satisfy the guessing probability requirement, what is now the minimum number of code digits required?

(c) The engineer must now add a second layer of security which requires a second type of code that is restricted to only 4 *unique* numbers yet must still satisfy the guessing probability requirement of the design, however the value of each number is no longer restricted from 0 to 9, but rather 1 to n. Determine the minimum value of n such that randomly guessing the 4-number code would have a probability less than 10^6.

10.5 Differentiation **[return to section]**

Problem 1: Showing each step, find the derivative $f'(x)$ of the function $f(x) = \mathrm{e}^{x^2+3x}$.

Problem 2: Showing each step, find the derivative $f'(x)$ of the function $f(x) = x^4 \cos(2x)$.

Problem 3: Showing each step, find the derivative $f'(x)$ of the function $f(x) = (x^2+1)/(\sqrt{x+2})$.

Problem 4: In an AC circuit, the capacitive reactance X_C is given by $X_C = \frac{1}{\omega C}$, where the signal angular frequency ω varies with time according to the function $\omega = 2\pi t$ and the capacitance C is a constant. Derive an expression for the rate of change of the reactance, $X_C'(t)$.

Problem 5: The power P in a circuit is given by $P = IV$, where I is the current and V is the voltage. If the voltage varies with respect to time as $V(t) = t^3 - t$ and the current exponentially decays over time as $I(t) = \mathrm{e}^{-t}$, derive an expression for the rate of change of the power, $P'(t)$.

Problem 6: The amplitude A of a signal in a circuit depends on the frequency f according to the equation $A(f) = \ln(1+f^2)$. If the frequency of the signal changes with respect to time according to $f(t) = \sin(t)$, derive an expression for the rate of change of the amplitude over time, $A'(t)$.

10.6 Vectors **[return to section]**

Problem 1: Given a vector $\mathbf{v} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$, find the unit vector $\mathbf{u}$ in the direction of $\mathbf{v}$.

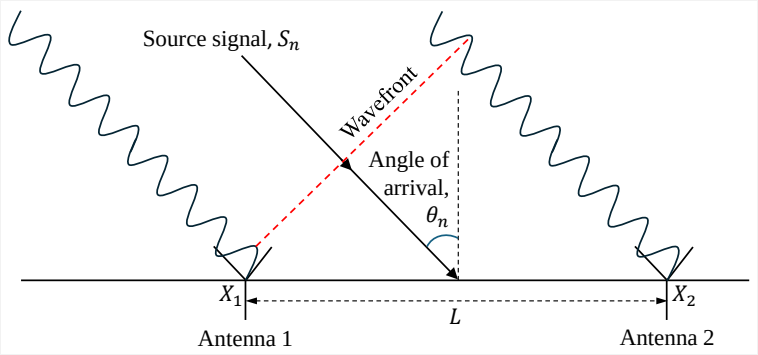
Problem 2: Calculate the inner product of vectors $\mathbf{a} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} -2 \\ 0 \\ 3 \end{bmatrix}$.

Problem 3: Consider two complex *orthogonal* vectors $\mathbf{v} = \begin{bmatrix} 3+4\mathrm{i} \\ a \end{bmatrix}$ and $\mathbf{u} = \frac{1}{2}\begin{bmatrix} 1+\mathrm{i} \\ b \end{bmatrix}$. Solve for all possible values of a and b, assuming that $\mathbf{u}$ is a unit vector and $b \in \mathbb{R}$.

10.7 Matrices [return to section]

Problem 1: Using MATLAB or some other language of choice (e.g., Python, Julia, etc.), Construct an $N \times N$ uniform random matrix. In MATLAB, this can be done with the `rand(N)` function (i.e., real entries in a full (*non-sparse*) matrix). Use a timing function (e.g., MATLAB's `tic` and `toc` functions) to calculate the length of time required to invert the matrix. Now loop this process as a function of the size of the square $N \times N$ matrix, where N ranges from 100 to 1,000 and store the time required to invert each matrix in an array. Plot the time array as a function of N two different ways: using a linear scale for both x and y axes (a *linear-linear* plot via the `plot(x,y)` command if using MATLAB) and using a base-10 logarithmic scale on both axes (a *log-log* plot via the `loglog(x,y)` command if using MATLAB).
How does the inversion time scale with respect to N for each type of plot, and what does this say about the computational *complexity* of matrix inversion (that is, the amount of resources required to invert a matrix)? Based on the inversion timescale you determined from your numerical experiments, estimate the size of the largest $N \times N$ matrix you could invert in 1 minute or less.

Problem 2: Two identical antennas, labeled 1 and 2, are separated in free-space by distance L as shown below.



If the antennas receive an electromagnetic signal from source S_n that has angular frequency ω_n then, as a function of time t, a unit-amplitude source signal is $S_n(t) = e^{\mathrm{i}\omega_n t}$. If the nth signal $S_n(t)$ is a plane wave and has an angle of arrival θ_n measured anticlockwise from normal incidence then there is a relative phase difference of ϕ_n between the contribution of $S_n(t)$ arriving at antenna 1 and 2. In general the relationship between angle of arrival of θ_n of the nth signal and the phase difference ϕ_n is

$$\phi_n = \frac{2\pi L}{\lambda_n} \sin(\theta_n) \tag{168}$$

for a signal of wavelength $\lambda_n = 2\pi c/\omega_n$, where c is the speed of light. If there are only two plane-wave sources, $S_1(t)$ and $S_2(t)$, then each antenna receives the sum of the two signals and at any given time this sum may be written in matrix form as $\mathbf{X} = \mathbf{AS}$ where vector $\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$ describes signal $X_1(t)$ received at antenna 1 and signal $X_2(t)$ received at antenna 2, $\mathbf{S} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$ describes sources $S_1(t)$ and $S_2(t)$, and the time-independent complex mixing matrix is $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$.

(a) Find the matrix elements of the mixing matrix, $\mathbf{A}$.
(b) Find the inverse matrix $\mathbf{A}^{-1}$ and find the conditions when it is *not* possible to separate the source signals using $\mathbf{S} = \mathbf{A}^{-1}\mathbf{X}$.
(c) In a typical wireless receiver system implementation the complex signals are separated into their real (in-phase, I) and imaginary (quadrature, Q) components at each antenna relative to a reference local oscillator. This doubles the size of the mixing matrix. Find the matrix elements for the mixing matrix in this case.
(d) Can an RF receiver be used to directly measure the electromagnetic field?

10.8 Beam splitter numerical error **[return to section]**

Problem 1: Consider an ideal, lossless, symmetric, 50:50 beam splitter with a total number of n_{tot} identical and indistinguishable photons.

(a) Run MATLAB script `Chapt11Fig7.m` for $n_{\text{tot}} = 8$ and $n_{\text{tot}} = 110$. Compare and explain the results.

(b) Calculate the average deviation from *zero* for probabilities of observing *odd* numbers of photons exiting output port 3 or 4 when $n_1 = n_{\text{tot}}/2$. Range the total input photon number parameter $12 \leq n_{\text{tot}} \leq 128$ where n_{tot} is even, and plot the average error (deviation from zero) on a logarithmic scale as a function of n_{tot} value on a linear scale. Comment on how this error varies as a function of n_{tot}.

(c) Repeat part (b), but now use single precision to compute the probability amplitude. How does this result differ from double precision?

(d) Using Mathematica and table 3 as a guide, derive simplified analytic expressions of Eq. (11.31) which can be used to accurately calculate the photon detection probabilities in output port 3 for the two extremal cases where $n_1 = n_{\text{tot}}$ and $n_1 = n_{\text{tot}}/2$. Use these simplified expressions to plot the probability distributions for the two aforementioned cases when $n_{\text{tot}} = 1000$ and comment on the results.