# Modeling and Simulation Lecture Notes - 4

June 2023

## Building Mathematical Models: Differential Equations for Time-Continuous Systems

Mathematical models are fundamental tools for understanding and analyzing the behavior of systems in fields as diverse as physics, engineering, biology, and social sciences. Often, these models are expressed in the form of differential equations.

### Time-Continuous Systems

A time-continuous system is a system where the variables change with time. This is in contrast to discrete systems where the state of the system changes at specific time intervals. In time-continuous systems, differential equations are typically used to describe the system's behavior.

A differential equation is an equation that includes the derivatives of a function. The order of the highest derivative present in the equation defines the order of the differential equation. First order differential equations can be solved analytically with basic techniques, while higher order equations might require more advanced methods.

### Practical Applications in Electronics

In electronics, differential equations can model the behavior of circuits. Consider the simple RC (Resistor-Capacitor) circuit. The current flowing through the circuit over time can be modeled by the first-order differential equation $V' = -\frac{1}{RC}V$, where $V$ is the voltage across the capacitor and $RC$ is the time constant of the circuit.

### Practical Applications in Mechanics

In mechanics, differential equations are used to model a variety of phenomena, including the motion of pendulums, the vibration of strings, or the flight of projectiles. For example, the motion of a simple pendulum can be described by the second-order differential equation $\theta'' + \frac{g}{L} \sin\theta = 0$, where $\theta$ is the pendulum's angular displacement, $L$ is the pendulum's length, and $g$ is the acceleration due to gravity.

## Examples of Continuous Systems

**Example 1: Population growth**
A fundamental model of population growth assumes that the rate of change of the population $P$ over time is proportional to the current population. This can be written as a first order differential equation:

$$\frac{dP}{dt} = rP$$

where $r$ is the constant of proportionality that represents the rate of growth. If we solve this equation, we find that the population grows exponentially over time, which is a common pattern in nature.

**Example 2: RL Circuit**
An RL circuit consists of a resistor and inductor in series. If we apply Kirchhoff's voltage law to this circuit, we get a first order differential equation:

$$L\frac{di}{dt} + Ri = V$$

where $V$ is the applied voltage, $R$ is the resistance, $L$ is the inductance, and $i$ is the current through the circuit. Solving this equation gives us insight into how the current changes over time in response to a step in voltage, among other things.

## Importance of Continuous Systems in Modeling

Many systems in nature and technology are inherently continuous. For instance, populations change continuously over time, voltages and currents in an electrical circuit vary continuously, and physical quantities like velocity and displacement are also continuous. Therefore, we often need to use differential equations to accurately model and understand these systems.

From a mathematical perspective, continuous models often capture the fundamental laws of nature more directly than discrete models. For instance, Newton's second law of motion, Maxwell's equations of electromagnetism, and the Schrödinger equation of quantum mechanics are all inherently continuous. When we use a discrete model, it is often an approximation (sometimes a very good one) to an underlying continuous model.

## Solving Mathematical Models with SciPy

The SciPy library in Python provides powerful tools for solving differential equations, which often arise in the mathematical modeling of continuous systems.

For example, the following Python code shows how we can use ‘odeint‘ to solve the population growth model:

```
from scipy.integrate import odeint

def model(P, t):
    r = 0.01  # growth rate
    dPdt = r * P
    return dPdt

P0 = 1000  # initial population
t = np.linspace(0, 100, 100)  # time grid

# solve the differential equation
P = odeint(model, P0, t)
```

This code defines a function model that implements the differential equation, then uses odeint to solve this equation over a grid of time points.

## Analogies between Different Physical Systems

Understanding the analogies between different physical systems can be highly beneficial, especially when it comes to problem-solving and design in engineering. For example, the mathematical forms of the differential equations describing an RL circuit (from electrical engineering) and a mass-spring system (from mechanical engineering) are identical.

The voltage across an inductor in an RL circuit corresponds to the velocity of a mass in a mass-spring system. Similarly, the current through the inductor corresponds to the displacement of the mass. Recognizing these analogies allows us to apply intuition and problem-solving techniques from one domain to problems in another domain.

This understanding can also make it easier to learn new material. For instance, a student who understands mechanical systems might find it easier to learn about electrical systems by drawing analogies between the two.

In the end, the understanding of differential equations and their solutions, the recognition of analogies between different physical systems, and the ability to implement and solve these systems in Python using libraries like SciPy, form a strong foundation for the understanding and application of mathematical modeling and simulation in engineering.

| | **RL Circuit** | **Mass-Spring System** |
|---|---|---|
| **Displacement-like quantity** | Charge, $Q$ | Displacement, $x$ |
| **Inertial-like element** | Inductor, $L$ | Mass, $m$ |
| **Force-like quantity** | Voltage, $V$ | Force, $F$ |
| **Velocity-like quantity** | Current, $I = \frac{dQ}{dt}$ | Velocity, $v = \frac{dx}{dt}$ |
| **Dissipative element** | Resistor, $R$ | Damping coefficient, $b$ |
| **Dissipative force** | Ohm's Law, $V_R = I \cdot R$ | Damping force, $F_d = b \cdot v$ |
| **Conservative force** | Faraday's Law, $V_L = L\frac{dI}{dt}$ | Hooke's Law, $F_s = k \cdot x$ |
| **Differential Equation** | $\frac{dI}{dt} + \frac{R}{L}I = \frac{V}{L}$ | $m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = 0$ |

Table 1: Analogies between RL Circuit and Mass-Spring System

# Laplace Transforms with Python

The Laplace Transform is a powerful mathematical technique used to simplify the solving of differential equations. It transforms complex time-domain equations into simpler s-domain equations, which can be easily solved.

$$F(s) = \mathcal{L}f(t) = \int_0^\infty f(t)e^{-st}dt \tag{1}$$

where:

$f(t)$ is a given function of time $F(s)$ is the Laplace transform of $f(t)$ $s$ is a complex number.

## Application of Laplace Transforms to Solve Differential Equations

By converting a differential equation into the s-domain using the Laplace transform, we can turn the calculus problem into an algebra problem. Let's consider a first-order differential equation:

$$\frac{dy(t)}{dt} + ay(t) = 0 \tag{2}$$

Taking the Laplace transform of both sides:

$$sY(s) - y(0) + aY(s) = 0 \tag{3}$$

Solving for $Y(s)$ gives the solution in the frequency domain:

$$Y(s) = \frac{y(0)}{s + a} \tag{4}$$

Taking the inverse Laplace transform, we can obtain the solution in the time domain.

The Laplace transform is a powerful tool, especially in the field of engineering, because it can simplify the process of dealing with differential equations. It converts differential equations, which can be challenging to solve directly, into algebraic equations in the s-domain, which are often much simpler to handle.

The Laplace transform has a few key advantages:

Converting Differential Equations into Algebraic Equations: The calculus involved in differential equations can become very complex very quickly, especially with higher-order differential equations. When you apply the Laplace transform to a differential equation, the differential terms (derivatives) are transformed into multiplication operations in the s-domain. This conversion often results in algebraic equations that are easier to solve. Once we have the solution in the s-domain, we can use the inverse Laplace transform to get the solution in the time-domain.

Handling Initial Conditions Naturally: In many physical problems, we have to solve differential equations with given initial conditions. The Laplace transform incorporates these initial conditions in the transformation process, and so the solutions automatically satisfy the initial conditions.

Solving Linear Time-Invariant (LTI) Systems: Many systems in engineering are modeled as LTI systems. The Laplace transform, together with the transfer function representation of a system, provides a very convenient way to analyze LTI systems.

Solving Integral Equations: Some problems are naturally expressed as integral equations. The Laplace transform can simplify the solution of these equations too.

Convolution Theorem: The convolution theorem says that under certain conditions the Fourier transform of a convolution of two signals is the pointwise product of their Fourier transforms. In other words, convolution in one domain (e.g., time domain) equals multiplication in the other domain (e.g., frequency domain). This property is very useful in system analysis and differential equations.

**Example:**

Consider the differential equation:

$$y''(t) + 3y'(t) + 2y(t) = e^t \tag{5}$$

with initial conditions $y(0) = 0$, and $y'(0) = 0$. Here, the prime notation (') denotes a derivative with respect to $t$.

The Laplace transform of the function $y(t)$ is defined as

$$Y(s) = \mathcal{L}\{y(t)\} = \int_0^\infty y(t)e^{-st}dt \tag{6}$$

For the Laplace transform of the derivative of a function, the following property holds:

$$\mathcal{L}\{y'(t)\} = sY(s) - y(0) \tag{7}$$

For the second derivative, the property is:

$$\mathcal{L}\{y''(t)\} = s^2Y(s) - sy(0) - y'(0) \tag{8}$$

Now we apply the Laplace transform to both sides of the differential equation. The left side becomes:

$$\mathcal{L}\{y''(t)+3y'(t)+2y(t)\} = s^2Y(s)-sy(0)-y'(0)+3[sY(s)-y(0)]+2Y(s) \tag{9}$$

Substituting the initial conditions $y(0) = 0$, and $y'(0) = 0$, we obtain:

$$s^2Y(s) + 3sY(s) + 2Y(s) \tag{10}$$

For the right side of the equation, the Laplace transform of $e^t$ is $1/(s-1)$ (which can be found in standard Laplace transform tables), thus:

$$\mathcal{L}\{e^t\} = \frac{1}{s-1} \tag{11}$$

Therefore, the transformed equation is:

$$s^2Y(s) + 3sY(s) + 2Y(s) = \frac{1}{s-1} \tag{12}$$

This equation can be rearranged to solve for $Y(s)$:

$$Y(s) = \frac{1}{(s-1)(s^2+3s+2)} \tag{13}$$

The next step would involve finding the inverse Laplace transform to obtain $y(t)$, which is done using either standard Laplace transform tables or a computer algebra system.

**Example:**
Consider the differential equation:

$$y''(t) - 2y'(t) + y(t) = e^{2t} \tag{14}$$

With initial conditions $y(0) = 1$, and $y'(0) = 0$.

Taking the Laplace transform of both sides of the differential equation and using the properties of the Laplace transform we obtain:

$$s^2Y(s) - sy(0) - y'(0) - 2sY(s) + 2y(0) + Y(s) = \frac{1}{s-2} \tag{15}$$

Substituting the initial conditions $y(0) = 1$, $y'(0) = 0$ and simplifying, we get:

$$Y(s) = \frac{s+1}{(s-1)^2+1} \tag{16}$$

This function can be inverted using standard Laplace transform tables or a computer algebra system to obtain the solution in the time domain.

The Laplace transform method allows us to solve this differential equation algebraically, which can be easier than solving it in the time domain, especially for more complex differential equations.

This example shows the power of the Laplace transform as a tool for solving differential equations, especially when the equations are complicated or have non-zero initial conditions.

**Example:**

In the field of classical mechanics, a common example that could benefit from Laplace transform is the driven harmonic oscillator. This physical situation can be described by the second-order differential equation:

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = F(t)$$

where:

- $m$ is the mass of the object,
- $x$ is the position of the object,
- $b$ is the damping coefficient,
- $k$ is the spring constant, and
- $F(t)$ is the external force acting on the object.

The Laplace transform can be used to simplify the process of solving this differential equation, especially when $F(t)$ is a piecewise function or has a complex form.

Firstly, we take the Laplace transform of both sides:

$$L\{m\frac{d^2x}{dt^2}\} + L\{b\frac{dx}{dt}\} + L\{kx\} = L\{F(t)\}$$

If we take the Laplace transform of a derivative, we get:

$$L\left\{\frac{d}{dt}f(t)\right\} = sF(s) - f(0)$$

where F(s) is the Laplace transform of f(t).

If we take the Laplace transform of a second derivative, we get:

$$L\left\{\frac{d^2}{dt^2}f(t)\right\} = s^2F(s) - sf(0) - f'(0)$$

Using the linearity of the Laplace transform and the known transforms of derivatives, we get:

$$m[s^2X(s) - sx(0) - x'(0)] + b[sX(s) - x(0)] + kX(s) = F(s)$$

We will assume that the initial conditions $x(0) = x_0$ and $x'(0) = v_0$ are given. The equation can then be rearranged to solve for $X(s)$, the Laplace transform of $x(t)$:

$$X(s) = \frac{F(s) + msx(0) + mx'(0) - bsx(0)}{ms^2 + bs + k}$$

Now, $X(s)$ can be found if $F(s)$, the Laplace transform of the forcing function, is known. Once $X(s)$ is found, the inverse Laplace transform can be taken to find $x(t)$, the solution to the original differential equation.

**Example:**

Consider an RC circuit where we want to find the voltage across the capacitor, $V_C(t)$, in response to a step input voltage, $V_s(t)$. The differential equation for this system is:

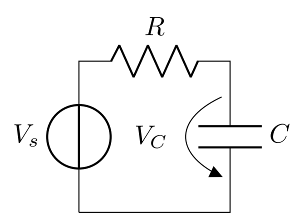
$$V_C'(t) + \frac{1}{RC}V_C(t) = \frac{1}{RC}V_s(t) \tag{17}$$

Let's assume $V_s(t) = u(t)$, where $u(t)$ is the unit step function.

We can solve this problem by first taking the Laplace transform of the differential equation, then solving for $V_C(s)$, and finally taking the inverse Laplace transform to get $V_C(t)$. This can be implemented in Python using the scipy library.

**Example: Solving a first-order RC circuit**

Consider the first order RC circuit below:




The governing differential equation for this circuit is:

$$RC\frac{dV_C}{dt} + V_C = V_s \tag{18}$$

We can solve this using Laplace transforms. Taking the Laplace transform of both sides and using the fact that $\mathcal{L}df/dt = sF(s) - f(0)$, we get:

$$RCsV_C(s) + V_C(s) = \frac{1}{s} \tag{19}$$

Assuming that the initial voltage across the capacitor is zero, this simplifies to:

$$V_C(s) = \frac{1}{s(RCs+1)} \tag{20}$$

The solution in the time domain can be found by taking the inverse Laplace transform.

## Transfer Functions, Poles, Zeros, and Limit Value Sets

To begin with, let's understand why these concepts are important in modeling and simulation.

Modeling involves the construction of an abstract representation (a model) to represent real-world or theoretical situations. Simulation, on the other hand, is the process of using a model to study the behavior and performance of an actual or theoretical system. Both modeling and simulation are crucial in various fields like physics, engineering, statistics, economics, etc., as they allow for the prediction of behavior without testing on the actual system.

For example, consider designing a bridge. A model of the bridge can be created and various simulations run to see how it would respond to different stresses and strains, without the need to actually build and test the bridge, which would be both costly and time-consuming.

The **Transfer Function** is a mathematical representation that we use to relate the output or response of a system to the input or the cause. It plays a crucial role in the analysis and design of systems in many branches of engineering. The concept of a transfer function is a fundamental building block in control theory and signal processing.

From a mathematical perspective, the transfer function of a system is the Laplace transform of the system's impulse response, divided by the Laplace transform of the input signal (assuming zero initial conditions). It is a complex function of a complex variable, which encodes a lot of information about the system's dynamics.

Let's understand why we need the **Poles and Zeros**. When analyzing or designing a system, it's essential to understand how the system behaves over time or how it responds to different inputs. Differential equations describe the dynamics of systems, but they can be challenging to solve or interpret.

This is where the concepts of poles and zeros come in. Poles and zeros provide a method to analyze system dynamics without solving differential equations completely. The locations of the poles and zeros in the complex plane can tell us a lot about the system's stability and frequency response, which are vital characteristics of any system.

For instance, poles allow us to determine the stability of a system. If all poles of a system lie in the left half of the complex plane (their real parts are negative), the system response decays over time, and the system is stable. However, if any pole has a positive real part, the system response grows over time, leading to an unstable system. The zeros of a transfer function, on the other hand, indicate the frequencies that the system attenuates.

Finally, the concept of the **Limit Value Set** is crucial in understanding the potential behavior of the system under all possible inputs. It helps us understand the capabilities and limitations of a system. In simple terms, the limit value set is the set of all possible outputs of the system. It's like looking at the "boundary" of what the system can do.

All these concepts come together in the analysis, design, and simulation of systems. They provide tools to understand the behavior of systems and make them easier to analyze and predict, which is crucial in many fields of engineering and science.

### Impulse Response Function

In control theory, signal processing, and related fields, the **impulse response** of a system is essentially its output when presented with a very short input signal called an impulse. More technically, the impulse response is the output resulting from the system when an input is a Dirac delta function.

The Dirac delta function is a mathematical construct which is used to model an impulse. It is zero everywhere except at zero, and has an 'area' of one, symbolically represented as:

$$\delta(t) = \begin{cases} 0, & \text{if } t \neq 0 \\ \infty, & \text{if } t = 0 \end{cases} \tag{21}$$

with

$$\int_{-\infty}^{+\infty} \delta(t)dt = 1 \tag{22}$$

If we denote the impulse response of a system as $h(t)$, then for an input impulse at time $t = 0$, the output of the system would be the impulse response $h(t)$.

**Now, why is the impulse response important?**

The importance of the impulse response stems from an integral theorem known as the **convolution theorem**. This theorem states that the response of a linear, time-invariant system (a common assumption for many systems we deal with in engineering and physics) to any input can be found by convolving the input signal with the system's impulse response.

In other words, if you know a system's impulse response, you can predict the system's output for any arbitrary input!

Mathematically, if $x(t)$ is an arbitrary input to the system and $y(t)$ is the output, this relationship can be written as:

$$y(t) = (h * x)(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau \tag{23}$$

This equation is known as the **convolution integral**, where the asterisk denotes convolution.

Hence, the impulse response function is a critical tool in system analysis and understanding the behavior of a system. It's a bridge between the input and output of a system, and hence, fundamental to the transfer function concept.

### Transfer Function

The **transfer function** of a linear, time-invariant system with input $x(t)$ and output $y(t)$ is the Laplace transform of the impulse response of the system.

Assuming all initial conditions are zero, the transfer function $H(s)$ can be written as:

$$H(s) = \frac{Y(s)}{X(s)} \tag{24}$$

where $Y(s)$ is the Laplace transform of the output $y(t)$, and $X(s)$ is the Laplace transform of the input $x(t)$.

The transfer function describes the response of the system for each possible exponential input $e^{st}$. The complex variable $s = \sigma + j\omega$ corresponds to an exponential input at the rate of $s$.

In many systems, the transfer function can be represented as a **ratio of polynomials** in $s$:

$$H(s) = \frac{b_n s^n + b_{n-1}s^{n-1} + \ldots + b_1 s + b_0}{a_m s^m + a_{m-1}s^{m-1} + \ldots + a_1 s + a_0} \tag{25}$$

Here, the $a_i$'s and $b_i$'s are the coefficients of the polynomials in the numerator and denominator of $H(s)$, respectively.

The roots of the numerator are called the **zeros** of the transfer function, and the roots of the denominator are called the **poles**.

Zeros and poles are important because they allow us to analyze and understand the behavior of the system without solving the complete differential equation. They give us information about the stability and the frequency response of the system:

- **Poles**: When the denominator of the transfer function is zero, the overall transfer function goes to infinity. These points are called poles. If any pole is located in the right half of the complex plane (real part $> 0$), the system is unstable.

- **Zeros**: These are the values of $s$ that make the transfer function go to zero. They represent the natural response of the system to decay away.

The set of all possible outputs of the system as the input varies over all possible inputs is called the **limit value set** of the system. This concept can help us understand the potential behavior of the system, even for inputs we haven't specifically considered.

In sum, the transfer function and its poles and zeros give us a powerful tool to analyze the behavior of a system in the frequency domain. This is very useful in control theory, communication systems, signal processing, and many other fields.

Let's look at an example to understand this better. Suppose we have a system with the following transfer function:

$$H(s) = \frac{s+2}{s^2+s-2} \tag{26}$$

In this case, the system has one zero at $s = -2$ and two poles at $s = 1$ and $s = -2$. The system would be unstable because there is a pole in the right half of the complex plane (s = 1).

We can also look at the frequency response of the system (i.e., the transfer function evaluated at $s = j\omega$, with $\omega$ being the frequency) to understand how the system will respond to different frequency inputs. The frequency response can tell us whether the system will amplify or attenuate signals at different frequencies, and whether it will introduce any phase shift between the input and output signals.

### Example: First Order Low-Pass Filter

A common example of a first order low-pass filter is the RC (resistor-capacitor) circuit. The differential equation describing this system is:

$$RC\frac{dV_{out}(t)}{dt} + V_{out}(t) = V_{in}(t) \tag{27}$$

where:

- $V_{in}(t)$ is the input voltage,

- $V_{out}(t)$ is the output voltage,

- $R$ is the resistance, and
- $C$ is the capacitance.

The process of finding the impulse response for a system involves solving the given differential equation with the input set as the Dirac delta function.

The impulse response $h(t)$ of this system can be found by solving the above differential equation with the input $V_{in}(t) = \delta(t)$, where $\delta(t)$ is the Dirac delta function.

We find that:

$$h(t) = \frac{1}{RC} e^{-\frac{t}{RC}} u(t) \tag{28}$$

where $u(t)$ is the unit step function.

The Laplace Transform of the impulse response $h(t)$ is the transfer function $H(s)$:

$$H(s) = \frac{1}{RCs + 1} \tag{29}$$

This shows a pole at $s = -1/RC$ and no zeros.

In Python, you could find the impulse response numerically. Here's an example code snippet:

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from scipy import signal

# define the system of ODEs
def RC_low_pass(y, t, R, C):
    Vin = 0 if t < 0 else 1
    return (Vin - y) / (R*C)

# parameters
R = 1.0  # resistance
C = 1.0  # capacitance

# initial conditions
Vout0 = 0.0 # initial output voltage

# time points
t = np.linspace(-1, 10, 1000)
```

```
# solve ODE
Vout = odeint(RC_low_pass, Vout0, t, args=(R, C))

# plot impulse response
plt.figure()
plt.plot(t, Vout)
plt.xlabel('Time')
plt.ylabel('Vout(t)')
plt.title('Impulse response of the RC circuit')
plt.grid(True)

# define the transfer function
s1 = signal.lti([1], [R*C, 1])

# calculate frequency response
w, mag, phase = signal.bode(s1)

# plot magnitude response
plt.figure()
plt.semilogx(w, mag)    # Bode magnitude plot
plt.xlabel('Frequency [Hz]')
plt.ylabel('Gain [dB]')
plt.title('Frequency response of the RC circuit')
plt.grid(True)

# plot phase response
plt.figure()
plt.semilogx(w, phase)  # Bode phase plot
plt.xlabel('Frequency [Hz]')
plt.ylabel('Phase [degrees]')
plt.grid(True)

plt.show()
```

**Question:**

Consider a simple RL series circuit where the inductor (L) and the resistor (R) have values of 0.1 H and 100 Ohm, respectively. The circuit is driven by an input current that changes abruptly from zero to one (an impulse). Calculate the current through the inductor over time (the impulse response) in the time domain.

**Solution:**

The impulse response of the RL circuit can be found by first taking the Laplace Transform of the differential equation describing the circuit:

$$L\frac{di}{dt} + Ri = \delta(t) \tag{30}$$

This yields the following in the frequency (s) domain:

$$LsI(s) + RI(s) = 1 \tag{31}$$

Solving for $I(s)$ gives:

$$I(s) = \frac{1}{Ls + R} \tag{32}$$

This is the transfer function of the system. To find the impulse response, we need to calculate the inverse Laplace Transform of this transfer function.

Using Sympy, we find that:

$$i(t) = e^{-t/R}u(t) \tag{33}$$

This is the impulse response of the RL circuit in the time domain. It describes how the current through the inductor evolves over time after the impulse at time zero. The step function $u(t)$ ensures that the response is zero for $t < 0$.

```
import sympy as sp

t = sp.symbols('t')
s = sp.symbols('s')

# parameters
R = 100  # resistance
L = 0.1  # inductance

# transfer function
H = 1 / (s*L + R)

# impulse response
h = sp.inverse_laplace_transform(H, s, t)

print(h)
```

**Example: Simple Harmonic Oscillator**

A simple harmonic oscillator, such as a mass on a spring, can be modeled by the following differential equation:

$$m\frac{d^2x}{dt^2} + kx = f(t) \tag{34}$$

where:

- $x(t)$ is the displacement of the mass,
- $f(t)$ is the external force,
- $m$ is the mass, and
- $k$ is the spring constant.

If we apply an impulse $f(t) = \delta(t)$, the system will start to oscillate. The solution to this equation with this input is the impulse response of the system.

$$h(t) = \frac{1}{m\omega} e^{-\frac{t}{2\zeta\omega}} sin(\omega t) u(t) \tag{35}$$

where $\omega = \sqrt{k/m}$ is the natural frequency of the oscillator and $\zeta = b/(2\sqrt{km})$ is the damping ratio (assuming a damping term $b\frac{dx}{dt}$ is present in the differential equation).

The transfer function $H(s)$ can be found by taking the Laplace transform of $h(t)$:

$$H(s) = \frac{1}{ms^2 + ks} \tag{36}$$

This shows two poles at $s = \pm j\omega$ and no zeros.

**How did we find the impulse response?**
The impulse response of a system is the output when the input is a Dirac delta function, $f(t) = \delta(t)$.

For the simple harmonic oscillator, this means we solve the following differential equation:

$$m\frac{d^2x}{dt^2} + kx = \delta(t) \tag{37}$$

Since $\delta(t)$ is zero everywhere except $t = 0$, the solution will be the same as the homogeneous solution, $x(t) = A\cos(\omega t) + B\sin(\omega t)$, for $t \neq 0$.

The constants $A$ and $B$ are found using the initial conditions. The impulse at $t = 0$ imparts a velocity, but not a displacement, so $x(0) = 0$. The velocity at $t = 0$ is found by integrating the impulse, $\int_0^\epsilon \delta(t)dt = 1$, so $\dot{x}(0) = 1/m$.
Plugging these into our homogeneous solution gives us $A = 0$ and $B = 1/(m\omega)$, so our impulse response is:

$$h(t) = \frac{1}{m\omega}\sin(\omega t)u(t) \tag{38}$$

where $u(t)$ is the unit step function, which is zero for $t < 0$ and one for $t \geq 0$, ensuring that $h(t)$ is also zero for $t < 0$.

**How did we find the transfer function H(s)?**

The transfer function is the Laplace transform of the impulse response. So we take the Laplace transform of $h(t)$:

$$H(s) = \mathcal{L}\{h(t)\} = \frac{1}{m\omega}\mathcal{L}\{\sin(\omega t)u(t)\} = \frac{1}{m\omega} \cdot \frac{\omega}{s^2 + \omega^2} = \frac{1}{ms^2 + k} \tag{39}$$

The Laplace transform of a function is defined as:

$$\mathcal{L}\{f(t)\} = \int_0^\infty e^{-st} f(t)dt \tag{40}$$

For the Laplace transform of a sine function, we have a standard result:

$$\mathcal{L}\{\sin(at)\} = \frac{a}{s^2 + a^2} \tag{41}$$

Applying this result to our function $h(t)$ gives us:

$$\mathcal{L}\{h(t)\} = \frac{1}{m\omega}\mathcal{L}\{\sin(\omega t)u(t)\} = \frac{1}{m\omega} \cdot \frac{\omega}{s^2 + \omega^2} \tag{42}$$

Simplifying this expression gives us:

$$H(s) = \frac{1}{ms^2 + k} \tag{43}$$

Where we used the fact that $\omega^2 = k/m$.

So the transfer function $H(s)$ of the simple harmonic oscillator is:

$$H(s) = \frac{1}{ms^2 + k} \tag{44}$$

This transfer function has poles where the denominator is zero, or $s = \pm j\omega$, which are on the imaginary axis in the complex plane, reflecting the fact that the system oscillates forever when given an impulse. There are no zeros, as the numerator is a constant.

**Example: A simple RC circuit**

Consider a simple RC (Resistor-Capacitor) circuit where a resistor with resistance $R$ and a capacitor with capacitance $C$ are connected in series to a voltage source $V_{in}(t)$. The output voltage is the voltage across the capacitor $V_{out}(t)$.

The governing differential equation for this system can be derived using Kirchhoff's voltage law and is given by:

$$V_{in}(t) = R\frac{dI}{dt} + \frac{1}{C}\int_0^t I(\tau)d\tau$$

Taking the Laplace Transform and solving for $V_{out}(s) = \frac{1}{C}\frac{1}{s}I(s)$ we get the transfer function:

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{RCs + 1}$$

The transfer function has one pole and no zeros.

The pole is obtained by setting the denominator of the transfer function equal to zero and solving for $s$:

$$RCs + 1 = 0 \Rightarrow s = -\frac{1}{RC}$$

Since the pole is in the left-half plane, the system is stable.

**Limit Value Set**

The limit value set of a system is the set of all possible outputs as the input ranges over all possible inputs.

In the case of a linear time-invariant (LTI) system, if the system is stable (i.e., all the poles of the transfer function are in the left half of the complex plane), then the limit value set of the system is the entire space of finite-energy signals.

In the case of our RC circuit, the system is stable and thus the limit value set is the set of all finite-energy signals.

**Practical Applications of Laplace Transforms**

The Laplace Transform is a powerful tool for solving differential equations, such as the one governing our RC circuit.

By transforming the differential equation into an algebraic equation, we were able to solve for the output $V_{out}(s)$ in terms of the input $V_{in}(s)$ in the complex frequency domain.

This allows us to analyze the system's behavior for different inputs without having to solve the differential equation for each one.

Moreover, we could determine the system's poles, which tell us about the system's stability, and the limit value set, which tells us about the possible outputs of the system.

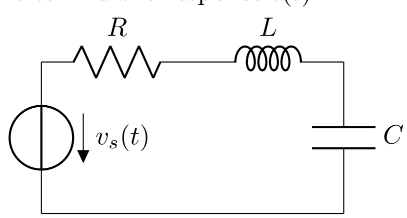
Finally, by taking the inverse Laplace Transform of $V_{out}(s)$, we could find the output $V_{out}(t)$ in the time domain for a given input $V_{in}(t)$.

**Example:**

A simple series RLC circuit consists of a resistor (R), an inductor (L), and a capacitor (C) connected in series. The governing equation for this circuit is given by the second order differential equation:

$$L\frac{d^2i}{dt^2} + R\frac{di}{dt} + \frac{1}{C}i = v_s(t) \tag{45}$$

Suppose that we have a unit step function as input, $V_s(t) = u(t)$, and we want to find the response $i(t)$.




In order to solve this, we will first take the Laplace transform of the equation, then solve for $I(s)$, and finally take the inverse Laplace transform to find $i(t)$.

$$\begin{aligned}
Ls^2I(s) + RsI(s) + \frac{1}{C}I(s) &= \frac{1}{s} \\
(s^2LC + sRC + 1)I(s) &= \frac{1}{s} \\
I(s) &= \frac{1}{s(s^2LC + sRC + 1)}
\end{aligned}$$

Now, we will decompose this into partial fractions and find the inverse Laplace transform:

$$\begin{aligned}
I(s) &= \frac{1}{s} - \frac{s + \frac{R}{L}}{s^2 + \frac{R}{L}s + \frac{1}{LC}} \\
i(t) &= \mathcal{L}^{-1}\{I(s)\} = 1 - e^{-\frac{R}{2L}t}\left[\cos\left(\frac{\sqrt{4LC - R^2}}{2L}t\right) + \frac{R}{\sqrt{4LC - R^2}}\sin\left(\frac{\sqrt{4LC - R^2}}{2L}t\right)\right]
\end{aligned}$$

The response of the circuit to a unit step input is given by:

$$i(t) = 1 - e^{-\frac{R}{2L}t}\left[\cos\left(\frac{\sqrt{4LC - R^2}}{2L}t\right) + \frac{R}{\sqrt{4LC - R^2}}\sin\left(\frac{\sqrt{4LC - R^2}}{2L}t\right)\right] \tag{46}$$

The second term represents a damped oscillation, with the damping factor and frequency determined by the values of $R$, $L$, and $C$.

The poles of the transfer function $H(s) = I(s)/(1/s)$ are the roots of the denominator $s^2LC + sRC + 1$, which are:

$$-\frac{R}{2L} \pm j\frac{\sqrt{4LC - R^2}}{2L} \tag{47}$$

These are complex conjugate poles, indicating that the system will oscillate. The oscillation will be damped if $R > 0$, and the system will be stable as long as $LC > R^2/4$. If $LC = R^2/4$, the system will be critically damped, and if $LC < R^2/4$, the system will be overdamped. The system has a zero at $s = 0$.

This example illustrates how Laplace transforms can be used to analyze and solve electrical circuits, and how the poles and zeros of the transfer function relate to the system's response.

Here's a Python script which calculates the response, plots it, and also calculates the poles of the system:

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as signal
import cmath

# Define parameters
R = 1.0
L = 1.0
C = 1.0

# Define system
num = [1]
den = [L*C, R*C, 1]
system = signal.lti(num, den)

# Time array
t = np.linspace(0, 10, 1000)

# Step response
t, y = signal.step(system, T=t)

# Poles and zeros
```

```
poles, zeros, gain = signal.tf2zpk(num, den)

# Plot response
plt.figure(figsize=(12,6))
plt.plot(t, y)
plt.title('Step response of RLC circuit')
plt.xlabel('Time [s]')
plt.ylabel('Current [A]')
plt.grid(True)
plt.show()

# Print poles and zeros
for p in poles:
    print("Pole: ", p)

for z in zeros:
    print("Zero: ", z)
```

This script first defines the parameters of the system (R, L, and C) and the Laplace transform of the system. It then generates a time array from 0 to 10 seconds.

The step function of the signal package is used to calculate the step response of the system, which is then plotted.

Finally, the poles and zeros of the system are calculated and printed.