Discrete-Time Solution of Ordinary Differential Equations (ODEs)

Tuesday, January 6, 2026 12:14 PM

1. Introduction

Ordinary Differential Equations (ODEs) are fundamental in modelling dynamic systems in engineering and science. In digital control, signal processing, and numerical simulation, continuous-time models must be converted into discrete-time representations for implementation on digital platforms.

This document presents a clear and systematic explanation of commonly used numerical discretization methods for solving first-order ODEs:

1. Forward Euler method
2. Backward Euler method
3. Trapezoidal rule (also known as Tustin or Bilinear Transform)
4. Zero-Order Hold (ZOH) / Exponential Mapping (briefly referenced)

Before discussing the discrete implementations, essential stability concepts and the physical interpretation of numerical integration are reviewed.

2. Stability Considerations in Continuous and Discrete Domains

2.1 Continuous-Time Stability

Consider a continuous-time transfer function in the Laplace (s) domain:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1}{s+a}, \qquad a > 0$$

The pole of the system is located at $s = -a$, which lies in the left half of the s-plane. Hence, the system is stable. The corresponding homogeneous time-domain response is:

$y(t) = Ce^{-at}$,

Where C is a constant. Since the exponential term decays to zero as $t \to \infty$, the system response decays to zero with time, confirming the stability of the system.

2.2 Discrete-Time Stability

Now consider a discrete-time transfer function in the z-domain:

$$H(z) = \frac{Y(z)}{U(z)} = \frac{1}{z+b}$$

The system pole is located at $z = -b$. A discrete-time system is stable if all poles lie strictly inside the unit circle, i.e.,

$|z| < 1 \Rightarrow |-b| < 1$

For a homogeneous discrete system, the difference equation is:

$zY(z) = -bY(z) \ \Rightarrow\ y_{k+1} = -by_k$

If $|-b| > 1$, the magnitude of the output increases with each sampling instant, causing the response to grow unbounded and the system to become unstable. Therefore, for stability, all discrete-time poles must lie strictly within the unit circle i.e., $|z| < 1$.

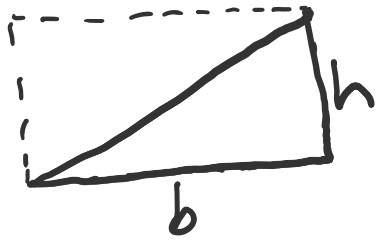
3. Geometric Interpretation of Numerical Integration

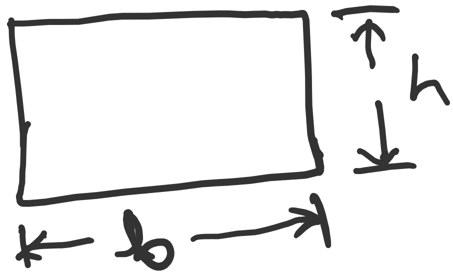
Numerical integration methods are fundamentally based on approximating the area under a curve.

- The area of a rectangle with base $\boldsymbol{b}$ and height $\boldsymbol{h}$ is $\boldsymbol{bh}$.

- The area of a triangle is $\frac{1}{2}\boldsymbol{bh}$.

These geometric principles form the basis for approximating integrals in discrete-time systems.





4. Physical Basis of ODE Solution Methods

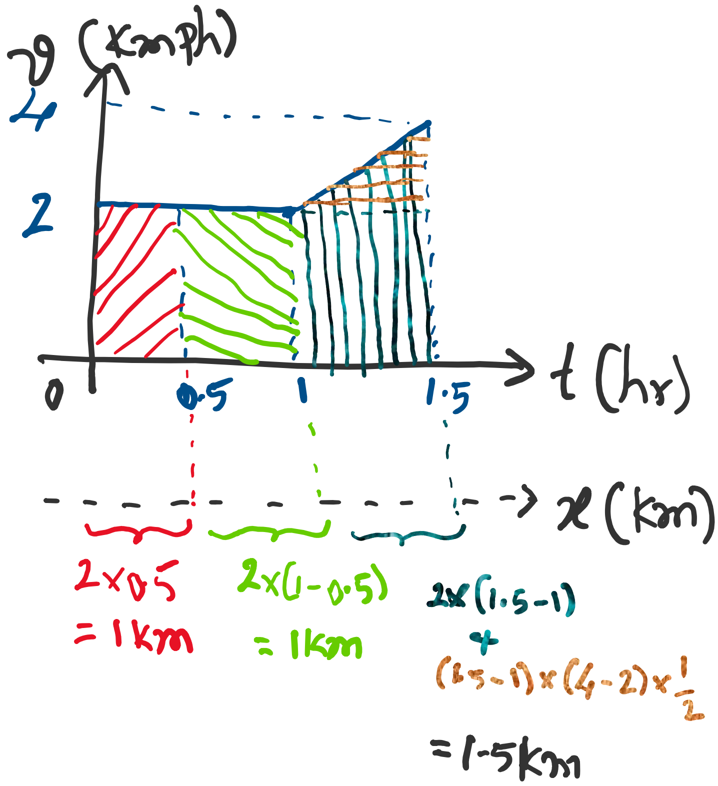
Velocity is defined as the rate of change of distance:

$$\frac{dx}{dt} = v \qquad (1)$$

The distance travelled over a time interval $[t_i, t_f]$ is equal to the area under the velocity-time curve during that interval.

$$x = \int_{t_i}^{t_f} v\, dt$$

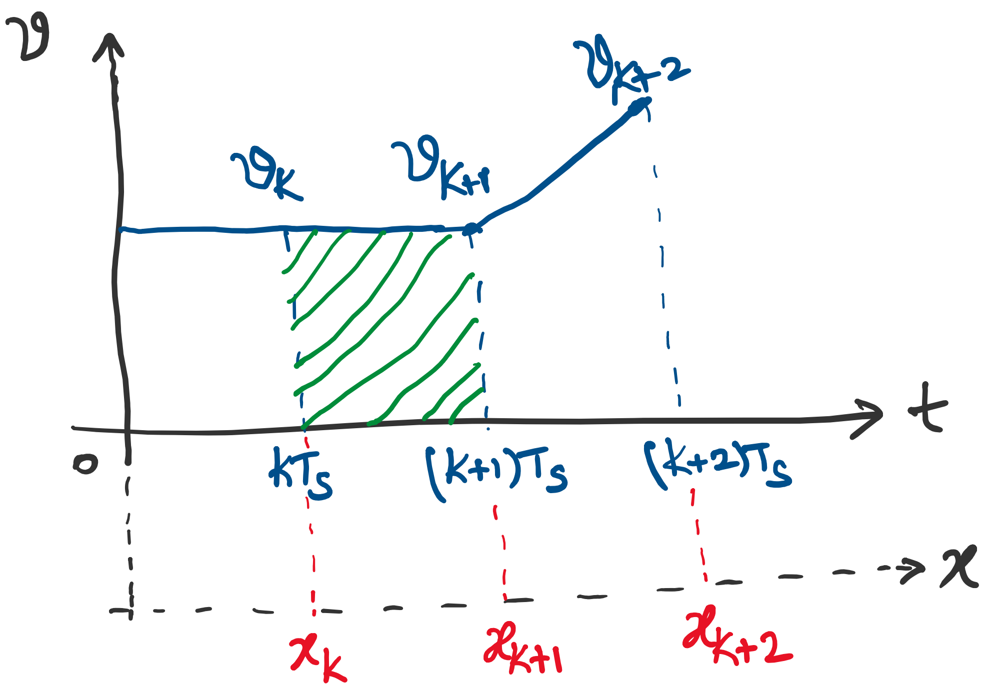
Consider a velocity–time plot, as shown below, in which the distance travelled is estimated at discrete intervals of half an hour. The distance covered during each interval is equal to the area under the velocity curve over that time period.



For a constant velocity of 2 km/h, the area under the curve over a half-hour interval is 1 km. If the same velocity is maintained for the next half hour, the total distance covered in one hour is therefore 2 km.

After one hour, the velocity increases linearly and reaches 4 km/h at 1.5 hours. Over this interval, the area under the velocity curve consists of a rectangle and a triangle. The sum of these areas corresponds to a distance of 1.5 km, indicating that the distance covered during the last half-hour interval is 1.5 km.

For generalization, let the sampling time be denoted as $T_s = 0.5$ hours, and let k represent the discrete-time sample index.



The objective is to compute the distance traveled between consecutive sampling instants kT_s and $(k+1)T_s$.

$x_{k+1} - x_k = \{(k+1)T_s \; - kT_s\} \, v_k \; = \; T_s v_k$ ------------> Forward Euler (2)

$x_{k+1} - x_k = \{(k+1)T_s \; - kT_s\} \, v_{k+1} = \; T_s v_{k+1}$ ------------> Backward Euler (3)

These two methods provide accurate results only when the velocity remains approximately constant over the sampling interval. When the velocity varies within the interval, both methods introduce approximation errors. To illustrate this, consider the area under the velocity curve between the time instants $(k+1)T_s$ and $(k+2)T_s$.

The Forward Euler method underestimates the actual area, whereas the Backward Euler method overestimates it. The exact area over this interval can be expressed as the sum of a rectangular and a triangular area:

$$x_{k+2} - x_{k+1} = \{(k+2)T_s \; - (k+1)T_s\} \, v_{k+1} \;\; + \frac{1}{2} \{(k+2)T_s \; - (k+1)T_s\} \, [v_{k+2} \; - \; v_{k+1}]$$

$$x_{k+2} - x_{k+1} = T_s * \tfrac{1}{2} \, [v_{k+1} + \; v_{k+2}]$$

$x_{k+1} - x_k = T_s * \frac{[v_k + v_{k+1}]}{2}$ ------------> Tustin (4)

5. Discrete Transfer Function Representation

The Laplace-domain representation of equation (1) is:

$$\frac{X(s)}{V(s)} = \frac{1}{s} \quad (5)$$

The corresponding z-domain transfer functions for the numerical methods, derived from (2)-(4), are given by:

- **Forward Euler**: $\frac{X(z)}{V(z)} = \frac{T_S}{z-1}$ (6)
- **Backward Euler**: $\frac{X(z)}{V(z)} = \frac{zT_S}{z-1}$ (7)
- **Tustin (Trapezoidal)**: $\frac{X(z)}{V(z)} = \frac{T_S}{2}\frac{z+1}{z-1}$ (8)

6. s-to-z Mapping Relationships

By comparing the continuous-time and discrete-time representations, the following mappings are obtained:

- **Forward Euler**: $s = \frac{z-1}{T_S}$ (9)
- **Backward Euler**: $s = \frac{z-1}{zT_S}$ (10)
- **Tustin (Bilinear Transform)**:
  $$\mathrm{s} = \frac{2}{T_s}\frac{z-1}{z+1} \quad (11a)$$
  Or equivalently,
  $$\mathrm{s} = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}} \quad (11b)$$

The inverse mappings are:

- **Forward Euler**: $z = 1 + sT_S$ (12)
- **Backward Euler**: $z = \frac{1}{1-sT_S}$ (13)
- **Tustin**: $z = \frac{1+\frac{sT_S}{2}}{1-\frac{sT_S}{2}}$ (14)

7. Application to a First-Order Low-Pass Filter

Consider a continuous-time low-pass filter:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{1}{s\tau + 1} \quad (15)$$

The pole is located at:

$$s = -\frac{1}{\tau}$$

Substituting this pole into the z-domain mappings yields the discrete-time poles for different methods.

7.1 Forward Euler

$$z = 1 - \frac{T_s}{\tau}$$

For stability:

$$\left|1 - \frac{T_s}{\tau}\right| < 1 \Longrightarrow 0 < T_s < 2\tau$$

Thus, Forward Euler becomes unstable for sufficiently large sampling times.

The discrete-time difference equation is obtained by substituting (9) into (15), yielding

$$y_{k+1} = \left(1 - \frac{T_s}{\tau}\right) y_k + \frac{T_s}{\tau} u_k \qquad (16)$$

7.2 Backward Euler

$$z = \frac{1}{1 + \frac{T_s}{\tau}}$$

For all positive values of T_s, $|z| < 1$. Hence, Backward Euler is unconditionally stable for this system.

The discrete-time difference equation is obtained by substituting (10) into (15), yielding

$$y_{k+1} = \frac{1}{1 + \frac{T_s}{\tau}} y_k + \frac{T_s/\tau}{1 + \frac{T_s}{\tau}} u_{k+1} \qquad (17)$$

7.3 Tustin Method

$$z = \frac{1 - \frac{T_s}{2\tau}}{1 + \frac{T_s}{2\tau}}$$

Again, $|z| < 1$ for any positive sampling time, indicating unconditional stability.

The discrete-time difference equation is obtained by substituting (11a) into (15), yielding

$$y_{k+1} = \frac{2\tau - T_s}{2\tau + T_s} y_k + \frac{T_s}{2\tau + T_s} \left(u_k + u_{k+1}\right) \qquad (18)$$

The above discrete-time difference equation requires the future sampled input u_{k+1}, which is not directly available and therefore must be estimated. A common estimation approach assumes that the input varies linearly over successive sampling intervals, leading to

$$u_{k+1} = u_k + \left(u_k - u_{k-1}\right) = 2u_k - u_{k-1}$$

This approximation is valid only when the sampling rate is sufficiently high such that the input can be assumed to be linear within each sampling interval. If this condition is not satisfied, the estimation introduces errors in u_{k+1}. To avoid this limitation, an alternative discrete-time difference equation can be derived by substituting (11b) into (15).

$$y_k = \frac{2\tau - T_s}{2\tau + T_s} y_{k-1} + \frac{T_s}{2\tau + T_s} \left(u_{k-1} + u_k\right) \qquad (19)$$

8. ZoH/Exponential Mapping

When the derivative term (velocity in the previous discussion) is neither constant nor varies linearly but follows a general curved profile, simple numerical integration methods such as Forward Euler, Backward Euler, and Tustin may introduce approximation errors.

In such cases, the **Zero-Order Hold (ZOH)** method is preferred. The ZOH approach discretizes the **exact continuous-time solution** of a linear time-invariant (LTI) system under the assumption that the input remains constant over each sampling interval. This results in an exact discrete-time representation at the sampling instants.

8.1 ZOH Mapping for a Homogeneous System

Consider the continuous-time system

$$\frac{dy}{dt} = s * y$$

The homogeneous solution:

$$y_h(t) = Ce^{st}$$

Since there is no external input, no particular solution exists. Therefore, the general solution is

$$y(t) = y_h(t) + y_p(t) = Ce^{st} + 0$$

Applying the initial condition, we obtain $y_0 = C \Longrightarrow C = y_0$

So: $y(t) = y_0 e^{st}$

Sampling the solution at $t = T_s$ gives the discrete-time equation

$$y_{k+1} = y_k e^{sT_s} \tag{20}$$

The general discrete-time representation is

$$y_{k+1} = zy_k \tag{21}$$

Comparing (20) and (21), the s-z mapping for ZoH method is obtained as

$$\boldsymbol{z = e^{st}}$$

8.2 ZOH Discretization of a First-Order Low-Pass Filter

Follow the same approach to discretize the LPF system of (15).

$$\tau\frac{dy}{dt} + y = u$$

$$\frac{dy}{dt} = -\frac{y}{\tau} + \frac{u}{\tau} \tag{22}$$

The homogeneous solution:

$$y_h(t) = Ce^{-\frac{t}{\tau}}$$

The particular solution:

Assume zero-order hold input, i.e., $u(t) = u_k\ (constant)\ \forall\ t\ \epsilon\ \left[kT_s, (k+1)T_s\right]$

Since the input is constant over the sampling interval, a constant particular solution is assumed:

$$y_p(t) = K$$

Substituting into (22), we get

$$0 = -\frac{K}{\tau} + \frac{u_k}{\tau} \Longrightarrow K = u_k$$

General solution:

$$y(t) = y_h(t) + y_p(t) = Ce^{-\frac{t}{\tau}} + u_k$$

Applying the initial condition, $y_0 = C + u_k \Longrightarrow C = y_0 - u_k$

So: $y(t) = y_0 e^{-\frac{t}{\tau}} + \left(1 - e^{-\frac{t}{\tau}}\right) u_k$

Sampling at $t = T_s$, the discrete-time difference equation becomes

$$y_{k+1} = y_k e^{-\frac{T_s}{\tau}} + \left(1 - e^{-\frac{T_s}{\tau}}\right) u_k \tag{23}$$

Let

$$\mathrm{a} = e^{-\frac{T_s}{\tau}},\quad \alpha = 1 - e^{-\frac{T_s}{\tau}}$$

Then (23) can be written as

$$y_{k+1} = ay_k + (1-a)u_k \tag{24a}$$

Or equivalently,

$$y_{k+1} = (1-\alpha)y_k + \alpha u_k \Longrightarrow y_{k+1} = y_k + \alpha(u_k - y_k) \qquad (24b)$$

8.3 Corelate with Velocity Examples

Case 1: Constant Velocity
Consider a constant velocity given by

$$\frac{dx}{dt} = 2$$

The corresponding continuous-time solution is

$$x(t) = x_0 + 2t$$

Sampling the solution at discrete instant $t = T_s$, the discrete-time difference equation becomes

$$x_{k+1} = x_k + T_s * 2$$

This result is **exact** and is equivalent to the Forward Euler, Backward Euler, and Tustin methods given in (2)-(4).

Case 2: Linearly Increasing Velocity
Now consider a linearly increasing velocity

$$\frac{dx}{dt} = 2 + 4t$$

The corresponding continuous-time solution is

$$x(t) = x_0 + 2t + 2t^2$$

Sampling the solution at discrete instant $t = T_s$, the discrete-time difference equation becomes

$$x_{k+1} = x_k + T_s * 2 + 2 * {T_s}^2$$

This result is **exactly captured by the Tustin (trapezoidal) method**, whereas the Forward and Backward Euler methods introduce approximation errors for linearly varying velocity.

These examples illustrate that Euler methods are exact only for constant derivatives, while the Tustin method is exact for derivatives that vary linearly over the sampling interval. In contrast, the Zero-Order Hold (ZOH) method yields an exact discrete-time representation for linear time-invariant systems under piecewise-constant inputs, even when the resulting system response varies exponentially.

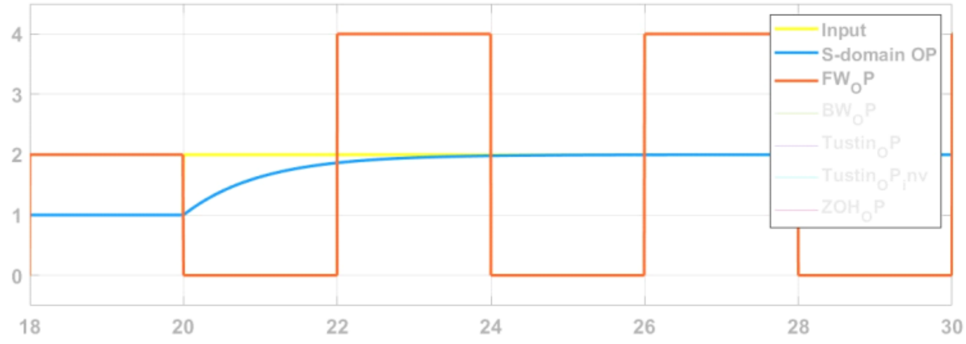
9. Simulation Study

The discrete-time low-pass filter (LPF) derived from the continuous-time s-domain model, along with all four discretization methods, is implemented in **MATLAB/Simulink** using the difference equations (15)-(19) and (24).

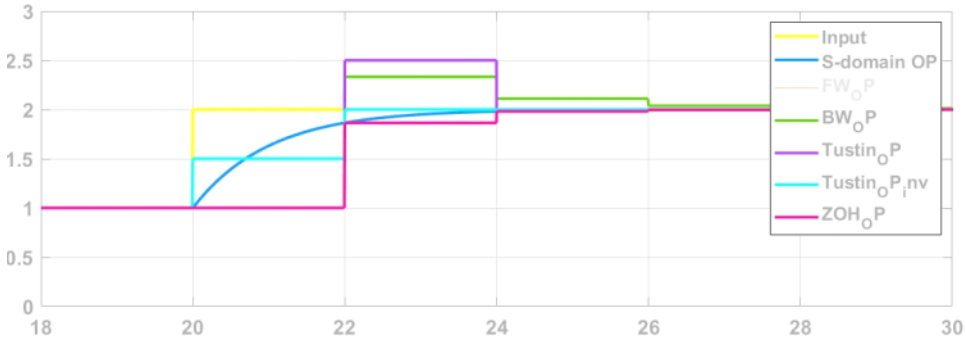
A step change in the input signal from 1 to 2 is applied, and the corresponding LPF outputs obtained using different discretization methods are observed. The simulation step size is set to 10e-6 sec to accurately capture the continuous-time dynamics. The sampling time of the discrete filters is varied across different cases to study the effects of discretization. The time constant of the LPF is fixed at $\tau = 1$.

Case 1: Sampling Time $T_s = 2$ sec
The selected sampling time corresponds to the **stability boundary** of the Forward Euler method. As expected, the Forward Euler–based discrete LPF exhibits oscillatory behaviour at a sampling time of 2 s, indicating marginal stability. This behaviour is illustrated in the corresponding simulation results.

The continuous-time s-domain LPF response, shown by the **blue curve**, is used as a reference for comparison with all discrete-time filter implementations.



The outputs of the remaining discretization methods are shown below for comparison with the continuous-time reference.

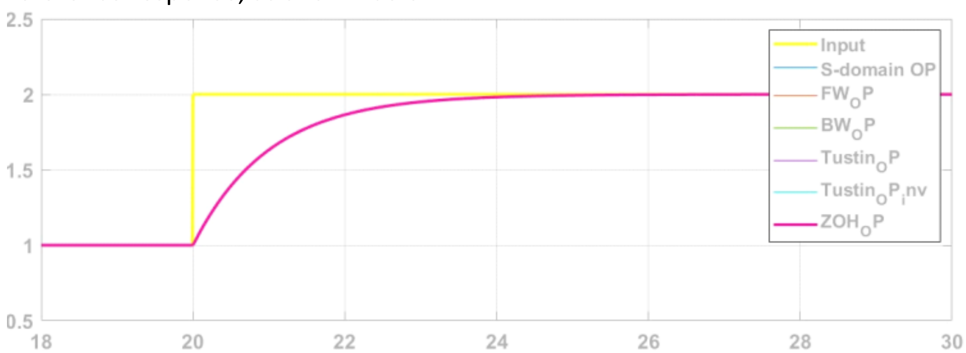


Although all discretization methods are able to track the reference response, a noticeable tracking error persists for all methods except the ZOH method. During the interval from 20 s to 26 s, the ZOH-based output exactly coincides with the continuous-time reference at each sampling instant, whereas the outputs obtained using other methods deviate from the reference.

This behaviour highlights that the ZOH method provides an exact discrete-time representation of the continuous-time system response under piecewise-constant inputs, even when the underlying response exhibits curvature between sampling instants. The output of the ZOH method corresponds to the sampled values of the continuous-time response held constant over each sampling interval, which motivates the term *zero-order hold*.

Case 2: Sampling Time $T_s = 10e^{-6}$ sec

At this very small sampling interval, the variation of the signal within each sample period is negligible, and the response can be approximated as locally linear. Consequently, all discretization methods accurately track the reference response, as shown below.



10. Conclusion

This document presented a structured and physically intuitive explanation of common numerical methods used to discretize continuous-time ODEs. While Forward Euler is simple and computationally efficient, it imposes strict stability constraints on the sampling time. Backward Euler and Tustin methods offer improved stability properties, with Tustin providing better accuracy by preserving stability and offering a closer approximation of the continuous-time frequency response.

In addition, the Zero-Order Hold (ZOH) or exponential mapping method was shown to yield an exact discrete-time representation for linear time-invariant systems under piecewise-constant inputs, even when the system response varies exponentially. Simulation results validated the theoretical analysis and highlighted the trade-offs among accuracy, stability, and computational complexity for the different discretization techniques.

These insights are particularly important in digital control, power electronics, and real-time simulation applications where numerical stability and accuracy are critical.