

CHAPTER OVERVIEW

10: Numerical Integration of ODEs

This article describes the numerical methods for solving the **initial-value problem**, which is a standard type of problem appearing in many fields of physics. Suppose we have a system whose state at time t is described by a vector $\vec{y}(t)$, which obeys the first-order ordinary differential equation (ODE) for the form:

$$\frac{d\vec{y}}{dt} = \vec{F}(\vec{y}(t), t). \quad (10.1)$$

Here, \vec{F} is some given vector-valued function, whose inputs are (i) the instantaneous state $\vec{y}(t)$ and (ii) the current time t . Then, given an initial time t_0 and an initial state $\vec{y}(t_0)$, the goal is to find $\vec{y}(t)$ for subsequent times.

Conceptually, the initial value problem is distinct from the problem of solving an ODE discussed in the article on finite-difference equations. There, we were given a pair of boundaries with certain boundary conditions, and the goal was to find the solution between the two boundaries. In this case, we are given the state at an initial time t_0 , and our goal is to find $\vec{y}(t)$ for some set of future times $t > t_0$. This is sometimes referred to as "integrating" the ODE, because the solution has the form

$$\vec{y}(t) = \vec{y}(t_0) + \int_{t_0}^t dt' \vec{F}(\vec{y}(t'), t'). \quad (10.2)$$

However, unlike ordinary numerical integration (i.e., the computing of a definite integral), the value of the integrand is not known in advance, because of the dependence of \vec{F} on the unknown $\vec{y}(t)$.

[10.1: Example- Equations of Motion in Classical Mechanics](#)

[10.2: Forward Euler Method](#)

[10.3: Backward Euler Method](#)

[10.4: Adams-Moulton Method](#)

[10.5: Runge-Kutta Methods](#)

[10.6: Integrating ODEs with Scipy](#)

This page titled [10: Numerical Integration of ODEs](#) is shared under a [CC BY-SA 4.0](#) license and was authored, remixed, and/or curated by [Y. D. Chong](#) via [source content](#) that was edited to the style and standards of the LibreTexts platform.