Visual Computing - Notes Week 13

Ruben Schenk, ruben.schenk@inf.ethz.ch

December 20, 2021

8.4 Physics-Based Animation

8.4.1 Introduction

We first differentiate between two important terms in the field of physics and physics-based animation:

- **Kinematics:** The branch of mechanics concerned with the motion of objects without reference to the forces which cause the motion.
- **Dynamics:** The branch of mechanics concerned with the motion of bodies under the action of forces.

8.4.2 The Animation Equation

We have already seen the *rendering equation* which is concerned with rasterization and path tracing which give approximate solutions to the rendering equation.

The **animation equation** is concerned with the large spectrum of physical systems and phenomena, such as solids, fluids, elasticity, etc. For animations, the connection between force and motion is essential:

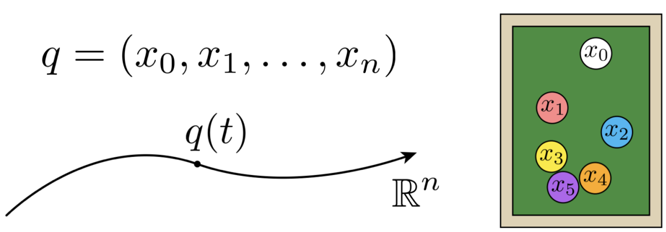
> A change in motion is proportional to the motive force impressed and takes place along the straight line in which that force is impressed. - Sir Isaac Newton, 1687

However, there is more to be said than $F = ma$:

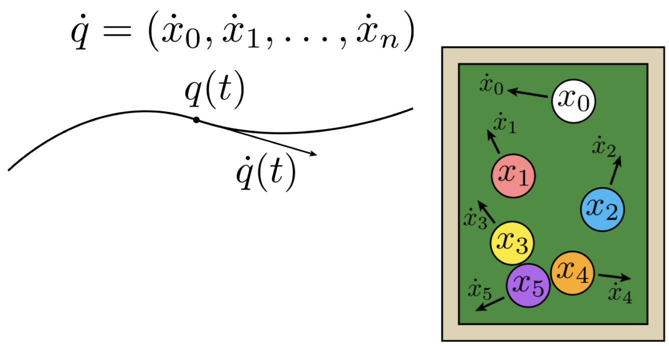
- Every system has a *configuration* $q(t)$
- It also has a *velocity* $\dot{q} := \frac{d}{dt}q$
- It has some kind of *mass* M
- There are *forces* F acting on the system

8.4.3 Generalized Coordinates

In physics, we often need to describe a system with many moving parts, e.g. a collection of billiard balls, each with position x_i. We usually collect them all into a single vector of **generalized coordinates.**



We can think of q as a single point moving along some trajectory in $\mathbb{R}^n$. If we take the time derivative of the generalized coordinates, we get **generalized velocity:**



8.4.4 Ordinary Differential Equations

Many dynamical systems can be described via an **ordinary differential equation:**

$$\frac{d}{dt}q = f(q,\, \dot{q},\, t),$$

where $\frac{d}{dt}$ is the change in configuration over time and f is the velocity function.

Example: Assume we have a function where the rate of growth is proportional to the value, i.e.

$$\frac{d}{dt}u(t) = au,$$

then our solution is given by $u(t) = be^{at}$.

Note that Newton's 2nd law is an ODE as well, i.e. $\ddot{q} = F/m$. We can also write this as a system of two first order ODEs, by introducing new variables for velocity:

$$\dot{q} = v, \quad \dot{v} = \frac{F}{m}\frac{d}{dt}\begin{bmatrix} q \\ v \end{bmatrix} = \begin{bmatrix} v \\ F/m \end{bmatrix}.$$

8.4.5 Solving ODEs Numerically

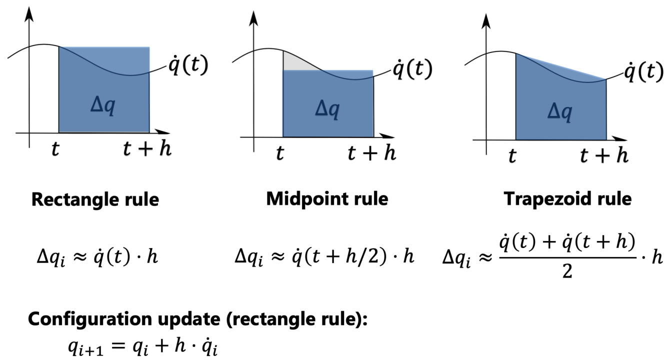
When we are talking about solving ODEs we mean that given some initial conditions $q(0)$ and $\dot{q}(0)$, we want to find the function $q(t)$. Solving ODEs *numerically* means solving numerical time integration:

$$q(t+h) = q(t) + \int_t^{t+h} \dot{q}(t)\, dt$$

We use some discrete approximation of the form

$$\Delta q_i \simeq \int_t^{t+h} \dot{q}(t)\, dt,$$

and then apply the following **numerical integration rules:**



8.4.6 Forward Euler

Forward Euler describes a simple scheme: We evaluate the derivative at the current configuration and write the new state explicitly in terms of known data:

$$q_{i+1} = a_i + h \cdot \dot{q}_i \dot{q}_{i+1} = \dot{q}_i + h \cdot \ddot{q}_i = \dot{q}_i + hM^{-1}F(q_i,\, \dot{q}_i)$$

Example: Assume some simple linear ODE, i.e. $\dot{u} = -au$, $a > 0$. The exact solution to this ODE would be $u(t) = u(0)e^{-at}$, so $u_k \to 0$ as $l \to \infty$. The forward Euler approximation is given by:

$$u_n = (1 - ha)^n u_0,$$

from where we can derive that this decays only if $|1 - ha| < 1$, or equivalently $h < 2/a$, so in practice we many need very small time-steps!

8.4.7 Backward Euler

We might try something else and evaluate the velocity at some new configuration. This scheme is also known as **backward Euler.** The new configuration is then implicit, and we must solve for it:

$$q_{i+1} = q_i + h \cdot \dot{q}_{i+1} \dot{q}_{i+1} = \dot{q}_i + h \cdot \ddot{q}_{i+1} = \dot{q}_i + hM^{-1}F(q_{i+1},\, \dot{q}_{i+1})$$

We can again observe the stability of the backward Euler with our previous example, i.e. $\dot{u} = -au$, $a > 0$. The backward Euler approximation is given by:

$$u_n = \left(\frac{1}{1 + ha}\right)^n u_0,$$

which decays if $|1 + ha| > 1$, which is always true! Backward Euler is *unconditionally stable* for linear ODEs.