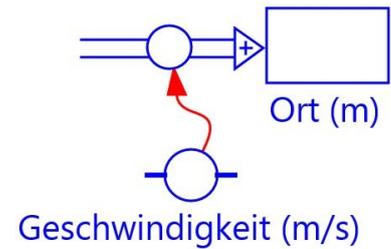


Aufgabe 1: Konstante Geschwindigkeit

a) Zeichne das zeitdiskrete Modell einer Bewegung mit konstanter Geschwindigkeit.

b) Setze das Modell iterativ um. Betrachte hierzu das Video [03_04_nvm_a_konstante_geschwindigkeit_xx](#) (xx ist coach oder geogebra)

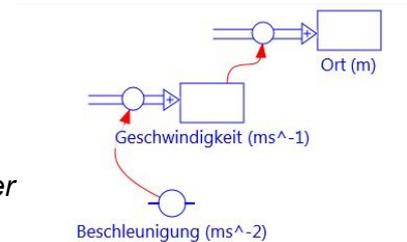


Aufgabe 2: Konstante Beschleunigung

a) Zeichne das zeitdiskrete Modell einer Bewegung mit konstanter Beschleunigung.

b) Setze das Modell iterativ um. Betrachte hierzu das Video

[03_04_nvm_b_konstante_beschleunigung_xx](#) (xx ist coach oder geogebra)



Aufgabe 3: Genauigkeit

Betrachte das Video [03_04_nvm_d_beschleunigung_eulerverfahren_und_genauigkeit_xx](#) (xx ist coach oder geogebra)

Notiere Dir die wesentliche Erkenntnis hieraus im Heft unter der Überschrift „Genauigkeit bei Simulationen“.

Kleinere Zeitschritte verbessern die Simulation.

Der Computer rechnet nur bis auf eine bestimmte Stellenanzahl. Bei jeder Rechenoperation verändert er also das tatsächliche Ergebnis durch Rundung.

Überlege Dir, was passiert, wenn man viele kleine Ungenauigkeiten addiert und

- nur wenige Rechenschritte macht.
- sehr viele Rechenschritte macht.

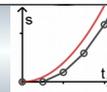
Fasse Deine Erkenntnis aus dem Video und Deine Überlegungen zusammen und notiere sie in Deinem Heft.

- Kleinere Zeitschritte verbessern die Simulation, bedeuten aber auch eine höhere Rechenleistung und auch mehr Rechenzeit!**
- Die Genauigkeit kann durch kleiner werdende Zeitschritte nicht beliebig erhöht werden, da sich bei jedem Rechenschritt Rundungsfehler aufsummieren. Wächst die Anzahl der Rechenschritte, so nimmt die Genauigkeit auf Grund der Rundungsfehler ab.**

Die Genauigkeit hat also ein Maximum, das sich aus den Rundungsfehlern und dem Zeitschritt ergibt.

Es gibt noch weitere Möglichkeiten, die Genauigkeit bei nicht zu vielen Rechenschritten zu erhöhen. Schau Dir das Video [03_04_e_beschleunigung_mit_runge_kutta](#) an und notiere im Heft, was Du für wichtig hältst.

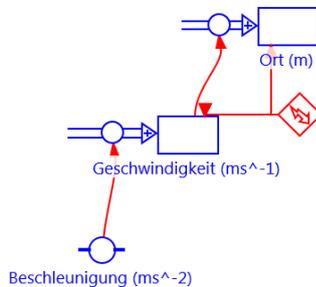
Mit Hilfe von mathematischen Verfahren kann man die iterative Umsetzung verbessern.



Vertiefungen

Programmiere die Bewegung eines hüpfenden Flummis (Gummiballs)

- a) Zunächst soll die Modellbildung den Flummi einige Male wieder auf die gleiche Höhe springen lassen. Dokumentiere verschiedene Simulationsparameter durch Screenshots der Schaubilder.



Definition

Triggerbedingung:

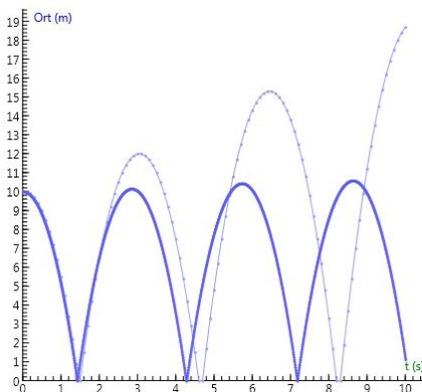
Einmal Ort <= 0

Aktionen:

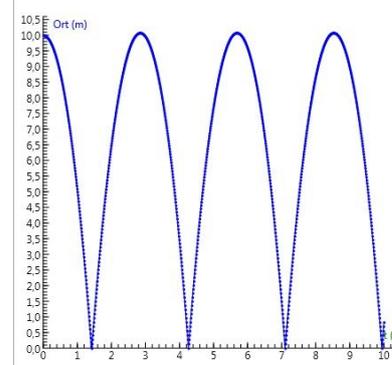
Ort = 0

Tue Geschwin... = -Geschwindigkeit

Hinzufügen... Entferne Formel



Euler mit $Dt = 0,1s$ und $Dt = 0,01s$



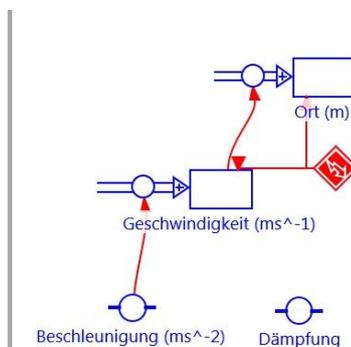
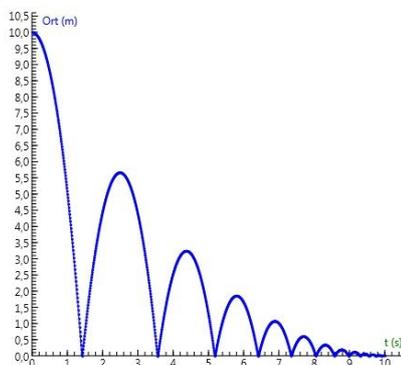
und RK4 mit 0,01s

- b) Der Flummi hüpfet bei der Simulation bei jedem nachfolgenden Mal immer höher als davor. Begründe dies mit Hilfe der Idee des Euler-Verfahrens.

Bei der vorgeschlagenen Programmierung wird bei einer negativen Höhe das Vorzeichen der Geschwindigkeit geändert. Somit hat der Flummi nach dem Aufprall eine höhere Geschwindigkeit als davor. Dies ist natürlich falsch und führt zu einer größeren Sprunghöhe als die Fallhöhe.

Um den Vorgang besser zu simulieren müsste der Aufprall genauer erfasst werden.

- c) Verändere die Modellbildung so, dass bei jedem Aufprall Energie entwertet wird und der Flummi dann nicht mehr so hoch springt.



Berechnet durch RK4

Definition

Triggerbedingung:

Einmal Ort <= 0

Aktionen:

Ort = 0

Tue Geschwin... = -Geschwindigkeit * Dämpfung

Bemerkung: Mit dem „Dämpfungsfaktor“ wird das vorherige Problem des nicht genau erfassten Aufpralls nicht behoben und der Dämpfungsfaktor ist somit größer, als der tatsächliche Dämpfungsfaktor.