

Spezielle Relativität

Jonas Berggren

February 26, 2020

Contents

1	Relativität nach Newtonscher Physik	3
1.1	Bezugssystem	3
1.2	Wechsel von Bezugssystemen	4
1.3	Minkoswsky Raumzeit diargamme	4
2	spezielle Relativität	4
2.1	Herleitung	5
2.2	transformation zwischen Bezubssytemen	5
2.3	Raumzeititnervall als erhaltene Größe	5
2.4	Implikationen	6
2.5	Einschränkungen dieses Modells	6
3	Programm	7
3.1	Nutzung	7
3.2	Button	7
3.2.1	Methoden	7
3.3	Input	8
3.3.1	Methoden	8
3.4	Box	8
3.4.1	Methoden	9
3.5	Obj	9
3.5.1	Methoden	9
3.6	Funktionen	9

4	Quellen	9
5	Notizen	10

Abstract

In diesem Dokument erkläre ich wie ich ein Programm entwickelt habe, was Albert Einsteins spezielle Relativität visualisiert. Das Programm arbeitet anhand von Minkowski-Raumzeitdiagrammen, und nutzt Objektorientierte Programmierung.

Außerdem erkläre ich die darunter liegende Physik und leite Lorentztransformation her. Dabei werde ich auf Relativität vorr Einstein, auf Minkowski-Raumzeit-diagramme, auf Transformationen zwischen Bezugssystemen, Auf die Implikation Spezieller Relativität, und desse Einschränkungen.

1 Relativität nach Newtonscher Physik

Bevor wir über Einsteins spezielle Relativität reden können, müssen wir das Konzept von Raum, Zeit und bewegung klarstellen

1.1 Bezugssystem

Zunächst muss klar gestellt werden wie Position, Zeit und Geschwindigkeit gemessen werden. Dazu muss ein Koordinatensystem Räumlich und Zeitlich definiert werden. Das Koordinatensystem hat einen Ursprung mit $x = 0, y = 0, z = 0$ und $t = 0$. Hierbei ist der Ursprung des Koordinatensystems in der Regel auf ein Objekt zu Beginn des Beobachtungszeitraums bezogen. Für diese Betrachtung müssen folgende Bedingungen erfüllt sein:

- Das Bezugssystem muss inertial(unbeschleunigt) sein
- Die Raumzeit muss flach sein, es darf keine Gravitation wirken, ART

Durch die Tatsache, dass in allen Inertialen Bezugssystemen die gleichen physikalischen Gesetze gelten, sind alle Bezugssysteme gleich gültig. Es ist keine Universal gültige Aussage über die Position oder Geschwindigkeit eines Körpers, oder Zeitpunkt eines Ereignisses möglich. Demnach ist es nichts sagend zu sagen, man hätte zum Zeitpunkt t die Position x, y, z und bewege sich mit Geschwindigkeit \vec{v} . Es muss immer ein Bezugspunkt gewählt werden z.b. Erdmittelpunkt Bezugssysteme können sich also relativ zu einander bewegen und dennoch gleichermaßen gültig das selbe Ereignis beschreiben.

1.2 Wechsel von Bezugssystemen

Ich werde mich im folgenden auf eine Raumdimension beschränke. Das Hinzufügen der anderen Raumdimensionen, kann durch Ersetzen der Richtungsabhängigen Größe n , durch Vektoren. x wird demnach zu $\vec{0p}$, v zu \vec{v} . dabei ist zu beachten dass die gerichteten Relativistischen Effekte nur entlang der Bewegungsrichtung auftreten.

Es wird zunächst ein Bezugssystem gewählt mit den Größen x, t und v . Anschließend wird ein gestrichenes Bezugssystem gewählt mit den Größen x', t' und v' , wobei $t = t'$ gilt. Aus unserer alltäglichen Erfahrung geht hervor, dass für die Position eines, zu dem ungestrichenen System statischen Objekts gilt: $x' = x - vt$ Genau so gilt für Geschwindigkeiten:

$$u' = u - v \quad (1)$$

Hierbei ist v die Geschwindigkeit des gestrichenen Bezugssystems und u die Geschwindigkeit des betrachteten Objekts.

1.3 Minkowsky Raumzeit Diagramme

Das Minkowski Raumzeit Diagramm betrachtet, in seiner üblichen Form, Objekte in einer Raumdimension und Zeit. Hierzu wird die Zeit auf die vertikale Achse gelegt und die Position auf die horizontale. Für die Betrachtung von spezieller Relativität werden die Einheiten einfachheitshalber so gewählt, dass die Lichtgeschwindigkeit $c = 1$ und $x = t$.

2 spezielle Relativität

Die Spezielle Relativität fügt ein entscheidendes Postulat hinzu:

- Die Lichtgeschwindigkeit ist eine universelle Konstante

Dies wirft direkt eine Frage auf: Wenn jemand mich mit einer Taschenlampe anleuchtet während ich mich auf ihn zu bewege, wie kann es dann sein, dass wir beiden den exakt gleichen Wert für die Geschwindigkeit dieses Lichts messen? Das Postulat ist also nicht mit der Formel 1 vereinbar.

2.1 Herleitung

Es werde zwei Bezugssysteme betrachtet, die sich relativ zu einander mit der Geschwindigkeit $v \neq 0$ bewegen. In dem ungestrichenen Bezugssystem fliegt ein Photon zwischen zwei Spiegeln hin und her, die l von einander entfernt sind. Um von einem Spiegel zum anderen zu gelangen muss das Licht vom gestrichenen System aus gesehen eine größere Strecke d zurücklegen, nämlich:

$$d^2 = l^2 + (vt')^2 \quad (2)$$

Da $d \neq l$ gilt aber beide Strecken mit der gleichen Geschwindigkeit zurückgelegt werden muss $t' \neq t$ gelten. Drückt man d und l in c aus so gilt:

$$c^2 t'^2 = c^2 t^2 + v^2 t'^2 \quad (3)$$

Dies kann dann durch subtrahieren von $v^2 t'^2$ um ausklammern von t'^2 nach t' aufgelöst werden. dann gilt:

$$t' = \frac{t}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (4)$$

Teilt man nochmal durch t ergibt sich der Relativitätsfaktor γ :

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (5)$$

2.2 transformation zwischen Bezugssystemen

2.3 Raumzeitintervall als erhaltene Größe

Nach der speziellen Relativität sind Raum, Zeit und sogar die Reihenfolge von Ereignissen relativ. Es wirkt so als nur die Lichtgeschwindigkeit konstant. Es gibt aber eine weitere Größe die unter Lorentztransformation invariant bleibt: Die sogenannte Eigenzeit

$$\tau^2 = (\Delta ct)^2 - \Delta x^2 - \Delta y^2 - \Delta z^2 \quad (6)$$

Wobei Δt der Zeitleihen und $\Delta x, y, z$ den räumlichen Abstand zwischen zwei Ereignissen darstellen. τ kann im Minkowski-Diagramm als Vektor dargestellt werden der die Ereignisse verbindet.

Diese Größe kann durch die Kombination aus negativen Raumkomponente und positiver Zeitkomponenten im gesamten Reellen Bereich definiert werden. Was für ein Wert diese Größe hat, hat auf die Wirkung darauf, worüber sich Beobachter uneinig sein können.

Ist τ negativ, können sich Beobachter über die Reihenfolge der Ereignisse uneinig sein. Außerdem können sich die beiden Ereignisse nicht beeinflussen. Dies ist im Diagramm daran zu erkennen, dass der Winkel zwischen τ und der ct -Achse kleiner als 45° ist. Somit sind auch Gleichzeitigkeitslinien zulässig, die steiler sind als τ . Weltlinien, die flacher sind als τ , sind jedoch unzulässig.

Gilt $\tau = 0$, so kann das eine Ereignis das andere durch ein Lichtteilchen beeinflussen, jedoch nicht durch ein massebehaftetes Signal.

Ist τ positiv, ist die Reihenfolge der Ereignisse universell. In dem Fall entspricht τ der Zeit, die eine Uhr zwischen den Ereignissen messen würde, die sich von dem ersten zum zweiten Ereignis bewegt.

2.4 Implikationen

2.5 Einschränkungen dieses Modells

Spezielle Relativität beschreibt nur inertielle Bezugssysteme in einer flachen Raumzeit.

Inertielle Bezugssysteme sind unbeschleunigt. Ein Ball, der mit einer Geschwindigkeit gleich null losgelassen wird, behält seine Position bei.

Dies löst zum Beispiel das bekannte Zwillingsparadoxon: Ein Zwilling fährt auf eine Raummission, bei der er mit beispielsweise halber Lichtgeschwindigkeit durch das All fliegt, während sein Bruder auf der Erde verbleibt. Als er zur Erde zurückkehrt, stellt sich die Frage, wer nun älter ist. Nach der Speziellen Relativität können sich beide Zwillinge für den Zeitraum der Mission als statisch betrachten und den anderen als bewegt. Demnach kommen beide zum Schluss, dass sie selbst jünger sein sollten.

Dabei wird jedoch vernachlässigt, dass der Astronaut beispielsweise zu Alpha Centauri fliegt, dort umkehrt und wiederkommt. Dabei erfährt er eine Beschleunigung, die die Spezielle Relativität nicht vorsieht.

Nehme man an, der Astronaut führe bei Alpha Centauri ein Swingby-Manöver durch und ändere so seine Richtung, so würde er sich nicht in einer flachen Raumzeit bewegen.

Diese Überlegung führt uns in die Allgemeine Relativität, die sich maßgeblich von Newtons Gravitation unterscheidet. Gravitation wird nicht mehr

als Kraftfeld beschrieben sondern als Raumzeitkrümmung, die die inertielle Laufbahn von Objekten verändert.

3 Programm

Das Programm basiert auf Objekt orientierter Programmierung und nutzt das pygame Modul für die Grafik. Ich habe insgesamt vier Klassen entwickelt: Eine für die Darstellung der Bezugssysteme im Diagramm, eine für die Darstellung von Buttons und zwei für die Darstellung der Eingabefelder. Außerdem habe ich jeweils eine Funktion zum wechseln von Bezugssystem, und zum Handhaben von unzulässigen Nutzereingaben.

3.1 Nutzung

Diese Programm in python2.7 ausgeführt werden. In dem rechten Feld können Geschwindigkeit und Startposition von Objekten eingegeben werden. Beim klicken der send-Taste wird die Weltlinie, und die dazu gehörige Gleichzeitigkeitslinie des Objektes in das Koordinatensystem eingezeichnet. Es können nach belieben Objekte hinzugefügt oder entfernt werden. Alle sichtbaren Weltlinien können angeklickt werden um die Lorentz-Transformation in das jeweilige Bezugssystem beobachten zu können.

3.2 Button

Die Klasse `Button` dient zur Darstellung und Handhabung aller Buttons. Diese Klasse nimmt einen Typ und optional, einen parent als Argument.

3.2.1 Methoden

In dem Konstruktor wird die Box in Abhängigkeit von Typ definiert. Anschließend werden weitere Instanzvariablen, für Farbe, Aktivität, Text, Typ, Mutterobjekt definiert.

Die Methode `handle` wird in der Hauptschleife aufgerufen und nimmt pygame ereignisse als Argumente. In Abhängigkeit vom Typ wird der Ortsvektor der Maus angepasst. Wenn sich die Maus über ein dem Button befindet wird der Aktivitätszustand auf True gesetzt, andernfalls auf False. Je nach

Aktivitätszustand wird in der Methode `draw` die Farbe ändert, was ein Hoverfunktion darstellt. Befindet sich die Maus über `self`, wird dann bei einem Klick, zwische den Typen unterschieden. Ist `self.type` gleich `add`, wird der Liste der `objs` eine Instanz von `Obj` hinzugefügt. Außerdem wird die Position von `self.rect` angepasst.

Ist `self.type` gleich `send` wird der index von `self` ermittelt, und der Methode `enter` übergeben, die in Abschnitt 3.3.1 näher erläutert wird.

Ist `self.type` gleich `ok` wird die globale Variable `err` Nne gesetzt.

Ist `self.type` gleich `x` wird der index von `self` ermittelt. Anschließend wird dieser Index genutzt, um die jeweiligen einträge in dem Listen `objs`, `delbuttons`, `inputs` und `sendbuttons`. anschließend werden die Position von den verbleibenden Elementen angepasst um das gelöschte Element aufzufüllen.

Die Methode `draw` wird in der Hauptschleife aufgerufen und zeichnet je nach Typ `self.rect` und `self.txt` auf die unterschiedlichen Flächen.

3.3 Input

Die Klasse `Input` erstellt Eingabefelder indem sie drei Instnazen der Klasse `Box` erstellt. Der Sinn dieser Klasse ist, die Liste der Eingabefelder besser zu organisieren damit indexe leichter gehandhbt werden können.

3.3.1 Methoden

Diese Klasse hat bis auf den Konstruktor nur die Methode `enter`. Sie beim Klicken des 'send' Buttons aufgerufen. Sie prüft die Eignaben in dem jeweiligen Feldern auf zulässigkeit. Es wird überprüft ob Parameter die Zahlen sein sollen Zahlen sind, und das keine Geschwindigkeit über der Lichtgeschwindigkeit eingegeben wird. Wir ein unzulässige Eingabe erkannt wird die Funktion `err` aufgerufen, die eine Fallspezifische Fehlermeldung anzeigt. Sind Felder leer, weredn Standardwert ergenzt. Anschließend wird mit ermittelten den Parametern eine neue Instanz von `Obj` erstellt.

3.4 Box

Die Klasse `Box` ist die eingentlihe Maschienrie hinter den Eingabefeldern. Jenachde was für Parameter übergeben werden, wir `self` zum Namens-

Positions- oder Geschwindigkeitsfeld. Demnach werden Position, und Standardinhalt angepasst.

3.4.1 Methoden

Die Methode `handle` wird in der Hauptscheife aufgerufen, und nimmt `pygameevents` als Parameter. wird das Feld angeklickt, wird Der Standardinhalt entfernt und `self.active` gleich `True` gesetzt. Wird woanders hingeklickt, werde sie Veränderungen wieder rückgängig gemacht. Demnach wird auch die Farbe des Feldes angepasst. Ist anschließend `self.active` `True` werden die Tastatureingaben gespeichert.

`draw` Zeigt dann den passenden Text und das Felde in der richtigen Farbe.

3.5 Obj

`Obj` ist das Herzstück des Programms. Es handhabt alle Einträge in das Diagramm. Es Arguente, die den Namen, den Index, die Position, die Geschwindigkeit, den Farbindex und $t = 0$ definieren.

3.5.1 Methoden

3.6 Funktionen

Die Funktion `error` wird von der Funktion `enter` bei jedem Fehlerfall aufgerufen. Es wird ein Fallspezifischer Fehlerwert übergeben. Die Funktion `error` erstellt ein Fenster auf dem eine fehlerspezifische Nachricht angezeigt wird. Außerdem wir ein `'ok'` button erstellt mit dem das fenster wieder geschlossen werden kann.

4 Quellen

Loedel-Minkowski-Diagramm; zweidimensionale Raumzeit(Zugriff: 06.09.2019):

<https://stackoverflow.com/questions/46390231/how-to-create-a-text-input-box-with-pygame>
<https://www.youtube.com/playlist?list=PLD9DDFBDC338226CA>

5 Notizen

$x = ct$ $x' = ct - vt = (c-v)t'$ speed addition Raum, Zeit, Gleichzeitigkeit, Reihenfolge Invariante Proper time propertime, if $\Delta x = 0$ they can be connected by a lightbeam spacelike, timelike lightcone Fourvector Fourvelocity Gleichzeitigkeit $t = vx$ Linie der Gleichzeitigkeit ist die Spiegelung der Weltline entlang der Lichtlinie