

PO7: Simulatie van de speciale relativiteitstheorie

Koen, Nico, Mitchell, Sven en Simone

6Vin1 / 6Vc

**Inhoudsopgave**

[**Inleiding**](#_yc7fcqt38bob)2

[**Hoofdvraag**](#_il2lbn4ytf6k)2

[**Deelvragen**](#_vwszrdkhmcc8)3

[**Kunnen wij de theorie achter de speciale relativiteitstheorie begrijpen en beschrijven?**](#_gi9rsvmwb72q)3

[Lichtsnelheid](#_dmcmywmtb85u) 3

[Gammafactor / lengtecontractie](#_ju5u9fnx2dat) 4

[**Kunnen wij de theorie achter de speciale relativiteitstheorie toepassen in Python?**](#_rr24sa1r56mi)5

[**Kunnen wij door middel van code in Python een overzichtelijk beeld creëren die de effecten van de speciale relativiteitstheorie duidelijk visualiseert?**](#_uawusznhllmt)6

[**Gebruiksaanwijzing**](#_xrad9s4uecjd)7

[**Conclusie**](#_2953s0ihglst)7

[**Discussie**](#_hb39plc313jl)8

[**Verwijzingen**](#_8qtin1pr0uvv)9

[**Bronnen**](#_f9oooxo8453z)9

# Inleiding

Voor de laatste praktische opdracht van informatica willen wij een simulatie maken van de speciale relativiteitstheorie van Einstein. Deze theorie hebben wij voor de tweede SE-week moeten leren en wij vonden deze theorie zeer interessant. In deze theorie wordt het gedrag van objecten beschreven als deze objecten met hele hoge snelheden bewegen. Met hoge snelheden wordt 10% of meer van de lichtsnelheid bedoeld. Objecten vertonen raar gedrag rond deze snelheden, zo kan de tijd anders gaan lopen en kan de lengte van dat object veranderen. Deze twee verschijnselen willen wij ook gaan simuleren in onze simulatie. Eerst willen we informatie vinden over deze theorie en hoe deze precies werkt, zodat we die goed kunnen simuleren. ***Het probleem dat wij willen oplossen is dat het voor mensen lastig is om de speciale relativiteitstheorie te visualiseren.***

# Hoofdvraag

In de inleiding hebben wij ons doel geformuleerd, dat is dat het voor mensen lastig is om de speciale relativiteitstheorie te visualiseren. Door dit doel zijn we op de volgende hoofdvraag gekomen: ***kunnen wij een simulatie maken voor de speciale relativiteitstheorie van Einstein, zodat mensen de theorie beter kunnen visualiseren?*** Wij willen deze deelvraag onderzoeken door middel van **literatuuronderzoek**.

# Deelvragen

Om het onderzoek overzichtelijker te maken, wordt het onderzoek verdeeld in deelvragen. In totaal hebben wij drie deelvragen:

* ***Kunnen wij de theorie achter de speciale relativiteitstheorie begrijpen en beschrijven?***

Deze deelvraag willen wij beantwoorden, omdat we eerst wat meer te weten willen komen over de speciale relativiteitstheorie voordat we daar een simulatie van maken.

* ***Kunnen wij de theorie achter de speciale relativiteitstheorie toepassen in Python?***

Deze vraag willen wij beantwoorden om te kijken of het überhaupt mogelijk is om de speciale relativiteitstheorie toe te passen in een Python programma.

* ***Kunnen wij door middel van code in Python een overzichtelijk beeld creëren die de effecten van de speciale relativiteitstheorie duidelijk visualiseert?***

Deze vraag willen wij beantwoorden om te kijken of wij een goed beeld kunnen creëren van hoe de speciale relativiteitstheorie eruit ziet.

# Kunnen wij de theorie achter de speciale relativiteitstheorie begrijpen en beschrijven?

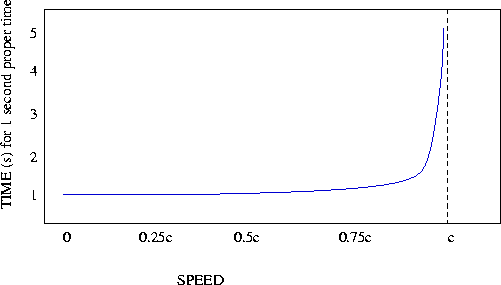
## Lichtsnelheid

De lichtsnelheid staat centraal bij de speciale relativiteitstheorie. Al vanaf de zeventiende eeuw hebben natuurkundigen gepoogd de lichtsnelheid te meten. De eerste was Ole Rømer die de stand van Jupiter gebruikte. Jupiter heeft een veel langere omlooptijd om de zon, waardoor Jupiter weinig is verschoven als de de aarde een half jaar heeft afgelegd, aan de andere kant van de zon staat en dus verder van Jupiter staat. Met het gemeten verschil in tijd en het te berekenen verschil in afstand kon Christiaan Huygens later met de gegevens van Rømer de lichtsnelheid benaderen (Flokstra et al, 2020).

In 1862 wist Léon Foucault de lichtsnelheid indrukwekkend nauwkeurig te benaderen. Hij gebruikte een opstelling bestaand uit een lichtbron, een draaiende spiegel en een holle spiegel. Aan de hand van de draaiing van de spiegel en het verschil in locatie van de aankomstplek van de uitgezonden lichtbundels, kon Foucault de lichtsnelheid bepalen. Hij kwam uit op 298.000 km/s (Price & Zizka, 2007). Later zou dit worden verbeterd tot 299.798 km/s.

Met de lichtsnelheid bekend kon Albert Einstein aan de slag met zijn speciale relativiteitstheorie. Deze theorie stoelde op twee postulaten (Flokstra at al, 2020):

1. de lichtsnelheid is altijd constant en de maximaal haalbare snelheid.
2. de natuurkundige wetten gelden altijd.



Met deze postulaten konden twee verschijnselen worden beredeneerd: tijddilatatie en lengtekrimp. Met tijddilatatie wordt bedoeld dat de tijd die wordt gemeten in een referentiesysteem B langzamer verloopt dan de tijd in het eigen referentiesysteem A, omdat het referentiesysteem B met snelheden dichtbij de lichtsnelheid beweegt ten opzichte van referentiesysteem A.. Het licht dat de waarnemer in referentiesysteem A waarneemt

*Tijddilatatie bij verschillende lichtsnelheden*

in B, moet een langere afstand afleggen, terwijl het met een constante snelheid gaat. Snelheid is afgelegde afstand per tijdseenheid. Als de snelheid constant is, zoals in dit geval de lichtsnelheid, en de afstand groter is, zal de tijdseenheid dus ook groter moeten zijn. Daardoor gaat de tijd in referentiesysteem B langzamer, aangezien er moet worden voldaan aan de constante lichtsnelheid (Flokstra et al, 2020).

Lengtecontractie treedt op als licht horizontaal beweegt in referentiesysteem B, dat nog steeds met een snelheid dichtbij lichtsnelheid beweegt ten opzichte van referentiesysteem A. Het is handig om voor te stellen dat het licht in referentiesysteem B naar voren beweegt voor een bepaalde afstand en dan diezelfde afstand weer terug beweegt naar het startpunt van dit licht. Voor de waarnemer in referentiesysteem A geldt echter ook dat het licht beweegt in een bewegend referentiesysteem. Stel dat referentiesysteem B met het licht mee beweegt op de heenweg. Het licht zal er dan langer over doen om een even grote afstand af te leggen, omdat de lichtsnelheid maar een klein beetje groter is dan de snelheid waarmee referentiesysteem B beweegt. Op de terugweg zal het licht er dan heel kort over doen een even grote afstand af te leggen gezien vanuit referentiesysteem A, omdat het licht tegen de beweging van referentiesysteem B in beweegt. De kortere tijd op de terugweg compenseert alleen niet volledig voor de langere tijd op de heenweg: de totale reistijd is langer dan het geval zou zijn geweest als referentiesysteem B stil stond ten opzichte van referentiesysteem A, gezien vanuit referentiepunt A. Voor de waarnemer van referentiesysteem A zal de reistijd van het licht dus langer zijn dan voor de waarnemer in bewegend referentiesysteem B. De tijd voor de waarnemer in referentiesysteem A gaat dus sneller dan in referentiesysteem B, maar de lichtsnelheid is gelijk in elk referentiesysteem. Voor de waarnemer in referentiesysteem A lijkt het dus dat de tijd in referentiesysteem B slomer gaat, waardoor er dus voor A een kleinere t is. Volgens de formule , waarbij geldt , oftewel de snelheid is de lichtsnelheid. Als *t* dan kleiner wordt, moet de lengte *s* ook kleiner worden om aan de constante lichtsnelheid te voldoen, waardoor het object voor de waarnemer in referentiesysteem A kleiner wordt (Greene, 2014).

## Gammafactor / lengtecontractie

=

De gammafactor wordt ook wel de lorentzcontractie factor genoemd. Deze factor geeft de lengtemaat aan die een bepaalde waarnemer ziet. Iedere waarnemer heeft een ander referentiekader en neemt daardoor dingen anders waar. Een waarnemer kan niet vaststellen of hij degene is die beweegt of dat de dingen om hem heen bewegen. Echter, kan lengtecontractie alleen plaatsvinden bij snelheden die richting de lichtsnelheid gaan. (De Jong, 2008)

# Kunnen wij de theorie achter de speciale relativiteitstheorie toepassen in Python?

Om een simulatie van de speciale relativiteitstheorie mogelijk te kunnen maken in Python gebruiken wij Pygame, dat is een library/framework voor Python. Pygame maakt het makkelijk om plaatjes, vormen en tekst te tekenen in een venster. Zonder Pygame is het veel lastiger om dat voor elkaar te krijgen en dat zou ook tijdverspilling zijn en we willen ons kunnen focussen op het maken van de simulatie en niet op het uitzoeken hoe we dingen kunnen tekenen met Python.

De theorie passen wij in onze simulatie op twee manieren toe door middel van twee verschillende verschijnselen die optreden als objecten hele hoge snelheden behalen. Deze twee verschijnselen zijn tijddilatatie en lengtecontractie, oftewel verandering in de tijd waarin een object zich bevindt en de verandering van de lengte van dat object.

Het toepassen van de gammafactor is natuurlijk erg eenvoudig in Python, omdat het een wiskundige formule is.

|  |
| --- |
| gammafactor = 1/((1-(v/c)\*\*2)\*\*0.5) |

In deze formule doen we eerst 1 - (v/c)\*\*2, dit is hetzelfde als daarna doen we die waarde \*\*0.5, oftewel tot de macht 0,5. Dit is hetzelfde als het nemen van de wortel, dus staat er eigenlijk . Daarna delen we 1 door deze waarde en dan krijgen we dus dit: , dat is precies de formule voor de gammafactor. Deze gammafactor kunnen we gebruiken om de twee verschijnselen die we willen laten zien te kunnen laten zien. Zoals bij de theorie is uitgelegd geeft de gammafactor namelijk aan hoe sterk de lengte van een object verandert en hoe sterk de tijd langzamer zal gaan. Wij gebruiken de gammafactor dus om de lengte van bijvoorbeeld een plaatje van een trein te veranderen of om bijvoorbeeld een klok langzamer te laten lopen. De waarde van de snelheid () van het object kan in deze formule niet gelijk zijn aan de lichtsnelheid (), stel dat dat wel zo is, dan staat er eigenlijk en delen door 0 mag niet.

De gamma factor is heel belangrijk om de speciale relativiteitstheorie te kunnen visualiseren. Met de gammafactor kunnen we de lengte van de trein berekenen en ook de snelheid waarop de klokken lopen berekenen. Voor het berekenen van de lengte van de trein hebben wij deze code:

|  |
| --- |
| self.new\_length = self.length / gammafactor self.percentage = (1 / gamma\_factor)\*100 |

In de eerste regel delen wij de huidige lengte van de trein door de gammafactor, waardoor new\_length kleiner wordt dan length. Ook rekenen wij op de tweede regel uit hoeveel procent de nieuwe lengte is van de echte lengte. Dit percentage laten wij dan zien in de simulatie als extra informatie, verder heeft het geen invloed. Daarna gebruiken wij deze code:

|  |
| --- |
| self.rectangle = pygame.Rect(  self.x\_mid\_position-(self.new\_length / 2),  self.y\_mid\_position-(self.height / 2), self.new\_length, self.height  ) |

Deze code gebruikt de Rect functie van Pygame om een rechthoek te maken waarin de trein past. Voor de lengte van deze rechthoek wordt new\_length gebruikt. Hierdoor krijgen we een rechthoek met de juiste lengte voor de ingestelde snelheid van de trein.

Voor de klokken hebben we de volgende code gebruikt:

|  |
| --- |
| self.time += delta\_time / gammafactor |

self.time geeft hier aan hoe lang de klok nu eigenlijk al loopt en elke hoeveelheid tijd doen we daar een waarde, delta\_time, bij. delta\_time wordt gedeeld door de gammafactor, zodat de klok langzamer loopt, omdat deze sneller gaat. Vrij simpel dus. We laten echter twee klokken zien en niet één, maar bij de andere klok delen we de delta\_time niet door de gammafactor, zodat de gebruiker kan zien hoe de tijd normaal zou lopen als de klok stil zou staan. We laten ook nog twee analoge klokken zien, voor deze klokken is het ook nodig om de radialen te berekenen, zodat we weten waar de wijzer heen moet wijzen.

Dit is natuurlijk niet alle code die we hebben gebruikt voor onze simulatie, maar dit is wel de belangrijkste code die we hebben gebruikt, omdat deze code het mogelijk maakt om de speciale relativiteitstheorie te laten zien.

# Kunnen wij door middel van code in Python een overzichtelijk beeld creëren die de effecten van de speciale relativiteitstheorie duidelijk visualiseert?

Wij willen de theorie toepassen op een visuele manier, waardoor het makkelijk te begrijpen wordt hoe lengtecontractie en tijddilatatie werken. We hebben in ons programma verschillende schermen gemaakt. Op deze schermen zijn een trein, analoge en digitale klokken en een treinstation te zien. Boven de trein, de klokken en het station staat een slider. Met de muis of met de pijltjestoetsen (links en rechts) kan de gebruiker de snelheid waarmee de trein, de rechter klok of het station gaan aanpassen van 0% tot en met 99,9999% van de lichtsnelheid (c). De snelheid van een object met massa mag nooit gelijk zijn aan de lichtsnelheid (Van de Heisteeg, 2021). Als dat wel mogelijk zou zijn leidt dat tot een deling door 0 in de formule van de gammafactor, zoals hiervoor al is uitgelegd. De trein zal na het selecteren van een snelheid lengtecontractie vertonen, doordat de lengte van de trein verandert en de rechter klok zal tijddilatatie vertonen, door op een andere manier te gaan lopen; dat zijn precies de twee verschijnselen die wij willen tonen.

Met Python is het dus mogelijk om een visuele interface te maken waarmee de gebruiker kan spelen. Onze interface bevat bovenaan een slider waarmee de gebruiker de snelheid aan kan passen. Boven die slider staan de snelheid in drie eenheden: het deel van de lichtsnelheid, meter per seconde en kilometer per uur. De laatste twee eenheden staan in wetenschappelijke notatie. Onder deze slider staat de trein of staan de klokken. De trein verandert van lengte en de klokken gaan verschillend lopen bij het veranderen van de snelheid. In totaal hebben we vier klokken gemaakt: twee klokken met een ronde wijzerplaat en een wijzer en twee digitale klokken met alleen getallen. Boven de trein staat ook het resterende percentage van de lengte.

Zoals eerder gezegd kan de snelheid van de trein niet gelijk zijn aan de lichtsnelheid (oftewel 1c). In de simulatie laten we echter wel zien wanneer de maximale snelheid is bereikt, terwijl dat eigenlijk is. Dit hebben wij gedaan om de interface niet te onrustig te maken. Als de gebruiker op d drukt op het toetsenbord ziet de gebruiker wel staan.

De gebruiker kan daarnaast ook op de spatieknop drukken om de simulatie te pauzeren. Om van de trein naar de klokken te gaan, gebruikt de gebruiker het pijltje dat omhoog wijst, andersom gebruikt de gebruiker het pijltje dat naar beneden wijst. Met de pijltjes naar links en naar rechts kan de gebruiker de snelheid aanpassen, maar dat kan ook met de muis.

# Gebruiksaanwijzing

Dit is de gebruiksaanwijzing voor onze simulatie.

**Spatie** = zet de klokken op pauze of start de klokken

**R** = reset de timers

**P** = wisselt van perspectief

**D** = het aantal decimalen van de snelheid met de eenheid

**Pijltje naar boven** = volgende simulatie

**Pijltje naar onder** = vorige simulatie

**Klikken of ingedrukt houden op de snelheids-slider** = verandert de snelheid naar de muispositie

**Pijltje naar rechts** = (afgerond) bij de snelheid toevoegen

**Pijltje naar links** = (afgerond) bij de snelheid toevoegen

**Simulatie 1** = tijddilatatie

**Simulatie 2** = lengtecontractie

**Simulatie 3** = tijddilatatie + lengtecontractie

Linksboven is huidige de simulatie + het perspectief te zien als je vanuit perspectief A kijkt. Als het perspectief B is, dan is de huidige simulatie en perspectief aan de rechterkant te zien. Dit komt overeen met de klokken die links (klok A) en rechts (klok B) staan.

# Conclusie

Wij kunnen positief antwoorden op alle drie de deelvragen. We hebben een degelijke beschrijving kunnen geven over het effect van een snelheid dichtbij op tijd en lengte. We gaan diep in op de lichtsnelheid zelf en tijddilatatie en lengtecontractie om de lezer een begrip te geven van deze verschijnselen. Het is ook gelukt om de theorie achter de speciale relativiteitstheorie toe te passen in Python. Bij gebruik van de simulaties gedraagt het programma zich zoals de speciale relativiteitstheorie voorschrijft: tijd gaat slomer en lengte wordt korter bij snelheden die dichtbij de lichtsnelheid liggen. Het ons gelukt om dit overzichtelijk te maken zodat de gebruiker zich de speciale relativiteitstheorie uitstekend kan visualiseren, hoewel deze laatste deelvraag meer subjectief is.

# Discussie

**Koen:**  
In mijn optiek is dit project zeer soepel verlopen. De samenwerking liep goed door en de groepsleden waren bereid toereikend arbeid te verrichten. Het resultaat was erg bevredigend. De simulatie en het verslag is hier een uiting van. Met dit project hebben wij de speciale relativiteitstheorie het digitale tijdperk ingetrokken. De het coderen van de simulatie in python verliep erg goed. Natuurlijk liep ik af en toe vast met het coderen, maar dat hoort nou eenmaal bij programmeren. Ik vind dat het eindresultaat erg goed gelukt is.

**Nico:**

Mijns inziens is het project opperbest gegaan. We hebben veel informatie kunnen vinden over de speciale relativiteitstheorie en we hebben deze theorie ook goed kunnen benutten in Python. We hebben een fraaie simulatie kunnen maken waarin de verschijnselen van deze theorie goed zichtbaar worden gemaakt, waardoor een gebruiker de theorie geriefelijk kan begrijpen. Het was wel netelig om dienstige informatie te vinden over de speciale relativiteitstheorie, omdat veel van de bronnen die wij hadden gevonden erg gecompliceerd waren en heel diep op het onderwerp in gingen. Hierdoor is onze uitleg ook erg breedvoerig geworden, eigenlijk iets te breedvoerig voor waar we de theorie eigenlijk voor wilden benutten.

**Mitchell:**

Naar mijn mening is het resultaat van dit project geworden zoals we gehoopt hadden. De samenwerking was goed en iedereen wist wat hij/zij kon doen. De simulatie is met elke versie steeds uitgebreider geworden, omdat we onze focus legde op altijd een versie te hebben die helemaal af is. Ik ben tevreden over hoe we de speciale relativiteitstheorie in een simulatie hebben kunnen weergegeven.

**Sven:**

Het verslag is naar mijn mening erg goed gelukt. De uitleg van de theorie is uitgebreid en gebaseerd op goede bronnen. De simulatie is erg goed gelukt en het is duidelijk hoe deze is gecreëerd. De simulatie is bovendien erg gebruiksvriendelijk met duidelijke gebruiksaanwijzingen in dit document. Het enige dat misschien iets beter kon is dat de stukjes over de speciale relativiteitstheorie is simpeler uitgelegd konden worden, maar aan de andere kant is dat altijd erg lastig bij een onderwerp als de speciale relativiteitstheorie.

**Simone:**

In mijn opinie is dit project een goede voorbereiding op het studeren, omdat je zelf onderzoek moest doen en betrouwbare bronnen moest zoeken. We waren allemaal zeer gemotiveerd om aan dit project te werken, omdat wij onze fantasie mochten gebruiken voor het onderwerp hiervan. Zo konden we een onderwerp kiezen dat ons allemaal aansprak en dat geeft meer motivatie. De samenwerking was ook zeer goed, we hebben alle stappen die we wilden zetten overlegt en samen uitgedacht. Naar mijn idee is onze hoofdvraag ook goed uitgepakt in de simulatie. De impact van de lengtecontractie en de tijddilatatie is ook voor mij nog verder verduidelijkt door de gemaakte simulatie.

# Verwijzingen

**GitHub repository:**

<https://github.com/Koen36/PO7informatica>

Hierin staat de code en het programma om de simulatie uit te voeren. Dezelfde bestanden zitten in de inleveropdracht in het zip-bestand.

**Trello bord:** <https://trello.com/invite/b/AVEv4pjI/d894c06f6cfa8acf2abf80f127e69e91/informatica-po7>

# Bronnen

De Jong, S. (2008, 21 augustus). *Speciale Relativiteitstheorie.* Radboud University. Geraadpleegd op 9 februari 2022, van <https://www.hef.ru.nl/~sijbrand/srt.pdf>

Flokstra, J., Groenewold, A., Hooyman, K., Kootwijk, C., Kortland, K., Bosman, M., … Vink, H. (2020). *K4 Relativiteitstheorie.* Geraadpleegd op 9 februari 2022.

Greene, B. (2014, 14 januari). *Special Relativity.* Geraadpleegd op 7 maart 2022, van https://worldscienceu.com/courses/special-relativity-world-science-u

Price, R., & Zizka, J. (2007, 16 oktober). *Experimental Determination of the Speed of Light by the Foucault Method.* Kansas State University. Geraadpleegd op 9 februari 2022, van <https://phys.ksu.edu/personal/rprice/SpeedofLight.pdf>

Van de Heisteeg, D. (2021, 1 oktober). *Sneller dan het licht?* the Quantum Universe. Geraadpleegd op 8 maart 2022, van <https://www.quantumuniverse.nl/sneller-dan-het-licht>