



Universiteit Antwerpen

FACULTEIT WETENSCHAPPEN

DEPARTEMENT FYSICA

ACADEMIEJAAR 2017-2018

Bachelorproef:
Simulatie van elektron optische elementen

Auteur
Robin Brabants

Promotor
Johan Verbeeck

14 mei 2018

Abstract

The goal of this theses was to develop an open source software packet for the simulation of magnetic and electric fields for a given setup of field producing elements and the trajectories of particles through these fields. This program distinguishes itself from commercial packages in the way these fields are calculated and in the possibility and simplicity to adapt and improve the code. An analytical approach for the calculation of magnetic fields was chosen and for the electric fields a gridless numerical method, called Walk on Spheres, was tested out. The programming language Python was chosen for this software and XML files were used for data input. The trajectory of the particle and some more info about the subsequent coordinates of the trajectory will be saved as a csv file which is easily read in by other applications. The chosen numerical method (WOS) proved to be difficult to expand for a correct and fast calculation of electric fields. In this report the structure and implementation of the program itself will be discussed and reviewed. The analytical expressions for magnetic fields will be derived and some important functions will be explained. Some output results of the program will be shown as well.

Inhoudsopgave

1	Inleiding	3
2	Methode	4
2.1	Implementatie	4
2.2	Functionaliteiten	7
2.2.1	Analytische velden	7
2.2.1.1	Magnetisch veld eindige rechte geleider	7
2.2.1.2	Magnetisch veld rechthoekige spoel	10
2.2.1.3	Magnetisch veld cirkelvormige geleider	12
2.2.1.4	Magnetisch veld cirkelvormige spoel	15
2.2.2	Walk on Spheres methode	16
2.2.3	Beweging deeltje in magnetisch en elektrisch veld	17
2.2.4	Vector aanmaken a.d.h.v. 2 hoeken	20
2.2.5	Afstandsbepaling	22
2.2.5.1	Afstand tot een cirkelschijf	22
2.2.5.2	Afstand tot een cilinder	22
2.3	Discussie	23
2.3.1	Walk on Spheres	23
2.3.2	Optimalisaties en uitbreidingen	25
3	Resultaten	26
4	Conclusie	33

1 Inleiding

Het doel van deze bachelorproef is het ontwikkelen van een open source software pakket voor de simulatie van magnetische en elektrische velden rond een willekeurige opstelling van stroomdoorlopen geleiders en potentiaaloppervlakken. Ook zal het pad van geladen deeltjes doorheen deze velden numeriek bepaald worden. Er bestaan al tal van commerciële programma's die gelijkaardige functionaliteiten hebben, echter hier ligt de nadruk op het ontwikkelen van een open source pakket dat eenvoudig uit te breiden is en aan te passen voor eigen specifiek gebruik in o.a. elektronen microscopie, deeltjesversnellers... Ook werd er hier niet gekozen voor traditionele numerieke technieken zoals de Finite Difference, Finite Element, Charge Simulation Method... Er werd eerder geopteerd voor gridloze methoden vermits de velden niet in de gehele 3D ruimte bepaald moeten worden, maar enkel in de punten nodig voor de deeltjesbanen te bepalen. Dit zou onnodig rekenwerk kunnen besparen. Voor de magnetische velden werden er analytische uitdrukkingen gebruikt vermits er met een beperkt aantal elementen zoals rechte en kromlijnige geleiders er al vele bruikbare configuraties nauwkeurig uitgerekend konden worden. Voor elektrische velden werd er gekozen om de numerieke, gridloze Walk On Spheres methode uit te proberen o.a. omdat dit een groot aantal mogelijke potentiaaloppervlakken kan uitrekenen zolang de minimale afstand tot deze oppervlakken kan bepaald worden.

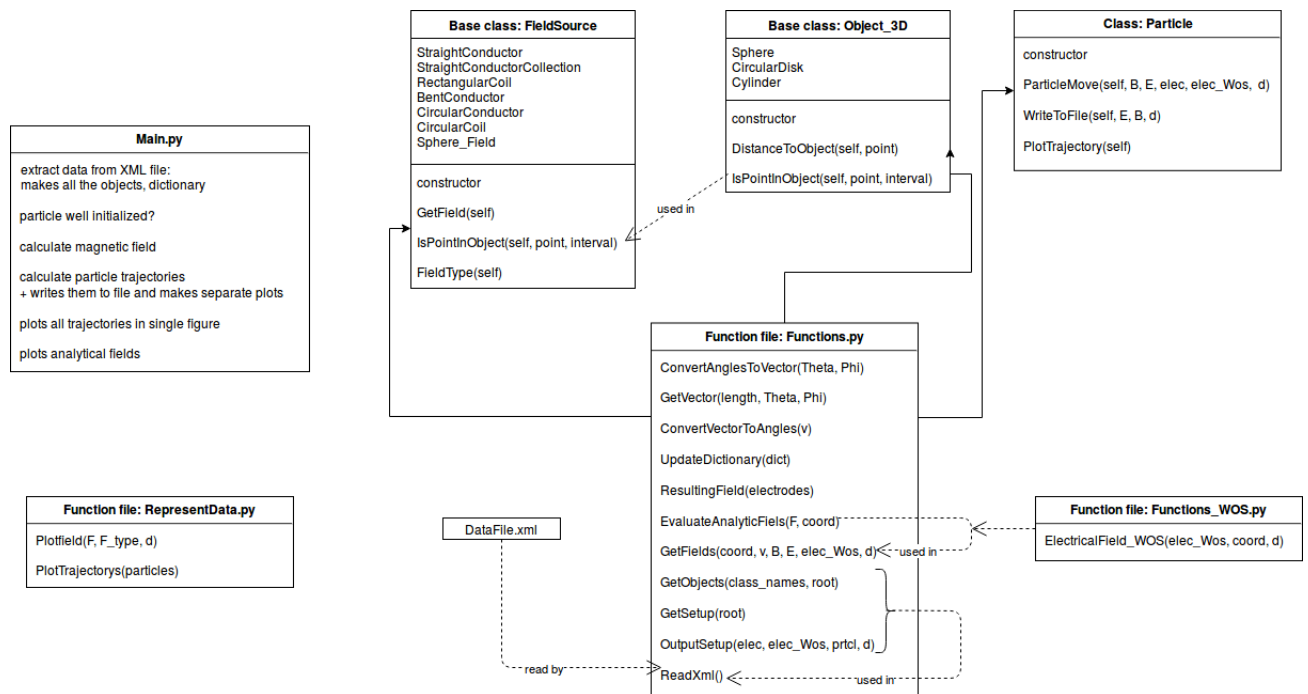
Er werd geopteerd om te programmeren in Python voornamelijk omdat dit een heel populaire introductietaal is in informatica cursussen. Ook de eenvoudige implementatie en de goede leesbaarheid van Python code maakt dit programma voor een breder publiek toegankelijk. Er werd ook gekozen voor data-invoer a.d.h.v. een xml-file. Dit zorgt voor een snellere uitvoering van het programma en een makkelijkere herbruikbaarheid van de opstelling vergeleken met invoer van data via een interface. Ook werd er geen gebruik gemaakt van een interface, maar eerder van de terminal vermits een interface voorlopig nog niet noodzakelijk was voor een simpele uitvoering van het programma.

In dit verslag zal kort en bondig de structuur van het programma besproken worden en enkele belangrijke analytische velduitdrukkingen, belangrijke functies... toegelicht worden. De code van het programma zelf is ook voorzien van voldoende commentaar om de structuur en werking ervan te begrijpen. Ook zal er een sectie gewijd worden aan de bespreking van enkele huidige beperkingen, disfunctionaliteiten en mogelijke uitbreidingen. Daarna zullen er ook nog enkele resultaten van berekende deeltjesbanen getoond worden. Een beperkte kennis van Python is vereist om de sectie "Implementatie" door te nemen.

2 Methode

2.1 Implementatie

In deze sectie wordt de structuur en de werking van het programma kort toegelicht. Voor meer informatie kan er naar de commentaren in de code zelf gekeken worden of naar de sectie "Functionaliteiten" waar enkele belangrijke delen van de code worden toegelicht. In figuur 1 is er een schematische weergave van het programma te zien als wijze van overzicht waarnaar soms zal worden verwezen.



Figuur 1 schematische weergave van het programma

Zoals vermeld in de inleiding wordt er gebruik gemaakt van een XML-file om al de informatie nodig voor de uitvoering van het programma in te geven. Onder de tag *Elements* wordt er informatie opgegeven omtrent de magnetische veldproducerende objecten en onder de tag *Elements_WOS* worden de elektrische veldproducerende objecten beschreven. Er wordt telkens informatie opgegeven over de oriëntatie van de objecten in de 3D ruimte en hun vorm, alsook hun stroom of potentiaal. Er kunnen meerdere objecten van dezelfde soort worden opgegeven: zo kunnen er bijvoorbeeld 3 *Straight-Conductors* worden geïnitieerd met een verschillende oriëntatie/vorm/stroom. Verder wordt er ook nog informatie opgegeven omtrent de deeltjes waarvan de banen zullen worden bepaald. Tot slot worden er onder de tags *Trajectory*, *WOS* en *Output* nog waarden opgegeven die respectievelijk te maken hebben met de numerieke methode voor het berekenen van de deeltjesbanen, de Walk on Spheres methode en de plots/files die het programma produceert. Voor meer informatie omtrent de verschillende opgegeven variabelen kan er naar het bestand *DataFile.xml* gekeken worden.

Al de verschillende functionaliteiten van het programma zullen worden aangeroepen vanuit Main.py waarvan een korte beschrijving te zien is in het schema. Ten eerste zal de XML file (bv.: DataFile.xml) worden ingelezen door de functie ReadXml die respectievelijk de functies GetObjects, GetSetup en OutPutSetup aanroept. Deze datafile moet op dezelfde hoogte worden geplaatst in de folder als Main.py en zijn naam wordt gebruikt in het commando om het programma uit te voeren (zie later). GetObjects genereert een lijst van klasse objecten a.d.h.v. de datafile (veldproducerende objecten en de deeltjes waarvan de banen berekend zullen worden). Dit gebeurt door te kijken of er ingeschakelde elementen in de datafile zijn wiens namen overeenstemmen met de klasse Particle of de klassen die erven van de basis klassen FieldSource en Object_3D (bv.: StraightCnductor). De constructors van deze klassen worden dan aangeroepen met als input de self variabelen (variabelen die behoren tot een klasse object) die automatisch uit de datafile worden gehaald (indien een self variabele niet overeenstemt met een variabele uit de datafile, dan geeft het programma een error weer en wordt zijn uitvoering stop gezet). De naam van een klasse object is de naam van de klasse zelf met een nummer achteraan toegevoegd (Bv.: StraightConductor1, StraightConductor2, BentConductor1). GetSetup genereert een dictionary met alle resterende informatie uit de datafile die nog niet is opgeslagen als self variabelen van klasse objecten.

OutputSetup zal informatie op het scherm printen en naar een bestand schrijven omtrent de veldproducerende objecten en de deeltjes en vragen aan de gebruiker of de getoonde informatie correct is alvorens er verder wordt gegaan met de uitvoering van het programma.

ReadXml geeft dus alle geïnitieerde objecten en een dictionary met informatie terug.

Voor het programma van start gaat met de berekeningen wordt er nog even nagegaan of de startposities van de deeltjes correct zijn. Zo mogen de deeltjes zich niet in een veldproducerend object bevinden of starten buiten het opgegeven gebied waarin de deeltjesbanen worden bepaald (TrajectoryBoundaries).

Daarna zal het magnetische veld worden bepaald a.d.h.v. de ResultingField functie die als input de lijst van geïnitieerde magnetische veldproducerende objecten meekrijgt (objecten wiens klasse erft van de FieldSource basis klasse). Deze functie roept telkens de memberfunctie GetField aan op al de opgegeven objecten en sommeert al de velden. GetField geeft een analytische uitdrukking terug in functie van de 3 ruimtecoördinaten x , y en z . De achterliggende theorie van deze veldfuncties wordt besproken in de sectie "Analytische velden". De Sphere_Field klasse geeft het elektrisch veld van een sfeer weer en is enkel aanwezig voor het uittesten van de Walk on Spheres methode (zie de sectie: "Discussie").

Het elektrische veld wordt hier nog niet bepaald vermits deze enkel in een bepaald punt in de ruimte zal geëvalueerd worden door de ElectricalField_WOS functie. De elektrische veldproducerende objecten zijn de objecten wiens klasse erft van de Object_3D basis klasse.

Nu zullen er voor alle deeltjes objecten trajecten worden berekend a.d.h.v. de memberfunctie ParticleMove (zie de sectie: "Beweging deeltje in magnetisch en elektrisch veld").

De magnetische en elektrische velden in een bepaald punt worden geëvalueerd a.d.h.v. de Getfields functie. Deze functie roept op zijn beurt de EvaluateAnalyticField functie aan die het magnetisch veld in een bepaald punt evalueert a.d.h.v. de eerder berekende analytische uitdrukking. Eventuele integralen die voorkomen in de uitdrukking worden numeriek bepaald. Het elektrische veld wordt bepaald a.d.h.v. de ElectricalField_WOS functie (zie de secties: "Walk on Spheres methode" en "Discussie"). De Walk on Spheres methode heeft functies nodig om de minimale afstand van een oppervlak tot een punt in de ruimte te berekenen. Dit gebeurt door de memberfunctie DistanceToObject (zie de sectie: "Afstandsbepaling") aan te roepen op de lijst van elektrische veldproducerende objecten. Indien de afstand negatief is, wilt dit zeggen dat het deeltje zich in een object bevindt. Tot slot wordt er ook nog een relativistische correctie doorgevoerd in de GetFields functie.

Gedurende de iteraties van de ParticleMove functie wordt er nagegaan of een deeltje botst met een veldproducerend object door de IsPointInObject memberfunctie aan te roepen op de lijst van objecten. Deze functie bepaalt eerst de afstand tot een object a.d.h.v. de DistanceToPoint functie. Indien de afstand een negatieve waarde heeft, bevindt het deeltje zich in een object en heeft het deeltje dus een botsing ondergaan. Echter door de discretisatie van de deeltjesbaan kan het zijn dat een deeltje gebotst heeft met een object, maar zich in geen enkel berekend punt van de baan in een object heeft bevonden. Daarom wordt gezegd dat een deeltje ook een botsing heeft ondergaan indien de afstand van een deeltje tot een object kleiner is dan een bepaalde waarde (interval). Voor elektrische veldproducerende objecten kan de memberfunctie IsPointInObject rechtstreeks worden aangeroepen, echter voor magnetische veldproducerende objecten wordt er eerst een overeenkomstig (zelfde geometrie) Object_3D object geïnitieerd om daarop dan de IsPointInObject memberfunctie aan te roepen. De opeenvolgende posities van een deeltje en extra informatie in deze punten zoals snelheid, versnelling en veldsterkte wordt opgeslagen in een dictionary. Deze dictionary wordt een self variabele van het deeltje object (self.Trajectory)

De memberfunctie WriteToFile schrijft de informatie opgeslagen in self.Trajectory naar een csv bestand dat makkelijk te importeren is in excel, matlab... Zoals de naam doet vermoeden, plot de PlotTrajectory memberfunctie de berekende deeltjesbaan.

De functies PlotTrajectorys en Plotfield maken respectievelijk 3D grafieken van alle deeltjesbanen te samen en het magnetisch veld.

Functions.py bevat nog enkele andere functies zoals ConvertAnglesToVector (zie de sectie: "Vector aanmaken a.d.h.v. 2 hoeken"), GetVector, ConvertVectorToAngles en UpdateDictionary die in verschillende onderdelen van het programma gebruikt worden. De werking van deze functies spreekt meestal voor zichzelf.

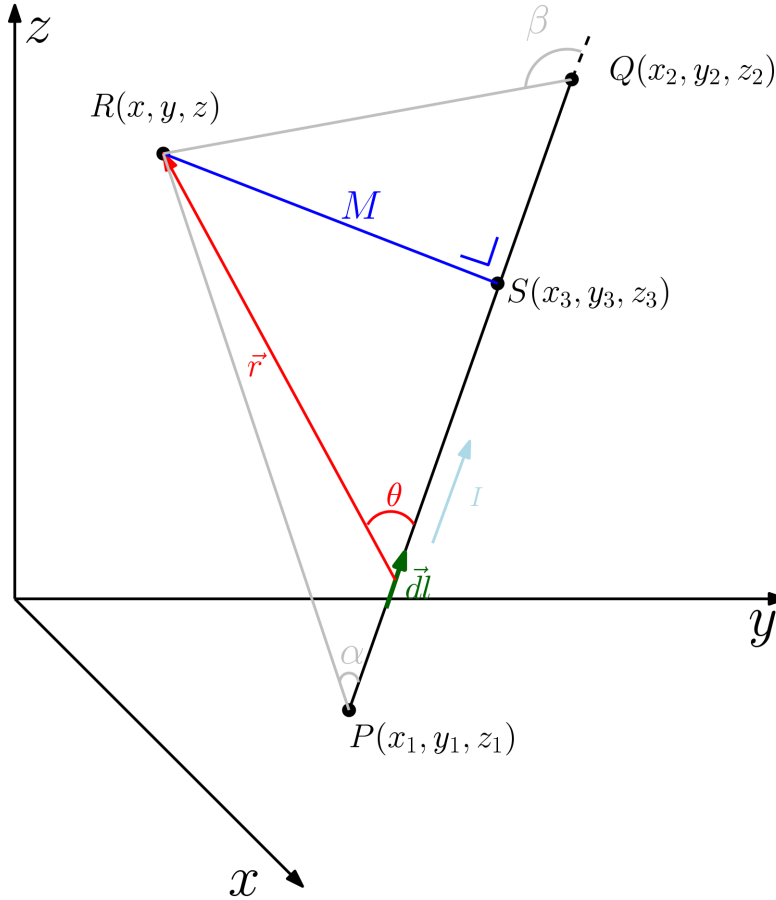
Het programma wordt uitgevoerd a.d.h.v. een terminal door eerst te navigeren naar de folder van het programma (code) en vervolgens het commando "python3.6 Main.py <naam van het XML bestand>" in te geven. Voor een betere weergave van enkele tussenresultaten, zal de terminal best in volledige schermweergave worden gebruikt. Dit programma gebruikt enkele externe modules waarvan het installatieproces voor Linux gebruikers beschreven staat in de README file.

2.2 Functionaliteiten

2.2.1 Analytische velden

2.2.1.1 Magnetisch veld eindige rechte geleider

Het is de bedoeling om een analytische uitdrukking te vinden voor het magnetisch veld veroorzaakt door een eindige rechte geleider in een 3D ruimte. Een visualisatie van het probleem is terug te vinden in figuur 1.



Figuur 1 Rechte geleider $[PQ]$ waardoor een stroom I loopt

Hierbij is de geleider gedefinieerd als het lijnstuk $[PQ]$ waarbij de conventionele stroomzin gelegen is in de richting PQ . Het punt R is het punt waarin men het magnetische veld gaat onderzoeken. De doelstelling is om een uitdrukking te vinden voor het magnetisch veld i.f.v. x , y en z waarbij de coördinaten van P en Q als bekend worden beschouwd. S is het snijpunt van de rechte PQ met de rechte door het punt R loodrecht op PQ .

Het magnetisch veld $d\mathbf{B}$ in het punt R veroorzaakt door het stroomdoorlopen elementje $d\mathbf{l}$ kan bepaald worden a.d.h.v. de regel van Laplace (in de vrije ruimte):

$$d\mathbf{B} = \frac{\mu_0 I}{4\pi} \frac{d\mathbf{l} \times \mathbf{e}_r}{r^2} \quad (1)$$

met μ_0 de permeabiliteit van het vacuüm en I de stroomsterkte.

De grootte van $d\mathbf{B}$, genoteerd als dB , wordt als volgt berekend:

$$dB = \frac{\mu_0 I}{4\pi} \frac{\sin \theta dl}{r^2} \quad (2)$$

met θ de hoek tussen de vectoren \mathbf{PQ} en \mathbf{r}

Vermits de vectoren $d\mathbf{B}$ allen dezelfde richting hebben (loodrecht op $d\mathbf{l}$ en \mathbf{e}_r), kan de grootte van het resulterende veld, B , als volgt bepaald worden:

$$B = \frac{\mu_0 I}{4\pi} \int_a^b \frac{\sin \theta dl}{r^2} \quad (3)$$

l is een grootte op de rechte PQ met als nulpunt het punt S . l neemt toe volgens de stroomzin I . Het maximum van l wordt bereikt in Q en wordt gelijkgesteld aan b (positief), het minimum van l wordt bereikt in P en wordt gelijkgesteld aan a (negatief). Merk dus op dat geldt: $|a| + |b| = L$ (L is de lengte van de geleider).

De lengte van de vector \mathbf{r} , genoteerd als r , kan nu als volgt uitgedrukt worden:

$$r = \frac{M}{\sin \theta} \quad (4)$$

met $M = |RS|$.

Een uitdrukking voor l wordt als volgt bepaald:

$$\tan \theta = \frac{M}{-l} \quad (5)$$

$$l = -\frac{M}{\tan \theta} \quad (6)$$

$$l = -M \cot \theta \quad (7)$$

Door differentiatie van vergelijking (7) wordt dl bekomen:

$$dl = \frac{M}{\sin^2 \theta} d\theta \quad (8)$$

Invullen van de vergelijkingen (8) en (4) in vergelijking (3) resulteert in de volgende vergelijking (de integratiegrenzen a en b worden nu vervangen door de hoeken α en β (zie Figuur 1)):

$$B = \frac{\mu_0 I}{4\pi M} \int_{\alpha}^{\beta} \sin \theta d\theta \quad (9)$$

Voor de grootte van het magnetisch veld wordt dus volgende uitdrukking verkregen:

$$B = \frac{\mu_0 I}{4\pi M} (\cos \alpha - \cos \beta) \quad (10)$$

De grootheden M , $\cos \alpha$ en $\cos \beta$ zullen nu nog worden uitgedrukt i.f.v. de coördinaten van de punten P , Q en R . Hiervoor zullen eerst de coördinaten van het punt S berekend worden.

De parametervergelijking van het punt S dat zich op het lijnstuk $[PQ]$ bevindt, is hieronder weergegeven: ($i = 1, 2, 3$ met $(x)_1 = x, (x)_2 = y, (x)_3 = z$):

$$(x_3)_i = (x_1)_i + t[(x_2)_i - (x_1)_i] \quad (11)$$

met $t \in [0, 1]$. Indien $t = 0$ bevindt men zich in punt P en indien $t = 1$ bevindt men zich in punt Q . De parameter t moet nog bepaald worden om de coördinaten van S eenduidig vast te leggen. Vermits $PQ \perp RS$, gelden de volgende vergelijkingen:

$$\mathbf{PQ} \cdot \mathbf{RS} = 0 \quad (12)$$

$$(x_2 - x_1, y_2 - y_1, z_2 - z_1) \cdot (x - x_3, y - y_3, z - z_3) = 0 \quad (13)$$

$$\sum_{i=1}^3 ((x_2)_i - (x_1)_i) \cdot ((x)_i - (x_1)_i - t[(x_2)_i - (x_1)_i]) = 0 \quad (14)$$

$$\frac{\sum_{i=1}^3 ((x_2)_i - (x_1)_i) \cdot ((x)_i - (x_1)_i)}{\sum_{i=1}^3 ((x_2)_i - (x_1)_i)^2} = t \quad (15)$$

En dus:

$$(x_3)_i = (x_1)_i + \left(\frac{\sum_{i=1}^3 ((x_2)_i - (x_1)_i) \cdot ((x)_i - (x_1)_i)}{\sum_{i=1}^3 ((x_2)_i - (x_1)_i)^2} \right) [(x_2)_i - (x_1)_i] \quad (16)$$

Voor M geldt:

$$M = \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2} \quad (17)$$

Voor $\cos \alpha$ geldt:

$$\cos \alpha = \frac{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2}}{\sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}} \quad (18)$$

Voor $\cos \beta$ geldt:

$$\cos \beta = -\frac{\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}}{\sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2}} \quad (19)$$

De grootte van \mathbf{B} is bepaald, nu zal de richting hiervan bepaald worden. De richting van \mathbf{B} staat loodrecht op de vectoren \mathbf{PQ} en \mathbf{PR} en heeft dus dezelfde richting als $\mathbf{PQ} \times \mathbf{PR}$ (genoteerd als \mathbf{n}).

$$\mathbf{PQ} \times \mathbf{PR} = (x_2 - x_1, y_2 - y_1, z_2 - z_1) \times (x - x_1, y - y_1, z - z_1) \quad (20)$$

$$\mathbf{n} = \begin{bmatrix} (y_2 - y_1) \cdot (z - z_1) - (z_2 - z_1) \cdot (y - y_1) \\ (z_2 - z_1) \cdot (x - x_1) - (x_2 - x_1) \cdot (z - z_1) \\ (x_2 - x_1) \cdot (y - y_1) - (y_2 - y_1) \cdot (x - x_1) \end{bmatrix} \quad (21)$$

Om uiteindelijk een uitdrukking voor \mathbf{B} te bekomen zal de vector in vergelijking (21) genormaliseerd worden. De eenheidsvector volgens de richting van \mathbf{B} wordt genoteerd als $\mathbf{e_B}$:

$$\mathbf{e_B} = \frac{1}{K} \cdot \begin{bmatrix} (y_2 - y_1) \cdot (z - z_1) - (z_2 - z_1) \cdot (y - y_1) \\ (z_2 - z_1) \cdot (x - x_1) - (x_2 - x_1) \cdot (z - z_1) \\ (x_2 - x_1) \cdot (y - y_1) - (y_2 - y_1) \cdot (x - x_1) \end{bmatrix} \quad (22)$$

met K de grootte van de vector \mathbf{n} .

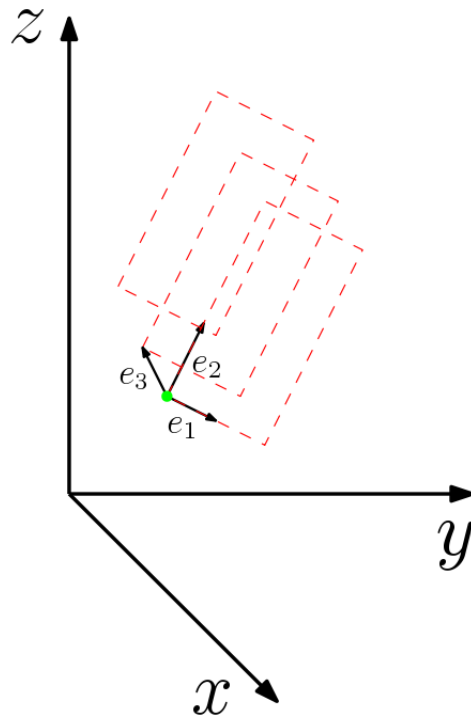
$$K = \left([(y_2 - y_1) \cdot (z - z_1) - (z_2 - z_1) \cdot (y - y_1)]^2 + [(z_2 - z_1) \cdot (x - x_1) - (x_2 - x_1) \cdot (z - z_1)]^2 + [(x_2 - x_1) \cdot (y - y_1) - (y_2 - y_1) \cdot (x - x_1)]^2 \right)^{\frac{1}{2}} \quad (23)$$

Om nu uiteindelijk een analytische uitdrukking te vinden voor het \mathbf{B} -veld in de 3D ruimte in functie van de plaatscoördinaten x, y en z , zullen volgende stappen doorlopen worden:

1. Maak de constanten π en μ_0 aan.
2. Initialiseer de geleider met de eindpunten P en Q door respectievelijk hun coördinaten (x_1, y_1, z_1) en (x_2, y_2, z_3) op te geven. Initialiseer ook de stroomsterkte I .
3. Maak de formules voor $(x_3)_i$ (16), M (17), $\cos \alpha$ (18), $\cos \beta$ (19) en \mathbf{e}_B (22) aan (deze zijn allen afhankelijk van (x, y, z) die momenteel als onbekende variabelen worden beschouwd).
4. Maak de vector \mathbf{B} aan door te stellen: $\mathbf{B} = B(10) \cdot \mathbf{e}_B(22)$.
5. Nu is het vectorveld \mathbf{B} aangemaakt veroorzaakt door een specifieke geleider i.f.v. de plaatscoördinaten (x, y, z) . Vectorvelden veroorzaakt door andere geleiders kunnen hierbij worden opgeteld als vectoren door het superpositiebeginsel. Indien het vectorveld in een bepaald punt (x, y, z) gekend moet zijn kan men de coördinaten van dit punt ingeven.

2.2.1.2 Magnetisch veld rechthoekige spoel

Het is de bedoeling om een analytische uitdrukking te vinden voor het magnetisch veld veroorzaakt door een rechthoekige spoel in een 3D ruimte. Een visualisatie van het probleem is terug te vinden in figuur 1.

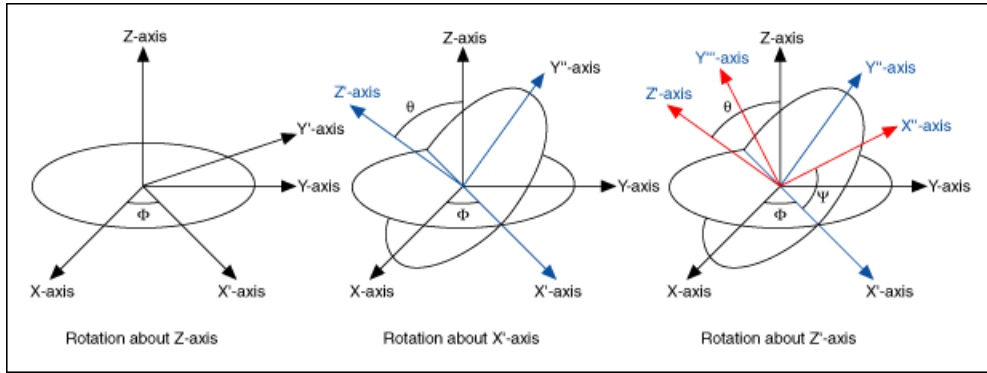


Figuur 1 Rechthoekige spoel waardoor een stroom I loopt

In deze schets van het probleem stellen de rode stippellijnen de windingen van de spoel voor. In werkelijkheid zouden deze windingen niet in 1 vlak mogen liggen, maar eerder bestaan uit aaneensluitende rechte geleiders.

De vectoren \mathbf{e}_1 en \mathbf{e}_2 stellen het vlak voor waarin de rechthoekige windingen liggen. Deze twee vectoren moeten natuurlijk loodrecht staan op elkaar. De vector \mathbf{e}_3 staat loodrecht op zowel \mathbf{e}_1 als \mathbf{e}_2 . De zin van \mathbf{e}_3 wordt zodanig gekozen dat de drie vectoren een rechtshandig assenstelsel voor de spoel vormen.

De gebruiker geeft de oriëntatie van het assenstelsel van de spoel t.o.v. het vastliggende assenstelsel van de opstelling op a.d.h.v. de drie hoeken van Euler (ϕ , θ en ψ). In figuur 2 wordt dit verduidelijkt: op de rechtse afbeelding stelt het zwarte assenstelsel het vastliggende stelsel voor en het rode assenstelsel diegene van de spoel.



Figuur 2 De hoeken van Euler [1]

Het assenstelsel van de spoel wordt dus bekomen door eerst een rotatie uit te voeren om de Z-as over ϕ , gevolgd door een rotatie om de X' -as over θ en ten slotte een rotatie om de Z' -as over ψ (zie figuur 2).

De eenheidsvectoren \mathbf{e}_1 , \mathbf{e}_2 en \mathbf{e}_3 in de basis van het vastliggende assenstelsel (\mathbf{e}_x , \mathbf{e}_y en \mathbf{e}_z) die respectievelijk overeenstemmen met de richtingen van X'' , Y'' en Z' uit figuur 2 worden bekomen uit de hoeken van Euler[2] :

$$\begin{aligned}\mathbf{e}_1 &= (\cos(\phi) \cos(\psi) - \sin(\phi) \sin(\psi) \cos(\theta)) \mathbf{e}_x + (\sin(\phi) \cos(\psi) + \cos(\phi) \sin(\psi) \cos(\theta)) \mathbf{e}_y + (\sin(\theta) \sin(\psi)) \mathbf{e}_z \\ \mathbf{e}_2 &= (-\cos(\phi) \sin(\psi) - \sin(\phi) \cos(\psi) \cos(\theta)) \mathbf{e}_x + (-\sin(\phi) \sin(\psi) + \cos(\phi) \cos(\psi) \cos(\theta)) \mathbf{e}_y + (\cos(\psi) \sin(\theta)) \mathbf{e}_z \\ \mathbf{e}_3 &= (\sin(\phi) \sin(\theta)) \mathbf{e}_x + (-\cos(\phi) \sin(\theta)) \mathbf{e}_y + (\cos(\theta)) \mathbf{e}_z\end{aligned}$$

De lengte (l), breedte (b) en hoogte (h) van de rechthoekige spoel worden opgegeven door de gebruiker en stellen respectievelijk de dimensies van de spoel voor in de \mathbf{e}_1 , \mathbf{e}_2 en \mathbf{e}_3 richting.

De oorsprong van het assenstelsel van de spoel (startpunt) wordt door de gebruiker opgegeven. Het aantal windingen (N) wordt ook door de gebruiker opgegeven.

Nu zal er bepaald worden op welke manier de rechte geleiders moeten opgesteld worden om een rechthoekige spoel te verkrijgen. Hiervoor worden simpelweg de coördinaten van enkele punten bepaald om a.d.h.v. deze punten een StraightConductorCollection te initialiseren en hierop de GetField functie aan te roepen (zie code: Elements). Een StraightConductorCollection zal in principe simpelweg de bepaalde punten verbinden met elkaar a.d.h.v. rechte geleiders waarvan de analytische velduitdrukking gekend is. Om dit uit te voeren worden er 4 vectoren aangemaakt:

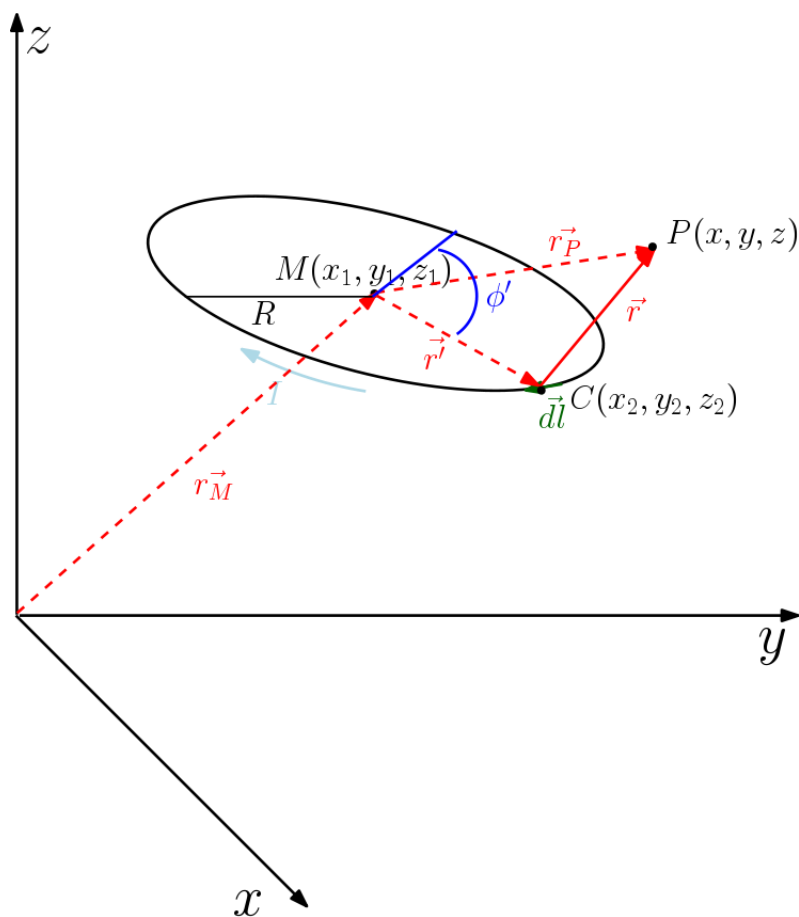
$$\begin{aligned}
\mathbf{a}_1 &= l\mathbf{e}_1 \\
\mathbf{a}_2 &= b\mathbf{e}_2 \\
\mathbf{a}_{3_1} &= \frac{l}{2b+2l} \frac{h}{N} \mathbf{e}_3 \\
\mathbf{a}_{3_2} &= \frac{b}{2b+2l} \frac{h}{N} \mathbf{e}_3
\end{aligned}$$

Deze vergelijkingen zijn geldig indien de spoel in tegen wijzerzin wordt opgewonden. Indien de spoel in wijzerzin wordt opgewonden, moeten de laatste subindices, 1 en 2, worden omgedraaid in bovenstaande vergelijkingen.

De vectoren a_1 en a_2 geven de "horizontale" dimensies van de spoel weer en de vectoren a_3 geven de fractie van "verticale" stijging weer per $\frac{1}{4}$ de winding respectievelijk voor de lengte en breedte "dimensie" van de spoel. Het is nu eenvoudig te zien dat de punten die verbonden moeten worden door rechte geleiders te bepalen zijn door te beginnen vanuit het startpunt voor de windingen. Het startpunt is natuurlijk het eerste punt waarbij de vectoren a_1 en a_{3_1} bij moeten worden opgeteld om het tweede punt te bekomen. Het derde punt wordt gevonden door de vectoren a_2 en a_{3_2} op te tellen bij het tweede punt. De vector a_1 wordt van het derde punt afgetrokken en de vector a_{3_1} wordt hierbij opgeteld om het vierde punt te bekomen. Het eindpunt en tevens het startpunt van de volgende winding wordt bekomen door a_2 van het vierde punt af te trekken en hierbij a_{3_2} op te tellen. Dit proces wordt herhaald totdat het opgegeven aantal windingen, N , bereikt is. Nu kunnen deze punten in de StraightConductorCollection functie gestoken worden om het totale veld van de spoel te bepalen.

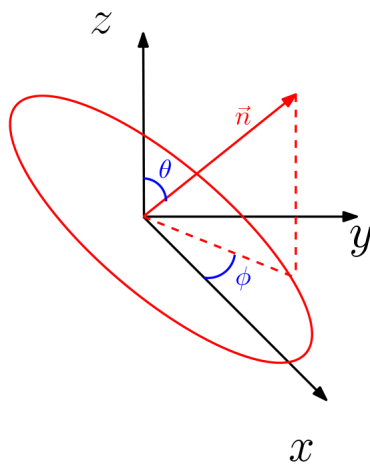
2.2.1.3 Magnetisch veld cirkelvormige geleider

Het is de bedoeling om een analytische uitdrukking te vinden voor het magnetisch veld veroorzaakt door een cirkelvormige geleider in een 3D ruimte. Een visualisatie van het probleem is terug te vinden in figuur 1.



Figuur 1 Cirkelvormige geleider waardoor een stroom I loopt

Hierbij is de cirkel gedefinieerd door zijn middelpunt M , zijn straal R en de normaalvector \mathbf{n} op de cirkel. Deze normaalvector is een eenheidsvector die de oriëntatie van de cirkel in de ruimte weergeeft. \mathbf{n} is beschreven door de hoeken ϕ en θ zoals te zien is in figuur 2.



Figuur 2 Cirkel met zijn normaalvector \mathbf{n}

De ingave van de oriëntatie van de normaalvector door de gebruiker wordt uniform gemaakt aan de andere elementen. De gebruiker geeft de hoeken θ_{user} en ϕ_{user} op zoals weergegeven in de sectie "Vector aanmaken a.d.h.v. 2 hoeken". Deze ingegeven hoeken worden dan getransformeerd naar de hoeken gebruikt in deze sectie (alsook in figuur 2) als volgt:

$$\theta = 90 - \phi_{user} \quad (1)$$

$$\phi = \theta_{user} \quad (2)$$

Het magnetisch veld \mathbf{dB} in het punt P veroorzaakt door het stroomdoorlopen elementje \mathbf{dl} kan bepaald worden a.d.h.v. de regel van Laplace (in de vrije ruimte):

$$\mathbf{dB} = \frac{\mu_0 I}{4\pi} \frac{\mathbf{dl} \times \mathbf{r}}{r^3} \quad (3)$$

met μ_0 de permeabiliteit van het vacuüm en I de stroomsterkte. De vector $\mathbf{r_P}$ verbindt het middelpunt van de cirkel met het punt P en wordt bekomen door de vectorsom van de vectoren \mathbf{r}' en \mathbf{r} (zie figuur 1).

Eerst zal de vector \mathbf{r}' bepaald worden. Deze vector wordt bekomen door de coördinaten van het punt C op de cirkel te bepalen.

Gebruik makende van de parametrische vergelijking van een cirkel [3], geldt er:

$$x_2 = -R \cos(\phi') \sin(\phi) + R \sin(\phi') \cos(\theta) \cos(\phi) + x_1 \quad (4)$$

$$y_2 = R \cos(\phi') \cos(\phi) + R \sin(\phi') \cos(\theta) \sin(\phi) + y_1 \quad (5)$$

$$z_2 = -R \sin(\phi') \sin(\theta) + z_1 \quad (6)$$

De hoek ϕ' wordt gemeten vanaf het blauwe lijnstuk in figuur 1 dat het middelpunt van de cirkel verbindt met een punt op de cirkel gelegen op dezelfde hoogte in wijzerzin tot aan de vector \mathbf{r}' . Het blauwe lijnstuk kan in principe bekomen worden door het punt M te verbinden met het punt C indien ϕ' gelijk wordt gesteld aan 0. De vector \mathbf{r}' is nu simpelweg:

$$\begin{aligned} \mathbf{r}' = & (-R \cos(\phi') \sin(\phi) + R \sin(\phi') \cos(\theta) \cos(\phi)) \cdot \mathbf{e}_x \\ & + (R \cos(\phi') \cos(\phi) + R \sin(\phi') \cos(\theta) \sin(\phi)) \cdot \mathbf{e}_y + (-R \sin(\phi') \sin(\theta)) \cdot \mathbf{e}_z \end{aligned} \quad (7)$$

Het stroomelementje $I \mathbf{dl}$ kan geschreven worden als volgt:

$$I \mathbf{dl} = I \frac{d\mathbf{r}'}{d\phi'} d\phi' \quad (8)$$

$$\begin{aligned} = & IR d\phi' [\sin(\phi') \sin(\phi) + \cos(\phi') \cos(\theta) \cos(\phi)] \cdot \mathbf{e}_x \\ & + (-\sin(\phi') \cos(\phi) + \cos(\phi') \cos(\theta) \sin(\phi)) \cdot \mathbf{e}_y + (-\cos(\phi') \sin(\theta)) \cdot \mathbf{e}_z \end{aligned} \quad (9)$$

Er geldt ook:

$$\mathbf{r_P} = (x - x_1) \cdot \mathbf{e}_x + (y - y_1) \cdot \mathbf{e}_y + (z - z_1) \cdot \mathbf{e}_z \quad (10)$$

De relatieve positievector \mathbf{r} tussen het punt P en het stroomelementje $I \mathbf{dl}$ is als volgt te bepalen:

$$\mathbf{r} = \mathbf{r_P} - \mathbf{r}' \quad (11)$$

$$\begin{aligned} = & (x - x_1 + R \cos(\phi') \sin(\phi) - R \sin(\phi') \cos(\theta) \cos(\phi)) \cdot \mathbf{e}_x \\ & + (y - y_1 - R \cos(\phi') \cos(\phi) - R \sin(\phi') \cos(\theta) \sin(\phi)) \cdot \mathbf{e}_y \\ & + (z - z_1 + R \sin(\phi') \sin(\theta)) \cdot \mathbf{e}_z \end{aligned} \quad (12)$$

Indien de vergelijkingen (9) en (12) worden ingevuld in vergelijking (3), kan het magnetisch veld \mathbf{dB} in het punt P veroorzaakt door het stroomdoorlopen elementje $d\mathbf{l}$ bepaald worden. Het vectorproduct en de magnitude van de vector \mathbf{r} worden in het programma bepaald.

Indien nu het magnetisch veld veroorzaakt door de gehele cirkelvormige geleider bepaald moet worden, dan zal de vergelijking voor \mathbf{dB} geïntegreerd moeten worden over $d\phi'$ van 0 tot 2π :

$$\mathbf{B} = \int_0^{2\pi} \mathbf{dB} d\phi' \quad (13)$$

Deze integraal zal numerisch geëvalueerd moeten worden door het programma.

Het is nu ook eenvoudig in te zien dat het veranderen van de integratiegrenzen in vergelijking (13) resulteert in het magnetisch veld van een stuk cirkelvormige geleider of m.a.w. een gebogen geleider met een kromtestraal R :

$$\mathbf{B} = \int_{\phi_1}^{\phi_2} \mathbf{dB} d\phi' \quad (14)$$

met $0 \leq \phi_1 < \phi_2 \leq 2\pi$

Het "start-" en eindpunt" van dit stukje gebogen geleider kunnen bekomen worden door respectievelijk ϕ_1 en ϕ_2 in te vullen in de vergelijkingen (4), (5) en (6).

2.2.1.4 Magnetisch veld cirkelvormige spoel

Het veld van een cirkelvormige spoel kan bekomen worden a.d.h.v. stukjes gebogen geleider waarvoor geldt $\phi_1 - \phi_2 = \pi$ (zie de sectie: "magnetisch veld cirkelvormige geleider"). Dit is gelijkaardig aan het veld van een rechthoekige spoel bekomen uit rechte geleiders. Een cirkelvormige spoel bestaat uit helixvormige windingen. Zo'n winding kan bekomen worden door twee gebogen geleiders (met bovenstaande voorwaarde) aan elkaar te schakelen. De implementatie hiervan is terug te vinden in het programma (CircularCoil). Voor elke gebogen geleider zal een nieuwe kromtestraal, middelpunt en normaalvector berekend worden volgens een bepaald algoritme. Dit algoritme valt buiten de scope van dit verslag, maar is terug te vinden in het programma.

2.2.2 Walk on Spheres methode

De walk on Spheres methode is een numeriek probabilistisch algoritme (Monte-Carlo methode) dat toegepast wordt op differentiaalvergelijkingen met een randvoorwaardeprobleem. Het kan onder andere gebruikt worden om de vergelijking van Laplace op te lossen ($\Delta\phi = 0$). Deze methode simuleert paden van Brownse bewegingen door de uitgangspunten van opeenvolgende sferen probabilistisch te bepalen. Dit is een gridloos algoritme.

Het elektrisch potentiaalprobleem in de ladingsvrije ruimte wordt beschreven door een vergelijking van Laplace:

$$\Delta V = 0 \quad (1)$$

met V de potentiaal. De randvoorwaarden bij dit probleem zijn de verschillende 3D objecten op wiens oppervlakken telkens een welbepaalde potentiaal is aangebracht.

[4] Indien de potentiaal in een welbepaald punt in de ruimte (P) bepaald moet worden, kan het Walk on Spheres algoritme doorlopen worden. Beschouw de grootste mogelijke sfeer ($S(P)$) met zijn middelpunt in P die geen 3D objecten omvat nog snijdt en dus het dichtsbijzijnd oppervlak van een 3D object raakt. Deze sfeer zal dus een straal hebben die wordt bepaald door het minimum van de minimale afstanden van het punt P tot aan de oppervlakken van de 3D objecten. Nu geldt er dat de potentiaal in het punt P gelijk is aan het gemiddelde van de potentiaal over het oppervlak van de initiële sfeer. Nu kan er een random punt (P_1) op de sfeer gekozen worden a.d.h.v. een random number generator. Rond dit punt zal weer de grootste mogelijke sfeer ($S(P_1)$) geconstrueerd worden. Er kan nu weer een random punt bepaald worden, maar dan op $S(P_1)$. Dit proces zal blijven herhaald worden totdat er een punt bereikt wordt dat dicht genoeg bij een oppervlak van een 3D object ligt. De potentiaal van dit oppervlak zal dan als "schatting" voor de potentiaal in het punt P_1 genomen worden. Dit proces wordt voor vele random punten op de initiële sfeer ($S(P)$) herhaald totdat er een gewenste nauwkeurigheid bereikt is. Indien het gemiddelde van al de geschatte waarden voor de potentiaal in de punten op de initiële sfeer wordt genomen, zal er uiteindelijk een benaderende waarde voor de potentiaal in het punt P worden gevonden.

Het is dus noodzakelijk dat het programma de minimale afstand van een punt in de ruimte tot aan een 3D object kan bepalen.

Een random punt op een sfeer kan als volgt bepaald worden [5]:

Laat een random number generator twee waarden, w_1 en w_2 , tussen 0 en 1 genereren en maak dan de sferische coördinaten θ en ϕ aan:

$$\theta = 2\pi w_1 \quad (2)$$

$$\phi = \cos^{-1}(2w_2 - 1) \quad (3)$$

Indien de sfeer een middelpunt $M(x_1, y_1, z_1)$ en een straal R heeft, geldt er voor de coördinaten van het random punt op de sfeer (x, y, z) dat:

$$x = x_1 + R \cos \theta \sin \phi \quad (4)$$

$$y = y_1 + R \sin \theta \sin \phi \quad (5)$$

$$z = z_1 + R \cos \phi \quad (6)$$

In dit programma zal niet de potentiaal in een bepaald punt bepaald moeten worden, maar eerder het elektrische veld. Zoals besproken wordt in de sectie "Discussie" blijkt dit voorlopig niet nauwkeurig te lukken.

2.2.3 Beweging deeltje in magnetisch en elektrisch veld

Er wordt een deeltje met lading q (in coulomb) en massa m (in kilogram) beschouwd. Dit deeltje beweegt zich voort met een initiële snelheid v_0 (in $\frac{m}{s}$) in een magnetisch veld \mathbf{B} en in een elektrisch veld \mathbf{E} .

Het deeltje wordt beschouwd op zijn beginpositie $\mathbf{r}(0)$ op tijdstip $t = 0$:

$$\mathbf{r}(0) = A_1\mathbf{e}_x + B_1\mathbf{e}_y + C_1\mathbf{e}_z \quad (1)$$

Het deeltje heeft een snelheidsvector $\mathbf{v}(0)$ op $t = 0$. Deze snelheidsvector wordt geconstrueerd door de eenheidsvector van de snelheid te vermenigvuldigen met v_0 :

$$\mathbf{v}(0) = v_0 \cdot \mathbf{e}_v \quad (2)$$

$$= A_2\mathbf{e}_x + B_2\mathbf{e}_y + C_2\mathbf{e}_z \quad (3)$$

Het magnetisch veld $\mathbf{B}(0)$ en het elektrisch veld $\mathbf{E}(0)$ op $t = 0$ in een bepaald punt in de ruimte worden berekend in het programma.

De formule voor de Lorentzkracht, $F = q(\mathbf{v} \times \mathbf{B} + \mathbf{E})$, kan omgevormd worden door de substitutie $F = m\mathbf{a}$. De volgende formule voor de versnelling $\mathbf{a}(0)$ op $t = 0$ is dan geldig:

$$\mathbf{a}(0) = \frac{q}{m}(\mathbf{v}(0) \times \mathbf{B}(0) + \mathbf{E}(0)) \quad (4)$$

$\mathbf{a}(0)$ uitgedrukt in zijn componenten:

$$\mathbf{a}(0) = A_3\mathbf{e}_x + B_3\mathbf{e}_y + C_3\mathbf{e}_z \quad (5)$$

Na een tijd t_1 (in sec.) (waarbij t_1 zo klein mogelijk moet zijn omdat anders de fout op de berekening te groot kan worden) zal het deeltje zich in de positie $\mathbf{r}(t_1)$ bevinden. Vermits $\frac{d\mathbf{r}}{dt} = \mathbf{v}$, geldt (invullen vgl. (3)):

$$\mathbf{r}(t_1) = \int_0^{t_1} \mathbf{v}(t) dt \quad (6)$$

$$\approx \int_0^{t_1} \mathbf{v}(0) dt \quad (7)$$

$$= A_2t_1\mathbf{e}_x + B_2t_1\mathbf{e}_y + C_2t_1\mathbf{e}_z + \mathbf{c}_1 \quad (8)$$

met \mathbf{c}_1 een constante. Merk op dat hier een benadering werd gemaakt: de snelheid van het deeltje op een tijdstip t ($\mathbf{v}(t)$) werd als constante beschouwd over de tijdsperiode $0 \rightarrow t_1$ (zie vgl. (7)). Enkel in het volgende punt (op tijdstip t_2) zal de snelheidsvector van het deeltje 'geüpdated' worden (zie verder). Dit is de reden waarom t_1 voldoende klein moet gekozen worden, zeker als het deeltje met een grote snelheid reist en zijn versnelling ten gevolge van het magnetisch veld groot is.

Vermits $\mathbf{r}(0) = \mathbf{c}_1$, geldt (zie vergelijking (1)):

$$\mathbf{c}_1 = A_1 \mathbf{e}_x + B_1 \mathbf{e}_y + C_1 \mathbf{e}_z \quad (9)$$

Indien nu vergelijkingen (8) en (9) gecombineerd worden, wordt de volgende vergelijking bekomen voor $\mathbf{r}(t_1)$:

$$\mathbf{r}(t_1) = (A_1 + A_2 t_1) \mathbf{e}_x + (B_1 + B_2 t_1) \mathbf{e}_y + (C_1 + C_2 t_1) \mathbf{e}_z \quad (10)$$

Na een tijd t_1 bevindt het deeltje zich dus in de nieuwe positie $\mathbf{r}(t_1)$. De snelheid van het deeltje na t_1 seconden, $\mathbf{v}(t_1)$, wordt als volgt berekend ($\frac{d\mathbf{v}}{dt} = \mathbf{a}$) (invullen vgl. (5)):

$$\mathbf{v}(t_1) = \int_0^{t_1} \mathbf{a}(t) dt \quad (11)$$

$$\approx \int_0^{t_1} \mathbf{a}(0) dt \quad (12)$$

$$= A_3 t_1 \mathbf{e}_x + B_3 t_1 \mathbf{e}_y + C_3 t_1 \mathbf{e}_z + \mathbf{c}_2 \quad (13)$$

met \mathbf{c}_2 een constante. Merk op dat hier weer een benadering werd gemaakt: de versnelling van het deeltje op een tijdstip t ($\mathbf{a}(t)$) werd als constante beschouwd over de tijdsperiode $0 \rightarrow t_1$ (zie vgl. (12)). Enkel in het volgende punt (op tijdstip t_2) zal de versnellingsvector van het deeltje 'geüpdated' worden (zie verder).

Vermits $\mathbf{v}(0) = \mathbf{c}_2$, geldt (zie vergelijking (3)):

$$\mathbf{c}_2 = A_2 \mathbf{e}_x + B_2 \mathbf{e}_y + C_2 \mathbf{e}_z \quad (14)$$

Indien nu vergelijkingen (13) en (14) gecombineerd worden, wordt de volgende vergelijking bekomen voor $\mathbf{v}(t_1)$:

$$\mathbf{v}(t_1) = (A_2 + A_3 t_1) \mathbf{e}_x + (B_2 + B_3 t_1) \mathbf{e}_y + (C_2 + C_3 t_1) \mathbf{e}_z \quad (15)$$

Implementatie in het programma:

Het magnetisch veld in een punt kan door het programma analytisch bepaald worden. Het elektrisch veld wordt a.d.h.v. de walk on Spheres methode bepaald. Er wordt nog een relativistische correctie doorgevoerd voor het magnetisch/elektrisch veld dat het deeltje 'waarneemt'. De velden \mathbf{E} en \mathbf{B} zijn de velden in het referentiestelsel van de veldproducerende objecten en zijn dus de velden berekend door het programma. De velden \mathbf{E}' en \mathbf{B}' zijn de velden gekoppeld aan het inertiaal referentiestelsel van het deeltje en zijn dus diegenen die gebruikt zullen worden om het traject van het deeltje te berekenen. Dit resulteert in een transformatie van \mathbf{E} en \mathbf{B} d.m.v. de Bernoulli-Joules vergelijkingen[6]:

$$\mathbf{E}' = \gamma(\mathbf{E} + \mathbf{v} \times \mathbf{B}) - (\gamma - 1)(\mathbf{E} \cdot \mathbf{e}_v) \mathbf{e}_v \quad (16)$$

$$\mathbf{B}' = \gamma(\mathbf{B} - \frac{\mathbf{v} \times \mathbf{E}}{c^2}) - (\gamma - 1)(\mathbf{B} \cdot \mathbf{e}_v) \mathbf{e}_v \quad (17)$$

met \mathbf{v} de snelheid van het bewegend deeltje en \mathbf{e}_v de eenheidsvector hiervan ($\mathbf{v} = v \mathbf{e}_v$). γ is hier de Lorentz-factor ($\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}}$) met c de lichtsnelheid in vacuum.

Het programma start met een deeltje in een welbepaald punt met een bepaalde snelheidsvector. Om nu het volgende punt waarin het deeltje zich zal bevinden en zijn snelheid te bepalen wordt volgend algoritme doorlopen (hierbij werd gebruik gemaakt van een soort van predictor-corrector methode om de fout verkregen door de eindig kleine tijdstap te beperken):

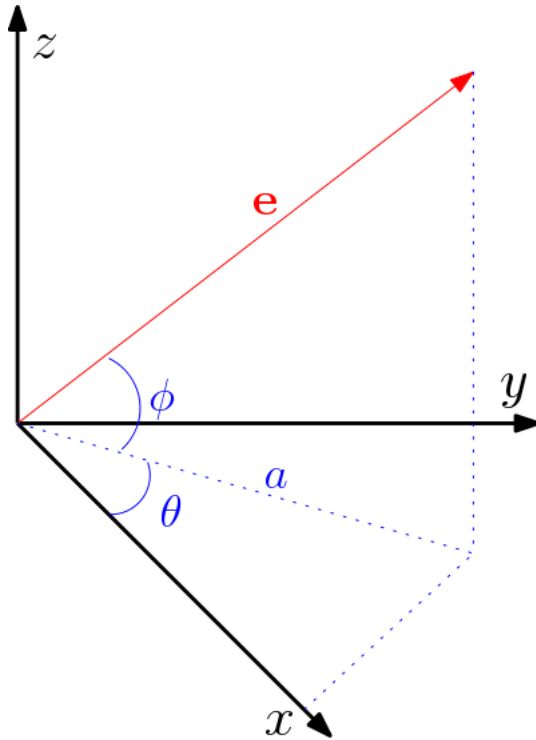
1. Het programma berekent de velden in het huidige punt waar het deeltje zich bevindt.
2. Er wordt een relativistische correctie doorgevoerd a.d.h.v. vergelijkingen (16) en (17)
3. De versnellingsvector van het deeltje in zijn huidig punt wordt berekend a.d.h.v. vergelijking (4)
4. Het volgende punt waarin het deeltje zich zal bevinden wordt bepaald a.d.h.v. vergelijking (10) die als input het huidige punt en de huidige snelheidsvector meekrijgt.
5. Het punt uit stap 4 wordt gebruikt om opnieuw de velden te berekenen in dit punt en om zodanig dus een nieuwe, tweede versnellingsvector te verkrijgen.
6. Er wordt nu een vectorgemiddelde genomen van de versnellingsvector uit stap 3 en stap 5 om zodanig een 'verbeterde' versnellingsvector te bekomen.
7. De initiële snelheidsvector en de gemiddelde versnellingsvector bekomen in stap 6 worden gebruikt om een nieuwe, tweede snelheidsvector te verkrijgen a.d.h.v. vergelijking (15).
8. Er wordt nu een vectorgemiddelde genomen van de initiële snelheidsvector en diegene uit stap 7 om zodanig een 'verbeterde' snelheidsvector te bekomen.
9. Het volgende punt waarin het deeltje zich zal bevinden wordt nu opnieuw bepaald, maar a.d.h.v. vergelijking (10) die nu als input het huidige punt en de gemiddelde snelheidsvector uit stap 8 meekrijgt.
10. Het programma berekent de velden in het volgende punt (bekomen in stap 9).
11. Er wordt een relativistische correctie doorgevoerd a.d.h.v. vergelijkingen (16) en (17)
12. De versnellingsvector van het deeltje in zijn volgend punt wordt berekend a.d.h.v. vergelijking (4)
13. Er wordt nu een vectorgemiddelde genomen van de initiële versnellingsvector en diegene uit stap 12 om zodanig een 'verbeterde' versnellingsvector te bekomen.
14. De initiële snelheidsvector en de gemiddelde versnellingsvector bekomen in stap 13 worden gebruikt om een snelheidsvector te verkrijgen a.d.h.v. vergelijking (15) op het volgende punt.

Het volgende punt en de snelheidsvector in dit punt zijn nu dus bepaald, alsook de velden in dit punt. Indien dus de stappen 3 → 14 in een loop gezet worden kunnen de opeenvolgende posities van het deeltje bepaald worden.

Men kan dit proces blijven herhalen totdat een bepaalde grenswaarde voor de totale tijdsperiode overschreden wordt of totdat het deeltje zich buiten een bepaald gebied begeeft. Dusdanig kan het hele traject van het deeltje bepaald worden.

2.2.4 Vector aanmaken a.d.h.v. 2 hoeken

De richtingen van bepaalde vectoren zullen gedefinieerd worden door 2 hoeken op te geven in het programma. A.d.h.v. deze 2 hoeken zal een eenheidsvector \mathbf{e} (zie figuur 1, eenheidsvector in rood aangeduid) geconstrueerd worden.



Figuur 1 constructie eenheidsvector uit 2 hoeken

De 2 hoeken zijn θ , gemeten vanaf de x-as in tegen-wijzer zin tot aan de projectie van \mathbf{e} op het xy-vlak (a op de figuur), en ϕ , gemeten vanaf a in tegen-wijzer zin tot aan \mathbf{e} .

De componenten van \mathbf{e} (e_x , e_y en e_z) voldoen aan de volgende voorwaarde:

$$e_x^2 + e_y^2 + e_z^2 = 1 \quad (1)$$

De volgende 3 vergelijkingen zijn geldig:

$$\cos \theta = \frac{e_x}{a} \quad (2)$$

$$\tan \phi = \frac{e_z}{a} \quad (3)$$

$$a = \sqrt{e_x^2 + e_y^2} \quad (4)$$

Uit vergelijking (1) en (4) volgt:

$$a^2 + e_z^2 = 1 \quad (5)$$

$$e_z^2 = 1 - a^2 \quad (6)$$

Vergelijking (6) invullen in (3) resulteert in:

$$\tan \phi = \frac{\sqrt{1-a^2}}{a} \quad (7)$$

$$\tan^2 \phi = \frac{1-a^2}{a^2} \quad (8)$$

$$\tan^2 \phi + 1 = \frac{1}{a^2} \quad (9)$$

$$a = \sqrt{\frac{1}{1+\tan^2 \phi}} \quad (10)$$

M.b.v. de waarde voor a die a.d.h.v. ϕ wordt bepaald (vgl. 10), kunnen nu de waarden voor de componenten van \mathbf{e} worden berekend:

uit (2) volgt:

$$e_x = a \cos \theta \quad (11)$$

uit (4) volgt:

$$e_y = \sqrt{a^2 - e_x^2} \quad (12)$$

Uit (3) volgt:

$$e_z = a \tan \phi \quad (13)$$

De componenten van de eenheidsvector en bijgevolg de eenheidsvector \mathbf{e} zelf werd geconstrueerd uit de 2 opgegeven hoeken.

2.2.5 Afstandsbepaling

2.2.5.1 Afstand tot een cirkelschijf

Hier wordt een infinitesimale cirkelschijf beschouwd met straal r , zijn middelpunt beschreven door de vector \mathbf{C} en zijn oriëntatie beschreven door de normaal \mathbf{N} loodrecht op het cirkeloppervlak. De afstand (d) van een punt in de ruimte, beschreven door de vector \mathbf{P} , tot de cirkelschijf zal nu bepaald worden. Stel dat $\Delta = \mathbf{P} - \mathbf{C}$, dan wordt de projectie \mathbf{Q} van \mathbf{P} op het vlak beschreven door de cirkelschijf gegeven door de volgende uitdrukking [7]:

$$\mathbf{Q} = \mathbf{C} + \Delta - (\mathbf{N} \cdot \Delta) \mathbf{N} \quad (1)$$

Indien de lengte van de vector $\mathbf{Q} - \mathbf{C}$ kleiner is dan de straal van de cirkel, dan zal de afstand van het punt \mathbf{P} tot de cirkelschijf logischer wijze gegeven worden door de grootte van de vector $\mathbf{P} - \mathbf{Q}$. Indien de lengte van $\mathbf{Q} - \mathbf{C}$ groter is dan de straal van de cirkel, dan wordt d bepaald door de afstand van \mathbf{P} tot aan de rand van de cirkelschijf[7]:

$$d = \sqrt{(\mathbf{N} \cdot \Delta)^2 + (|\mathbf{N} \times \Delta| - r)^2} \quad (2)$$

2.2.5.2 Afstand tot een cilinder

Het berekenen van de afstand van een punt in de ruimte tot een afgesloten cilinder is vrij triviaal nu dat de afstand tot een cirkelschijf gekend is. Voor de implementatie hiervan kan naar de code van het programma gekeken worden.

2.3 Discussie

2.3.1 Walk on Spheres

De Walk on Spheres methode blijkt goed te werken voor de bepaling van de elektrische potentiaal in een punt in de 3D ruimte. Echter zoals aangehaald werd in de sectie "Walk on Spheres methode" moet dit nog uitgebreid worden naar de bepaling van het elektrische veld. In eerste instantie werd er gekeken naar de paper van Ben Yu-Kuang Hu[8] om het elektrische veld te bepalen. Hierin staat vermeld dat het gemiddelde elektrische veld in een sfeer (\mathbf{E}) volledig bepaald wordt door de potentiaal over het oppervlak van de sfeer (S):

$$\mathbf{E} = \frac{-3}{4\pi R^3} \oint_S \Phi d\mathbf{a} \quad (1)$$

met R de straal van de sfeer en Φ de potentiaal over het oppervlak van de sfeer. Er staat niet vermeld wat de vector $d\mathbf{a}$ precies is. Bij nader inzicht zal deze formule niet in deze vorm toegepast kunnen worden op de Walk on Spheres methode, vermits er wordt gesproken van een gemiddeld elektrisch veld in een sfeer en niet in het middelpunt van de sfeer. Ook blijkt deze formule enkel toepasbaar te zijn indien de bronnen van het elektrische veld zich binnenin de sfeer bevinden. Desondanks werd deze formule hier toch vermeld, vermits deze het uitgangspunt vormde voor de proefondervindelijke zoektocht naar de uitbreiding van de Walk on Spheres methode om het elektrische veld in een punt te bepalen. Voorlopig werd er nog maar enkel getracht om de richting van de elektrische veldvector in P te bepalen en nog niet de lengte van deze vector.

De Walk on Spheres methode bepaalt de elektrische potentiaal in een punt (P) door het gemiddelde te nemen van "geschatte" waarden voor de potentialen in de punten op de initiële sfeer ($S(P)$) (zie de sectie: "Walk on Spheres methode"). Eerst werd er getracht om a.d.h.v. van deze "geschatte" waarden op de initiële sfeer ($S(P)$) het elektrische veld in P te bepalen. Vermits $\mathbf{E} = -\nabla V$, kan er beredeneerd worden om aan elke geschatte waarde voor de potentiaal op de sfeer een vector toe te kennen die vanaf het punt op $S(P)$ waarin de geschatte waarde bekomen werd naar het middelpunt P van de initiële sfeer wijst en evenredig is met de "geschatte" waarde voor de potentiaal. Indien al deze bekomen vectoren worden opgeteld en deze vectorsom gedeeld wordt door het aantal bekomen vectoren, zou dit mogelijk een schatting kunnen geven van het elektrische veld in P . Dit werd uitgetest in het programma door telkens de vectoren te bepalen die de geschatte potentiaalpunten op $S(P)$ verbinden met P , deze dan te normaliseren en te vermenigvuldigen met de geschatte potentiaal. Van deze vectoren werd dan de vectorsom genomen en gedeeld door het totaal aantal vectoren.

Deze methode werd getest door het programma zowel analytisch (a.d.h.v. de Sphere_Field klasse) als via de methode hierboven beschreven het elektrische veld in P te laten bepalen. Er werd telkens een sfeer genomen met zijn middelpunt in de oorsprong, een straal van $5m$ en een potentiaal van $40V$. Het elektrische veld werd in het punt $(3,2,10)$ bepaald. Er werden telkens 1000 iteraties genomen voor de Walk on Spheres (WOS) methode (er werden dus 1000 punten op de initiële sfeer bekomen). De resultaten zijn hieronder weergegeven voor 2 onafhankelijke uitvoeringen van het programma. De eerste lijn is de analytisch bekomen veldvector, de tweede lijn is bekomen a.d.h.v. de hierboven beschreven methode en de laatste lijn is een componentsgewijze vergelijking tussen de twee bekomen vectoren om een duidelijk beeld te verkrijgen of de richting van de elektrische veldvectoren overeenstemt.

analytical: $1.06194690265487.\mathbf{e}_x + 0.707964601769911.\mathbf{e}_y + 3.53982300884956.\mathbf{e}_z$
WOS: $0.742453435096501.\mathbf{e}_x + (-0.227918904376110).\mathbf{e}_y + 2.15827603884003.\mathbf{e}_z$
WOS/analytical: $0.699143651382539, -0.321935452431255, 0.609712980972309$

analytical: $1.06194690265487 \cdot \mathbf{e}_x + 0.707964601769911 \cdot \mathbf{e}_y + 3.53982300884956 \cdot \mathbf{e}_z$
WOS: $1.14930427142322 \cdot \mathbf{e}_x + 0.540186724584631 \cdot \mathbf{e}_y + 2.66168087542221 \cdot \mathbf{e}_z$
WOS/analytical: 1.08226152225686, 0.763013748475791, 0.751924847306775

Uit de vergelijking tussen de elektrische veldvectoren bekomen in het punt (3,2,10) op analytische wijze en a.d.h.v. de WOS methode, kan er besloten worden dat deze methode geen correcte resultaten bekamt. Ook is er geen consistentie tussen de 2 uitvoeringen van het programma. Deze methode werkt waarschijnlijk niet vermits aan de punten op de sfeer rond P eigenlijk geen werkelijke schattingen voor de potentiaal op de sfeer worden toegekend: deze "geschatte" waarden kunnen enkel gebruikt worden om uiteindelijk de potentiaal in P te bepalen en zijn niet representatief voor de potentiaal op de sfeer zelf. Vandaar werd er een tweede gelijkaardige methode uitgetest.

Deze methode berust op hetzelfde principe als de vorige methode, echter zal er nu per iteratie een andere vector worden toegekend. Deze vector is nog altijd evenredig met de "geschatte" waarde voor de potentiaal, maar heeft nu zijn oorsprong in het laatste random bekomen punt in een iteratie dat dicht genoeg bij het oppervlak van het 3D object lag waarvan de potentiaal als "schatte" werd gebruikt. Deze vector wijst nog altijd naar het punt P en zal genormaliseerd en vermenigvuldigd worden met de geschatte potentiaal. Ook werd hier terug de vectorsom genomen en gedeeld door het totaal aantal bekomen vectoren. Deze methode werd op dezelfde manier getest als de eerste:

analytical: $1.06194690265487 \cdot \mathbf{e}_x + 0.707964601769911 \cdot \mathbf{e}_y + 3.53982300884956 \cdot \mathbf{e}_z$
WOS: $4.75716339465419 \cdot \mathbf{e}_x + 2.96142803184802 \cdot \mathbf{e}_y + 16.1052863838474 \cdot \mathbf{e}_z$
WOS/analytical: 4.47966219663270, 4.18301709498532, 4.54974340343690

analytical: $1.06194690265487 \cdot \mathbf{e}_x + 0.707964601769911 \cdot \mathbf{e}_y + 3.53982300884956 \cdot \mathbf{e}_z$
WOS: $4.31652954106094 \cdot \mathbf{e}_x + 2.82735931818131 \cdot \mathbf{e}_y + 15.8193285344203 \cdot \mathbf{e}_z$
WOS/analytical: 4.06473198449906, 3.99364503693110, 4.46896031097375

Nu geeft de WOS methode al een betere richting van de elektrische veldvector weer in vergelijking met de eerste methode. Echter zoals te zien is, zit hier nog een vrij grote fout op en dit na maar liefst 1000 iteraties. Er zijn nog andere methoden uitgeprobeerd waarbij de vectoren bekomen in de iteraties geschaald werden met de afstand tot het oppervlak van het 3D object met gelijkaardige resultaten. Voorlopig werd er dus nog geen goede manier gevonden om de WOS methode uit te breiden om de richting van de elektrische veldvector in een punt te bepalen.

Mogelijk zou hier nog verder met geëxperimenteerd kunnen worden, maar het ziet er momenteel naar uit dat deze methode een doodlopende straat is om het elektrische veld nauwkeurig en relatief snel te bepalen in een punt in de ruimte. Een mogelijke bron van de onnauwkeurigheid op de elektrische veldvector zou misschien ook te wijten kunnen zijn aan de manier waarop er een random punt op de sfeer gekozen wordt (zie de sectie: "Walk on Spheres methode"). Voor de berekening van het elektrische veld zou er dus hoogstwaarschijnlijk op zoek moeten worden gegaan naar een andere gridloze methode (bv.: multipoolontwikkelingen).

2.3.2 Optimalisaties en uitbreidingen

De analytische uitdrukking voor het totale magnetische veld i.f.v. x , y en z is een vectorsom van de magnetische velden van de individuele veldproducerende elementen (zie de sectie: "Implementatie"). Hoe meer elementen er dus worden opgesteld, hoe groter de analytische uitdrukking van het totale magnetische veld wordt. Indien er bijvoorbeeld een rechthoekige spoel met 10 windingen wordt opgesteld, dan zal de analytische uitdrukking bestaan uit de som van de analytische uitdrukkingen van maar liefst 40 rechte geleiders. Het evalueren van deze lange analytische functie in een bepaald punt in de ruimte door de `EvaluateAnalyticField` functie zal relatief veel tijd in beslag nemen en het kan dus nuttig zijn om deze te optimaliseren. Het is niet mogelijk om deze analytische uitdrukking verder te vereenvoudigen, dus zal een optimalisatie van de `EvaluateAnalyticField` functie zich vooral moeten baseren op het evalueren van de analytische uitdrukking in een bepaald punt. Er kan naar bepaalde Python modules gezocht worden die analytische uitdrukkingen performant kunnen evalueren.

Vermits de `Walk on Spheres` methode niet goed werkt, werd de `Object_3D` klasse vrij beperkt gehouden (zie de sectie: "Implementatie"). Echter de `IsPointInObject` functie dat gebruik maakt van afstandsbepalingsfuncties (`DistanceToPoint`) wordt gebruikt in het bepalen of het deeltje al dan niet een botsing met een veldproducerend element heeft ondergaan. Het kan dus toch nog nuttig zijn om deze basis klasse verder uit te breiden met 3D objecten.

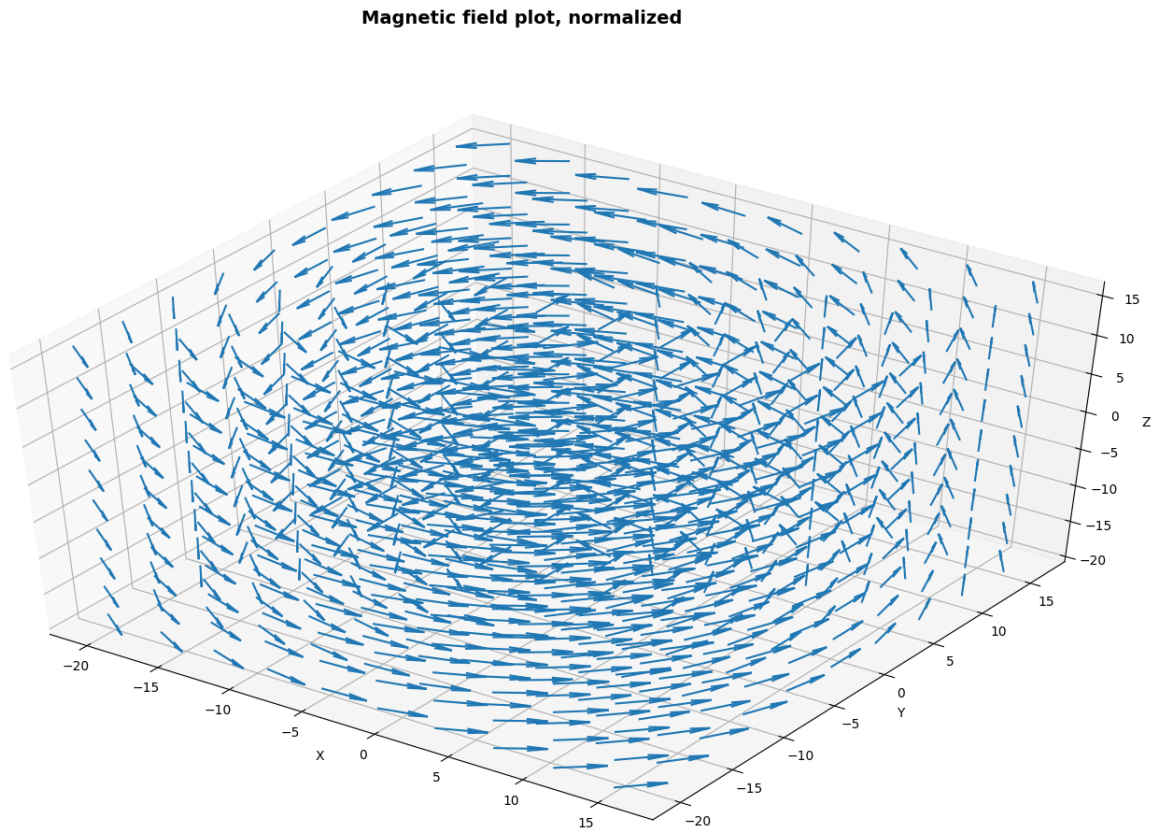
De klasse `FieldSource` zou eventueel ook nog kunnen worden uitgebreid. Zo werd er bijvoorbeeld nog geen analytische uitdrukking bepaald voor het magnetische veld van een stroomdoorlopen oppervlak (current sheet). Indien dit mogelijk zou zijn voor eindige oppervlakken, zou dit ook nog geïmplementeerd kunnen worden.

Er zijn natuurlijk nog veel meer andere uitbreidingen van de code te bedenken. Hier werden slechts enkelen aangehaald.

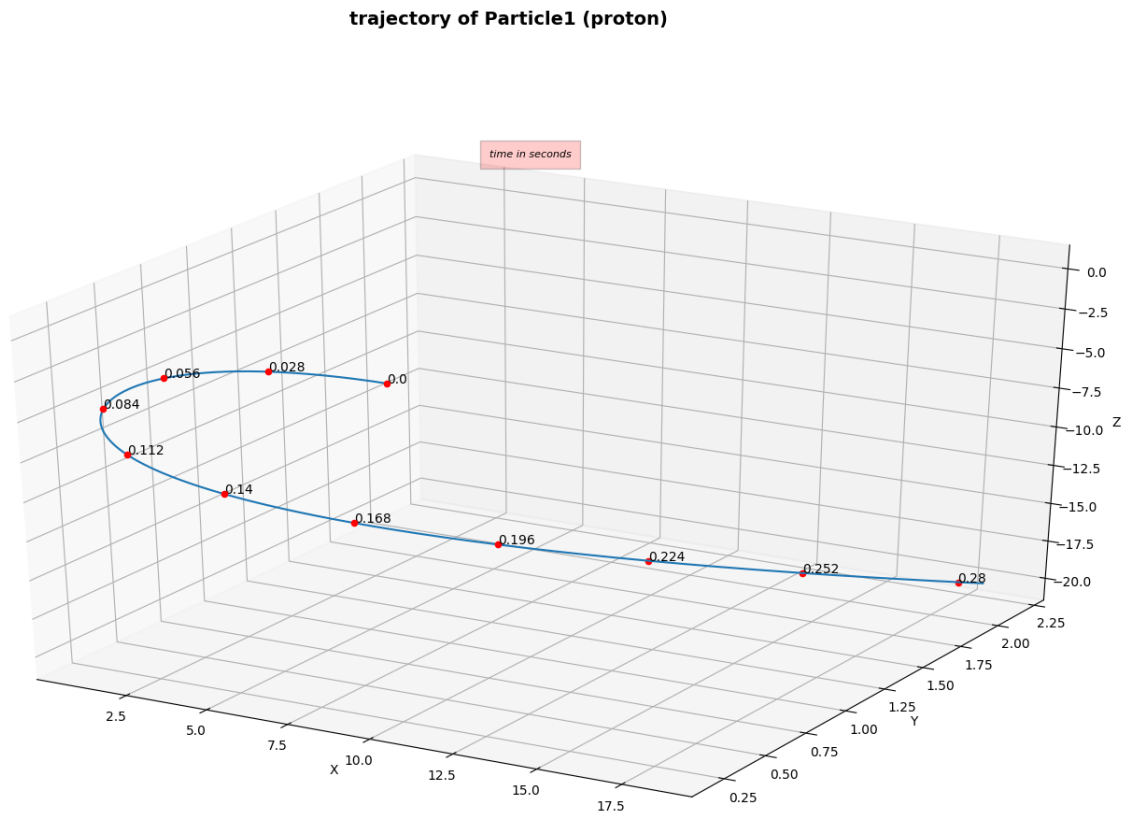
3 Resultaten

Voor deze sectie werden enkele configuraties met magnetische veldproducerende elementen uitgerekend door het programma. Er wordt telkens aangehaald welke elementen er werden opgesteld en van welke deeltjes er banen werden berekend. In elk van de voorbeelden zijn de deeltjes protonen die een initiële snelheidsvector hebben die ligt in de tegenstelde richting van de x-as. Voor elke configuratie worden er twee grafieken getoond die aangemaakt werden door het programma. Deze grafieken geven het uitgerekende magnetische veld weer en de trajecten die de geïnitieerde deeltjes in dit veld afleggen. Het magnetische veld werd telkens genormaliseerd voor een duidelijkere weergave.

In het eerste voorbeeld werd er een rechte geleider opgesteld tussen de punten $(-5, 0, -10)$ en $(-5, 0, 10)$ waardoorheen een stroom van $5A$ loopt in de z -richting. In figuur 1 wordt hiervan het genormaliseerde magnetische veld weergegeven. Het traject van een proton werd bepaald die een initiële snelheid heeft van $125 \frac{m}{s}$ en zich in de startpositie $(10, 0.2, 0)$ bevond. Dit traject is weergegeven in figuur 2.



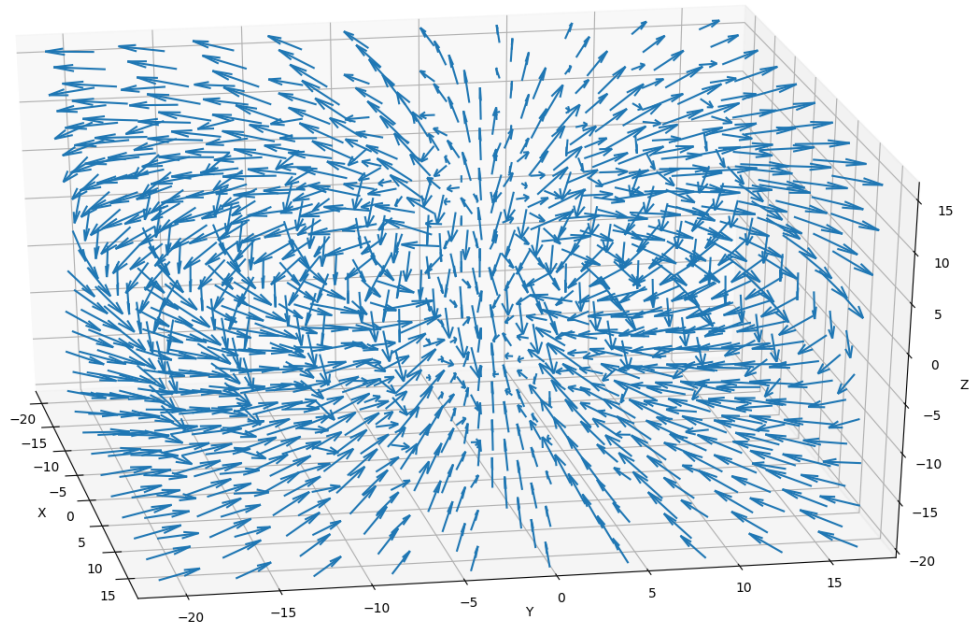
Figuur 1 genormaliseerd magnetisch veld voorbeeld 1



Figuur 2 deeltjesbaan voorbeeld 1

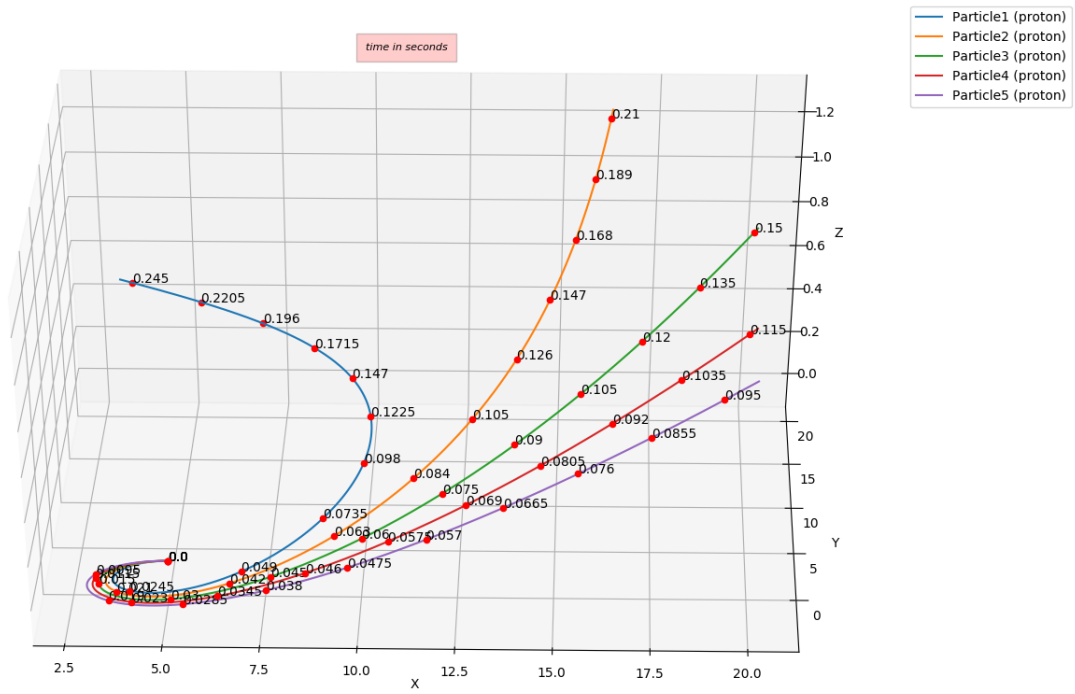
In het tweede en derde voorbeeld werd er een balkvormige spoel met als grondvlak een vierkant met zijde $5m$ opgesteld. De spoel staat gericht volgens de z -as, heeft een hoogte van $2m$ en bestaat uit 10 windingen die in anti-wijzerzin (van bovenaf bekeken) werden opgewonden. Het midden van het grondvlak van de spoel bevindt zich in het punt $(-7.5, 0, -1)$. In voorbeeld 2 loopt er doorheen de windingen een stroom van $5A$, terwijl de stroom in voorbeeld 3 slechts $1A$ bedraagt. De stroom loopt in anti-wijzerzin (van bovenaf bekeken) in beide voorbeelden. In figuur 3 wordt van beide voorbeelden het genormaliseerde magnetische veld weergegeven. De trajecten van 5 protonen werden bepaald die allen dezelfde startpositie $(5, 0, 0)$, maar verschillende initiële snelheden hebben. Deze snelheden bedragen respectievelijk (voor proton 1 t.e.m. 5) $125, 150, 175, 200$ en $225 \frac{m}{s}$. De deeltjesbanen van de protonen voor voorbeelden 2 en 3 worden respectievelijk in de figuren 4 en 5 getoond.

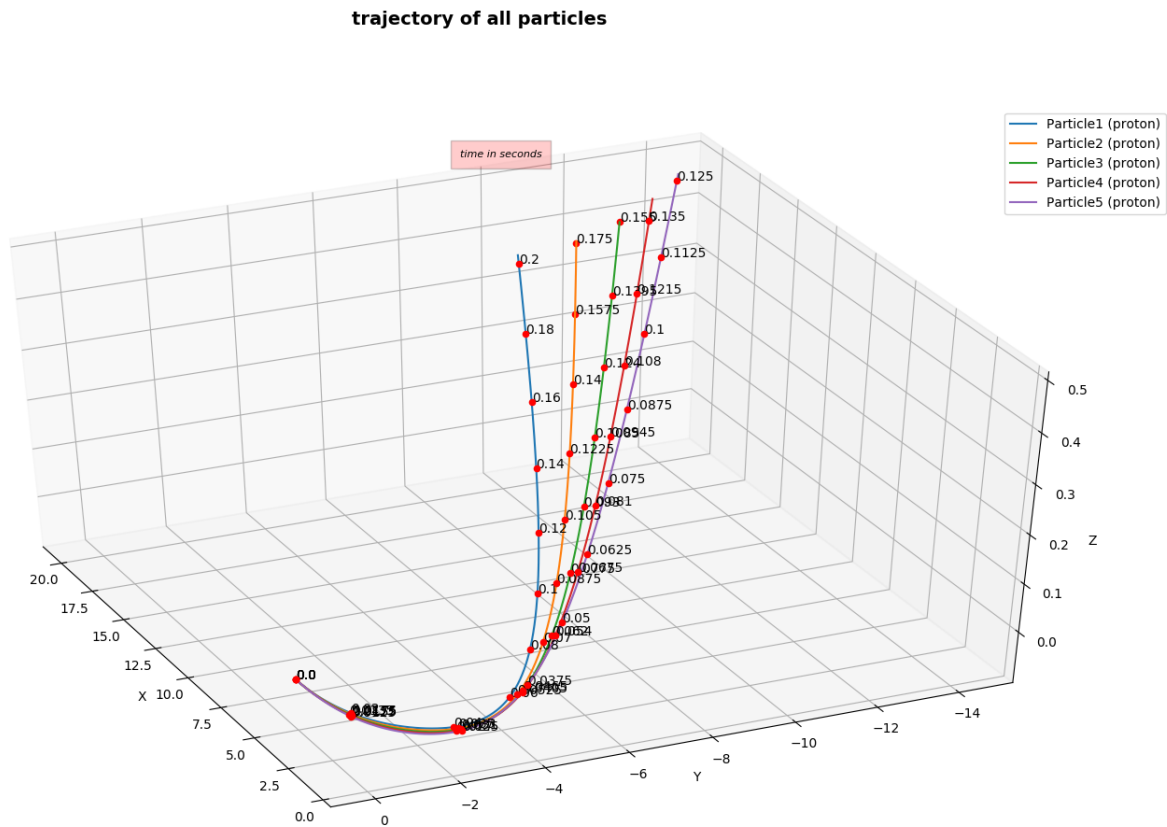
Magnetic field plot, normalized



Figuur 3 genormaliseerd magnetisch veld voorbeelden 2 en 3

trajectory of all particles

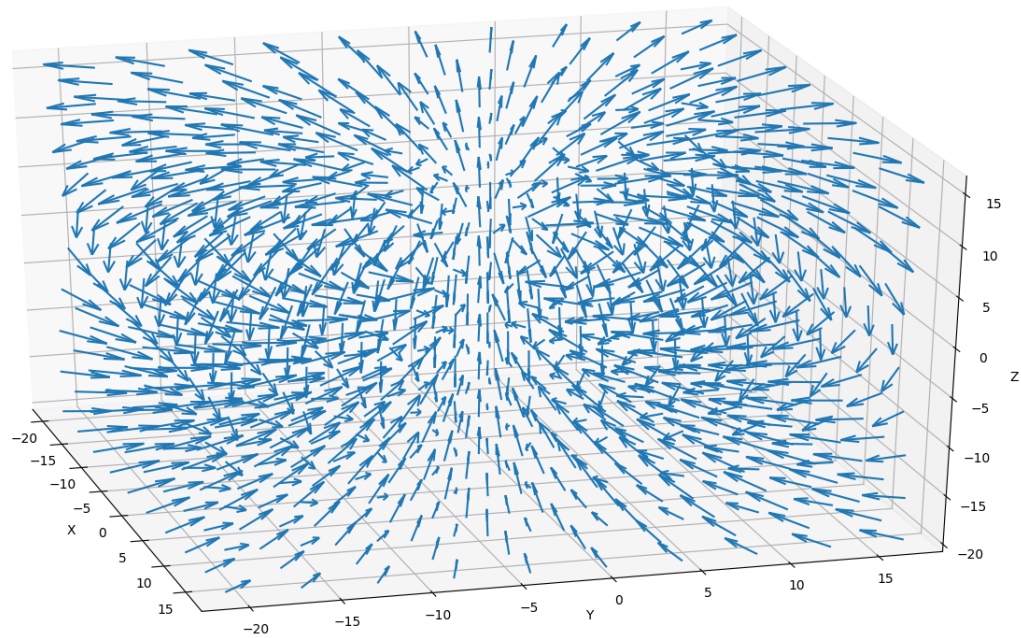




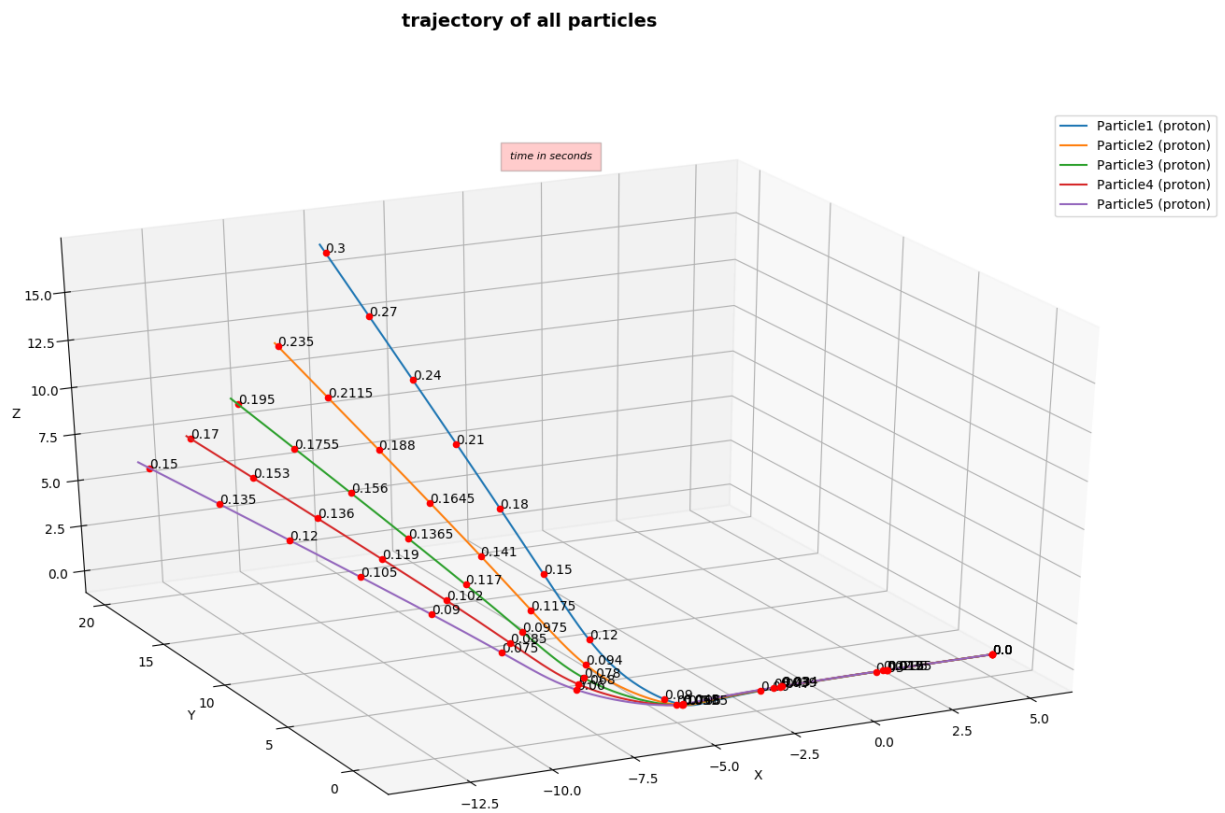
Figuur 5 deeltjesbanen voorbeeld 3

In het vierde en laatste voorbeeld werd er een cirkelvormige geleider met als middelpunt $(-7.5, 0, -0.5)$ en een straal van $2.5m$ opgesteld. De normaal op het cirkeloppervlak is gericht volgens de z -as. De stroom loopt in anti-wijzerzin (van bovenaf bekeken) en bedraagt $2A$. In figuur 6 wordt hiervan het genormaliseerde magnetische veld weergegeven. Zoals in voorbeelden 2 en 3 worden de trajecten van 5 protonen bepaald die allen dezelfde startpositie $(5, 0, 0)$, maar verschillende initiële snelheden hebben. Deze snelheden bedragen respectievelijk (voor proton 1 t.e.m. 5) $125, 150, 175, 200$ en $225 \frac{m}{s}$. De deeltjesbanen van de protonen zijn weergegeven in figuur 7.

Magnetic field plot, normalized



Figuur 6 *genormaliseerd magnetisch veld voorbeeld 4*



Figuur 7 deeltjesbanen voorbeeld 4

4 Conclusie

In deze thesis werd een software pakket ontwikkeld dat voorlopig enkel in staat is om de magnetische velden rond een willekeurige opstelling van stroomdoorlopen geleiders te bepalen, alsook het pad van geladen deeltjes in deze velden. De numerieke Walk on Spheres methode voor het bepalen van elektrische velden bleek echter geen goede resultaten te produceren. Deze methode kan eventueel nog verder onderzocht worden of er kan een alternatieve methode gebruikt worden. Er zijn nog enkele optimalisaties en uitbreidingen die kunnen gerealiseerd worden in de toekomst. Door de structuur van het programma is deze eenvoudig uit te breiden en aan te passen aan eigen noden. Zoals getoond werd, werkt het berekenen van deeltjesbanen in magnetische velden naar behoren. De XML bestanden die gebruikt werden voor de data invoer zijn handig om met te werken door hun herbruikbaarheid en overzichtelijkheid i.t.t. data invoer via interfaces. De output bestanden die het programma genereert zijn ook makkelijk te importeren in andere applicaties zoals excel of matlab.

Referenties

- [1] 3D Cartesian Coordinate Rotation (Euler) VI. March 2017. Austin (Texas, USA): National Instruments Corporation; [accessed 2018 April 30]
http://zone.ni.com/reference/en-XX/help/371361P-01/gmath/3d_cartesian_coordinate_rotation_euler/
- [2] Scheunders P. 2016. *Wiskundige methoden voor de fysica II*. Antwerpen: Universiteit Antwerpen (extract of pages 85-87)
- [3] Parametric Equation of a Circle in 3D. February 2014. Wolfram Demonstrations Project (Becker A.); [accessed 2018 May 6]
<http://demonstrations.wolfram.com/ParametricEquationOfACircleIn3D/>
- [4] Ragheb M. 2013. *MONTE CARLO SIMULATION: Science and Engineering Applications*. Urbana-Champaign (USA): University of Illinois. Chapter 6, POTENTIAL THEORY AND HEAT CONDUCTION DIRICHLET'S PROBLEM)
<http://mragheb.com/NPRE%20498MC%20Monte%20Carlo%20Simulations%20in%20Engineering/Potential%20Theory%20and%20Heat%20Conduction%20Dirichlet's%20Problem.pdf>
- [5] Sphere Point Picking. MathWorld—A Wolfram Web Resource (Weisstein, Eric W.); [accessed 2018 April 23]
<http://mathworld.wolfram.com/SpherePointPicking.html>
- [6] Daniel H. 1997. *Physik 2: Elektrodynamik - Relativistische Physik*. Berlin: Walter de Gruyter (extract of pages 360-361)
https://books.google.be/books?id=8vAC8YG41goC&pg=PA360&redir_esc=y#v=onepage&q&f=false
- [7] David E. 2015, *Distance to Circles in 3D*, Redmond WA 98052
<https://www.geometrictools.com/Documentation/DistanceToCircle3.pdf>
- [8] Yu-Kuang Hu B. 2008. *Averages of static electric and magnetic fields over a spherical region: a derivation based on the mean-value theorem*. Department of Physics, 250 Buchtel Commons, University of Akron, Akron, OH 44325-4001.
<https://arxiv.org/pdf/physics/0002021.pdf>