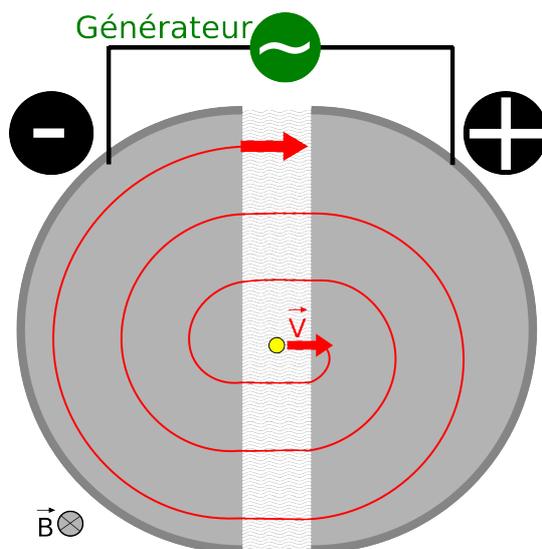
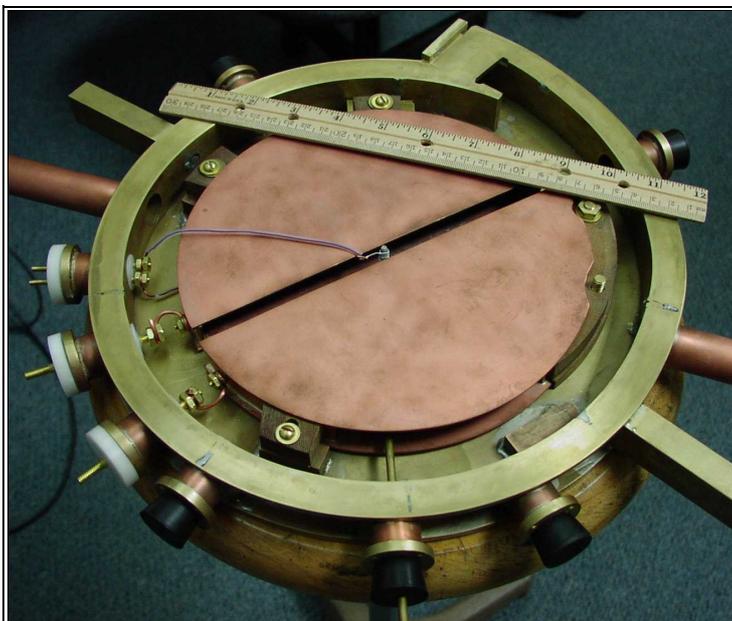


Simulation d'un accélérateur de particules



Simulation d'un accélérateur de particules à l'aide de Mathematica

Travail de maturité
par Axel Angel
3M7

assisté par Vincent de Coulon
Gymnase Auguste Piccard
Année 2009

0. Introduction

0.1 Résumé

Les accélérateurs de particules permettent d'accélérer à très grandes vitesses des particules électriquement chargées comme, par exemple, les électrons et les protons. Ils utilisent pour la plupart un condensateur et peuvent se servir d'un champ magnétique. Il existe deux catégories d'accélérateurs: les linéaires qui s'étendent sur plusieurs kilomètres de long et les circulaires qui peuvent être miniaturisés et qui forment la plus grande partie des accélérateurs en activité de nos jours. Ils sont utilisés principalement dans le domaine scientifique mais surtout dans le domaine de la physique en tant que collisionneurs. Ces derniers permettent avec une très grande énergie cinétique de faire éclater la matière et d'en explorer sa composition nucléaire. L'accélérateur qui nous intéresse est le cyclotron, inventé en 1929 par Ernest Orlando Lawrence. Il est circulaire et composé de deux dés creux de cuivre placés l'un en face de l'autre. Il utilise un champ électrique d'un condensateur à courant alternatif pour accélérer la particule entre les deux dés et un champ magnétique constant pour garder la particule dans l'appareil. Bien entendu le cyclotron a ses limites et il ne fait pas le poids face aux accélérateurs plus puissants comme le synchrocyclotron et le synchrotron. Nous avons choisi le cyclotron pour sa simplicité ce qui nous permettra de programmer d'abord un simulateur assez basique. En effet, d'un point de vue physique, il utilise les mouvements balistiques de base: Le MRUA (Mouvement Rectilignes Uniformément Accéléré) s'appliquant entre les dés lorsque la particule est soumise à l'accélération du champ électrique et le MCU (Mouvement Circulaire Uniforme) s'appliquant lorsque la particule se trouve à l'intérieur d'un dés soumis au champ magnétique. Les lois de Coulomb pour le champ électrique, la seconde loi de Newton pour les accélérations et la force de Lorentz seront nécessaires pour mener à bien les calculs de la simulation. Avec ces lois nous pourrons calculer la position de la particule à n'importe quel moment à l'aide des résultats de notre simulation. Le logiciel retenu pour la programmation est Mathematica, conçu par Stephen Wolfram. Ce logiciel est utilisé dans le domaine mathématique et scientifique en partie pour les calculs numériques et les calculs formels grâce à sa conception. Mathematica s'utilise avec son propre langage de programmation au travers de son interpréteur. Il est doté de nombreuses fonctions mathématiques, mais également graphiques. Notre code de simulation utilisera une petite quantité de fonctions, des boucles et des variables. Nous procéderons par étape successive en décomposant les mouvements. Nous commencerons par simuler une particule accélérée dans un condensateur. Puis nous programmerons une première passe de simulation, nécessaire à la simulation finale, qui calculera et stockera les caractéristiques de la particule aux points clés des mouvements. Nous débuterons en concevant notre simulateur d'abord classique, sans relativité, puis nous intégrerons les concepts de la relativité restreinte. S'en suivra la fabrication du simulateur final en utilisant les résultats de la première passe pour calculer la position à n'importe quel moment. Pour terminer, nous analyserons les différences entre les cas classiques et relativistes puis entre l'électron et le proton.

0.2 Motivation

Mon premier choix s'était porté sur la robotique car j'avais déjà programmé sur une plateforme facilitée de développement similaire et cela m'aurait permis d'aller plus loin. Malheureusement, le sujet n'étant plus disponible, mon choix s'est donc tourné vers la deuxième possibilité, celle développée dans ce dossier. Néanmoins, le sujet reste tout aussi intéressant, si ce n'est plus. En effet il comporte une partie de programmation avec le logiciel Mathematica, ce que j'apprécie puisqu'il est très utilisé dans le domaine scientifique et que cela me permettra de l'utiliser plus facilement dans le futur. Une autre partie, que j'apprécie également concerne la physique pure, qui me permet de revoir les notions de base dans un cas réel et d'aller plus loin en apprenant la relativité restreinte que nous n'étudierons, au programme, que bien plus tard. Ce dernier point me permet de prendre un peu d'avance, ce qui est fort appréciable. Je trouve également importante la partie concernant l'historique qui permet de se rendre compte des étapes de création, de l'évolution et des avantages ou utilités des accélérateurs dans la vie de tous les jours. De manière générale, j'ai choisi ce sujet pour le côté scientifique des recherches et des simulations ainsi que pour les parties de programmation qui sont très intéressantes lorsqu'on est déjà capable de programmer dans d'autres langages et que, de ce fait, on a la possibilité de se concentrer directement sur Mathematica. Dans ce travail, je trouve également intéressante la physique qui englobe le sujet, ajoutant de la matière fondamentale. En effet, c'est un sujet complexe mais tout à fait passionnant.

Sommaire :

0 Introduction	
0.0	Couverture 1
0.1	Résumé 2
0.2	Motivation 3
1 Les accélérateurs	
1.0	Introduction 5
1.1	Les applications 5
1.2	Le cyclotron 6
1.3	Le synchrocyclotron 7
1.4	Le synchrotron 7
2 Physique	
2.0	Introduction 8
2.1	Champ électrique 8
2.2	Champ magnétique 9
2.3	Application classique 10
2.4	Relativité restreinte 11
3 Mathematica	
3.0	Introduction 14
3.1	Fonctions 14
3.1.0	Catégorisation
3.1.1	Fonctions mathématiques
3.1.2	Fonctions graphiques
3.1.3	Fonctions de programmation
3.2	Application 17
3.2.0	Étape de création
3.2.1	Programmation de la simulation classique
3.2.1.1	Électron dans un condensateur
3.2.1.2	Pré-simulation, moments clés
3.2.1.3	Simulation finale
3.2.2	Programmation de la simulation relativiste
3.3	Analyse et comparaison des résultats 25
4 Conclusion et sources	
4.0	Conclusion 40
4.1	Liens et références 42

En annexe: un CD contenant le code des simulations au format PDF et Mathematica (.nb) ainsi que les tracer des trajectoires obtenues en grande résolution.

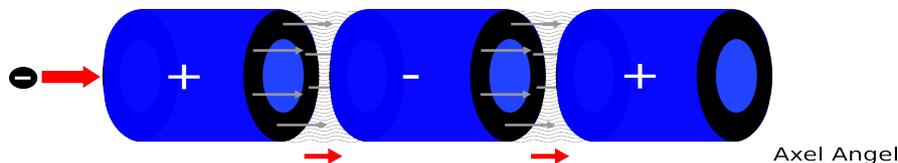
1. Les accélérateurs

1.0 Introduction

Les accélérateurs de particules servent à accélérer des électrons, des protons, ou de manière plus générale des particules électriquement chargées. Cette accélération permet d'atteindre de très grandes vitesses à l'aide de champ électriques alternatifs. Selon le type d'accélérateurs, chacun peut avoir recours à des champ magnétiques. Il existe deux catégories bien distinctes :

- Linéaire : L'accélération se passe généralement dans un long canon droit et linéaire. Le problème majeur de ce procédé est qu'il faut que celui-ci atteigne plusieurs kilomètres de long. L'accélérateur est composé d'une multitude de tubes polarisés de manière à ce que la particule soit attirée par le suivant grâce au champ électrique entres eux. Pour l'accélérateur à radiofréquences de Wideroë, la polarité est changée périodiquement et chaque tube est de plus en plus grand pour assurer une même périodicité quelque soit la vitesse de la particule.

Au laboratoire du *Centre de l'accélérateur linéaire de Stanford* aux États-Unis, les ingénieurs ont construit le plus grand accélérateur linéaire du monde qui s'étend sur plus de 3,2 kilomètres. Ils ont réussi à accélérer des électrons à des énergies avoisinant les 60 GeV, soit en théorie non-relativiste, une vitesse de 480 fois celle de la lumière. Bien entendu, en réalité il est impossible pour une particule de dépasser la vitesse de la lumière en tenant compte de la relativité. Voici un schéma d'un accélérateur de particules de type linéaire :



Axel Angel

- Circulaire : Cette catégorie peut être grandement miniaturisée à l'instar de la précédente. L'accélération se fait grâce à un unique champ électrique alternatif et un champ magnétique qui permet de garder la particule en rotation dans l'appareil. Il existe plusieurs variantes de celui-ci :
 - Cyclotron dont l'électron accéléré aura une trajectoire ressemblant à une spirale.
 - Synchrocyclotron est une version relativiste et améliorée du cyclotron.
 - Synchrotron est semblable au cyclotron mais dont la trajectoire est circulaire.
 Ceux-ci seront étudiés plus loin dans le chapitre.

1.1 Les applications

- Physique : Les scientifiques ont d'abord trouvé un intérêt pour l'accélérateur dans les recherches fondamentales de la physique (à but non lucratif) concernant les particules élémentaires. Ils ont inventé les collisionneurs qui, d'abord accélèrent celles-ci, puis les font s'écraser les unes contres les autres afin d'étudier leurs structures, leurs caractéristiques et leurs comportements, études qui ne peuvent pas être faites telles quelles dans la nature à

des vitesses faibles.

- Médicale : Les médecins ont trouvé les accélérateurs très utiles pour soigner certaines maladies comme, par exemple, le cancer par radiothérapie. Cela consiste à utiliser les particules à très grande vitesse afin de bloquer la capacité de multiplication des cellules malades grâce aux radiations; évidemment sans affecter les cellules saines. Cette méthode a fait ses preuves puisqu'elle permet de guérir pratiquement toute seule les tumeurs. Aujourd'hui elle est utilisée, en combinaison avec la chirurgie, sur plus de la moitié des malades ayant contracté un cancer . Les médecins utilisent les accélérateurs également pour stériliser des appareils chirurgicaux.
- Artistique : Il est possible par exemple, d'obtenir les caractéristiques d'une œuvre comme les composants chimiques ou la datation grâce aux radiations émises par les particules. Cela rend possible l'authentification sans avoir à détériorer l'objet.
- Militaire : Les accélérateurs permettent de simuler les armes nucléaires militaires dans le but de savoir si elles ont le comportement voulu, l'impact choisi ou encore pour tester leur dangerosité.
- Alchimie : Grâce au cyclotrons, il est devenu possible de transmuter des particules. En bombardant, par exemple, un noyau d'azote 14 avec des particules alpha, le scientifique Ernt Rutherford a réussi à obtenir de l'oxygène 17. De ce fait, les accélérateurs plus puissants permettent d'obtenir des isotopes artificiels ou rares d'une grande variété de noyaux combinés avec des particules alpha.

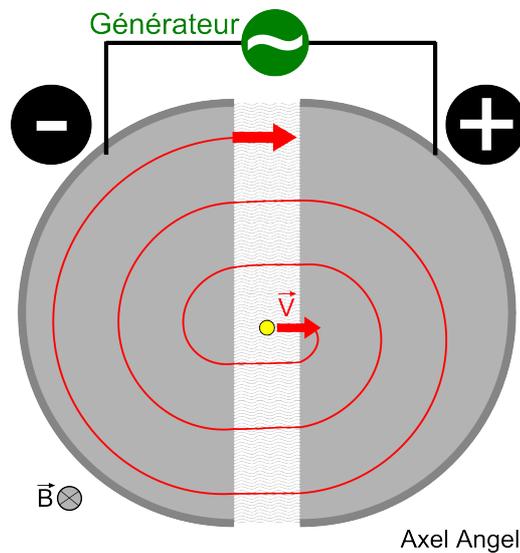
1.2 Cyclotron

Le cyclotron est un accélérateur, de particules chargées, de forme circulaire qui utilise de très fortes tensions alternatives et un champ magnétique. Il a été inventé par Ernest Orlando Lawrence en 1929 à l'université de Berkeley en Californie. C'est le premier accélérateur de particules réalisé au monde.

L'appareil est composé de deux dés creux à base de cuivre dont le rayon est en général de quelques mètres. Le cuivre a été choisi car il est diamagnétique, de ce fait il laisse passer le champ magnétique à l'intérieur des dés. Un générateur fournit une tension de l'ordre de 50kV relié à chacun des dés. Cette tension va s'alterner afin d'accélérer la particule progressivement mais uniquement lorsque celle-ci se trouve dans l'espace entre les deux dés. En effet, le champ électrique est nul dans les dés par effet de cage de Faraday et donc la particule ne sera pas influencée par ce dernier dans les dés. Le système doit comporter un champ magnétique perpendiculaire aux dés pour permettre à la particule de rester dans l'appareil le temps nécessaire pour qu'elle atteigne une vitesse déterminée. Bien entendu les dés, étant fait de cuivre, permettent au champ magnétique de passer au travers pour n'influencer que la particule. Après un certain temps, la particule aura acquis une vitesse si grande que le diamètre des dés sera insuffisant. C'est à ce moment là qu'elle devra être déviée sur une cible choisie. Avec cet accélérateur, il est possible d'obtenir une énergie de l'ordre de 30 MeV.

Les scientifiques, qui souhaitaient avoir un accélérateur plus puissant, se sont attelés au

synchrocyclotron, le successeur du cyclotron, pour fournir de plus grandes énergies aux particules.



Axel Angel

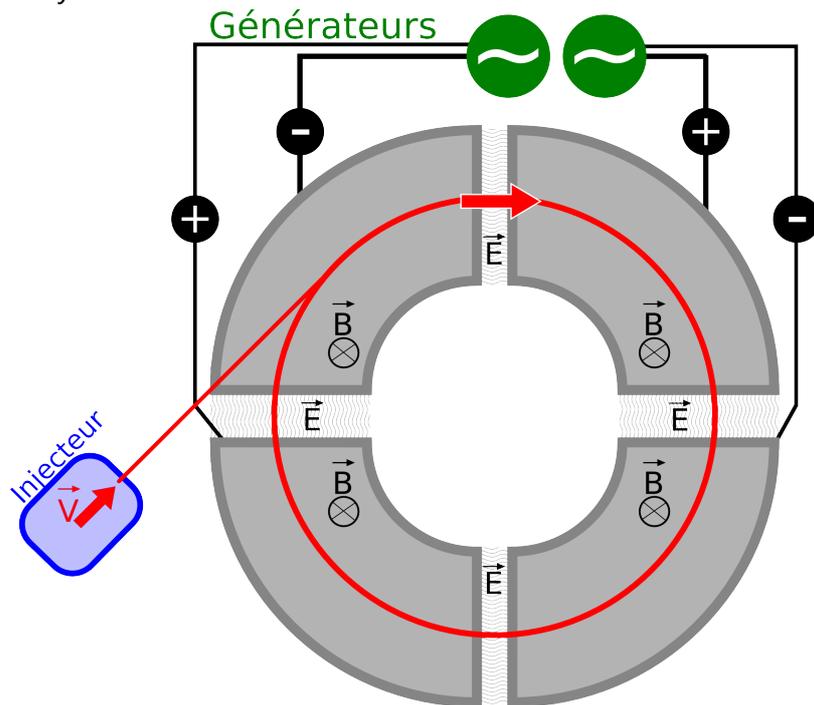
1.3 Synchrocyclotron

Le synchrocyclotron reprend toutes les caractéristiques de son aîné le cyclotron, à deux différences près, l'une étant qu'il n'a plus besoin de deux dés, un seul suffit et l'autre que son champ électrique va alterner de plus en plus lentement en fonction de la vitesse de la particule. Le but étant de s'adapter à l'augmentation de la masse de la particule en fonction de sa vitesse, qui s'explique par la loi de la relativité. Le premier atout de cet accélérateur est qu'il n'est pas limité dans le nombre de tours possibles tant que la fréquence du champ électrique peut être réduite pour s'accorder avec la vitesse. Le second est que l'intensité du champ magnétique peut être modifiée en fonction de la distance entre le potentiel positif et négatif des dés. L'espace qu'il y avait entre les dés du cyclotron n'est plus utile avec le synchrocyclotron parce qu'il n'y a plus besoin de fournir un puissant champ électrique pour faire de fortes accélérations. C'est pour cela qu'il est possible de n'utiliser plus qu'un seul dés, pour autant que l'extrémité du générateur soit relié à la terre. Avec ce type de machine, le CERN a pu fournir en 1958 une énergie de 600 MeV à des protons.

1.4 Synchrotron

Le synchrotron est un accélérateur reprenant la majeure partie des caractéristiques du cyclotron mais en plus puissant. Sa principale différence se situe dans le rayon fixe de l'orbite de la particule. En effet la particule va être forcée de tourner autour d'un centre bien déterminé. Ce qui va permettre de focaliser le champ magnétique sur une zone bien précise et de faire varier celui-ci selon l'énergie totale de la particule. Cette modification va imposer une fréquence variante du champ électrique et en conséquence du champ magnétique également. Ces deux champs ne seront plus indépendants comme ils l'étaient dans le cyclotron, on dit alors qu'ils doivent être synchronisés avec la particule, d'où le nom

de l'accélérateur: « synchro ».



Une autre différence est apparue sur l'injecteur de particules; Celui-ci n'est plus au centre des dés mais en dehors du système ce qui permet notamment de donner une vitesse initiale plus importante. Évidemment l'avantage du synchrotron par rapport au cyclotron est énorme puisque il tient compte de la relativité des masses par rapport à la vitesse des particules. À titre d'exemple, il a été possible d'atteindre avec l'accélérateur américain, Tévatron, les 1 TeV (soit 10^6 MeV) avec des électrons et le CERN a prévu d'obtenir dès 2010 une énergie proche des 7 TeV avec le LHC.

2. Physique

2.0 Introduction

Les accélérateur de particules ont recourt à une grande quantité de lois physiques pour assurer leur bon fonctionnement. Dans ce travail, nous nous focaliserons sur les principaux thèmes physiques du cyclotron comme l'accélération par un champ électrique et le magnétisme.

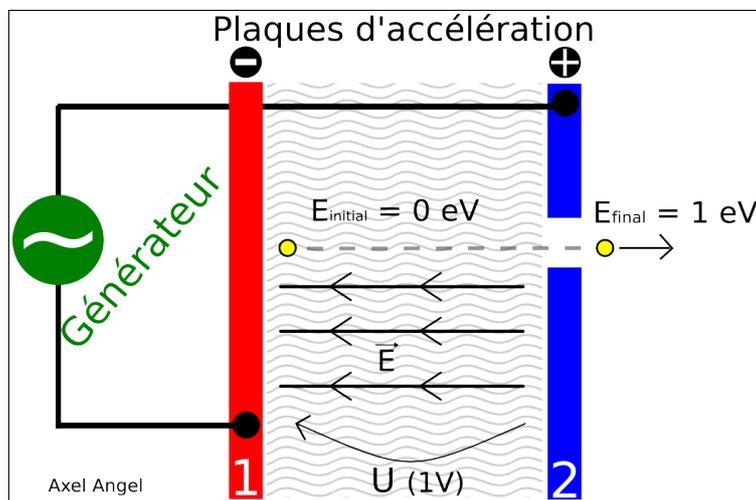
2.1 Champ électrique

Le champ électrique est une zone, où des particules immobiles de charges opposées sont éloignées et créent ainsi une tension. Toute particule se trouvant à l'intérieur d'un champ électrique subira une force d'attraction notée F_e de la loi de Coulomb. L'expression de cette force a une forte ressemblance avec celle de la force gravitationnelle F_g de la loi de Newton. En effet la loi de Coulomb, qui exprime l'attraction de particules ponctuelles chargées, dépend de la distance au carré, d'une constante et des charges s'attirant :

$$F_e = \frac{1}{\underbrace{4\pi\epsilon}_{\text{constante}}} \frac{q_1 q_2}{d^2} ; F_g = \frac{\underbrace{G}_{\text{constante}}}{d^2} m_1 m_2$$

Le champ électrique permet au cyclotron d'accélérer dans sa zone d'action une particule chargée. L'appareil utilise un condensateur, qui est composé d'une plaque chargée positivement et une seconde plaque chargée négativement. Elles sont reliées aux bornes d'un générateur fournissant une tension d'environ 75'000 Volt.

Dans le schéma ci-dessous, la particule qui est ici un électron (en jaune), est placée à l'intérieur d'un condensateur composé de deux plaques (en rouge et bleu) qui sont reliées à un générateur (en vert). L'électron va d'abord être attiré par la plaque positive vers la droite, accélérer et acquérir une énergie cinétique. Dans ce cas, les plaques ont une tension d'un volt. Après avoir atteint la sortie du condensateur, la particule aura acquis une énergie cinétique d'un électron-volt (eV).

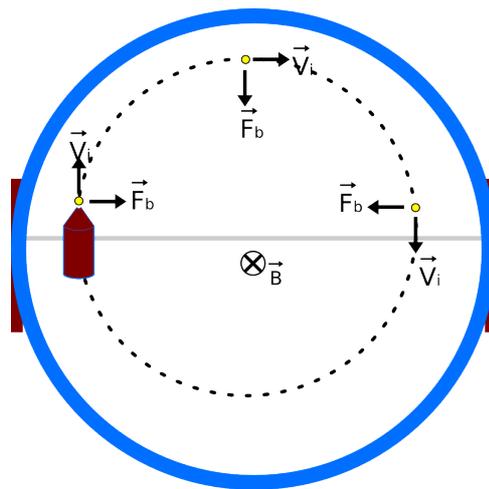


2.2 Champ magnétique

Le champ magnétique, qui est généré par un élément comme un aimant, un électroaimant ou un courant électrique, est une zone dans laquelle une particule chargée subira une force liée à sa vitesse et sa charge. Le champ magnétique est défini par un vecteur. Ces caractéristiques vont définir les vecteurs de forces agissant sur les particules influencées dans le champ. Dans le cas des accélérateurs de particules circulaires, le champ magnétique va obliger les particules à rester dans l'appareil grâce à la force de Lorentz :

$$\vec{F} = q\vec{E} + q\vec{V} \wedge \vec{B} \Rightarrow F = qE + qV B \sin(\alpha)$$

Bien évidemment il faut obligatoirement que le vecteur du champ soit perpendiculaire aux dés pour que la particule reste dans un même plan. Dans le cas contraire la particule sortira de l'appareil car la force de Lorentz fait apparaître une vitesse ayant une composante dans l'axe des Z (à cause du sinus de l'angle entre le vecteur vitesse et le vecteur champ).



Voici un exemple de cas avec un injecteur d'électrons en brun, un champ magnétique B homogène généré par des bobines d'Helmholtz et une trajectoire circulaire de particule. Dans ce cas-ci la particule va rester dans un même plan contenant l'injecteur car l'angle entre la vitesse et le champ est de 90° ; cela permettra à la particule d'avoir une trajectoire circulaire parfaite. C'est grâce au champ magnétique, ayant une direction s'enfonçant vers le bas, que la force agissant sur la particule sera toujours perpendiculaire à la vitesse, agissant comme une force centripète (MCU). Nous le verrons plus tard, mais c'est de cette manière que nous allons calculer la trajectoire dans les dés de l'accélérateur lorsque la particule subira la force du champ magnétique.

2.3 Application classique

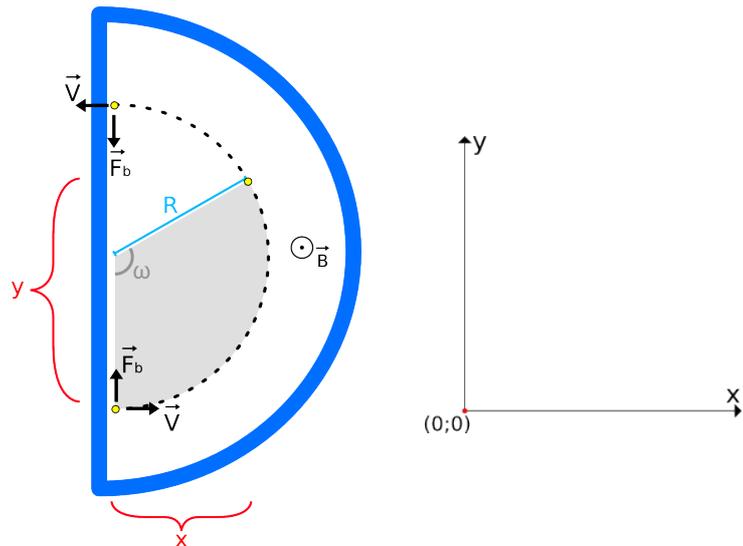
Dans le cadre du cyclotron, qui fait l'objet de cette étude, un champ électrique et un champ magnétique sont présents afin d'assurer respectivement l'accélération et le maintien de la particule dans l'appareil. D'un point de vue balistique non relativiste l'accélération de la particule dans le champ électrique entre les dés n'est qu'un simple MRUA. Son accélération peut être calculée à partir d'une équation basée sur la seconde loi de Newton (la force du champ électrique sera notée F_E). Nous obtenons:

$$\sum \vec{F} = m\vec{a} \Rightarrow F_E = qE = ma \Rightarrow a = \frac{qE}{m}$$

Ce qui nous donnera une équation balistique :

$$x = \frac{1}{2}at^2 + v_0t + x_0 = \frac{qE}{2m}t^2 + \underbrace{v_0t + x_0}_{0 \text{ à } t_0}$$

Maintenant, intéressons-nous au mouvement de la particule dans le champ magnétique à l'intérieur des dés. Sa trajectoire va former un demi cercle parfait :



Ce qui nous permet d'utiliser l'équation de trajectoire du MCU. Il nous faut avant tout trouver l'expression du rayon, en posant la seconde loi de Newton ainsi que la force de Lorentz (elle sera notée F_L). Il faut noter que l'angle entre la vitesse et la direction du champ magnétique est droit, par conséquent, le sinus vaut 1 donc nous pouvons l'omettre. Nous avons à droite de l'équation, l'accélération de type centripète expliqué par le MCU. Nous en déduisons le rayon en fonction de la vitesse, du champ magnétique, de la charge et de la masse :

$$\sum \vec{F} = m\vec{a} \Rightarrow F_L = qV B \underbrace{\sin(\alpha)}_1 = m \frac{V^2}{R} \Rightarrow R = \frac{mV}{qB}$$

Nous pouvons calculer la position à l'aide de l'équation du MCU, l'angle de position sera noté ωt :

$$\omega = \frac{V}{R} = \frac{qB}{m}$$

$$x = \sin(\omega t)R = \sin\left(\frac{qB}{m}t\right)\frac{mV}{qB}$$

$$y = (1 - \cos(\omega t))R = (1 - \cos\left(\frac{qB}{m}t\right))\frac{mV}{qB}$$

2.4 Relativité restreinte

La prochaine notion importante à voir est la relativité restreinte. En effet celle-ci va complètement bouleverser la compréhension de notre univers et permettre de revisiter les formules de la mécanique classique pour s'adapter à cette nouvelle notion. En somme, la relativité restreinte met une limite aux vitesses accessibles par les corps: la vitesse de la lumière ne pouvant pas être dépassée. Nous allons étudier la relativité d'Einstein qui est celle employée aujourd'hui, bien qu'il y ait celle de Galilée mais cette dernière est incorrecte pour des vitesses proches de c .

La relativité restreinte d'Einstein est très employée dans les systèmes de références

inertiels, c'est à dire dans tout système où la première loi de Newton est vérifiée. Le système d'un corps est inertiel lorsque la résultante des forces appliquées sur celui-ci est nulle, ce qui implique que ce dernier avance à vitesse constante.

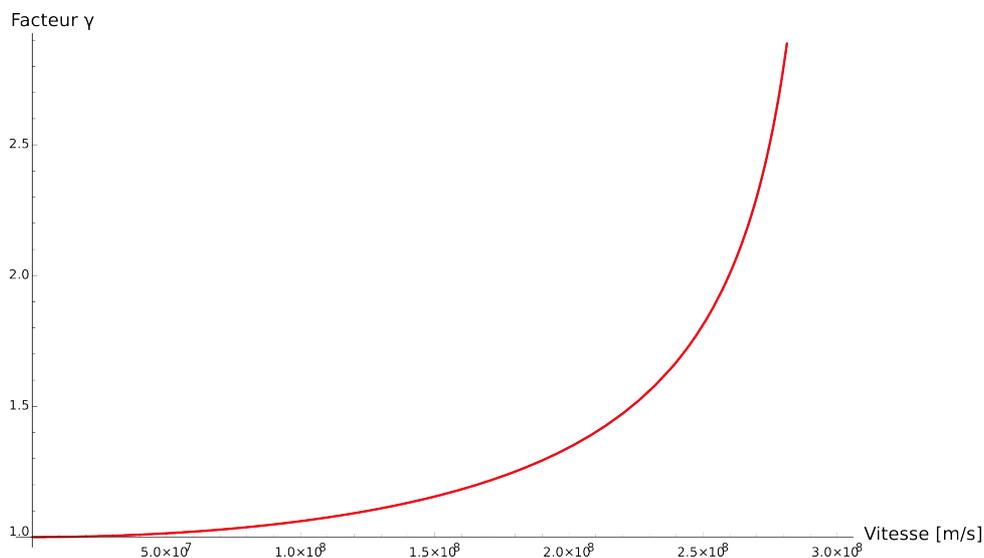
Tout d'abord, il faut savoir qu'en dynamique relativiste, les équations de mouvement font apparaître le facteur γ (prononcé gamma), qui diverge lorsque la vitesse de la particule, V , tend vers la vitesse de la lumière, c . Gamma s'exprime ainsi :

$$\gamma = \frac{1}{\sqrt{1 - \frac{V^2}{c^2}}} \text{ avec } |V| < c$$

Il force le facteur V à ne pas dépasser ni à égaler la vitesse c . En effet, d'une part, si V était égal à c alors le dénominateur serait nul et impliquerait un calcul impossible puisque la division par 0 n'est pas déterminée. D'autre part, si V était plus grand que c alors la valeur sous la racine serait négatif et, une fois encore, le calcul serait impossible.

$$\lim_{V \rightarrow 0} \gamma = \frac{1}{1} = 1 \text{ et } \lim_{V \rightarrow c} \gamma = \frac{1}{0} = +\infty$$

En étudiant la limite de l'expression γ pour les valeurs (positives) limites autorisées, nous remarquons qu'elle varie de 1 à l'infini positif. Ce facteur gamma est donc négligeable à des vitesses faibles mais devient très important à grandes vitesses. Nous pouvons le constater facilement en observant l'allure de γ en fonction de la vitesse de la particule :



Maintenant que nous avons défini et étudié gamma, nous pouvons chercher les expressions relativistes des mouvements. Tout d'abord, nous posons la seconde loi de Newton lorsque la particule se trouve entre les plaques du condensateur :

$$\sum \vec{F} = m \vec{a}$$

$$q \vec{E} = m \frac{d}{dt} (\gamma \vec{v})$$

La seule force agissant sur la particule, à ce moment, est celle du champ électrique. Nous posons ensuite que l'accélération est la dérivée de la vitesse qui comprend le facteur

gamma. Cette équation va nous permettre de trouver l'expression du mouvement rectiligne relativiste (MRR) comprenant le facteur γ . Nous poursuivons en isolant la dérivée à droite puis en intégrant de chaque côté.

$$\underbrace{\frac{qE}{m}}_a t + k_1 = \gamma v$$

Nous continuons à résoudre l'équation en trouvant le facteur d'intégration k_1 . Il suffit de prendre l'équation précédente au temps initial :

$$t=0 \Rightarrow v=v_0 \Rightarrow \underbrace{\frac{qE}{m}}_0 t + k_1 = \gamma_0 v_0$$

$$k_1 = V_0 \gamma(V_0) = V_0 \gamma_0$$

Il nous faut maintenant trouver le second facteur d'intégration, k_2 , en posant la condition initiale de position nulle :

avec : $x(0)=0$

$$x(0)=0 = \frac{c}{a} \sqrt{c^2 + k_1^2} + k_2 \Rightarrow k_2 = -\frac{c}{a} \sqrt{c^2 + V_0^2 \gamma_0^2} = -\frac{c}{a} \sqrt{\frac{c^4}{c^2 - V_0^2}} = -\frac{c^2}{a} \frac{c}{\sqrt{c^2 - V_0^2}}$$

$$k_2 = -\frac{c^2}{a} \gamma_0$$

Nous trouvons finalement les équations de mouvement attendues avec les constantes d'intégrations remplacées dans les formules :

$$x_{(t)} = \frac{c}{a} \sqrt{c^2 + \underbrace{(at + v_0 \gamma_0)}_{k_1}^2} + \underbrace{\frac{-c^2}{a}}_{k_2} \gamma_0$$

$$v_{(t)} = \frac{c(at + v_0 \gamma_0)}{\sqrt{c^2 + (at + v_0 \gamma_0)^2}}$$

Pour le mouvement circulaire relativiste (MCR), la démarche reste très semblable à celle que nous venons d'effectuer. Nous commençons par poser la seconde loi de Newton, en ajoutant à nouveau le facteur gamma à la vitesse :

$$m \frac{d}{dt} (\gamma \vec{v}) = q \vec{V} \wedge \vec{B}$$

$$m \gamma \underbrace{\frac{V^2}{R}}_{a_c} = q V B \text{ avec } \vec{V} \perp \vec{B}$$

Comme nous pouvons le constater ici, l'accélération est toujours la même que la version classique (non-relativiste), c'est à dire l'accélération centripète. Finalement nous retrouvons la même formule que pour le MCU standard à la différence du terme γ qui fait son apparition au côté de la vitesse :

$$R = \frac{m \gamma V}{q B} \text{ et } T = \frac{\pi m \gamma}{q B}$$

Sans surprise l'expression pour trouver le rayon est similaire à la version classique, néanmoins la période nous fait remarquer quelque chose d'intéressant. En effet, le terme gamma n'est plus un facteur de la vitesse mais de la masse. Il faut s'imaginer que la masse

est une énergie en relativité et que, plus elle grandit, plus le temps d'une période sera long à cause de l'inertie de la particule.

Le calcul de l'énergie cinétique en relativité démontre bien cette notion d'énergie massique car le carré de la vitesse de la lumière et le terme gamma sont utilisés comme facteur de la masse.

$$E_{cin} = E_{total} - E_0 = \gamma m c^2 - m c^2 = m c^2 (\gamma - 1)$$

Comme nous pouvons le voir ici, $mc^2\gamma$ est l'énergie totale atteinte au moment voulu et mc^2 est l'énergie au repos (cela vient de la loi d'Einstein qui pose $E = mc^2$ et qui le nomma énergie de masse). En calculant la différence ci-dessus, nous obtenons l'énergie pour passer de l'état au repos (A) à une vitesse donnée (B).

3. Mathematica

3.0 Introduction

Mathematica est un logiciel mathématique commercial conçu par Stephen Wolfram en 1988. Il est axé sur le calcul formel (les opérations sur des expressions littérales). Par exemple les simplifications, les intégrations, les dérivations ou encore les résolutions d'équations paramétriques. Il est très utilisé dans le monde scientifique car il intègre des outils de statistiques et de fabrication de graphiques. Il est doté d'une grande quantité de bibliothèques de fonctions utiles. De manière générale, il est optimisé et donc parfait pour les calculs complexes de la physique. C'est pour cette raison qu'il a été retenu pour traiter des accélérateurs de particules dans ce travail.

3.1 Fonctions

Parmi la grande quantité de fonctions qu'offre Mathematica, nous n'exploiterons qu'une quinzaine de celles-ci. Avant de les utiliser directement, nous allons les décrire dans la partie qui suit. Afin d'améliorer la compréhension, chaque fonction sera accompagnée de sa syntaxe et de ses paramètres d'entrées.

3.1.0 Catégorisation

Nous allons classer les différentes fonctions utiles à notre simulateur dans trois catégories bien distinctes.

3.1.1 Fonctions mathématiques

Cette catégorie regroupe les fonctions liées aux calculs mathématiques tels que le calcul numérique, différentielle, ou de modulo par exemple.

$$NSolve\left[\{ \text{équation}_1, \text{équation}_2, \text{équation}_n \}, \{ x_1, x_2, x_n \} \right]$$

La fonction *NSolve* (*Numerical Solve* traduit par résolution numérique) sert principalement à résoudre approximativement des équations numériques uniquement. Elle prend en 1er paramètre une ou plusieurs équations puis ensuite la ou les variables inconnues. Par exemple :

`NSolve[{x^2 - 6 x + 5 == 0}, {x}]` nous donnera : $\{\{x \rightarrow 1.\}, \{x \rightarrow 5.\}\}$

Abs[*value*]

La fonction *Abs* (*absolute* traduit par absolu) permet de retourner le paramètre *value* en valeur absolue.

Sqrt[*value*]

La fonction *Sqrt* (*square* traduit par racine) permet de retourner la racine carrée du paramètre *value*.

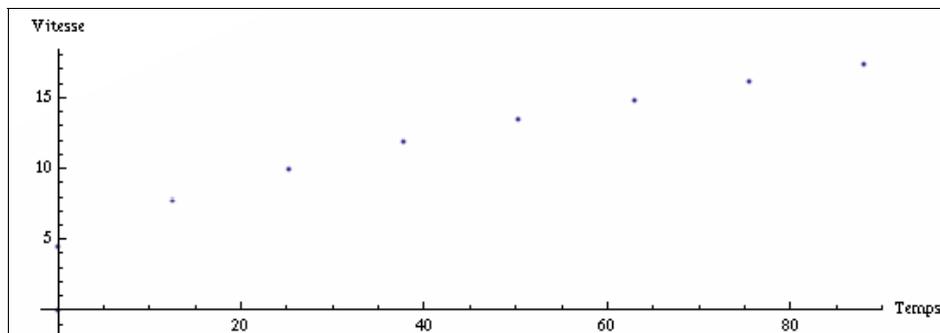
Mod[*dividende, diviseur*]

La fonction *Mod* (*modulo* comme en français) permet de retourner le reste de la division donnée par les paramètres *dividende* et *diviseur*.

3.1.2 Fonctions graphiques

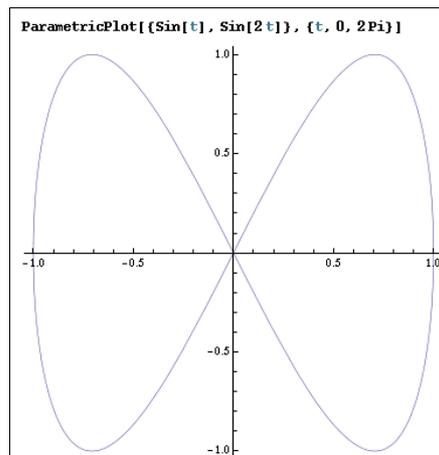
ListPlot[{ *liste₁*, *liste₂*, *liste_n* }]

La fonction *ListPlot* (traduit par tracer d'une liste) sert à afficher sur un graphique les points exprimés par les listes données en paramètres. Ce type de graphique est aussi appelé nuage de points. Voici un exemple avec la représentation graphique de la vitesse d'une particule accélérée dans un condensateur :



ParametricPlot[{ f_x , f_y }, { *valeur*, *valeur_{min}*, *valeur_{max}* }]

La fonction *ParametricPlot* (traduit par tracer paramétrique) permet de faire un graphique à partir de paires d'équations paramétriques donné par f_x et f_y . Ces deux fonctions doivent donner l'une la coordonnée en *x* et la deuxième la coordonnée en *y* en fonction d'un paramètre nommé *valeur*. Il faut spécifier l'intervalle de valeurs à afficher avec les deux derniers paramètres *valeur_{min}* et *valeur_{max}*. Cette fonction permet de dessiner des courbes qui ne peuvent être exprimés par une fonctions comme les cercles, les ellipses ou les tores par exemple. Elle nous permettra de dessiner la trajectoire de notre particule dans l'accélérateur.



3.1.3 Fonctions de programmation

`Print[expression]`

La fonction *Print* (traduit par *imprimer/afficher*) permet d'écrire dans la sortie standard une expression. Elle peut contenir du texte, des symboles ou des variables. Cette fonction permet de déboguer des instructions ou d'afficher un résultat textuelle proprement.

`If[{ condition , t , f }]`

La fonction *If* (traduit par *si*) sert à évaluer le contenu de *condition* en retournant la valeur *t* lorsque celle-ci est équivalente à vraie sinon *f*. Elle s'avèrera très utile pour définir quand l'accélération du champs électrique devra être appliquée ou pas.

`For[initialisation , condition , incrémentation , { instructions }]`

La fonction *For* (traduit par *pour*) est une commande de programmation de base permettant de faire des boucles. Le paramètre *initialisation* permet de définir la valeur de départ d'une variable qui sera incrémentée par l'instruction *incrémentation* avant chaque itération. Le second paramètre *condition* définit selon sa validité si la boucle doit continuer de s'itérer ou non. Le dernier paramètre *instructions* est le corps de la boucle contenant les instructions de chaque itération. Voici à droite un exemple simple avec une boucle affichant dix itérations.

```
For[i = 0, i < 10, i++; {
  Print["Itération numéro ", i]
}];
Itération numéro 1
Itération numéro 2
Itération numéro 3
Itération numéro 4
Itération numéro 5
Itération numéro 6
Itération numéro 7
Itération numéro 8
Itération numéro 9
Itération numéro 10
```

Append[*liste, élément*]

La fonction *Append* (traduit par appondre) retourne la liste ou le tableau spécifié en 1er paramètre en ajoutant la cellule donnée en second paramètre à la fin de celui-ci.

3.2 Application

3.2.0 Étape de création

Pour constituer une simulation de l'accélérateur de particules complet, il nous faut tout d'abord procéder par étapes successives. Nous allons commencer par simuler un électron dans un condensateur subissant une accélération constante sans effet relativiste. Cela nous permettra de l'intégrer facilement à la partie MRUA entre les dés dans le simulateur. En ce qui concerne le MCU, nous l'intégrerons directement au code. Suite à cela nous allons préparer une importante partie du simulateur qui s'occupe de faire une pré-simulation. Elle permet de calculer les temps critiques (ou moments clés) où la particule change de type de mouvement, dans un intervalle de temps donné. Elle permet également de calculer les caractéristiques à ces moments précis telles que la position et la vitesse. En pratique ces temps critiques sont présents aux instants où la particule se trouve à la sortie ou à l'entrée d'un dés. Ces temps nous serviront à simplifier et à accélérer les calculs du simulateur du cyclotron final. L'étape suivante consiste à combiner la simulation des mouvements MRUA et MCU en exploitant les informations calculées par la pré-simulation. Cette étape mène directement à un simulateur fonctionnel qui permet de connaître la position de la particule en un instant quelconque, pour autant que celui-ci soit compris dans l'intervalle de la pré-simulation établie précédemment.

3.2.1 Programmation de la simulation classique

Après avoir abordé les différentes fonctions utiles à la programmation, ainsi que l'approche à utiliser pour concevoir, par étapes, un simulateur non-relativiste complet d'un cyclotron, nous pouvons maintenant parler et commenter les codes de programmation des différentes étapes avec la matière traitée. Nous vous présenterons le code final de chaque étape tout en le détaillant.

3.2.1.1 Électron dans un condensateur

La première étape de la programmation consiste à attribuer à des variables de Mathematica, des valeurs globales qui seront utilisées fréquemment, par exemple des constantes ou des informations statiques. Il faut noter que Mathematica n'initialise pas la majeure partie des variables physiques connues, il faut les établir dans cette section. Nous avons dans cet exemple la charge de la particule (q), sa masse (m) et la tension de crête entre les plaques (U). La distance d représente l'espacement des plaques. Tandis que la variable Ec contient la valeur du champ électrique, la variable a est l'accélération dans le condensateur, X_0 et V_0 sont les paramètres initiaux du premier mouvement MRUA. Il faut noter que U et d utilisés dans cette partie sont inventés à des fins de vérifications.

```

q = -1.60217653 * 10^-19;
U = -10;
d = 5;
Ec = U / d;
m = 9.109382616 * 10^-31;
a = q * Ec / m;
Xo = 0;
Vo = 0;

```

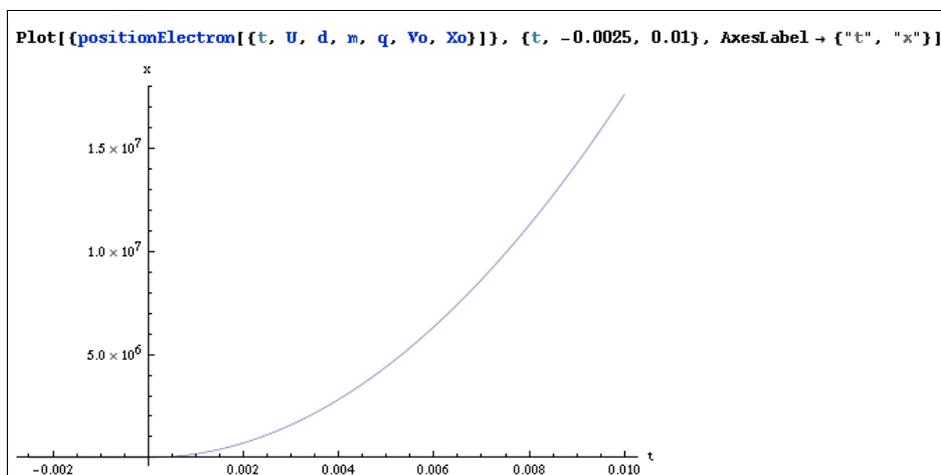
Cette section du code est importante puisqu'elle nous permet de définir des constantes à un seul endroit sans avoir à nous occuper du reste du code. En effet, un code ayant ses constantes à son en-tête est plus facile à entretenir et à comprendre. Lorsque nous souhaitons faire des essais avec des valeurs simples, il nous suffit de modifier cette partie afin de savoir si les résultats obtenus sont cohérents. Voici ce que nous obtenons avec les valeurs fictives :

```

accelerationDuChampsEl[{t_, U_, d_, m_, q_, Vo_, Xo_}] := If[t ≥ 0 and
  t ≤ NSolve[d == 1/2*a*X^2 + Vo*X + Xo, X][[2]][[1]][[2]], q*(U/d)/m, 0]
accelerationDuChampsEl[{0, U, d, m, q, Vo, Xo}]
3.51764 × 1011
positionElectron[{t_, U_, d_, m_, q_, Vo_, Xo_}] :=
  1/2*accelerationDuChampsEl[{t, U, d, m, q, Vo, Xo}]*t^2 + Vo*t + Xo
positionElectron[{0.5, U, d, m, q, Vo, Xo}]
4.39705 × 1010

```

La suite de la programmation consiste à calculer la position entre les dés avec les variables données ci-dessus, aux lignes 1 et 2. Nous définissons tout d'abord une fonction retournant l'accélération du champs en fonction du temps. Celle-ci résout en premier lieu l'équation de position afin de savoir si la particule se trouve bien dans le condensateur et, si c'est le cas, elle retourne l'accélération. Nous testons à la ligne 3 le bon fonctionnement de la fonction. Nous créons aux lignes 5 et 6, la fonction finale qui va utiliser la fonction d'accélération précédente pour calculer la position à n'importe quel moment de ce MRUA. Une fois encore, nous testons à la ligne 8, que la fonction retourne une valeur correcte.



Dans cette portion de code, nous affichons graphiquement le résultat de notre fonction de position. De cette manière nous pouvons remarquer visuellement que la particule n'est accélérée qu'à l'intérieur du condensateur en négligeant les effets avant et après les plaques. Une fois que toutes les lignes ont été vérifiées comme nous l'avons fait nous pouvons passer à l'étape suivante.

3.2.1.2 Pré-simulation, moments clés

Dans cette catégorie nous nous occupons de programmer la partie calculant le temps des moments critiques (ou moments clés) de notre simulation. Comme dit plus haut, ceux-ci se trouvent au moment où la particule change de mouvement, passant d'un MRUA à un MCU ou inversement. La pré-simulation va franchir un niveau de programmation plus complexe avec l'utilisation de boucles et de fonctions de vérification de condition. Voici les grandeurs physiques qui seront utilisées dans les sections suivantes, y compris la simulation finale :

```

q = 1.60217653 * 10^-19;
U = 75000;
d = 0.0004;
Ec = U / d;
m = 9.109382616 * 10^-31;
Ac = q * Ec / m;
Xo = 0;
Vo = 0;
B = 2.0;
Loop = 30;

```

Comme dans la partie précédente nous commençons toujours par définir dans les premières lignes, les variables constantes que nous utiliserons tout au long du code. Certaines ont déjà été introduites plus haut mais d'autres sont nouvelles. La variable B représente la valeur en Tesla du champ magnétique appliqué perpendiculairement aux dés, la variable Loop définit le nombre d'itérations que la boucle de la pré-simulation effectuera (une itération est composée d'un mouvement MRUA et un MCU).

La première passe de la simulation est composée d'une partie qui initialise les variables de stockage des données. Elles contiendront les caractéristiques de la particule aux moments clés.

```

V = Vo;
VArray = {V};
TempsArray = {0};
TempsTotal = 0;
TempsVitArray = {{0, V}};
(* Tableaux de positions *)
TempsPosXArray = { {0, 0} };
TempsPosYArray = { {0, 0} };
TempsPosArray = { {0, {0, 0}} };
RayonsArray = { };
actualPosX = 0;
actualPosY = 0;

```

La variable *V* conservera la vitesse courante de la particule. *Varray* et *TempsArray* sont des tableaux qui contiendront respectivement toutes les vitesses et les temps des moments clés. La variable *TempsTotal* conservera le temps actuel au cours de la pré-simulation,

TempsVitArray associe le temps à la vitesse, les tableaux *TempsPosXArray* et *TempsPosYArray* contiendront les coordonnées de la particule en X et en Y respectivement aux points critiques associés au temps. Il faut noter que les lignes grises commençant par les signes « (*) » et se terminant par « *) » sont des commentaires ignorés par Mathematica. *TempsPosArray* est un tableau redondant contenant les mêmes données que les deux tableaux précédents. *RayonsArray* est une variable stockant le rayon des mouvements MCU. En dernier, nous avons *actualPosX* et *actualPosY* qui stockeront les coordonnées courantes au fil de la pré-simulation. Toutes ces variables nous permettent maintenant de calculer les moments clés et d'enregistrer les caractéristiques acquises au cours du temps.

```

1 For[i = 0, i < Loop, i++, {
2   (* Calculs des temps des mouvements *)
3   (* -- X *)
4   tempsX = NSolve[d == 1/2*Ac*Inc^2 + Abs[V]*Inc + Xo, Inc][[2]][[1]][[2]];
5   (* -- Y *)
6   V = Sqrt[V^2 + (2*U*q/m)];
7   If[Mod[i, 2] == 0, V = V, V = -V]; (* Gère la vitesse signée *)
8   VArray = Append[VArray, V];
9   R = Abs[(m * V) / (q * B)];
10  RayonsArray = Append[RayonsArray, R];
11  tempsY = Abs[(Pi * R) / V];
12  (* Enregistre les informations dans les tableaux *)
13  (* TempsArray = Append[TempsArray, TempsTotal+tempsX]; *)
14  (* Mouvement MRUA *)
15  If[Mod[i, 2] == 0, actualPosX += d, actualPosX -= d];
16  TempsPosXArray = Append[TempsPosXArray, {TempsTotal + tempsX, actualPosX}];
17  TempsPosYArray = Append[TempsPosYArray, {TempsTotal + tempsX, actualPosY}];
18  (* -- Remplis le tableau de position complet *)
19  TempsPosArray = Append[TempsPosArray, {TempsTotal + tempsX, {actualPosX, actualPosY}}];
20  (* Mouvement MCU *)
21  If[Mod[i, 2] == 0, actualPosY += 2*R, actualPosY -= 2*R];
22  TempsPosXArray = Append[TempsPosXArray, {TempsTotal + tempsX + tempsY, actualPosX}];
23  TempsPosYArray = Append[TempsPosYArray, {TempsTotal + tempsX + tempsY, actualPosY}];
24  (* -- Remplis le tableau de position complet & stocke les données *)
25  TempsPosArray = Append[TempsPosArray, {TempsTotal + tempsX + tempsY, {actualPosX, actualPosY}}];
26  TempsArray = Append[TempsArray, TempsTotal + tempsX];
27  TempsVitArray = Append[TempsVitArray, {TempsTotal + tempsX, V}];
28  TempsVitArray = Append[TempsVitArray, {TempsTotal + tempsX + tempsY, -V}];
29  TempsArray = Append[TempsArray, TempsTotal + tempsX + tempsY];
30  (* Affichage des résultats *)
31  Print[" [", i, "] V: ", V, "; T(", (TempsTotal + tempsX), ", ", (TempsTotal + tempsX + tempsY), "); R: ", R, "; Pos(", actualPosX,
32  ", ", actualPosY, ") "];
33
34  TempsTotal += tempsX + tempsY;
35  }];

```

Cette portion du code est consacrée à la pré-simulation. Nous allons la voir en détail avec des commentaires dans le paragraphe suivant. Tout d'abord il faut noter que cette pré-simulation va s'occuper, dans une même itération, du couple MRUA et MCU de la particule. En prenant une base d'axes orientés verticalement vers la droite pour x et horizontalement vers le haut pour y, les déplacements de la particule seront une fois positifs et une fois négatifs. On notera aussi que la pré-simulation ne calculera pas toutes les positions durant les mouvements mais ne s'occupera que des points critiques. Cela simplifie grandement les opérations puisque nous sautons et simplifions une grande quantité de calculs.

La première ligne introduit la boucle d'itération de la pré-simulation qui attribue à la variable i le nombre d'itérations effectuées. Le nombre d'itérations, qui est défini par la

condition donnée en second paramètre de For, correspond à la valeur de Loop. À la quatrième ligne, nous calculons le temps nécessaire à la particule pour effectuer, le MRUA, soit la distance séparant les deux dés en prenant en compte les éventuelles positions ou vitesses de départ. Nous calculons à la sixième ligne l'énergie cinétique acquise après le mouvement. A la septième ligne nous contrôlons si les mouvements sont positifs ou négatifs selon nos axes, causant un changement de signe de la vitesse. A la ligne huit, nous stockons la vitesse atteinte à la fin d'un tableau, puis à la suivante nous nous occupons du MCU. De la ligne neuf à onze nous déterminons et stockons les caractéristiques de celui-ci, soit le rayon et la vitesse du mouvement. À la ligne quinze, nous vérifions le signe du delta x de position, selon les axes, que nous ajoutons à la position actuelle. Aux lignes seize à dix-neuf, nous stockons les coordonnées de la particule après le MRUA. À la ligne vingt et une et suivantes, nous calculons et stockons les coordonnées à la fin du MCU. Des lignes vingt-six à vingt-neuf, nous stockons les caractéristiques telles que les composantes de vitesse. Nous affichons les résultats à des fins de vérification à la ligne trente et une avec la commande Print. Finalement nous ajoutons les delta temps des deux mouvements à la variable de temps absolue TempsTotal pour que la boucle continue à partir de celle-ci. Voici un extrait de ce que donne la sortie après une pré-simulation.

```
[0] V: 1.62426 × 108; T(4.92531 × 10-12, 1.38563 × 10-11); R: 0.000461748; Pos(0.0004, 0.000923496)
[1] V: -2.29705 × 108; T(1.58964 × 10-11, 2.48274 × 10-11); R: 0.00065301; Pos(0., -0.000382524)
[2] V: 2.81331 × 108; T(2.63928 × 10-11, 3.53238 × 10-11); R: 0.000799771; Pos(0.0004, 0.00121702)
[3] V: -3.24853 × 108; T(3.66435 × 10-11, 4.55745 × 10-11); R: 0.000923496; Pos(0., -0.000629974)
[4] V: 3.63196 × 108; T(4.67372 × 10-11, 5.56682 × 10-11); R: 0.0010325; Pos(0.0004, 0.00143502)
[5] V: -3.97862 × 108; T(5.67193 × 10-11, 6.56503 × 10-11); R: 0.00113105; Pos(0., -0.000827069)
[6] V: 4.2974 × 108; T(6.66169 × 10-11, 7.55479 × 10-11); R: 0.00122167; Pos(0.0004, 0.00161627)
[7] V: -4.59411 × 108; T(7.64476 × 10-11, 8.53786 × 10-11); R: 0.00130602; Pos(0., -0.000995769)
[8] V: 4.87279 × 108; T(8.62237 × 10-11, 9.51546 × 10-11); R: 0.00138524; Pos(0.0004, 0.00177472)
[9] V: -5.13637 × 108; T(9.59539 × 10-11, 1.04885 × 10-10); R: 0.00146018; Pos(0., -0.00114563)
[10] V: 5.38707 × 108; T(1.05645 × 10-10, 1.14576 × 10-10); R: 0.00153144; Pos(0.0004, 0.00191726)
[11] V: -5.62661 × 108; T(1.15302 × 10-10, 1.24233 × 10-10); R: 0.00159954; Pos(0., -0.00128183)
[12] V: 5.85636 × 108; T(1.2493 × 10-10, 1.33861 × 10-10); R: 0.00166486; Pos(0.0004, 0.00204789)
[13] V: -6.07744 × 108; T(1.34531 × 10-10, 1.43462 × 10-10); R: 0.0017277; Pos(0., -0.00140752)
```

Cette pré-simulation nous a permis de stocker dans des variables tableaux notamment les caractéristiques aux points clés de la simulation, cela nous permettra de calculer plus rapidement les coordonnées de la particules à n'importe quel moment compris dans la pré-simulation. Grâce à cette étape, notre simulateur n'aura plus qu'à calculer, à partir du dernier point clé et de ses caractéristiques, le mouvement entre le point clé et le moment donné. Bien évidemment, cette méthode est optimisée pour des calculs uniques, il ne l'est plus dès que l'on souhaite faire un graphique ou un tableau des positions sur toute la durée de la simulation. Dans ce cas nous pourrions nous passer d'une pré-simulation et faire directement les calculs à la chaîne en une fois. Cependant nous préférons cette méthode car elle est plus simple à mettre en œuvre et la perte de temps encourue par ce choix est négligeable dans cette simulation (de l'ordre de quelques secondes).

3.2.1.3 Simulation finale

La partie de code ci-dessous est le simulateur complet. Il utilise les tableaux de la pré-simulation pour obtenir les résultats souhaités.

```

1 DEBUG = 1;
2 positionFinale[t_] := (
3   (* Cherche le temps de départ du mouvement cherché au temps t *)
4   For[i = 1, TempsArray[i] < t, i++];
5   (* Le i trouvé correspond à l'index de la cellule suivante du temps cherché *)
6   (* -- Et empeche qu'on soit "out of range" de la list *)
7   If[i > 1, i -= 1, i = 1];
8   If[DEBUG == 1, Print["i=", i]];
9   lastTemps = TempsArray[i];
10  lastX = TempsPosXArray[i][[2]];
11  lastY = TempsPosYArray[i][[2]];
12  If[DEBUG == 1, Print["lastTemps=", lastTemps, " , lastX=", lastX, " , lastY=", lastY]];
13  lastVit = TempsVitArray[i][[2]];
14  If[Abs[lastVit] == lastVit, currAccX = Ac, currAccX = -Ac];
15  If[DEBUG == 1, Print["currAccX=", currAccX]];
16  currX = 0;
17  currY = 0;
18  deltaT = t - lastTemps;
19  If[DEBUG == 1, Print["deltaT=", deltaT, " , lastVit=", lastVit]];
20  (* regarde si c'est MRUA/MCU et calcule la position *)
21  (* -- Mod i par 2 donne 0=MCU 1=MRUA *)
22  If[Mod[i, 2] == 1, {
23    (* MRUA *)
24    If[DEBUG == 1, Print["MRUA"]];
25    currX = 1/2 * currAccX * deltaT^2 + lastVit * deltaT + lastX;
26    currY = lastY;
27  },
28  {
29    (* MCU *)
30    If[DEBUG == 1, Print["MCU"]];
31    currRayon = RayonsArray[i][[2]];
32    vitAngulaire = lastVit / currRayon;
33    currAngle = vitAngulaire * deltaT;
34    If[DEBUG == 1, Print["currRayon=", currRayon, " , vitAngulaire=", vitAngulaire, " , currAngle=", currAngle, " deltaX=", deltaX]];
35    deltaX = Sin[currAngle] * currRayon;
36    deltaY = Abs[(1 - Cos[currAngle]) * currRayon];
37    currX = lastX + If[Mod[(i/2), 2] == 0, deltaX, deltaX];
38    currY = lastY + If[Mod[(i/2), 2] == 0, -deltaY, deltaY];
39  }
40 ];
41 If[DEBUG == 1, Print["Av [", TempsArray[i], "] X: ", lastX, " , Y: ", lastY, ""]];
42 If[DEBUG == 1, Print["Pd [", t, "] X: ", currX, " , Y: ", currY, ""]];
43 If[DEBUG == 1, Print["Ap [", TempsArray[i+1], "] X: ", TempsPosXArray[[i+1]][[2]], " , Y: ", TempsPosYArray[[i+1]][[2]], ""]];
44 {currX, currY}
45 );
46 positionFinaleX[t_] := (
47   positionFinale[t][[1]]
48 );
49 positionFinaleY[t_] := (
50   positionFinale[t][[2]]
51 );

```

Tout d'abord, à la ligne une, nous avons ajouté une variable qui nous permet de déboguer le code. La fonction écrit des lignes d'informations supplémentaires lorsque *DEBUG* est défini à 1 (que nous appellerons « mode verbeux »). Bien entendu, cette variable sera désactivée lors de simulation requérant une grande quantité d'appel à cette fonction sous peine de voir Mathematica planter à cause de la quantité de texte qui serait écrite par cette dernière. À la ligne suivante, nous définissons cette fonction avec le paramètre de temps. La ligne quatre permet de trouver la cellule du tableau contenant les caractéristiques du dernier moment clé afin de les utiliser comme paramètres initiaux des mouvements. Les lignes suivantes, de la septième à la treizième, nous permettent de récupérer les caractéristiques enregistrées par la pré-simulation, telles que le temps, les coordonnées et la vitesse de la particule au dernier moment clé dans des variables pour un accès plus facile. Notons que les lignes commençant par un if avec *DEBUG*, nous permettent d'afficher des informations lorsque nous sommes en mode de débogage. Nous ne les commenterons pas puisqu'elles ne participent pas directement à la simulation. À la ligne

quatorze, nous déterminons le sens de l'accélération du MRUA. Aux lignes seize et dix-sept, nous définissons les coordonnées initiales à zéro. La ligne dix-huit nous permet d'obtenir le delta temps à utiliser pour le mouvement actuel. À la ligne vingt-deux, nous déterminons par un modulo si celui-ci est un MRUA ou un MCU. En effet, puisque le type de mouvement alterne à chaque fois, un modulo deux nous indique quel mouvement la particule suit. Nous avons aux lignes suivantes une séparation entre le code du MRUA et celui du MCU. Aux lignes suivantes, entre accolades, nous avons la partie MRUA calculant la position au temps δt et nous définissons la position en Y ne changeant pas. Pour la partie MCU, nous récupérons le rayon enregistré par la pré-simulation puis nous calculons la vitesse angulaire et l'angle à l'instant δt . Il faut noter que les angles sont en radian par conséquent la particule accomplit un demi-tour soit un π pour chaque MCU. Aux lignes trente-sept et trente-huit, nous déterminons les nouvelles coordonnées de la particule grâce à l'angle calculé précédemment. Nous avons ajouté deux fonctions à la fin du code pour faciliter la récupération d'une coordonnée si nécessaire.

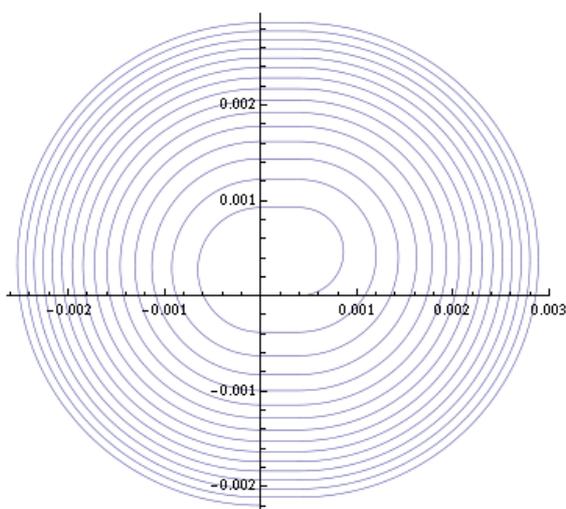
La prochaine étape de programmation que nous avons faite concerne un court test et un graphique de la trajectoire.

```

1  DEBUG = 1;
2  positionFinale[TempsArray[[(Loop + 1)]]] (* Prend un temps au milieu de la simulation *)
   i=30
   lastTemps=1.44109×10-10 , lastX=0.0004 , lastY=-0.00140752
   currAccX=3.29779×1019
   deltaT=8.93097×10-12 , lastVit=6.29074×108
   MCU
   currRayon=0.00178834 , vitAngulaire=3.51764×1011 , currAngle=3.14159 deltaX=-0.000807925
   Av [1.44109×10-10] X: 0.0004 , Y: -0.00140752
   Pd [1.5304×10-10] X: 0.0004 , Y: 0.00216916
   Ap [1.5304×10-10] X: 0.0004 , Y: 0.00216916
   {0.0004, 0.00216916}

5  DEBUG = 0;
6  lastTemp = TempsArray[[(Loop * 2 + 1)]];
7  ParametricPlot[positionFinale[t], {t, 0, lastTemp}]

```



Nous commençons aux deux premières lignes par appeler la fonction de simulation pour vérifier son bon fonctionnement. Nous utilisons le mode verbeux de la fonction et calculons la position de la particule précisément à la moitié du temps de la simulation. La fonction nous retourne des informations concernant le numéro d'itération auquel le temps correspond, les variables récupérées de la pré-simulation et le type de mouvement actuel. Elle retourne également des informations selon ce dernier; des caractéristiques, comme ici, le rayon, la vitesse angulaire et l'angle du MCU. La fonction nous renvoie, de plus, les informations concernant les points clés précédents et suivants l'instant choisi pour vérifier la cohérence des résultats.

Aux lignes cinq et suivantes, nous demandons à Mathematica de dessiner un graphique de notre fonction de simulation avec *ParametricPlot* qui trace la trajectoire en X et Y. Nous prenons le dernier temps disponible calculé par la simulation et faisons attention à désactiver le mode débogage. C'est ainsi que nous obtenons la trajectoire spiraloïde non-relativiste du cyclotron.

3.2.2 Programmation de la simulation relativiste

Dans cette partie, nous nous occuperons d'adapter les différentes parties du code vu précédemment aux lois relativistes liées aux mouvements que nous avons expliquées plus haut. Bien évidemment, nous n'allons pas aborder une seconde fois tout le code mais seulement les parties qui ont subies des modifications assez importantes. Notons que les caractères « [...] » veulent signifier une omission d'une partie inchangée de code à l'endroit indiqué.

```
(*Fonctions*) gamma[v_] := 1 / Sqrt[1 - v^2 / c^2];
xCond[t_, v_] := (c / Ac) * Sqrt[c^2 + (Ac t + v * gamma[v])^2] - c^2 / Ac * gamma[v];
vCond[t_, v_] := D[xCond[tf, vf], tf] /. {tf -> t, vf -> v}
[...]
For[i = 0, i < Loop, i++, {
  [...]
  VRel = Abs[VRel];
  tempsX = Sqrt[(d / c + c / Ac * gamma[VRel])^2 - (c / Ac)^2] - (VRel * gamma[VRel]) / Ac;
  (*--Y*) VRel = vCond[tempsX, VRel];
  If[Mod[i, 2] == 0, VRel = VRel, VRel = -VRel];
  [...]
  tempsY = Abs[(Pi * m * gamma[VRel]) / (q * B)];
```

Nous avons introduit dans l'en-tête du programme, des fonctions concernant les équations de trajectoire et de vitesse du MRR (notons que « Cond » signifie condensateur). Puis nous avons modifié, dans la pré-simulation, les calculs: des temps des mouvements balistiques, du rayon du MCU et de la vitesse. Nous avons ensuite modifié la ligne dans le simulateur final concernant le MRUA pour utiliser notre fonction comme suit:

```
positionFinaleRel[t_] := (
  [...]
  currX = xCond[deltaT, lastVit] + lastX;
```

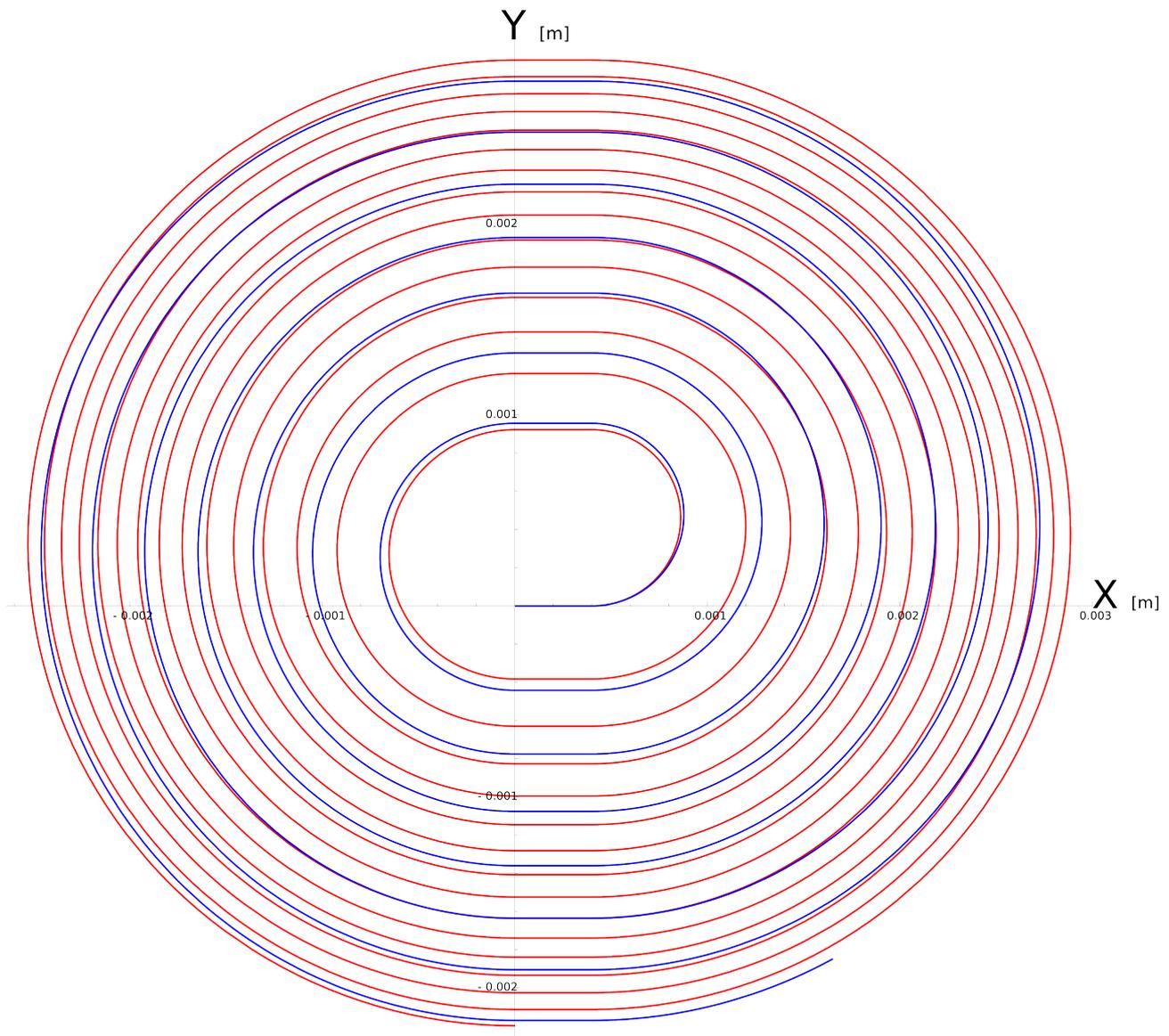
C'est ainsi que ces quelques adaptations nous permettent de tenir compte de la relativité dans notre simulateur.

3.3 Analyse et comparaison des résultats

Nous allons nous occuper dans cette section d'exploiter notre simulateur pour comparer l'impact de la relativité sur la trajectoire, la vitesse, le rayon et la période du MCU d'un électron. Nous comparerons également ces grandeurs dans le cas relativiste avec deux particules différentes, par exemple un électron et un proton. Pour terminer nous donnerons la taille de notre accélérateur, s'il avait existé et les grandeurs physiques obtenus pour un grand nombre de cycles pour avoir une idée plus proche d'un cas réel.

Voici tout d'abord les trajectoires finales que nous avons obtenues pour un électron une fois dans le cas classique et une fois dans le cas relativiste réunies sur un même graphique. Nous utiliserons, à l'avenir, sur nos graphiques, la couleur rouge pour le cas classique et bleue pour le cas relativiste. Les caractéristiques choisies pour cette simulation sont:

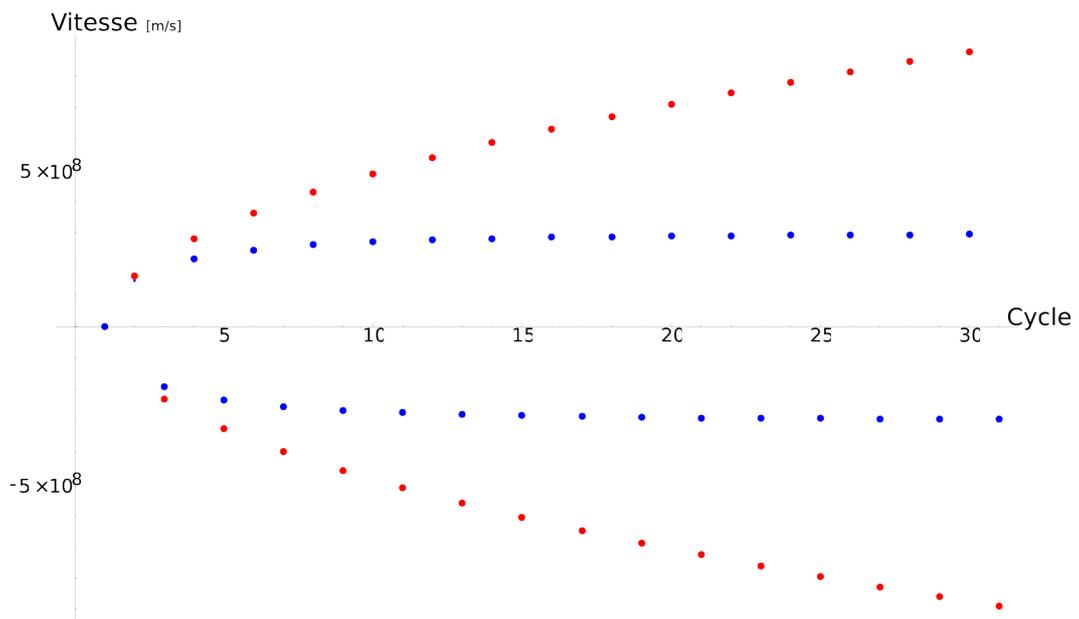
- Particules: électrons (masse: $9.11 \cdot 10^{-31}$ m)
- Tension du condensateur: 75'000 V
- Distance entre les plaques: $4 \cdot 10^{-4}$ m
- Champ magnétique: 2 T
- Cycles de simulation (1 MRR + 1 MCR): 30



Nous pouvons observer clairement dès le premier mouvement circulaire les effets de la relativité. En effet, nous remarquons que les rayons relativistes sont plus grands que les classiques et cela à cause du facteur gamma. Après quelques cycles, nous pouvons voir que l'électron classique doit faire plusieurs tours pour arriver à la même position en Y. Nous pouvons déjà anticiper en disant qu'il faudra un appareil plus gros pour le cas relativiste à cause du facteur gamma qui apparaît dans l'expression du MCR. Il faut noter que nous ne pouvons pas distinguer de différences dans les mouvements rectilignes parce que le changement que l'électron subit est purement temporel.

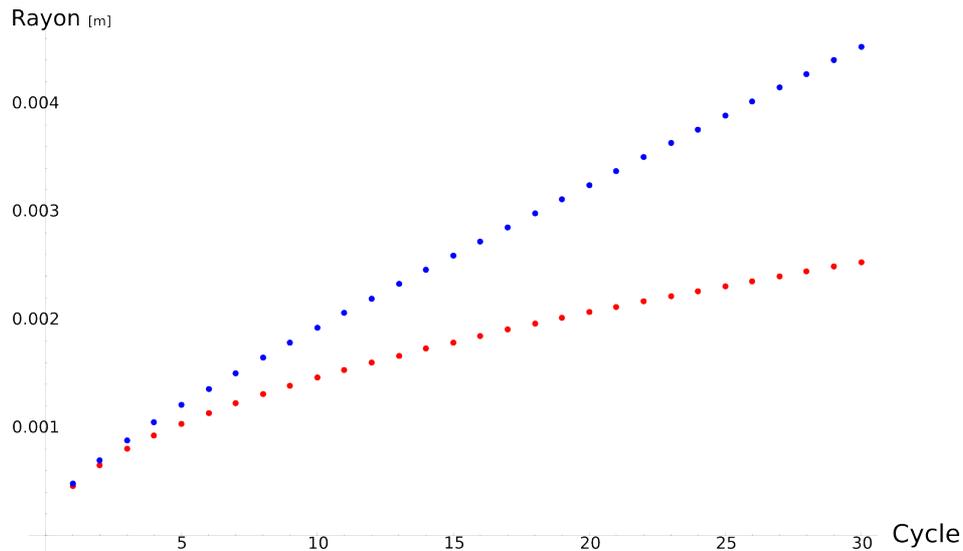
Nous allons maintenant commenter les vitesses obtenues dans les deux cas en fonction du nombre de cycles à l'aide d'un tableau de valeurs et d'un graphique. Le tableau permettra de se rendre compte des vitesses obtenues précisément et du rapport entre les vitesses relativistes et classiques. Le graphique nous permettra d'observer le type d'accroissement et les limites de vitesses.

Cycle	Vitesses [m/s]		
	Classiques	Relativistes	Rapport C/R
0	0,00E+000	0,00E+000	
1	1,62E+008	1,47E+008	1,11
2	-2,30E+008	-1,90E+008	1,21
3	2,81E+008	2,16E+008	1,30
4	-3,25E+008	-2,33E+008	1,40
5	3,63E+008	2,45E+008	1,48
6	-3,98E+008	-2,54E+008	1,57
7	4,30E+008	2,61E+008	1,65
8	-4,59E+008	-2,66E+008	1,73
9	4,87E+008	2,71E+008	1,80
10	-5,14E+008	-2,74E+008	1,87
11	5,39E+008	2,77E+008	1,94
12	-5,63E+008	-2,79E+008	2,01
13	5,86E+008	2,82E+008	2,08
14	-6,08E+008	-2,83E+008	2,15
15	6,29E+008	2,85E+008	2,21
16	-6,50E+008	-2,86E+008	2,27
17	6,70E+008	2,87E+008	2,33
18	-6,89E+008	-2,88E+008	2,39
19	7,08E+008	2,89E+008	2,45
20	-7,26E+008	-2,90E+008	2,51
21	7,44E+008	2,91E+008	2,56
22	-7,62E+008	-2,91E+008	2,62
23	7,79E+008	2,92E+008	2,67
24	-7,96E+008	-2,92E+008	2,72
25	8,12E+008	2,93E+008	2,77
26	-8,28E+008	-2,93E+008	2,82
27	8,44E+008	2,94E+008	2,87
28	-8,59E+008	-2,94E+008	2,92
29	8,75E+008	2,94E+008	2,97



Dans le tableau nous pouvons facilement observer les limites de vitesses de la simulation relativiste dès les premiers cycles qui finissent par d'infimes changements. On peut largement constater les phénomènes sur le graphique qui montrent l'asymptote horizontale de la vitesse relativiste qui s'est établie à c dès les dix premiers cycles. Évidemment, la vitesse classique continuera d'augmenter après les trente premiers cycles d'une façon importante alors que la vitesse relativiste ne variera que faiblement. Il est important de souligner que l'électron dans le cas classique dépasse la vitesse de la lumière dès le quatrième cycle. Cela montre bien que la relativité restreinte ne peut plus être négligeable dès les premiers cycles. Nous allons utiliser, à des fins de comparaisons, la formule de calcul d'énergie cinétique classique ($E_{cin} = \frac{1}{2} m v^2$) pour se rendre compte de l'impact de la relativité sur les énergies. Les énergies finales sont de $3.49 \cdot 10^{-13}$ J, soit 2.18 MeV, et $3.94 \cdot 10^{-14}$ J, soit 0.25 MeV, pour le cas classique et relativiste respectivement. Nous allons observer les rayons des mouvements circulaires puis comparer les deux cas avec un tableau et un graphique :

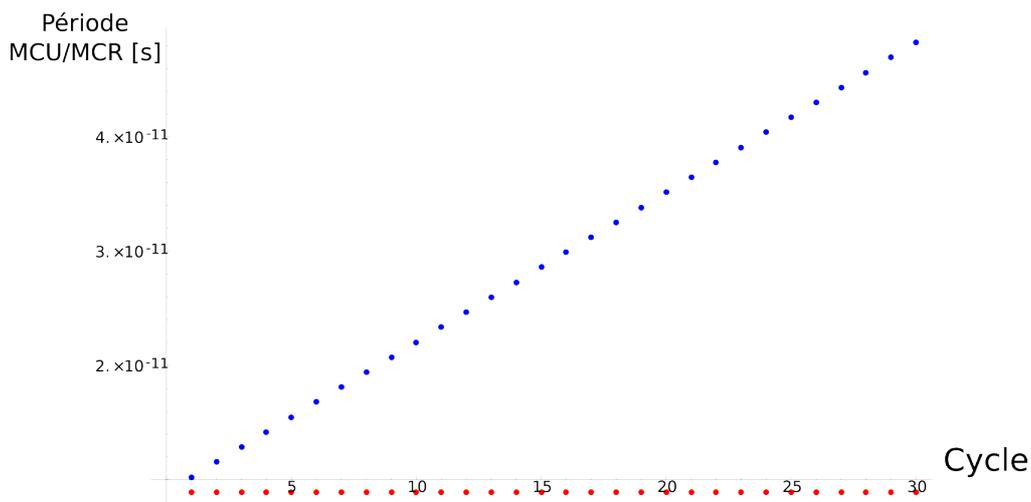
Cycle	Rayons [m]		
	Classiques	Relativistes	Rapport C/R
1	4,62E-004	4,78E-004	0,97
2	6,53E-004	6,99E-004	0,93
3	8,00E-004	8,83E-004	0,91
4	9,23E-004	1,05E-003	0,88
5	1,03E-003	1,21E-003	0,86
6	1,13E-003	1,36E-003	0,83
7	1,22E-003	1,50E-003	0,81
8	1,31E-003	1,65E-003	0,79
9	1,39E-003	1,79E-003	0,78
10	1,46E-003	1,92E-003	0,76
11	1,53E-003	2,06E-003	0,74
12	1,60E-003	2,19E-003	0,73
13	1,66E-003	2,33E-003	0,72
14	1,73E-003	2,46E-003	0,70
15	1,79E-003	2,59E-003	0,69
16	1,85E-003	2,72E-003	0,68
17	1,90E-003	2,85E-003	0,67
18	1,96E-003	2,98E-003	0,66
19	2,01E-003	3,11E-003	0,65
20	2,07E-003	3,24E-003	0,64
21	2,12E-003	3,37E-003	0,63
22	2,17E-003	3,50E-003	0,62
23	2,21E-003	3,63E-003	0,61
24	2,26E-003	3,76E-003	0,60
25	2,31E-003	3,89E-003	0,59
26	2,35E-003	4,02E-003	0,59
27	2,40E-003	4,14E-003	0,58
28	2,44E-003	4,27E-003	0,57
29	2,49E-003	4,40E-003	0,57
30	2,53E-003	4,53E-003	0,56



Encore une fois, nous pouvons remarquer la différence importante entre le cas classique et relativiste. Celle-ci vient du facteur gamma qui apparaît dans l'expression du rayon relativiste mais qui est absent dans la version classique. Comme nous l'avons vu il dépend de la vitesse et va croître en tendant vers l'infini. En effet, nous pouvons constater l'écart qui augmente rapidement au fil des cycles. Après trente cycles, nous remarquons que le rayon relativiste est quasiment deux fois plus grand que dans le cas classique et qu'elles tendent vers l'équation (où n est le nombre de cycle et ΔE l'énergie reçue par cycle):

$$R(n) \rightarrow n \frac{\Delta E}{qB} + \frac{mc}{qB} \quad \text{avec } \Delta E = qU$$

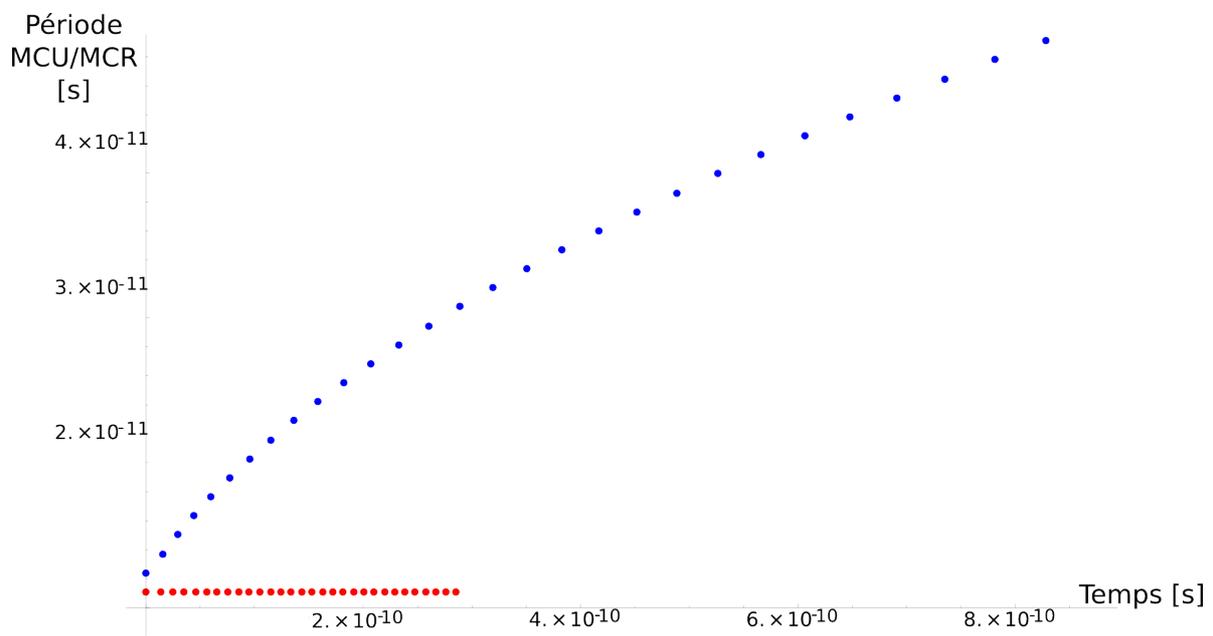
Avec ces informations, nous pouvons affirmer que dans la réalité, afin de procéder expérimentalement à cette accélération, il nous faudrait un cyclotron de rayon 2.53 mm dans le cas classique, alors que dans le relativiste, il faudrait 4.53 mm, soit presque le double comme nous l'avons dit. Bien entendu, les cyclotrons qui ont été construits (utilisant des protons) font plusieurs dizaines de centimètres à quelques mètres de diamètres alors que notre simulation (utilisant des électrons) tient sur 1 cm. Nous verrons plus loin que notre simulation avec un proton est plus intéressante que celle-ci parce que les énergies finales et les grandeurs obtenues ici sont trop faibles pour pouvoir s'imaginer précisément les résultats d'un cas réaliste.



Sur ce graphique nous pouvons observer la période du mouvement du MCU/MCR en fonction du nombre de cycles dans les deux cas. Évidemment, la période ne varie pas dans le cyclotron classique. Dans le cas relativiste nous remarquons que la période augmente linéairement avec le nombre de cycles parce que la vitesse n'augmente presque plus. Nous le constatons avec l'équation de la période relativiste :

$$T(n) = n \frac{\pi \Delta E}{q B c^2} + \frac{\pi m}{q B}$$

La pente étant constante, la période augmente selon une droite.



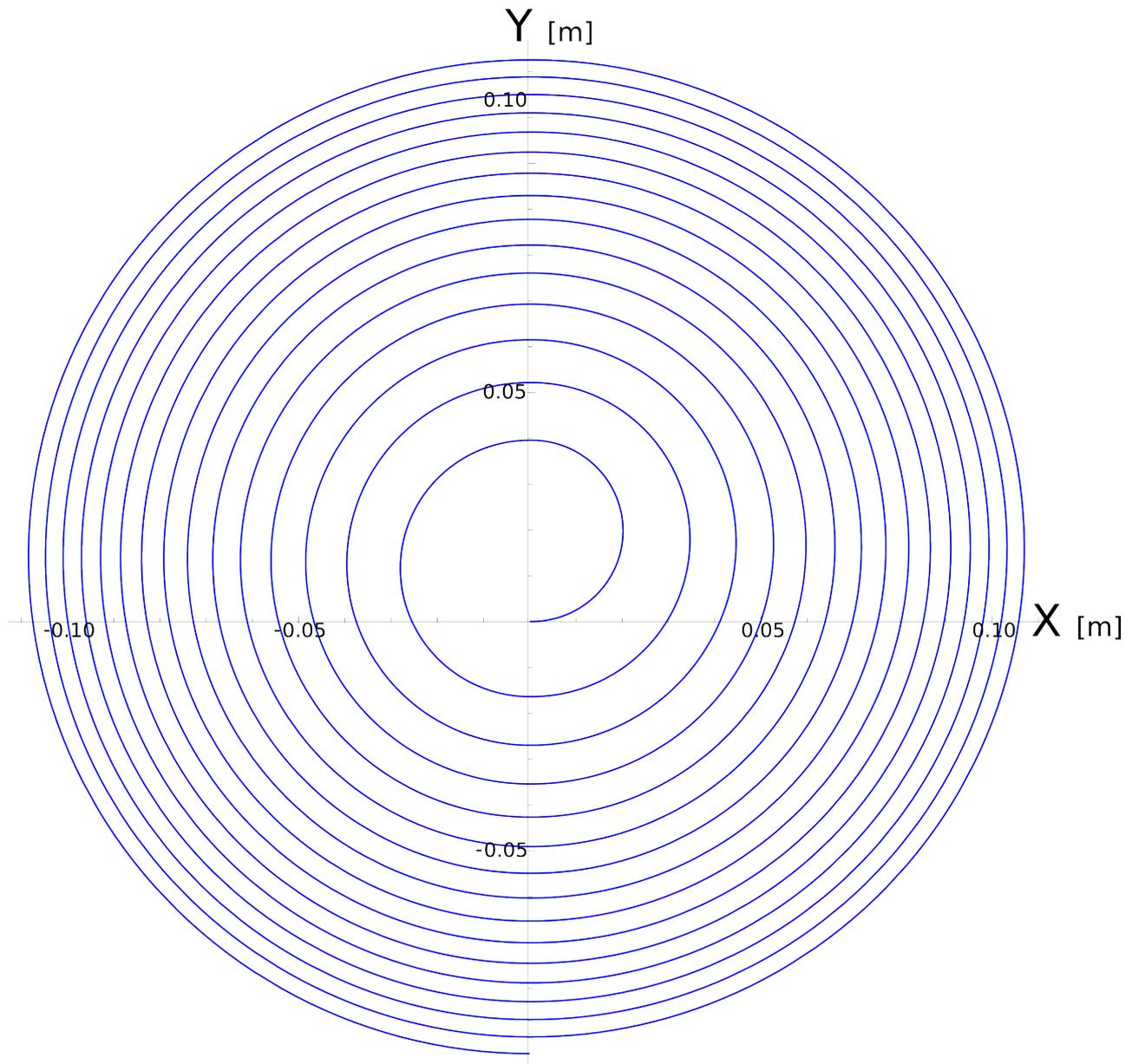
Néanmoins, si nous prenons le graphique ci-dessus, qui est la période MCU/MCR, cette fois-ci par rapport au temps, nous pouvons noter que non seulement le temps de la simulation est environ trois fois plus long que le classique pour un même nombre de cycles mais aussi que la période relativiste ne croît rapidement qu'au début. En effet, les dernières

valeurs nous indiquent que la période va croître plus faiblement vers la fin.

Voici, maintenant la comparaison des résultats entre un électron et un proton dans le cas relativiste exclusivement. Il nous est impossible d'utiliser un même graphique de trajectoires pour les deux particules, comme précédemment, à cause de l'énorme différence de grandeur qui les distinguent. En effet, la masse du proton est 1'759 fois plus grande que celle de l'électron, nous pouvons nous attendre à des résultats très différents et à ne distinguer qu'une trajectoire à la fois sur un même graphique. Nous garderons les caractéristiques de notre précédente simulation mais en ajoutant la masse de notre seconde particule, le proton:

- Particules: un électron (masse: $9.11 \cdot 10^{-31}$ kg) et un proton (masse: $1.67 \cdot 10^{-27}$ kg)

Comme dit plus haut, nous ne pouvons avoir les trajectoires de l'électron et du proton sur un même graphique. Cependant, nous avons pu faire un graphique de la trajectoire du proton seul afin d'observer sa forme et ses dimensions :

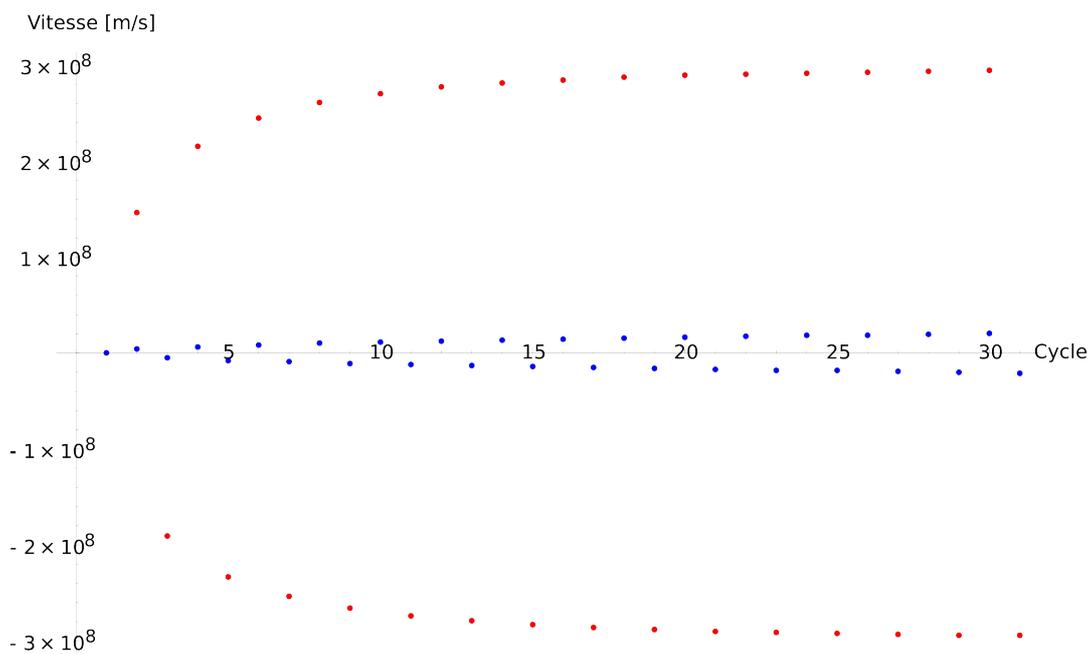


Comme nous pouvons le constater, le rayon de la trajectoire est beaucoup plus grand que celui de l'électron. Ce dernier était à vue d'œil de 0.003 m lors du dernier cycle, alors que le proton commence dès le premier cycle avec un rayon s'approchant des 0.04 m, soit un facteur de 13.3 environ. Nous pouvons noter que le mouvement MRR n'apparaît tout simplement pas dû au fait qu'il est bien trop petit comparé au MCR. En effet, les plaques étant séparées de 0.0004 m, il est quasiment impossible de voir cette grandeur sur ce graphique. Ceci nous donne l'impression d'une vraie spirale sans interruption, mais ce n'est pas le cas puisque celle-ci n'est en fait qu'une succession de demi-cercles de plus en plus grands. Nous pouvons porter notre attention sur les rayons des trajectoires, ce qui nous fait remarquer que celui du proton est environ 40 fois plus grand que celui de l'électron.

Nous allons maintenant comparer les vitesses atteintes par les deux particules en fonction

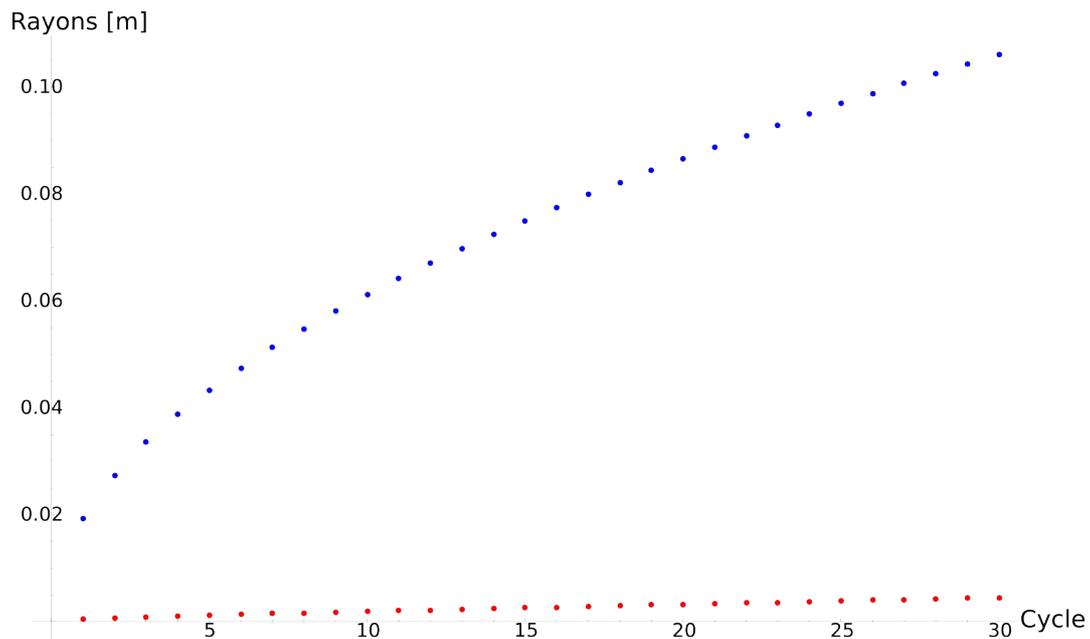
du cycle avec un tableau puis un graphique.

Cycle	Vitesses [m/s]		
	Électron	Proton	Rapport É/P
0	0,00E+000	0,00E+000	
1	1,47E+008	3,79E+006	38,72
2	-1,90E+008	-5,36E+006	35,48
3	2,16E+008	6,56E+006	32,87
4	-2,33E+008	-7,58E+006	30,71
5	2,45E+008	8,47E+006	28,90
6	-2,54E+008	-9,28E+006	27,35
7	2,61E+008	1,00E+007	26,01
8	-2,66E+008	-1,07E+007	24,84
9	2,71E+008	1,14E+007	23,80
10	-2,74E+008	-1,20E+007	22,88
11	2,77E+008	1,26E+007	22,05
12	-2,79E+008	-1,31E+007	21,30
13	2,82E+008	1,37E+007	20,61
14	-2,83E+008	-1,42E+007	19,99
15	2,85E+008	1,47E+007	19,42
16	-2,86E+008	-1,51E+007	18,89
17	2,87E+008	1,56E+007	18,40
18	-2,88E+008	-1,61E+007	17,94
19	2,89E+008	1,65E+007	17,52
20	-2,90E+008	-1,69E+007	17,13
21	2,91E+008	1,73E+007	16,75
22	-2,91E+008	-1,78E+007	16,41
23	2,92E+008	1,82E+007	16,08
24	-2,92E+008	-1,85E+007	15,77
25	2,93E+008	1,89E+007	15,47
26	-2,93E+008	-1,93E+007	15,20
27	2,94E+008	1,97E+007	14,93
28	-2,94E+008	-2,00E+007	14,68
29	2,94E+008	2,04E+007	14,44



Comme nous pouvons le constater avec le tableau ci-dessus, la vitesses du proton est beaucoup plus faible que celle de l'électron. En effet, à cause de sa masse importante et donc de son inertie, le proton atteindra une vitesse plus faible pour une même énergie cinétique que son homologue. Avec le graphique, ci-dessus, nous pouvons constater bien mieux que dans l'analyse précédente, le fait que les vitesse tendent vers une asymptote, ici valant c . Le proton voit sa vitesse stabilisée dès les trois premiers cycles. La vitesse de l'électron, tout comme celle du proton, tend vers c mais elle l'atteint beaucoup plus rapidement. Cette asymptote et l'inertie expliquent pourquoi les rapports sont grands au début de la simulation puis tendent vers un nombre fixe plus faible. En terme d'énergie cinétique relativiste (cette fois $E_{cin} = mc^2(\gamma - 1)$), nous obtenons comme énergies finales $3.94 \cdot 10^{-14}$ J, soit 0.25 MeV pour les deux particules. Les effets que nous observons s'explique par le fait que l'énergie cinétique croît lors d'un passage dans le condensateur et que le résultat final sur la vitesse n'est pas le même pour le proton que l'électron en raison de leur masse inerte différente. Voici maintenant la comparaison des rayons électron-proton :

Cycle	Rayons [m]		
	Électron	Proton	Rapport É/P
0	4,78E-004	1,98E-002	
1	6,99E-004	2,80E-002	2,50E-002
2	8,83E-004	3,43E-002	2,58E-002
3	1,05E-003	3,96E-002	2,65E-002
4	1,21E-003	4,42E-002	2,73E-002
5	1,36E-003	4,85E-002	2,80E-002
6	1,50E-003	5,24E-002	2,87E-002
7	1,65E-003	5,60E-002	2,94E-002
8	1,79E-003	5,94E-002	3,01E-002
9	1,92E-003	6,26E-002	3,07E-002
10	2,06E-003	6,56E-002	3,14E-002
11	2,19E-003	6,86E-002	3,20E-002
12	2,33E-003	7,14E-002	3,26E-002
13	2,46E-003	7,41E-002	3,32E-002
14	2,59E-003	7,67E-002	3,38E-002
15	2,72E-003	7,92E-002	3,44E-002
16	2,85E-003	8,16E-002	3,50E-002
17	2,98E-003	8,40E-002	3,55E-002
18	3,11E-003	8,63E-002	3,61E-002
19	3,24E-003	8,85E-002	3,66E-002
20	3,37E-003	9,07E-002	3,72E-002
21	3,50E-003	9,28E-002	3,77E-002
22	3,63E-003	9,49E-002	3,82E-002
23	3,76E-003	9,70E-002	3,88E-002
24	3,89E-003	9,90E-002	3,93E-002
25	4,02E-003	1,01E-001	3,98E-002
26	4,14E-003	1,03E-001	4,03E-002
27	4,27E-003	1,05E-001	4,08E-002
28	4,40E-003	1,07E-001	4,13E-002
29	4,53E-003	1,08E-001	4,17E-002



Les rayons obtenus, lors de cette simulation, nous montrent encore une fois le rôle essentiel de la masse dans les résultats. Comme pour les vitesses vues plus haut, l'inertie du proton va affecter de manière importante les mouvements qu'il fait. En effet, le rayon de son MCR est plus large que celui de l'électron. Sur le graphique, on remarque que celui de l'électron est plus modéré et semble dès le début avoir une pente d'accroissement constante. En effet en considérant :

$$R(n) \rightarrow n \frac{\Delta E c}{q B} + \frac{m c}{q B}$$

Nous pouvons l'observer, il y a une asymptote oblique quand le nombre de cycle est grand, mais nous pouvons déjà le voir après quinze cycles.

Nous savons que le rapport des rayons ne cessera pas d'augmenter et nous notons qu'en fin de simulation, ce rapport semble croître plus faiblement. Il faut ajouter qu'il nous faudrait maintenant, non plus un accélérateur d'un diamètre de 5 mm environ, utilisé pour l'électron, mais de 10.8 cm pour un proton, soit vingt fois plus.

Cycle	Périodes [s]		
	Électron	Proton	Rapport É/P
0	1,0242E-011	1,6400E-008	6,25E-004
1	1,1553E-011	1,6401E-008	7,04E-004
2	1,2863E-011	1,6403E-008	7,84E-004
3	1,4174E-011	1,6404E-008	8,64E-004
4	1,5485E-011	1,6405E-008	9,44E-004
5	1,6796E-011	1,6407E-008	1,02E-003
6	1,8107E-011	1,6408E-008	1,10E-003
7	1,9417E-011	1,6409E-008	1,18E-003
8	2,0728E-011	1,6410E-008	1,26E-003
9	2,2039E-011	1,6412E-008	1,34E-003
10	2,3350E-011	1,6413E-008	1,42E-003
11	2,4661E-011	1,6414E-008	1,50E-003
12	2,5972E-011	1,6416E-008	1,58E-003
13	2,7282E-011	1,6417E-008	1,66E-003
14	2,8593E-011	1,6418E-008	1,74E-003
15	2,9904E-011	1,6420E-008	1,82E-003
16	3,1215E-011	1,6421E-008	1,90E-003
17	3,2526E-011	1,6422E-008	1,98E-003
18	3,3836E-011	1,6424E-008	2,06E-003
19	3,5147E-011	1,6425E-008	2,14E-003
20	3,6458E-011	1,6426E-008	2,22E-003
21	3,7769E-011	1,6428E-008	2,30E-003
22	3,9080E-011	1,6429E-008	2,38E-003
23	4,0390E-011	1,6430E-008	2,46E-003
24	4,1701E-011	1,6431E-008	2,54E-003
25	4,3012E-011	1,6433E-008	2,62E-003
26	4,4323E-011	1,6434E-008	2,70E-003
27	4,5634E-011	1,6435E-008	2,78E-003
28	4,6945E-011	1,6437E-008	2,86E-003
29	4,8255E-011	1,6438E-008	2,94E-003

Sur ce tableau présentant les périodes MCR des deux particules en fonction du cycle, nous pouvons mentionner que celle du proton est bien plus élevée que celle de l'électron. Ces résultats nous laissent penser que la période est constante alors que la pente est faible. D'ailleurs, cette pente est la même pour l'électron et le proton car elle est indépendante de la masse, il n'y a que la position à l'ordonnée qui varie d'une particule à l'autre:

$$T(n) = n \underbrace{\frac{\pi \Delta E}{q B c^2}}_{\text{pente}} + \underbrace{\frac{\pi m}{q B}}_{\text{valeur initiale}}$$

Le rapport des périodes de l'électron sur celles du proton nous donne un facteur moyen sur trente cycles d'environ $1.78 \cdot 10^{-3}$. Il est difficile de percevoir les infimes variations à cause des pentes faibles, c'est surtout le cas pour le proton. De ce fait, nous avons constaté que la période du proton ne variait qu'au millième durant notre simulation tandis que celle de l'électron continuait d'augmenter de manière plus importante à cause des mêmes raisons évoquées précédemment avec les rayons.

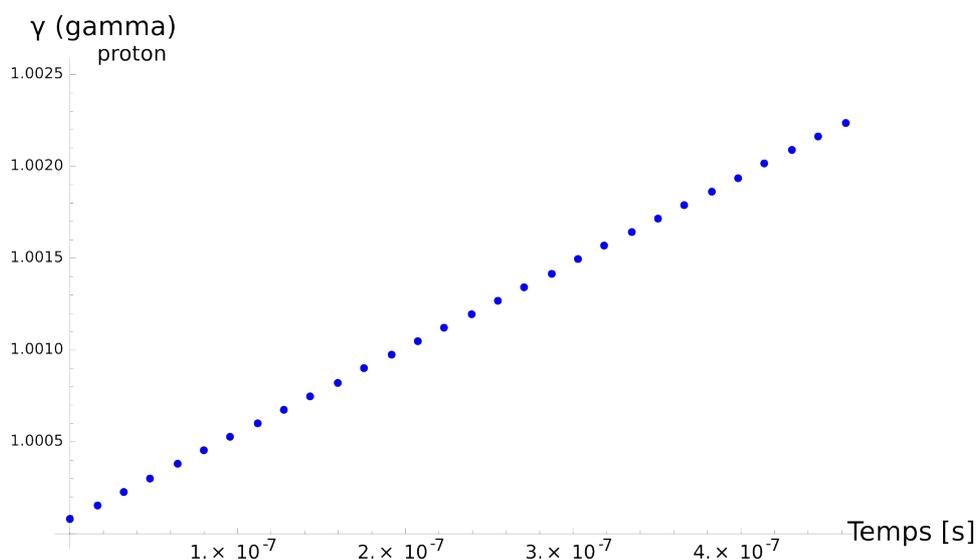
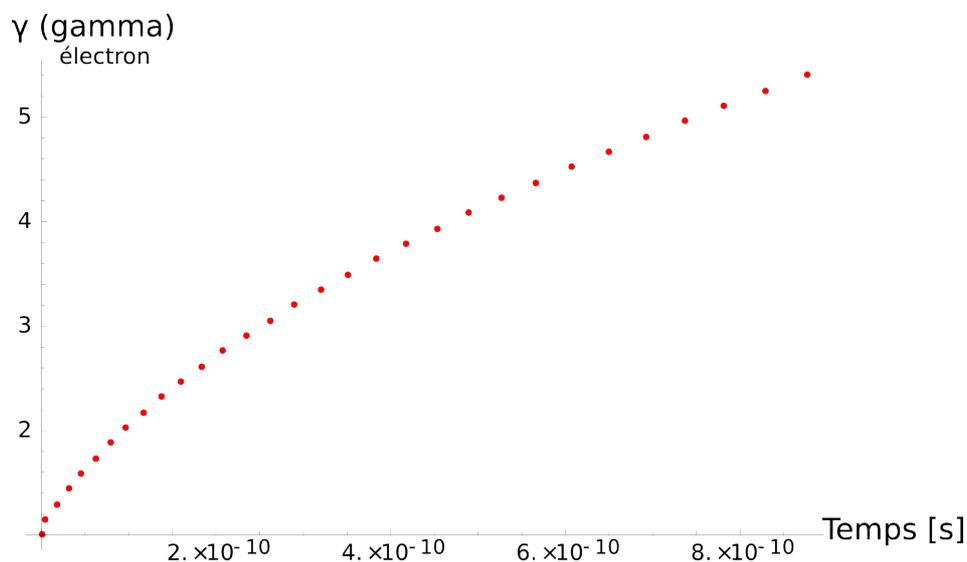
Concernant maintenant les temps obtenus pour chacune des particules pour effectuer les trente cycles établis, nous avons obtenus pour :

- l'électron: $9.26 \cdot 10^{-10}$ s
- le proton: $4.94 \cdot 10^{-7}$ s

En faisant le rapport du second sur le premier, nous obtenons un facteur de 510. Il faudra donc beaucoup plus de temps au proton qu'à l'électron pour atteindre une énergie donnée à la fin d'un nombre de cycles déterminé en raison de sa période plus grande causée par

sa grande masse (en négligeant le MRR, qui représente dans notre simulation 5.2% du temps total pour l'électron et 0.24% pour le proton). De plus, nous savons qu'il va demander un accélérateur d'un diamètre plus élevé que l'électron. Malgré cela, les chercheurs utilisent plus fréquemment, si ce n'est tout le temps, l'électron plutôt que le proton dans les cyclotrons. Les effets sur le proton, relevés dans notre analyse, sont dus à sa grande masse, mais ce sont sûrement des avantages pour ces chercheurs. En effet, une accélération et une vitesse plus faibles semblent être des atouts importants pour gérer au mieux une particule que si c'était le contraire. L'électron n'est pas adapté pour cet accélérateur à cause de sa grande vitesse qui fait intervenir la relativité dès les premiers cycles. Pourtant en terme d'énergie, les deux particules sont équivalentes à la fin de la simulation.

Une grandeur physique intéressante, liée à la relativité, n'a pas encore été comparée ici. Il s'agit du facteur gamma, γ . Voici les derniers graphiques représentant les facteurs gamma de l'électron et du proton en fonction du temps :

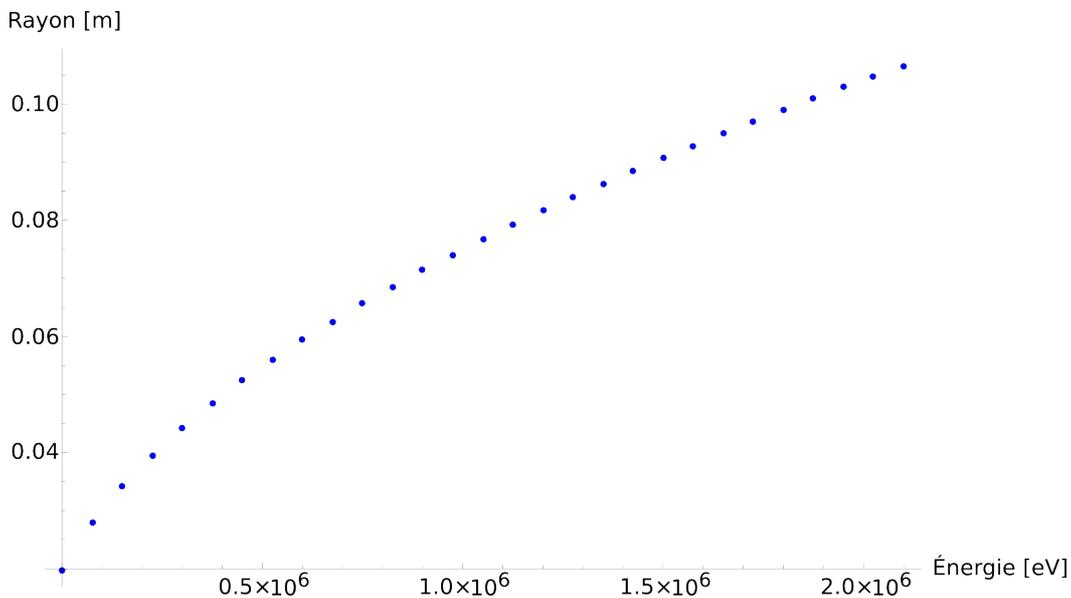


Comme nous pouvions nous y attendre, l'électron a un facteur γ qui s'accroît rapidement au début puis presque linéairement vers la fin. À cause de sa grande vitesse, son facteur est important contrairement au proton. Ce dernier, comme nous l'avons dit, a une vitesse relativement stable et il en résulte un facteur gamma augmentant linéairement. Comme nous pouvons le voir, il y a un facteur d'environ 5 entre les gammas du proton et de l'électron. En résumé, le facteur γ n'est pas négligeable lorsque l'accélérateur travaille avec une particule légère car il est important après une fraction de nanoseconde ($\gamma > 1.5$ après ~ 0.05 nanoseconde). Alors que le proton a un facteur relativement faible et négligeable, même à long terme, dans le cas considéré ici.

Nous retrouvons sur ce graphique le facteur gamma des deux particules en fonction du nombre de cycle cette fois-ci. D'une part nous constatons encore une fois l'écart important entre l'électron et le proton et, d'autre part, nous pouvons observer que les données forment deux droites décrites par l'équation:

$$\gamma(n) = n \underbrace{\frac{\Delta E}{m c^2}}_{\text{pente}} + \underbrace{1}_{\text{valeur initiale}}$$

Un dernier point intéressant, qui est à souligner dans cette analyse, c'est le lien qui unit l'énergie, dépendant de la vitesse et de la masse, au rayon de la trajectoire, soit le rayon de l'accélérateur qu'il faudrait créer. Nous avons choisi de ne présenter le résultat qu'avec le proton car c'est le cas le plus intéressant.



Nous constatons que, comme la plupart des graphiques concernant le proton, la courbe s'accroît rapidement puis tend à croître avec une pente linéaire. Il est tout à fait normal que plus la particule se déplace rapidement, plus l'espace requis à son accélération est important. Par conséquent nous pouvons prévoir, avec ce graphique, qu'elle serait la taille nécessaire pour atteindre une énergie déterminée, ici, en électron-volt. En effet, la courbe s'étant stabilisée, elle augmente linéairement avec l'énergie. Il serait donc facile de calculer la suite du graphique en approximant la courbe par l'asymptote :

$$R(E_{\text{cin}}) \rightarrow E_{\text{cin}} \frac{c}{qB} + \frac{mc}{qB}$$

Voici quelques énergies et rayons que nous obtenons précisément pour 200 cycles avec le proton :

Cycle	Proton		Cycle	Proton	
	E [eV]	R [eV]		E [eV]	R [eV]
1	7,500E+04	1,979E-02	101	7,575E+06	1,992E-01
6	4,500E+05	4,847E-02	106	7,950E+06	2,041E-01
11	8,250E+05	6,564E-02	111	8,325E+06	2,089E-01
16	1,200E+06	7,917E-02	116	8,700E+06	2,136E-01
21	1,575E+06	9,071E-02	121	9,075E+06	2,182E-01
26	1,950E+06	1,009E-01	126	9,450E+06	2,227E-01
31	2,325E+06	1,102E-01	131	9,825E+06	2,271E-01
36	2,700E+06	1,188E-01	136	1,020E+07	2,314E-01
41	3,075E+06	1,268E-01	141	1,058E+07	2,356E-01
46	3,450E+06	1,343E-01	146	1,095E+07	2,398E-01
51	3,825E+06	1,414E-01	151	1,133E+07	2,439E-01
56	4,200E+06	1,482E-01	156	1,170E+07	2,479E-01
61	4,575E+06	1,547E-01	161	1,208E+07	2,519E-01
66	4,950E+06	1,610E-01	166	1,245E+07	2,558E-01
71	5,325E+06	1,670E-01	171	1,283E+07	2,596E-01
76	5,700E+06	1,728E-01	176	1,320E+07	2,634E-01
81	6,075E+06	1,784E-01	181	1,358E+07	2,672E-01
86	6,450E+06	1,838E-01	186	1,395E+07	2,708E-01
91	6,825E+06	1,891E-01	191	1,433E+07	2,745E-01
96	7,200E+06	1,942E-01	196	1,470E+07	2,781E-01

Nous pouvons constater que les valeurs sont assez similaires avec ce que nous avons déjà vu et commentés précédemment. Il faut noter que nous avons pris un cycle différent pour l'énergie que pour le rayon. En effet, la vitesse, utilisée pour calculer l'énergie, est considérée au temps initial (avant que le proton ne passe dans le condensateur) tandis que le rayon vient du MCR qui se produit après l'accélération dans le MRR. Ceci vient de la conception du simulateur qui calcule et sauve la vitesse en début du cycle comme étant celle du cycle entier. Il faut donc décaler, comme nous l'avons fait, les énergies d'un cycle en avance pour obtenir les mesures au même moment que le MCR. Ceci fait, nous observons que les rayons et les énergies tendent à se stabiliser vers la fin de cette simulation. Au dernier cycle, nous obtenons une énergie de 14.7 MeV pour un rayon de 27.8 cm. Ces caractéristiques nous paraissent plus réalistes qu'avec seulement 30 cycles comme précédemment. Les énergies obtenus sont proches des résultats obtenus par les cyclotrons actuels, comme le *88-Inch Cyclotron* à Berkeley obtenant 55 MeV (avec un proton). Finalement, il faudrait quelques centaines de cycles supplémentaires pour obtenir une énergie finale de cette ordre.

4. Conclusion et sources

4.0 Conclusion

En abordant les différents types d'accélérateurs pour avoir une idée générale de leur utilisation à notre époque et expliqué les différents principes physiques nécessaires à la conception de notre simulateur en introduisant quelques concepts de la relativité restreinte dont nous avons besoin, nous constatons que nous avons beaucoup appris sur l'utilisation de ces lois et de leurs applications dans les accélérateurs de particules.

Par la suite, en continuant avec le chapitre principal de notre dossier sur Mathematica abordant la programmation et l'analyse de nos résultats, nous avons pu nous rendre compte de la puissance de ce programme, en apprécier sa facilité à créer des graphiques et nous avons pu découvrir sa souplesse à l'utilisation.

Nous avons appris à l'utiliser convenablement selon les cas et à atteindre nos objectifs, même si cela n'a pas toujours été si facile car il nous est arrivé d'être confronté à certains problèmes. Nous avons donc cherché et corrigé des fautes de syntaxes, des calculs mal écrits ou encore des résultats incohérents, mais nous avons toujours fini par trouver une solution à force de réflexions. C'est en comprenant et en corrigeant nos erreurs que nous avons pu continuer à avancer.

Nous avons terminé dans le dernier chapitre par l'analyse des différents cas. Nous aurions pu comparer le proton classique et relativiste mais nous avons estimé que la comparaison avec l'électron était suffisamment complète et pertinente. L'utilisation de notre simulateur s'est avéré simple et, une fois terminé, il nous a suffi de puiser dans les résultats pour en faire des graphiques, des tableaux et des analyses. Nous savons que certaines parties ne nécessitaient pas forcément un simulateur car nous avons tiré certaines expressions en fonction du cycle, comme le rayon, la période ou le facteur gamma. Cependant, les points forts du simulateur sont de pouvoir trouver facilement les grandeurs en fonction du temps et surtout de tracer avec précision un graphique des trajectoires.

Nous avons tout de même quelques regrets concernant cette dernière partie car nous aurions voulu utiliser nos résultats pour les comparer à un accélérateur célèbre, le *88-Inch Cyclotron*. Nous avons essayé de contacter le *Berkeley Accelerator Space Effects (BASE)* en Californie à la grande Université de Berkeley, lieu de naissance du synchrocyclotron et aussi du synchrotron, mais les responsables du projet n'ayant pas répondu, nous n'avons pas pu procéder à des comparaisons. Nous avons tenté aussi d'autres prises de contacts malheureusement elles n'ont pas eu plus de succès.

C'est dommage car il aurait été très intéressant de comparer les résultats de notre simulateur et de tester sa justesse par rapport à la réalité avec les paramètres utilisés par des chercheurs expérimentés. Nous aurions pu l'ajuster en intégrant au simulateur des concepts plus avancés. Il faut noter que, dans notre simulation, nous avons choisis le cas parfait qui est peu probable en situation réelle. En effet, nous considérons qu'aux bornes du condensateur c'est la tension crête qui s'applique lorsque la particule est accélérée. En réalité, puisque le générateur est alternatif, il y a de fortes chances pour que la tension ne soit pas à son maximum à ce moment là. Par conséquent la particule est accélérée différemment en fonction du cycle et rarement à son maximum. De plus, nous avons

considéré le champ magnétique comme étant parfait donc homogène. Mais ce n'est pas tout, nous avons placé notre particule à la sortie d'une plaque et avec une vitesse initiale connue, soit dans des conditions idéales. Dans la réalité, nous n'aurions pas de telles chances car l'injecteur envoie des particules n'importe où, n'importe quand et à n'importe quelle vitesse. Il n'y aurait qu'une faibles quantités de particules qui sortiraient de l'accélérateur. Si nous avions pu intégrer tous ces cas, notre simulateur aurait été bien plus réaliste. Nous aurions également pu, par la suite, essayer d'aborder d'autres idées plus générale, comme les impacts qu'auraient fait les particules contre des matériaux ou contre d'autres particules, ce qui aurait fait intervenir des lois assez complexes concernant les collisions.

Nos regrets sont toutefois atténués par le fait qu'un tel simulateur aurait sûrement pris des mois, voire des années de travail pour atteindre ce niveau de réalisme. Au final, cette première approche nous aura permis, premièrement, de mieux comprendre le fonctionnement théorique des accélérateurs et, deuxièmement, de comprendre en partie celui des collisionneurs tel le LHC mais aussi d'avoir une idée beaucoup plus nette de la complexité de ceux-ci dans des conditions réelles.

4.1 Références

Couverture du dossier :

<http://deptorg.knox.edu/physicsdept/Projects/Projects.htm> (5 oct. 2009)

Livres / Documents :

- «Physique Générale» par Douglas C. Giancoli, édition Boeck, Volume 3 pages 192 à 221.
- «Physique» par Eugene Hecht, Tamer Becherrawy, Joël Martin, chapitre 28 pages 1054 à 1080.
- Le « Help » fourni avec Mathematica

Liens / Sites :

<http://www.techno-science.net/?onglet=glossaire&definition=8097>

(24 sept. 2009)

http://fr.wikipedia.org/wiki/Accélérateur_de_particules

(21 sept. 2009)

<http://users.lal.in2p3.fr/mouton/miroir/accelerateur/cours/Les-cyclotrons-version4.pdf>

(13 oct. 2009)

<http://cyclotron.lbl.gov/>

(19 oct. 2009) – Concernant le cyclotron

<http://www.astrosurf.com/luxorion/relativite-restreinte.htm>

(16 sept. 2009) – Concernant la relativité restreinte

<http://thecyclotronkids.org/>

(19 oct. 2009) – Pour les caractéristiques et informations concernant le cyclotron

<http://books.google.fr/books>

(21 sept. 2009) – Pour la consultation de livres sur la physique.

http://webphysics.davidson.edu/physlet_resources/bu_semester2/c13_cyclotron.html

(30 août 2009) – Une simulation en Java d'un cyclotron

Remarques :

Tous les schémas et graphiques figurant dans ce dossier ont été conçus avec Inkscape et Mathematica pour celui-ci exclusivement.

Logiciels pour la conception :

- Calculs, Mathematica 7, pour Linux
- Suite bureautique, OpenOffice
- Dessin vectoriel, Inkscape
- Système d'exploitation, Gentoo Linux