

Національний університет «Києво-Могилянська академія»

Факультет природничих наук

Кафедра фізико-математичних наук

ЧИСЛОВЕ ДОСЛІДЖЕННЯ ДИНАМІКИ ЗАРЯДЖЕНИХ ЧАСТИНОК ПРИ  
ПРОХОДЖЕННІ ЦИКЛОТРОННОГО РЕЗОНАНСУ

Виконав студент 4 року навчання

спеціальності 104 Фізика та астрономія

Сечкар Григорій Сергійович

Керівник

доктор фізико-математичних наук,

доцент Яковенко Юрій Володимирович,

Інститут ядерних досліджень НАН України,

відділ теорії ядерного синтезу

Рецензент

Канд.фізико-матем.наук Р.М.Мельник

Кваліфікаційна робота захищена

з оцінкою 70 «задовільно»

Секретар ЕК Г.В.Оводенко

« 05 » липня 2022 р.

Київ – 2022

## ЗМІСТ

Вступ.....	3
Огляд літератури.....	6
Теоретична частина.....	
3.1.Циклотронний резонанс.....	9
3.2.Метод стаціонарної фази.....	12
Постановка задачі.....	
4.1.Нормалізація рівняння руху частинки в електромагнітному полі.....	13
4.2 Виведення формули для дослідження зміни енергії частинки під час проходження точки локального резонансу.....	15
Практична частина.....	
5.1. Двовимірний простір.....	17
5.2. Тривимірний простір.....	29
Висновок.....	35
Список використаних джерел.....	36
Додаток А.....	39
Додаток Б.....	45
Додаток В.....	49
Додаток Г.....	53

## ВСТУП

Терміни «циклотронна частота» та «циклотронний резонанс» походять від назви прискорювача заряджених частинок – циклотрона, в якому під дією електромагнітного поля заряджені частинки розганяються й набувають великих значень швидкості й енергії. Головний принцип дії циклотрона: заряджена частинка рухається по круговій траєкторії під дією магнітного поля і набуває прискорення внаслідок дії електричного поля. В свою чергу, циклотронний резонанс відбувається, коли провідник, поміщений в постійне магнітне поле, відбиває чи поглинає електромагнітні хвилі на частотах, які рівні або кратні циклотронній частоті носіїв заряду. Оскільки плазма є середовищем, що містить багато вільних заряджених частинок, явище циклотронного резонансу відіграє важливу роль у плазмі – космічній та лабораторній.

Перш за все, варто відзначити роль цього явища в збудженні хвиль у плазмі. Популяція частинок, енергія яких перевищує теплову, може стати джерелом вільної енергії для збудження хвилі, якщо вона увійде з цією хвилею в резонанс – зокрема, циклотронний. При цьому, циклотронна частота в більшості випадків лежить у радіодіапазоні.

Циклотронний резонанс також відповідає за збудження кількох типів хвиль у космічній плазмі. Наприклад, саме цей резонанс може збуджувати так звані вістлери – ймовірно, перші відомі хвилі в плазмі [1]. Інший приклад: плазма в керованому термоядерному синтезі ( КТС ), що відбувається за дуже високих температур[2]. Це дозволяє використовувати надлишкову енергію, що вивільняється в такому процесі, для інших цілей. Гаряча плазма в таких пристроях завжди містить популяції надтеплових йонів, які здатні збуджувати хвилі.

З іншого боку, циклотронний резонанс може відповідати за передачу енергії від хвилі до заряджених частинок і, відповідно, їхнього прискорення (саме це відбувається в циклотроні). Це становить інтерес як механізм згасання хвиль, так і метод нагрівання плазми. Електронно-циклотронне та йонно-

циклотронне нагрівання широко застосовуються в термоядерних пристроях (токамаках та стелараторах). Також йонно-циклотронний резонанс активно використовується для розробки нового покоління плазмових двигунів для космічних апаратів[3].

Проте важливою особливістю є те, що магнітні поля, як в космосі, так і в термоядерних пристроях, є просторово неоднорідними (на відміну від циклотрона). Це ускладнює процес резонансу: частинка, що рухається в неоднорідному магнітному полі, зазнає циклотронного резонансу впродовж короткого проміжку часу. У зв'язку з цим цікавою є динаміка заряджених частинок при проходженні через йонно-циклотронний резонанс з електромагнітними хвилями в однорідному та неоднорідному магнітному полях. Саме ці процеси досліджуються в даній роботі.

Мета роботи - вивчити динаміку руху зарядженої частинки під час її проходження через циклотронний резонанс. Для цього необхідно виконати наступні задачі:

- записати закон руху частинки в електромагнітному полі, а також всі сили, що діють на неї;
- провести нормалізацію даних рівнянь для зручності їх подальшого використання під час написання програми на мові Python;
- за допомогою Python змодельовати рух частинки у двовимірному просторі ( на площині) в однорідному магнітному полі; дослідити швидкість, траєкторію, стан кінетичної енергії частинки за відсутності електростатичної хвилі;
- вивчити зміну траєкторії частинки після появи електростатичної хвилі та в результаті підбору різних значень циклічної частоти  $\omega$  й початкових швидкостей  $V_{x0}$  та  $V_{y0}$ ; підібрати необхідні значення даних величин для того, щоб отримати явище циклотронного резонансу; переконатись, що траєкторією руху частинки є коло, що відповідає теорії;

- дослідити характер залежності зміни енергії частинки та модуля її швидкості від зміни амплітуди електростатичної хвилі;

- змодельовати рух частинки в тривимірному просторі за однорідного та неоднорідного магнітного поля, а також за наявності чи відсутності електростатичної хвилі;

- переконатись в наявності різкої зміни кінетичної енергії частинки під час проходження точки локального циклотронного резонансу; вивчити узгодженість між величинами «стрибків» енергії, отриманих емпіричним шляхом та за допомогою аналітичної формули.

Об'єктом дослідження є заряджена частинка, що рухається в електромагнітному полі. Предметом дослідження є параметри динаміки руху зарядженої частинки. Інструментом для проведення дослідження є мова програмування Python.

## ОГЛЯД ЛІТЕРАТУРИ

Явище, яке зараз називають циклотронним резонансом, було вперше запропоновано як метод прискорення заряджених частинок у статті [4], в якій було описано принцип роботи циклотрона ( власне, циклотронним цей резонанс стали називати пізніше). Циклотронний рух частинок у магнітному полі описано в багатьох підручниках, наприклад [5].

Як було згадано вище, циклотронний резонанс відіграє значну роль у космічній, лабораторній та технічній плазмах. Він може вести як і до передачі енергії від частинок до хвиль, тобто резонансного збудження хвиль та їх нестійкості, так і до поглинання енергії хвиль і, відповідно, нагрівання плазми.

Електронно-циклотронний резонанс розглядався як один з методів нагрівання плазми в її джерелах для обробки матеріалів та для плазмових двигунів (які є наразі основними двигунами, що застосовуються при корекції орбіт супутників) [6].

Також електронно-циклотронний та йонно-циклотронні резонанси відіграють роль у нагріванні плазми магнітосфери та йоносфери [7,8]. А в роботах [9,10,11] йонно-циклотронний резонанс розглядається як один з механізмів нагрівання плазми сонцевого вітру та корони через хвилі та турбулентність.

Явище циклотронного резонансу є важливим і в плазмі термоядерних пристроїв, адже воно відповідає за передачу енергії хвиль зарядженим частинкам при нагріванні такої плазми хвилями в циклотронному діапазоні частот. Результати експериментів з електронно-циклотронного резонансного нагрівання наведені у роботі [12], а з йонно-циклотронного – у [13]. Ці два види нагрівання плазми в термоядерних пристроях є одними з найефективніших і широко застосовуються.

У термоядерній плазмі нестійкості в циклотронному діапазоні частот збуджуються популяціями швидких ( надтеплових ) йонів, що віддають енергію хвилям, з якими вони резонують. Огляд робіт про швидкі йони у токамаках та

стелараторах, а також нестійкості, що збуджуються цими йонами, можна знайти в роботах [14,15,16].

З фізичної точки зору, найлегше розгойдати хвилю, якщо вона є захопленою в певному резонаторі, який не допускає або принаймні уповільнює випромінювання енергії хвилі назовні. У фізиці термоядерної плазми хвилі, які є стоячими в радіальному напрямку, звать власними модами (модами дискретного спектру; аналог – дискретний спектр у квантовій механіці). Важливими різновидами власних мод, що можуть збуджуватись швидкими йонами, є альфвенівські власні моди (дискретні моди альфвенівських хвиль)[17] та швидкі магнітно-звукові моди [18]. Найважливішими резонансами, через які ці моди можуть збуджуватись швидкими йонами, є черенковський резонанс (модифікований через складну геометрію системи) та циклотронний.

Природно, що циклотронний резонанс відіграє головну роль при збудженні нестійкостей, що спостерігаються в йонно-циклотронному діапазоні частот. Інтерес до таких нестійкостей з'явився після спостереження так званого йонно-циклотронного випромінювання (англ. ICE – ion cyclotron emission) – випромінювання на частотах, близьких до гармонік йонних частот [19]. Пізніше в цьому частотному діапазоні було спостережено багато нестійкостей з різноманітними властивостями.

В експериментах на сферичному торі (сферичному токамаку) NSTX спостерігалося парадоксальне явище: при нагрівання плазми інжекцією нейтрального пучка температура електронів не збільшувалася, а навіть дещо зменшувалася при значному підсиленні потужності нагрівання [20]. Це значно погіршувало утримання енергії в плазмі, що корелювало з появою нестійкостей в йонно-циклотронному діапазоні частот. Ці нестійкості було ідентифіковано як альфвенівські та швидкі магнітно-звукові моди. [21]. Як можливі пояснення було висунуто просторове каналювання енергії пучка власними модами (тобто моди забирають енергію швидких йонів усередині плазми і віддають її на

периферії) [22] або підсилення електронного перенесення цими модами [23]. Це викликало інтерес до власних мод у йонно-циклотронному діапазоні частот.

При русі частинок у неоднорідному магнітному полі циклотронний резонанс спостерігається лише локально, на певних частинах траєкторій. Це веде до стрибкоподібної зміни енергії частинки. Такий характер взаємодії є важливим для характеру взаємодії частинка-хвиля, як лінійної [18], так і нелінійної [24]. Зручним математичним методом для дослідження поведінки частинки при її проходженні через локальний резонанс є метод стаціонарної фази [25, 26], який буде застосовуватись в даній роботі.

## ТЕОРЕТИЧНА ЧАСТИНА

### 3.1. Циклотронний резонанс

Циклотронний резонанс відбувається при взаємодії заряджених частинок, які рухаються по круговій траєкторії в магнітному полі, та зовнішніх сил, що діють на частинки. Його назва походить від циклотрону - прискорювача заряджених частинок - який використовує осцилююче електричне поле з відповідною фазою для прискорення частинок та надання їм кінетичної енергії. Якщо частота хвилі або іншого збурення електромагнітного поля збігається з так званою «циклотронною частотою» - частотою кругового руху частинки в магнітному полі – частинка відбирає енергію в хвилі або ж, навпаки, віддає її. Оскільки циклотронна частота відіграє ключову роль в нашому досліді, ми наводимо виведення виразу для її числового значення [5].

Відтак, векторне рівняння руху зарядженої частинки в постійному однорідному магнітному полі з напруженістю  $\vec{H} = (0;0;H)$  виглядає наступним чином:

$$m\ddot{\vec{r}} = \frac{q}{c}[\vec{r} \vec{H}].$$

Використовуючи паралельну до  $\vec{H}$  декартову систему координат з віссю  $OZ$ , силу Лоренца та рівняння руху можна представити так:

$$\vec{F} = \frac{q}{c} \begin{vmatrix} \vec{n}_x & \vec{n}_y & \vec{n}_z \\ \dot{x} & \dot{y} & \dot{z} \\ 0 & 0 & H \end{vmatrix} = \frac{qH}{c} (\dot{y}\vec{n}_x - \dot{x}\vec{n}_y), \quad \ddot{x} = \omega\dot{y}, \quad \ddot{y} = -\omega\dot{x}, \quad \ddot{z} = 0.$$

Тут  $\omega = \frac{qH}{mc}$  - циклотронна частота. Це частота (в Гц), яку має заряджена частинка з масою  $m$  та зарядом  $q$  при обертанні в магнітному полі з напруженістю  $\vec{H}$ .

Після інтегрування та врахування початкових умов, рівняння руху мають наступний вигляд:

$$\dot{x} = \omega(y - y_0) + \dot{x}_0, \quad \dot{y} = -\omega(x - x_0) + \dot{y}_0, \quad \dot{z} = \dot{z}_0.$$

Підставивши функцію  $y(x)$  в рівняння  $\ddot{x} = \omega y$ , отримаємо рівняння гармонічних коливань з правою частиною, яка є сталим числом:

$$\ddot{x} + \omega^2(x - x_0) = \omega \dot{y}_0.$$

Його загальним розв'язком є наступне рівняння :

$$x = x_0 + \frac{\dot{y}_0}{\omega} - a \cos(\omega t + \alpha).$$

Після його диференціювання отримуємо формулу для проекції швидкості частинки на вісь  $OX$ :

$$\dot{x} = a \omega \sin(\omega t + \alpha).$$

Використовуючи функції  $x(t)$  та  $\dot{x}(t)$ , визначення сталих  $a = \frac{1}{\omega} \sqrt{(\dot{x}_0^2 + \dot{y}_0^2)}$ ,  $\operatorname{tg} \alpha = \frac{\dot{x}_0}{\dot{y}_0}$  ( бо  $\sin \alpha = \frac{\dot{x}_0}{\sqrt{(\dot{x}_0^2 + \dot{y}_0^2)}}$ ,  $\cos \alpha = \frac{\dot{y}_0}{\sqrt{(\dot{x}_0^2 + \dot{y}_0^2)}}$  ) та після їх підстановки у функцію  $\dot{x}(y)$ , отримуємо

$y(t)$ :

$$y = y_0 - \frac{x_0}{\omega} + a \sin(\omega t + \alpha),$$
 звідки можна дістати формулу для проекції швидкості

частинки на вісь  $OZ$  -

$$\dot{y} = \alpha \omega \cos(\omega t + \alpha).$$

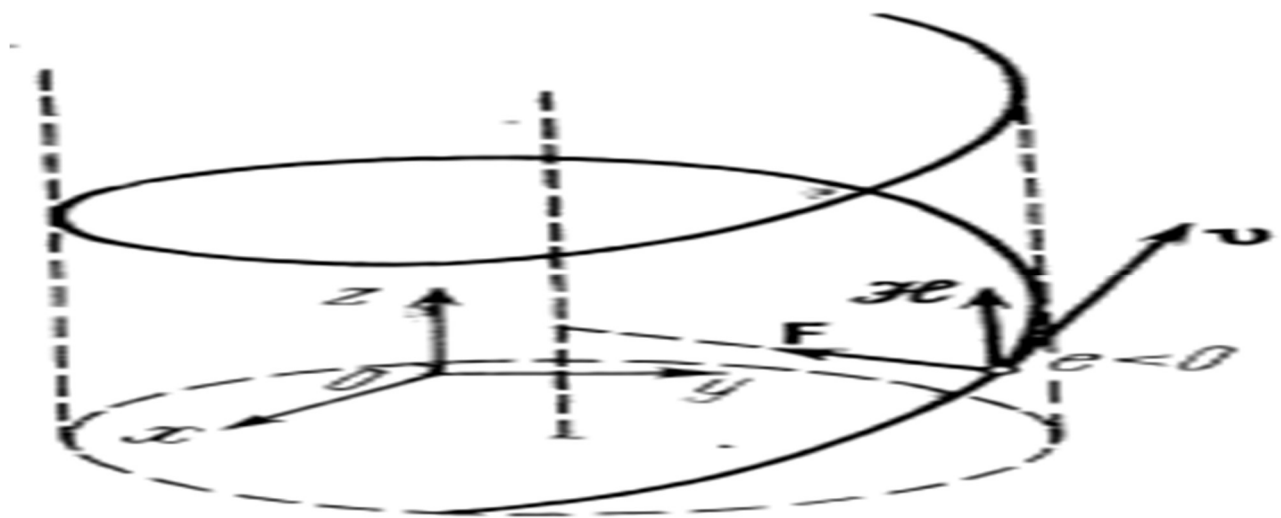


Рис. 1.22

Рис.3.1.1[5]. Траєкторія руху частинки в тривимірному просторі.

При цьому проекція матеріальної точки, що рухається по колу, на площину, перпендикулярну  $\vec{H}$ , має такий вигляд:

$$(x - x_0)^2 + (y - y_0)^2 = a^2, \text{ де } x_0 = x_0 + \frac{\dot{y}_0}{\omega}, y_0 = y_0 - \frac{\dot{x}_0}{\omega}.$$

Так як радіус кола визначається за допомогою проекції початкової швидкості на площину, яка перпендикулярна вектору напруженості  $\vec{H}$ , власне, самою напруженістю  $H$ , а також питомим зарядом  $\frac{q}{m}$ , то центр даного кола буде визначатись через взятими в початковий момент часу проекціями радіус-вектора й швидкості заряду на площину  $OXY$  – тобто, через  $x_0, y_0, \dot{x}_0, \dot{y}_0$  і циклотронну частоту  $\omega$ . Оскільки частинка в напрямку осі  $OZ$  рухається за інерцією зі швидкістю  $z_0$ , то траєкторія точки – це гвинтова лінія з постійним кроком  $z_0 \left( \frac{2\pi}{\omega} \right)$ . При цьому модуль швидкості точки та величина проекції швидкості на площину  $OXY$  є сталими величинами.

Щоб отримати проекції прискорення,  $\dot{x}(t)$  та  $\dot{y}(t)$  диференціюються за часом:

$$\ddot{x} = a\omega^2 \cos(\omega t + \alpha), \quad \ddot{y} = -\omega^2(y - y_0).$$

Виразивши ці рівняння через  $x$  та  $y$  ( $\ddot{x} = -\omega^2(x - x_0)$ ,  $\ddot{y} = -\omega^2(y - y_0)$ ), а також врахувавши, що проекція прискорення на вісь  $OZ$  рівна нулю, отримуємо формулу для абсолютного значення прискорення заряду:

$$\sigma = a\omega^2 = \sqrt{\omega(\dot{x}_0^2 + \dot{y}_0^2)}.$$

В даному випадку вектор прискорення направлений перпендикулярно до осі уявної циліндричної поверхні радіуса  $a$ , на яку «намотується» заряд.

Цікавою є властивість фокусування частинок магнітним полем, що розглядається. Вона полягає в тому, що частинки з однаковим питомим зарядом й однаковим початковим положенням, проте з різними початковими швидкостями, які перпендикулярні  $\vec{H}$ , досягають одного і того ж положення через період часу  $T = \frac{2\pi}{\omega}$ .

Отже, частинки в магнітному полі зазнають коливань з циклотронною частотою. Відповідно, вони будуть знаходитись в резонансі, якщо під впливом електромагнітних коливань їхня частота буде збігатись з циклотронною частотою. Ми покажемо, що у випадку неоднорідного магнітного поля параметри частинки( енергія, швидкість) будуть зазнавати змін. Оскільки отримати точний аналітичний розв'язок рівняння, що вираховує ці зміни в стані частинки складно, слід користуватись методом стаціонарної фази.

### 3.2.Метод стаціонарної фази

Метод стаціонарної фази використовується для наближеного обчислення інтегралів від швидкоосцилюючих функцій. Даний метод є одним із видів асимптотичного розкладу. Він полягає в тому, що розв'язок певного інтегралу отримується через виведення асимптот ряду й застосування члена старшого порядку як наближення, а члена наступного порядку як оцінки похибки. Цей метод застосовується як для одномірних, так і для багатомірних інтегралів Фур'є. Відповідно до назви методу, основний внесок в інтеграл надходить від точок, поблизу яких фазова функція повільно змінюється.

Скориставшись розв'язками, що викладені в четвертому розділі посібника Копсона[26], ми отримали наступне рівняння[27], яке необхідно було розв'язати для того, щоб провести порівняння змін енергій, отриманих аналітичним та числовим методами:

$$\int_{\mathbb{R}} g(x) \exp(ikf(x)) dx = g(x_0) \exp(ikf(x_0) + \operatorname{sgn}(f'(x_0))i\pi/4) \left( \frac{2\pi}{k|f''(x_0)|} \right)^{1/2} + o(k^{-1/2}) \quad (3.2.1.).$$

## ПОСТАНОВКА ЗАДАЧІ

### 4.1. Нормалізація рівняння руху частинки в електромагнітному полі.

Необхідними складовими частинами для моделювання руху зарядженої частинки в електромагнітному полі є:

1) рівняння магнітного поля

$$\begin{aligned} B_x &= -B_0 \alpha x \\ B_y &= 0 \\ B_z &= B_0(1 + \alpha z) \end{aligned} \quad \nabla * \vec{B} = \frac{\partial B_x}{\partial x} + \frac{\partial B_z}{\partial z} = 0,$$

де  $B_x, B_y, B_z$  - складові магнітного поля на осі OX, OY, OZ;  $\alpha$  - характерний параметр неоднорідності поля;

2) записане у системі СГС рівняння сили Лоренца, що діє на рухому частинку масою  $m$  та зарядом  $q$  -

$$\begin{cases} m \dot{\vec{v}} = q \vec{E} + [\vec{v} \times \vec{B}] \\ \dot{\vec{r}} = \vec{v} \end{cases};$$

3) рівняння електростатичної хвилі

$$\begin{aligned} \Phi &= \Phi_0 \cos(kx - \omega t) \\ \vec{E} = -\nabla \Phi &\Rightarrow \begin{cases} E_x = k \Phi_0 \sin(kx - \omega t) \\ E_y = E_z = 0 \end{cases} \end{aligned}$$

Електростатична хвиля є однією з найпростіших моделей хвилі, проте дає змогу показати ефект циклотронного резонансу. Вона рухається в напрямку осі OX зі швидкістю  $v = \frac{\omega}{k}$ . Якщо  $\omega = \omega_{B0}$ , де  $\omega_{B0} = \frac{q B_0}{mc}$  - циклотронна частота, то частинка опиняється в резонансі при проходженні точки  $z = 0$ , де  $B = B_0$ .

Рівняння сили Лоренца розписується в декартових координатах наступним чином:

$$\begin{cases} F_x = q(E_x + v_y B_z - v_z B_y), \\ F_y = q(E_y + v_z B_x - v_x B_z), \\ F_z = q(E_z + v_x B_y - v_y B_x) \\ \frac{dx}{dt} = v_x, \frac{dy}{dt} = v_y, \frac{dz}{dt} = v_z. \end{cases}$$

Розписавши перші три рівняння системи і підставивши масу  $m$ , заряд  $q$  та електростатичну хвилю, отримали наступну систему рівнянь:

$$\begin{cases} m \frac{dv_x}{dt} = qk\Phi_0 \sin(kx - \omega t) + \frac{q}{c} v_y B_z - \frac{q}{c} v_z B_y, \\ m \frac{dv_y}{dt} = \frac{q}{c} v_z B_x - \frac{q}{c} v_x B_z, \\ m \frac{dv_z}{dt} = \frac{q}{c} v_x B_y - \frac{q}{c} v_y B_x. \end{cases}$$

Далі необхідно здійснити нормалізацію даних рівнянь. Ця операція спрощує подальше написання програмного коду, адже всі величини приводяться до одного порядку і є безрозмірними, що дозволяє провести правильну візуалізацію.

Нормалізовані величини позначаються знаком «тильда» згори. Відтак,

$$\tilde{r} = \frac{\vec{r}}{\rho_0}, \quad \tilde{t} = \frac{t}{\omega_{BO}^{-1}}, \quad \tilde{v} = v_0 \vec{v}, \quad \tilde{B}_{x,y,z} = \frac{B_{x,y,z}}{B_0}, \quad \tilde{k} = k\rho_0, \quad \tilde{\omega} = \omega\omega_{BO}^{-1}. \quad \text{Тут } \rho_0 = \frac{v_0}{\omega_0} - \text{характерний}$$

ларморів радіус, що рівний  $\frac{v_0}{\omega_{BO}}$ ;  $B_0$  – характерна величина магнітного поля;  $v_0$  –

характерна швидкість;  $\omega_{BO} = \frac{qB_0}{mc}$  – циклотронна частота.

Для прикладу, підставимо нормалізовані величини в перше рівняння системи:

$$m \frac{v_0}{\omega_{BO}^{-1}} \frac{d\tilde{v}_x}{d\tilde{t}} = qk\Phi_0 \sin(k\rho_0\tilde{x} - \omega\omega_{BO}^{-1}\tilde{t}) + \frac{q}{c} v_0 B_0 \tilde{v}_y \tilde{B}_z - \frac{q}{c} v_0 B_0 \tilde{v}_z \tilde{B}_y.$$

Спростивши й скоротивши, маємо:

$$\frac{d\tilde{v}_x}{d\tilde{t}} = \frac{qk\Phi_0}{mv_0} \sin(\tilde{k}\tilde{x} - \tilde{\omega}\tilde{t}) + \tilde{v}_y \tilde{B}_z - \tilde{v}_z \tilde{B}_y, \quad \text{де } \tilde{k} = k\rho_0, \quad \tilde{\omega} = \omega\omega_{BO}^{-1} - \text{нормалізовані хвильове}$$

число та частота хвилі.

Після цього вводимо безрозмірну амплітуду хвилі  $A = \frac{qk\Phi_0}{mv_0}$  і отримуємо

необхідне нормалізоване рівняння

$$\frac{d\tilde{v}_x}{d\tilde{t}} = A \sin(\tilde{k}\tilde{x} - \tilde{\omega}\tilde{t}) + \tilde{v}_y \tilde{B}_z - \tilde{v}_z \tilde{B}_y.$$

Відповідним чином розписуються й нормалізуються друге й третє рівняння системи (при цьому слід враховувати, що  $E_y = E_z = 0$ , відтак перший доданок в цих рівняннях буде рівний нулю). В результаті отримуємо наступну систему диференціальних рівнянь, які необхідно розв'язати з допомогою Python для того, щоб змоделювати рух зарядженої частинки в електромагнітному полі при проходженні через циклотронний резонанс.

$$\left\{ \begin{array}{l} \frac{d\tilde{v}_x}{d\tilde{t}} = A \sin(\tilde{k}\tilde{x} - \tilde{\omega}\tilde{t}) + \tilde{v}_y \tilde{B}_z - \tilde{v}_z \tilde{B}_y, \\ \frac{d\tilde{v}_y}{d\tilde{t}} = \tilde{v}_z \tilde{B}_x - \tilde{v}_x \tilde{B}_z, \\ \frac{d\tilde{v}_z}{d\tilde{t}} = \tilde{v}_x \tilde{B}_y - \tilde{v}_y \tilde{B}_x, \\ \frac{d\tilde{x}}{d\tilde{t}} = \tilde{v}_x, \frac{d\tilde{y}}{d\tilde{t}} = \tilde{v}_y, \frac{d\tilde{z}}{d\tilde{t}} = \tilde{v}_z. \end{array} \right.$$

## 4.2. Виведення формули для дослідження зміни енергії частинки під час проходження точки локального резонансу

Для того, щоб дослідити зміну енергії частинки під час проходження точки локального резонансу, необхідно порівняти величини «стрибка» кінетичної енергії отримані за аналітичною формулою (метод стаціонарної фази) та розраховані емпіричним способом (у ході виконання практичного дослідження). Формулу 3.2.1. для розрахунку за методом стаціонарної фази наведено в розділі 3.2. Її слід використати для системи рівнянь в нашій задачі наступним чином.

У нормалізованих змінних швидкість зміни енергії дорівнює

$$\frac{d}{d\tilde{t}} (v_x^2 + v_y^2 + v_z^2) = 2v_{\perp} A \sin(kx - \omega t).$$

Введемо фазу циклотронного обертання  $\zeta$  співвідношеннями

$$v_x = v_{\perp} \cos(\zeta), v_y = v_{\perp} \sin(\zeta).$$

Легко побачити, що  $\dot{\zeta} = -B_z$ . Отже,

$$\dot{E} = 2v_{\perp} A \sin(kx - \omega t) \cos(\zeta) = v_{\perp} A [\sin(kx - \omega t + \zeta) + \sin(kx - \omega t - \zeta)].$$

Тоді резонанс можливий лише в першому доданку:

$$\dot{E} = \frac{v_{\perp} A}{2i} [e^{(ikx - i\omega t - i\zeta)} - e^{(ikx + i\omega t - i\zeta)} + \dots].$$

Далі нехтуємо  $kx$  та вводимо  $\xi = -\zeta - \omega t$ . Тоді  $\dot{\xi} = -\dot{\zeta} - \omega = B_z - \omega$ , так що  $\dot{\xi} = 0$  при  $z = 0$  (там, де має місце локальний циклотронний резонанс,  $B_z = \omega = 1$ ).

Тоді

$$\dot{E} = \frac{v_{\perp} A}{2i} \{e^{[i\xi(t)]} - e^{[-i\xi(t)]}\}.$$

Після інтегрування отримуємо стрибок енергії:

$$\Delta E = \int dt v_{\perp} A \operatorname{Im}\{e^{[i\xi(t)]}\} = \operatorname{Im}\{\int dt v_{\perp} A [e^{[i\xi(t)]}]\}.$$

Далі слід застосувати метод зупинки фази. Враховуючи, що

$$\ddot{\xi} = \frac{\partial B_z}{\partial z} \dot{z} = \alpha v_z < 0,$$

отримуємо

$$\int dt v_{\perp} A [e^{[i\xi(t)]}] = A v_{\perp}(t_*) \sqrt{\frac{2\pi}{|\ddot{\xi}(t_*)|}} e^{[i\xi(t_*) - \frac{i\pi}{4}]},$$

де «\*» позначає величину в точці зупинки фази, тобто  $\dot{\xi} = 0$ . Остаточоно записуємо

$$\Delta E = A v_{\perp*} \sqrt{\frac{2\pi}{|\alpha v_z|}} \sin(\zeta_* + \omega t_* + \frac{\pi}{4}). \quad (4.2.1.)$$

Отримали формулу 4.2.1., яку будемо застосовувати в розділі 5.2..

## ПРАКТИЧНА ЧАСТИНА

Практична частина роботи була виконана у текстовому редакторі Sublime Text, який використовується як інтерпретатор написаного тексту в програмний код Python. Для роботи були завантажені та підключені стандартні бібліотеки Python, як-от Matplotlib ( для створення графіків руху частинки в електромагнітному полі), SciPy ( для використання спеціального модуля `integ.solve_ivp`, який розв'язує системи лінійних диференціальних рівнянь), NumPy( для створення numpy-масивів даних, які є зручними у використанні), а також вбудований модуль `math`, який дозволяє застосовувати в обчисленнях число  $\pi$  та рахувати тригонометричні функції кутів.

### 5.1. Двовимірний простір

На першому етапі досліджується двовимірний простір  $(x, V_x, y, V_y)$ , в якому  $B=B_z=\text{const}$ ,  $B_x=B_y=0$ . Відповідно,  $z$  та  $V_z$  дорівнюють 0. Оскільки електростатична хвиля відсутня, амплітуда( $A$ ) також рівна нулю. Магнітне поле однорідне й направлене перпендикулярно до вектора швидкості. Детальний опис процесу створення програми наведено в додатку А.

Отже, зобразимо графіки залежностей різних величин під час руху частинки в такому полі.

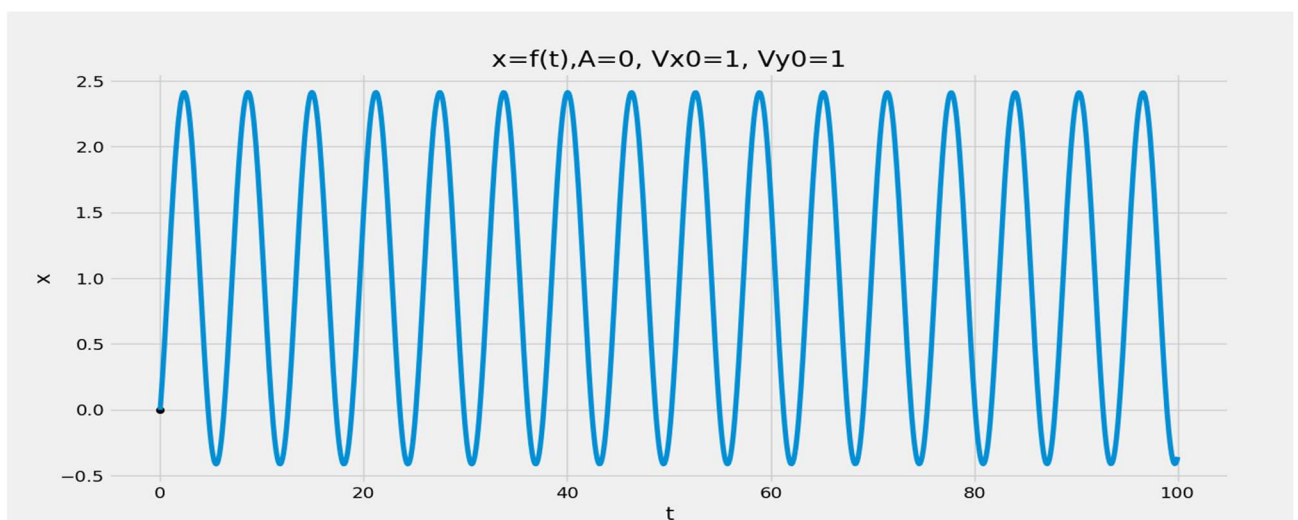
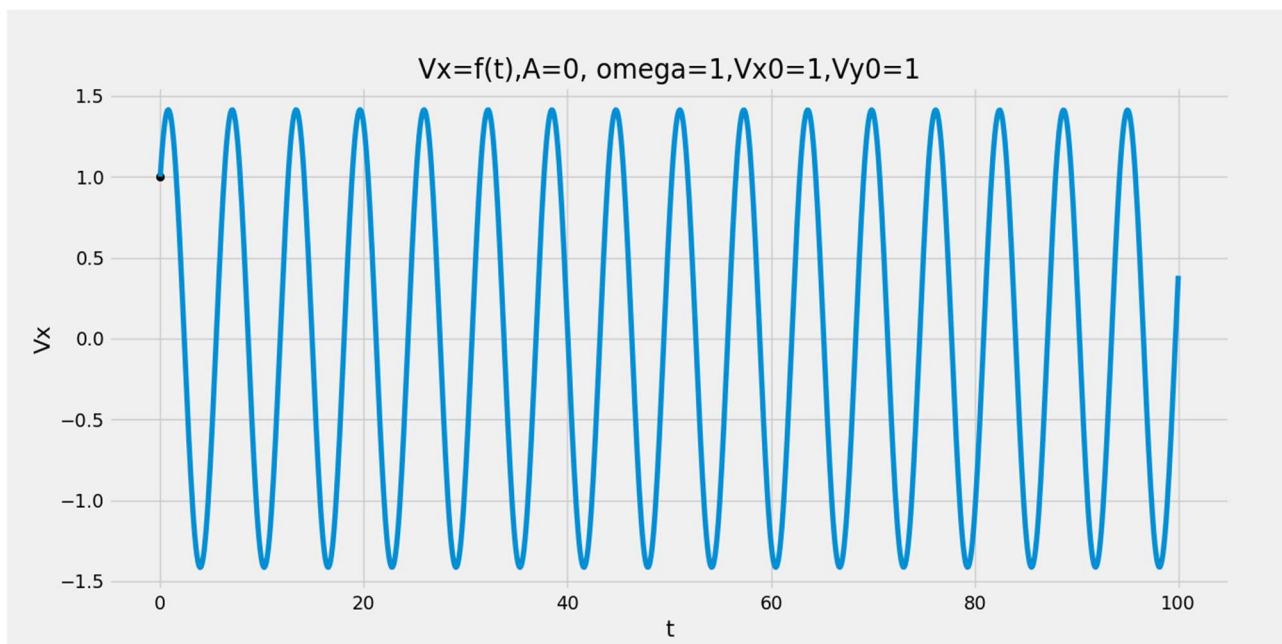
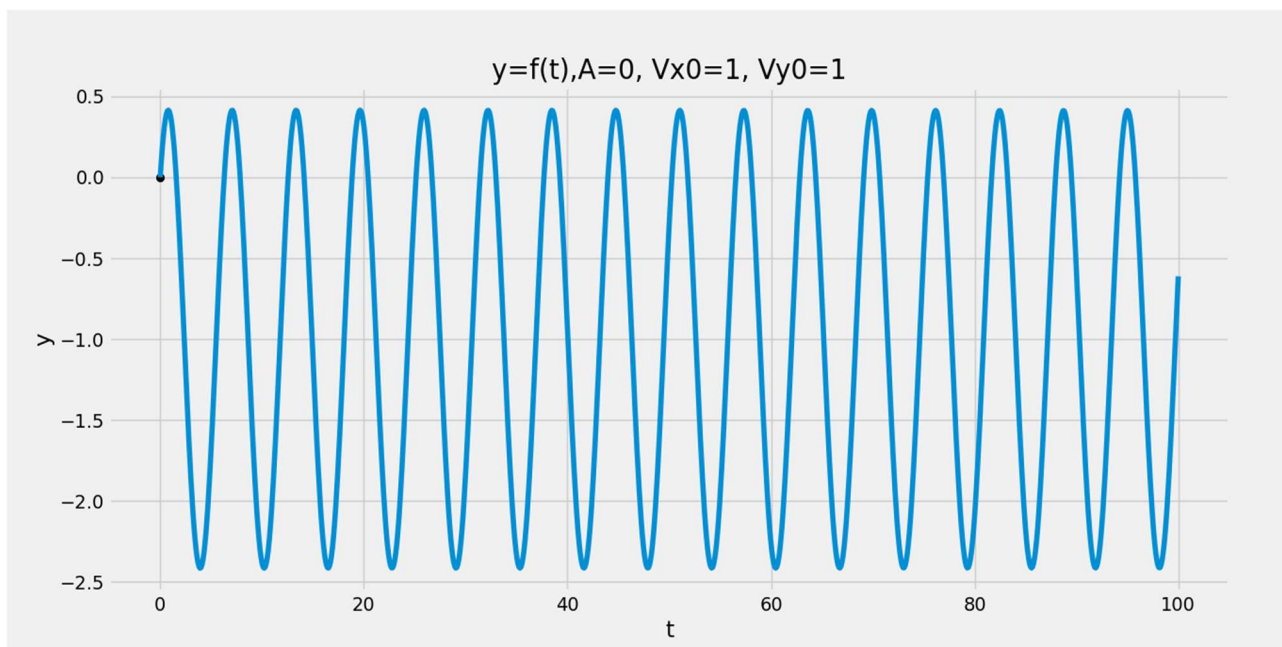


Рис.5.1.1.Графік  $x = f(t)$ .

Рис.5.1.2. Графік  $V_x = f(t)$ .

З графіків, зображених на Рис.5.1.1. та Рис.5.1.2., можна побачити, що координата та швидкість частинки по осі  $OX$  змінюються рівномірно. Те саме відбувається і з величинами по осі  $OY$ .

Рис.5.1.3. Графік  $y = f(t)$ .

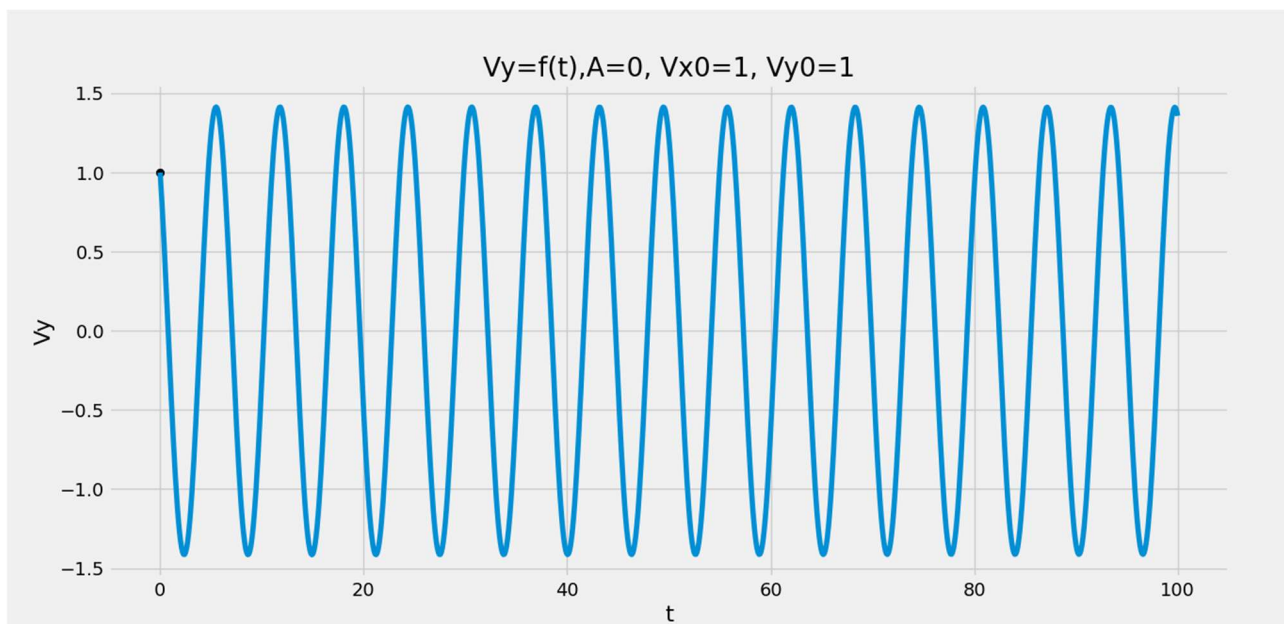


Рис.5.1.4. Графік  $V_y = f(t)$ .

Задля того, щоб дослідити траєкторію руху частинки в однорідному полі, створимо графіки залежностей швидкості  $V_x$  від координати  $x$ , а також  $u$  від  $x$ .

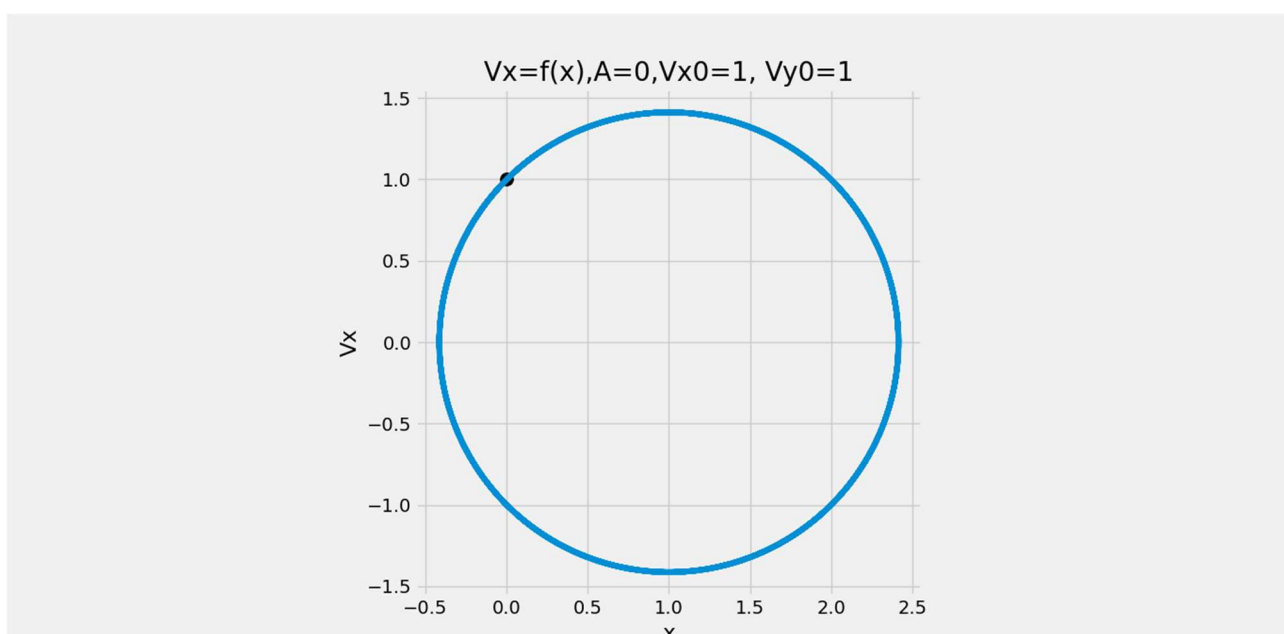


Рис.5.1.5. Графік  $V_x = f(x)$ .



Рис.5.1.6. Графік траєкторії руху частинки в площині  $OXY$ .

З наведених вище рисунків видно, що траєкторією руху частинки є коло ( за циклотронної частоти 1). Це узгоджується з теорією, яка наведена в теоретичній частині роботи, розділі 3.1.( Рис.3.1.1.). Крім того, даний запуск програми є тестом, щоб переконатись, що все працює правильно. Це необхідно для подальших дослідів.

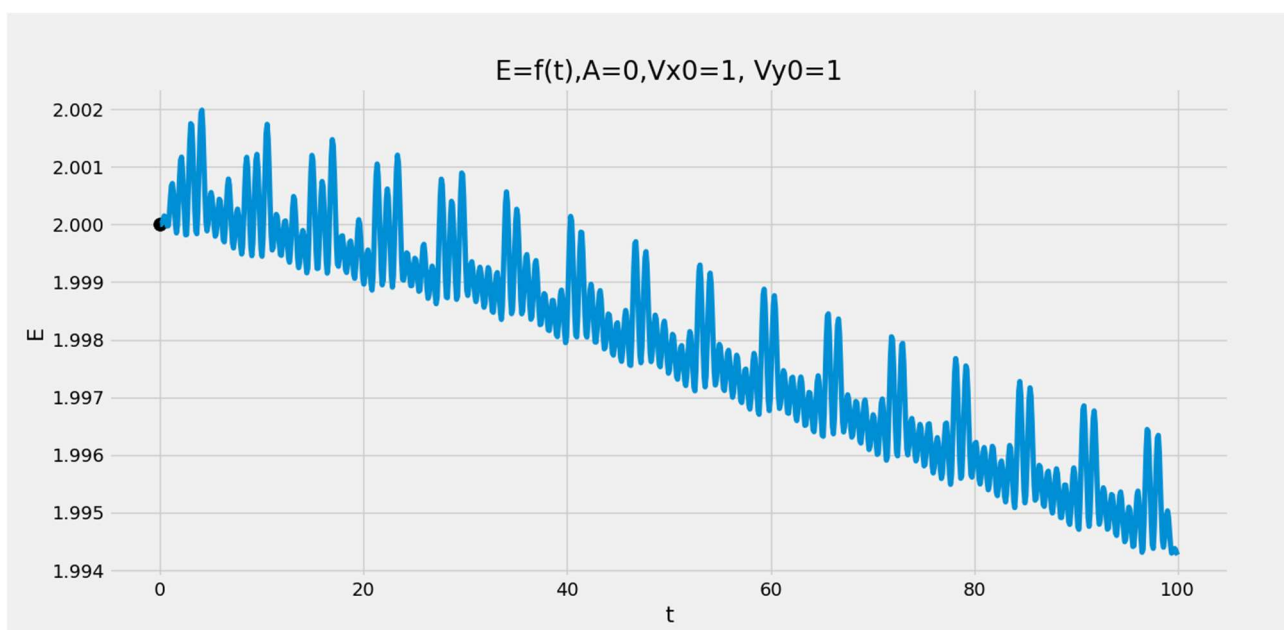


Рис.5.1.7.Графік  $E = f(t)$ .

З даного графіка видно, що кінетична енергія частинки залишається незмінною, хоч і присутня певна похибка числового методу яка, однак, є невеликою (0,3%).

Цікавим буде дослідити зміну траєкторії руху частинки вже за наявності електростатичної хвилі. Для цього візьмемо невелику амплітуду коливань хвилі  $A$ , що рівна 0.005, а також візьмемо 2 значення циклічної частоти  $\omega$  : 0,8 та одиницю, яка дорівнює значенню резонансної частоти. Також будемо підбирати різні значення початкових швидкостей  $V_x$  та  $V_y$ . Зобразимо серію графіків траєкторії руху частинки в площині  $OXY$ .



Рис.5.1.8. Графік траєкторії руху частинки в площині  $OXY; \omega=0,8; V_{x0}=5; V_{y0}=5$ .



Рис.5.1.9. Графік траєкторії руху частинки в площині  $OXY; \omega=0,8; V_{x0}=0; V_{y0}=5$ .

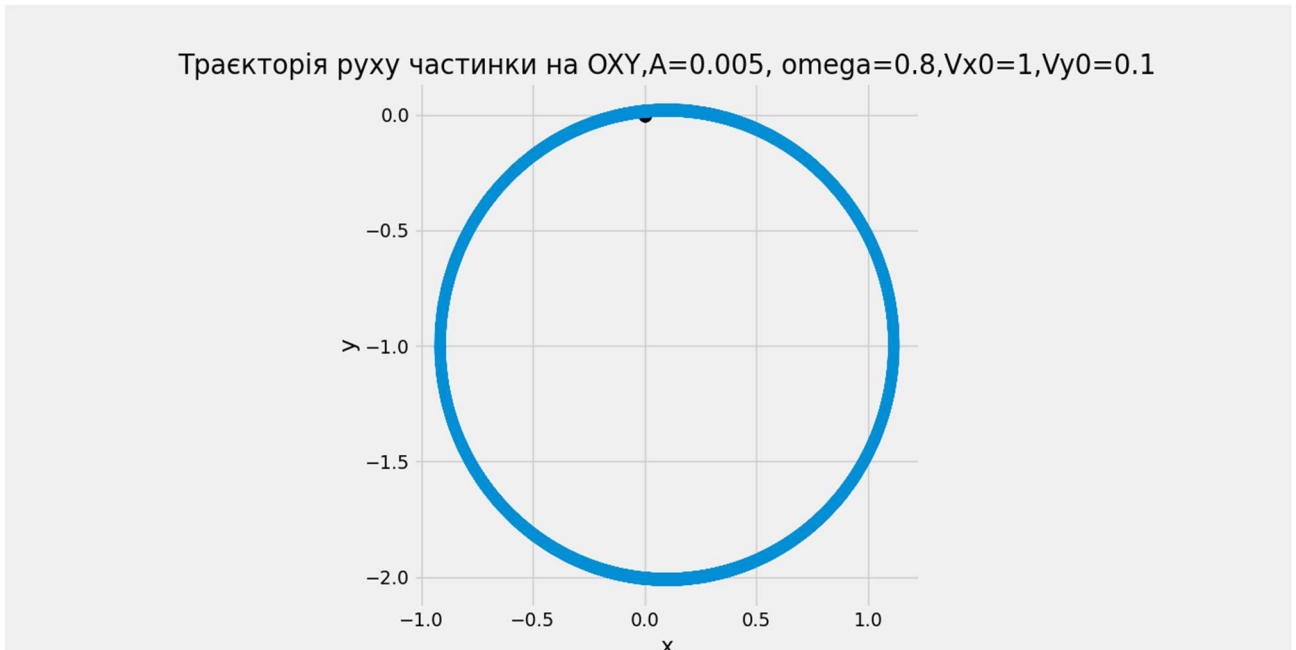


Рис.5.1.10. Графік траєкторії руху частинки в площині  $OXY$ ;  $\omega=0,8$ ;  $V_{x0}=1$ ;  $V_{y0}=0,1$ .

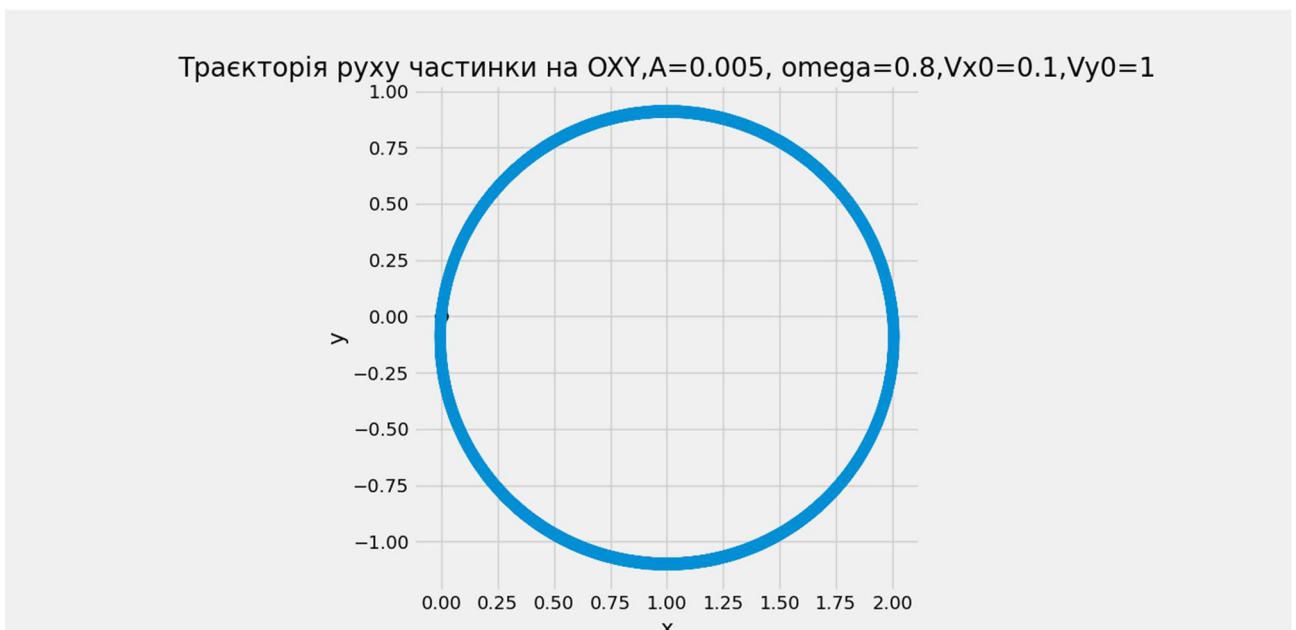


Рис.5.1.11. Графік траєкторії руху частинки в площині  $OXY$ ;  $\omega=0,8$ ;  $V_{x0}=0,1$ ;  $V_{y0}=1$ .

Аналізуючи останні 4 графіки, можна підсумувати, що якщо циклічна частота коливань  $\omega$  не дорівнює резонансній частоті, і, як мінімум, одна з компонент  $V_x$ ,  $V_y$  вектора швидкості  $V$  дорівнює або є більшою за одиницю, то траєкторією руху частинки в площині  $OXY$  буде коло.



Рис.5.1.12. Графік траєкторії руху частинки в площині  $OXY$ ;  $\omega=0,8$ ;  $V_{x0}=0,5$ ;  $V_{y0}=0,5$ .

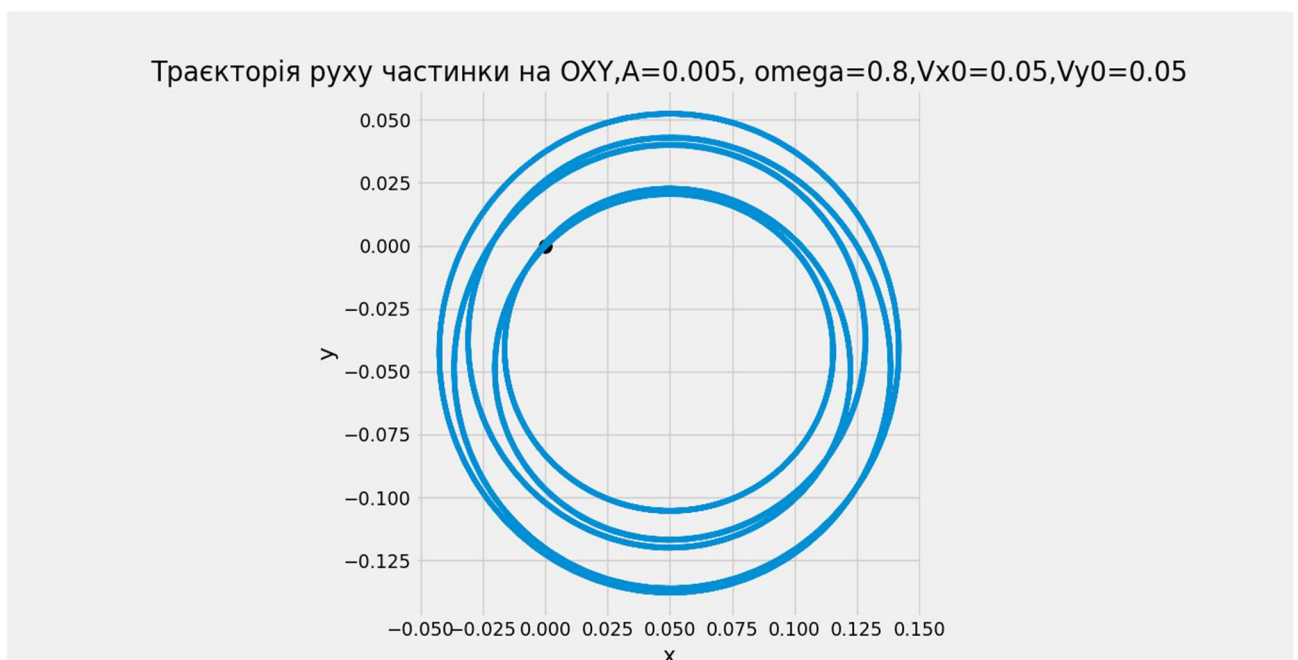


Рис.5.1.13. Графік траєкторії руху частинки в площині  $OXY$ ;  $\omega=0,8$ ;  $V_{x0}=0,05$ ;  $V_{y0}=0,05$ .

З останніх двох графіків помітно, що лише коли швидкості  $V_x$  та  $V_y$  мають близькі до нуля значення, то траєкторія руху частинки починає змінюватись з часом. Частинка починає «розкручуватись», і радіус кола починає збільшуватись. Якщо ж задати значення початкових швидкостей  $V_x$ ,  $V_y$  рівними

нулю, то отримаємо симетричну фігуру( див.Рис.5.1.14.), але при цьому чітко визначити траєкторію руху частинки складно.

Відтак, можна підсумувати, що чим меншу початкову кінетичну енергію має частинка, тим більш помітними є її коливання ( у випадку нерезонансної частоти  $\omega$ ). Поведінка частинки є детермінованою, тобто радіус кола, по якому рухається частинка, змінюється обмежено. Обміну енергією з хвилею немає, адже відсутнє постійне прискорення, по енергії частинка лише коливається.

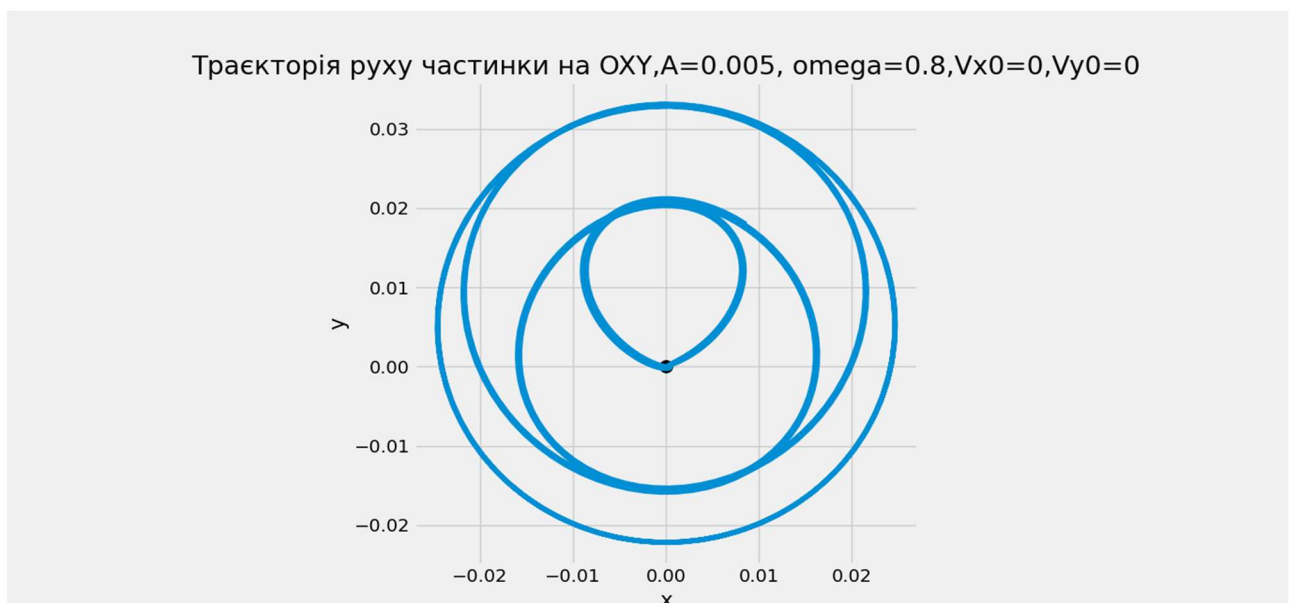


Рис.6.1.14. Графік траєкторії руху частинки в площині OXY;  $\omega=0,8$ ;  $V_{x0}=0$ ;  $V_{y0}=0$ .

Тепер цікавим буде дослідити траєкторію руху частинки при циклотронній частоті  $\omega$ , яка дорівнює резонансній частоті й, відповідно, рівна одиниці.

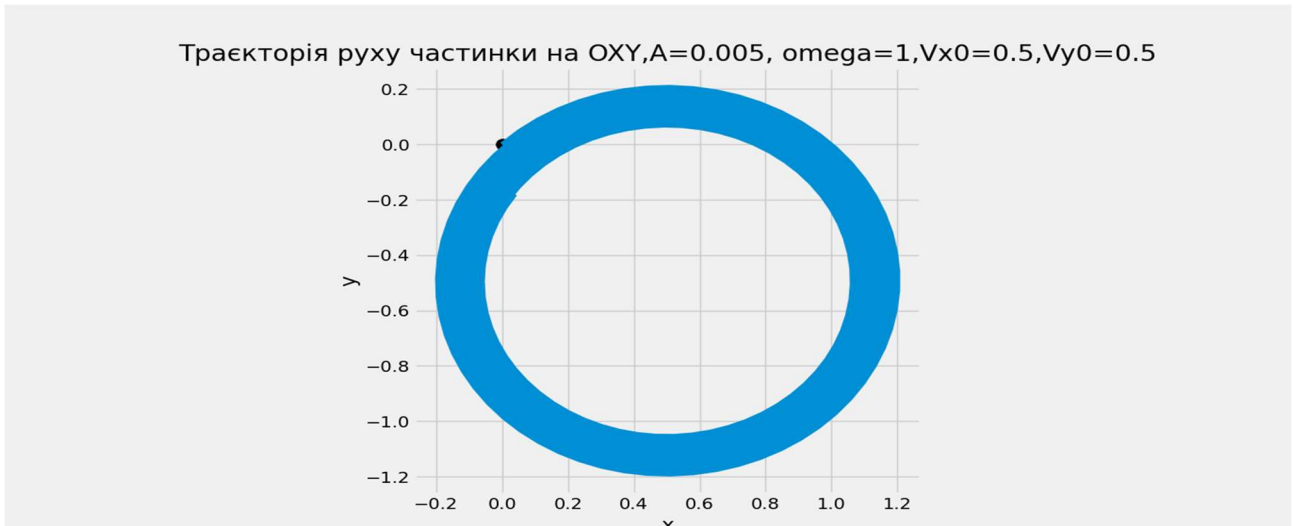


Рис.5.1.15. Графік траєкторії руху частинки в площині  $OXY$ ;  $\omega=1$ ;  $V_{x0}=0,5$ ;  $V_{y0}=0,5$ .

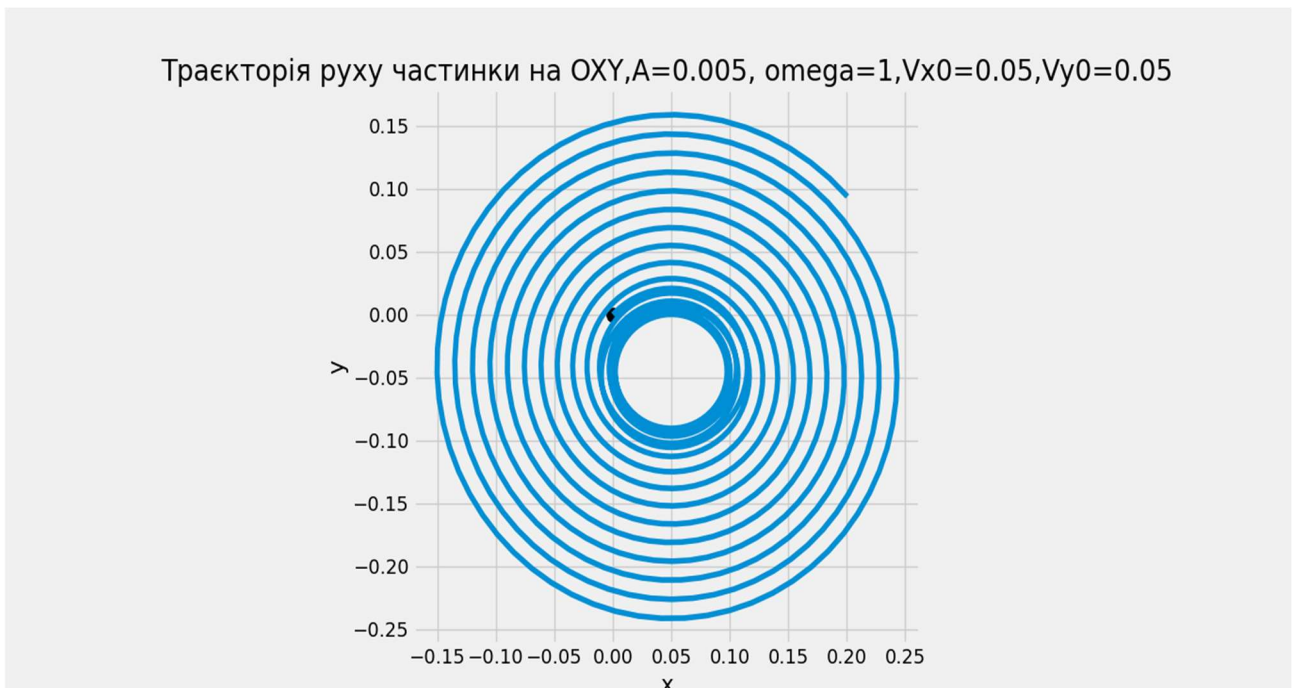


Рис.5.1.16. Графік траєкторії руху частинки в площині  $OXY$ ;  $\omega=1$ ;  $V_{x0}=0,05$ ;  $V_{y0}=0,05$ .

Неоднорідною є поведінка частинки у випадку, коли її компоненти швидкості по відповідним осям  $OX$ ,  $OY$  отримують початкові значення, які менші одиниці. На рис 5.1.15. видно, що коли  $V_x=0,5$ ;  $V_y=0,5$ , то радіус кола, який описує частинка, поступово зменшується. У випадку  $V_x=0,05$ ;  $V_y=0,05$  (Рис.5.1.16.), тобто значення близькі до нуля, частинка навпаки починає «розкручуватись» - на початковому етапі радіус кола збільшується

нерівномірно, але оскільки частинка весь час перебуває в резонансі, то в подальшому вона коливається рівномірно.

Якщо ж задати початкові значення  $V_x, V_y$  рівними нулю, то можна буде спостерігати( Рис.5.1.17) явище циклотронного резонансу. В такому випадку початкове прискорення частинки дорівнює нулю, а радіус спіралі рівномірно збільшується протягом усього часу дії системи. Швидкість руху частинки буде зростати рівномірно( Рис.5.1.18) і вона буде набувати постійного прискорення. Таким чином, частинка буде “відбирати” енергію у електростатичної хвилі.

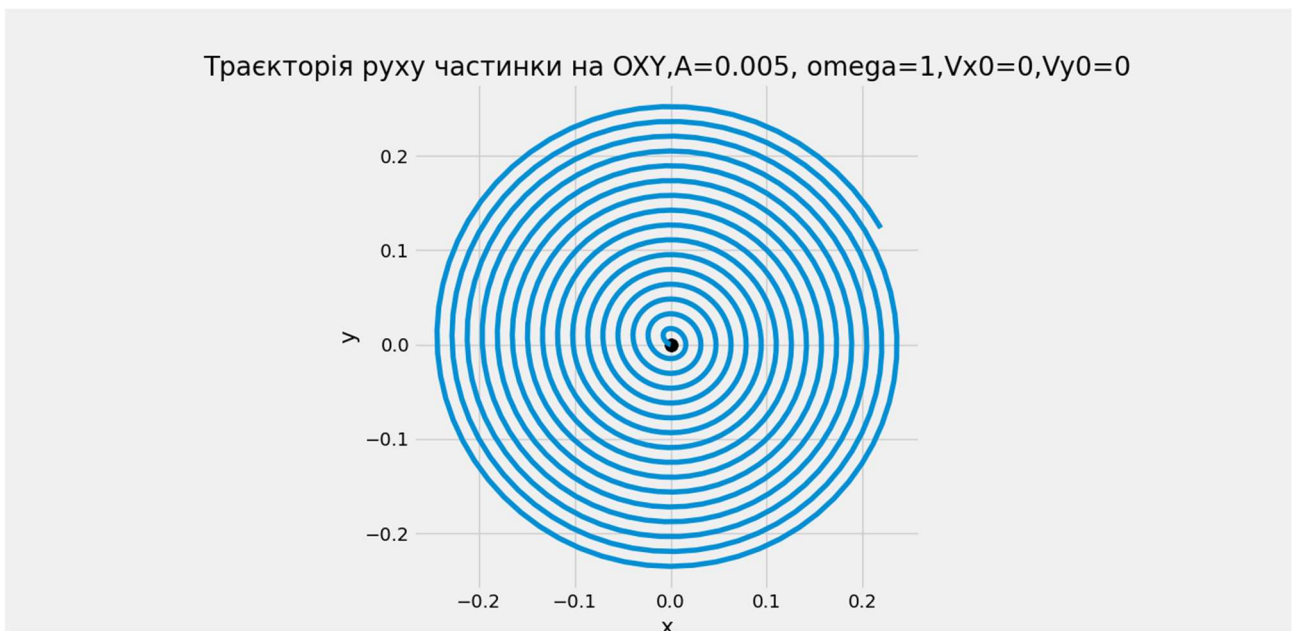


Рис.5.1.17. Графік траєкторії руху частинки в площині  $OXY; \omega=1; V_{x0}=0; V_{y0}=0$ .

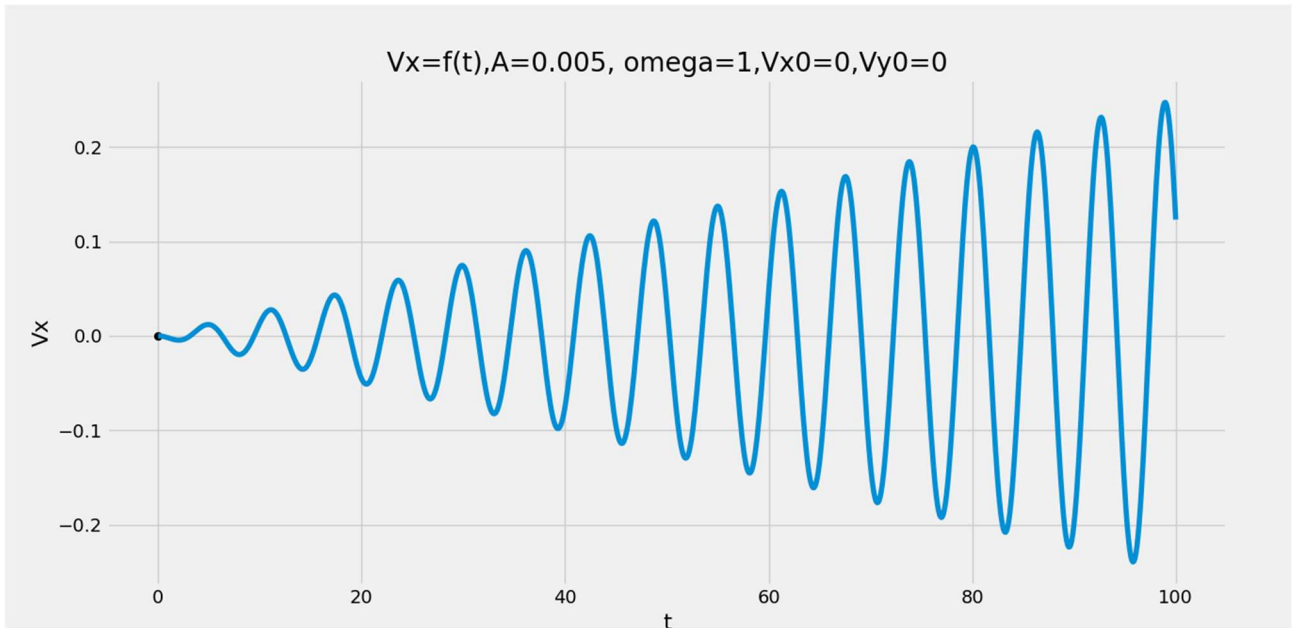


Рис.5.1.18. Графік  $V_x = f(t)$  у випадку циклотронного резонансу;  $\omega=1$ ;  $V_{x0}=0$ ;  $V_{y0}=0$ .

Цікаво буде дослідити залежність параметрів руху частинки, а саме: радіуса, швидкості та зміни енергії - від амплітуди коливань електростатичної хвилі під час циклотронного резонансу. Для цього внесемо певні зміни в програму ( див. додаток Б). Візьмемо 6 значень амплітуди коливань, починаючи з нуля і з кроком 0.005.

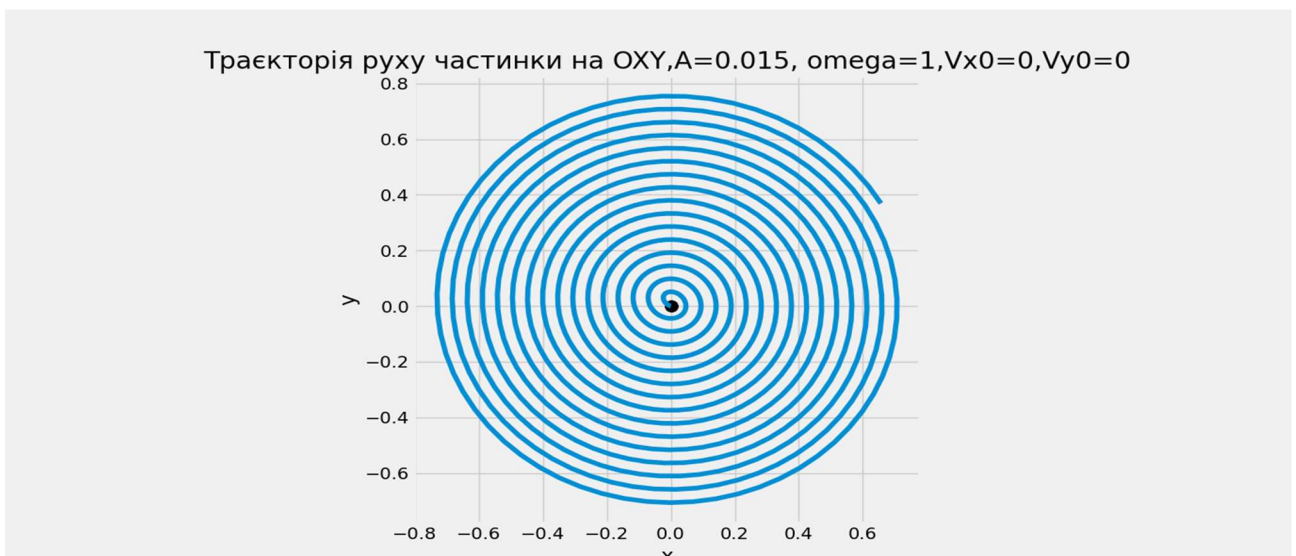


Рис.5.1.19. Графік траєкторії руху частинки в площині  $OXY$ ;  $A=0,015$ ;  $\omega=1$ ;  $V_{x0}=0$ ;  $V_{y0}=0$ .

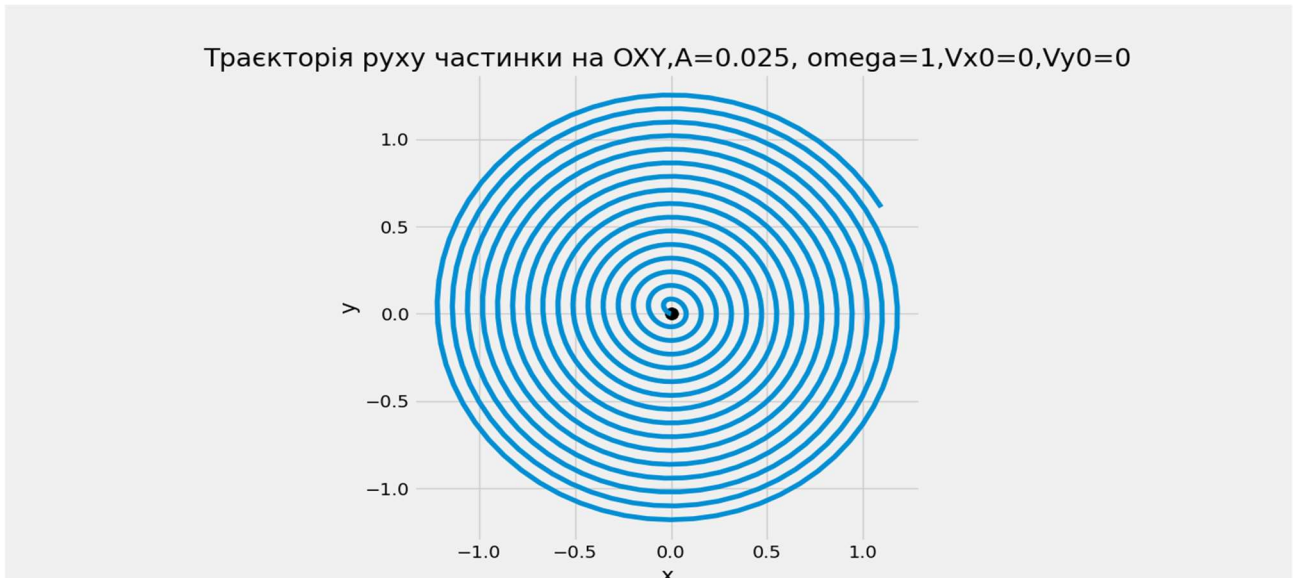


Рис.5.1.20. Графік траєкторії руху частинки в площині OXY;  
 $A=0,025$ ;  $\omega=1$ ;  $V_{x0}=0$ ;  $V_{y0}=0$ .

Проаналізувавши дані графіки, а також Рис.5.1.17, можна зробити висновок, що радіус руху частинки по колу під час циклотронного резонансу зростає разом зі збільшенням амплітуди.

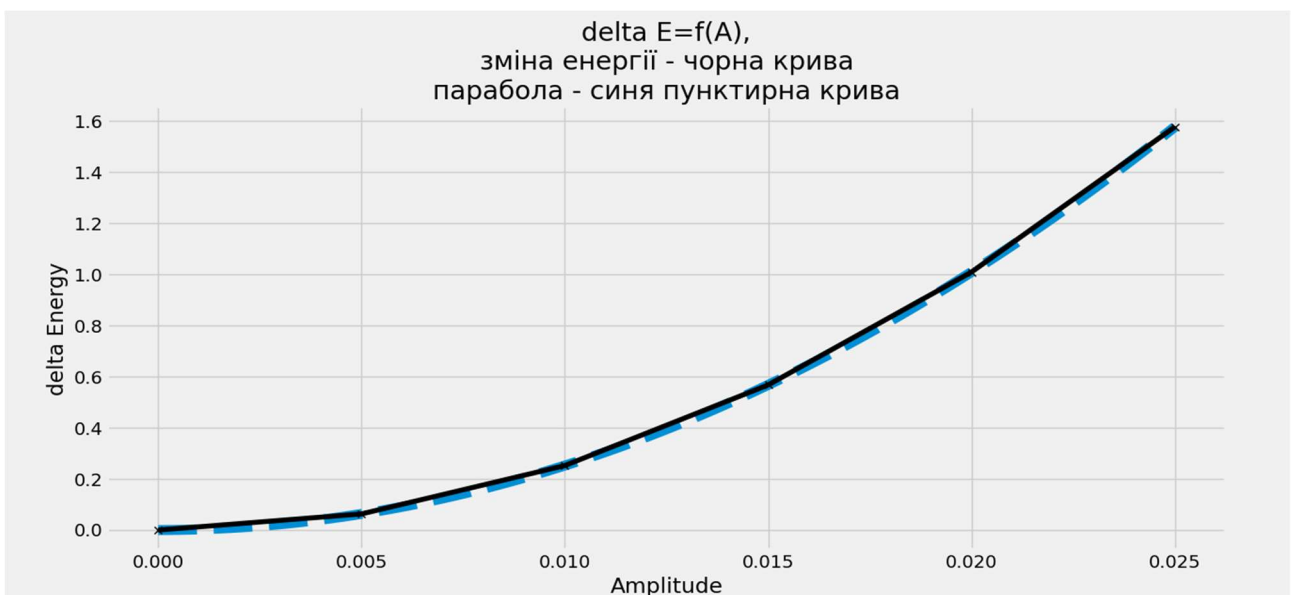


Рис.5.1.21. Зміна кінетичної енергії частинки від амплітуди коливань.

Для того, щоб точно визначити характер залежності зміни кінетичної енергії частинки від амплітуди, на графік також було нанесено параболу.

Оскільки дві криві майже ідентичні між собою, можна зробити висновок, що зміна енергії частинки квадратично залежить від амплітуди електростатичної хвилі. А ось модуль швидкості частинки  $V$ , який рівний  $V = \sqrt{V_x^2 + V_y^2}$ , від амплітуди залежить лінійно.

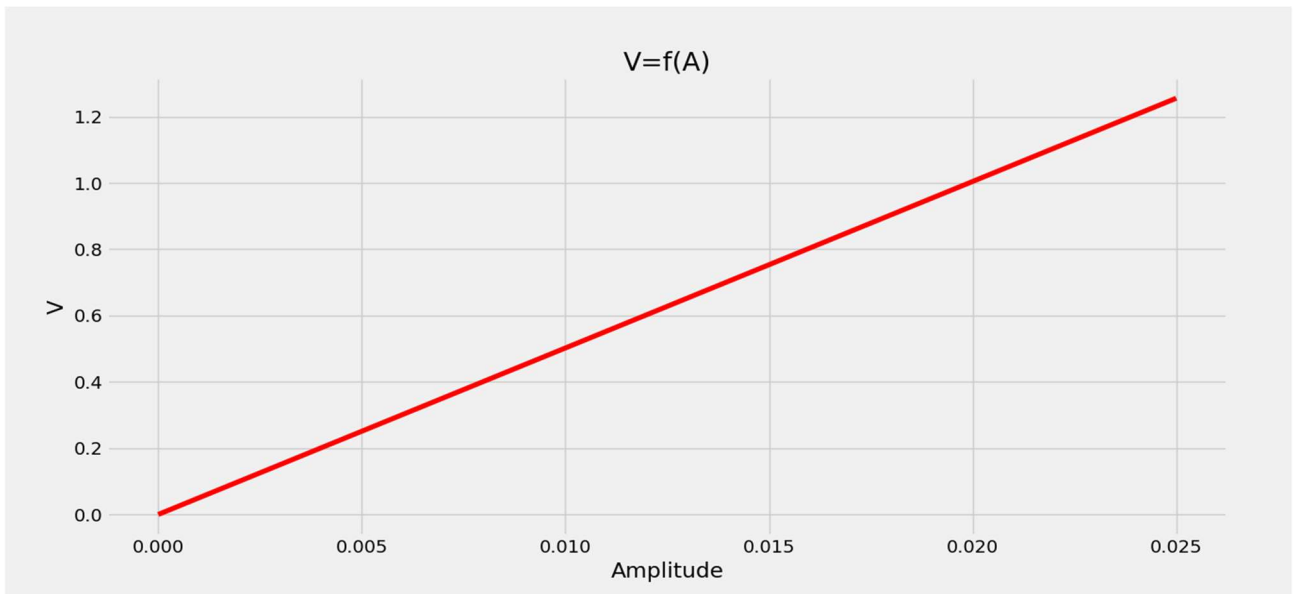


Рис.5.1.22. Залежність модуля швидкості частинки від амплітуди коливань.

## 5.2. Тривимірний простір

Друга частина дослідження полягає в тому, що тепер буде вивчатись динаміка руху зарядженої частинки у тривимірному просторі  $(x, V_x, y, V_y, z, V_z)$ . Отже, з'явиться координата  $z$  та компонента вектора швидкості  $V$  по осі  $OZ - V_z$ .

Для початку, проведемо дослідження руху частинки у системі координат  $OXYZ$  за відсутності електростатичної хвилі та за однорідного магнітного поля. Відповідно, амплітуда  $A$ , компоненти  $B_x$  та  $B_y$  магнітного поля будуть дорівнювати нулю,  $B_z$  - одиниці. Циклічна частота  $\omega$  протягом дослідження в просторі  $OXYZ$  буде рівна резонансній частоті, тобто одиниці. Детальний опис процесу створення програми наведено в додатку В.

Траєкторія руху частинки на  $OXYZ$ ,  $A=0$ ,  $V_{x0}=1$ ,  $V_{y0}=1$ ,  $V_{z0}=1$

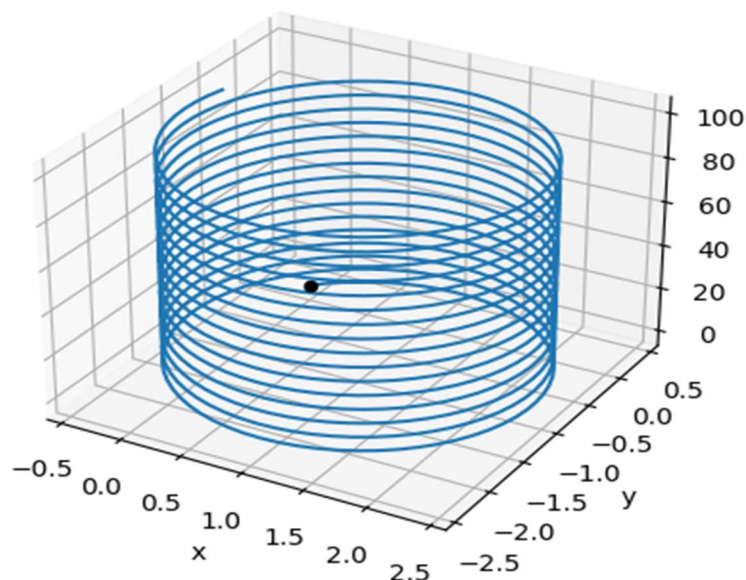


Рис. 5.2.1. Траєкторія руху частинки в просторі  $OXYZ$ ;  $A=0$ .

З графіка ми бачимо, що в однорідному магнітному полі, за відсутності електростатичної хвилі, траєкторією руху частинки є гвинтова лінія, що є узгоджується з теорією, наведеної в розділі 3.1. (Рис.3.1.1.).

Якщо ж додати електростатичну хвилю, і задати значення амплітуди рівним 0,05, то частинка продовжить рухатись по спіралі, але радіус її руху буде змінним.

Траєкторія руху частинки на  $OXYZ$ ,  $A=0.05$ ,  $V_{x0}=1$ ,  $V_{y0}=1$ ,  $V_{z0}=1$

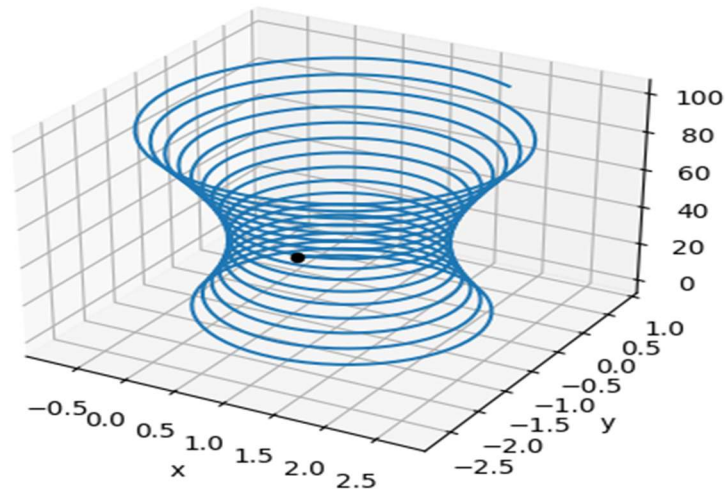


Рис.5.2.2. Траєкторія руху частинки у просторі  $OXYZ$ ;  $A=0,05$ .

Наступним дослідом стане «запуск» частинки вже в неоднорідному магнітному полі, тобто з'явиться компонента  $B_x$ , яка, разом із  $B_z$ , будуть запрограмовані як функції від вектора  $\mathbf{a}$  (детальніше в додатку Г). Також в цьому випадку буде присутня електростатична хвиля. Амплітуда  $A$  є невеликою і дорівнює 0,002. Окрім того, характеристичний параметр магнітного поля  $\alpha$  буде дорівнювати  $\approx 0,002$ . Також постійного значення набуде  $B_{z0}$ , яке характеризує магнітне поле по осі  $OZ$  в початковий момент часу.

Разом з тим, в цьому досліді нас цікавить не лише траєкторія руху частинки, а й «стрибок» її кінетичної енергії під час проходження точки резонансу. Плюс, необхідно було дослідити узгодження між величинами «стрибків» енергії, розрахованих аналітичним способом (за методом стаціонарної фази, див. розділи 3.2. та 4.2.) і отриманих практичним способом (у процесі «запуску» частинки.) Для цього в програму з додатку В було внесено кілька значних доповнень (див. додаток Г).

Важливу роль в даному досліді відіграє кут  $\zeta$  – початкова фаза запуску. Геометрично, це кут між напрямом модуля швидкості  $V_{\perp}$  ( $V_x^2 + V_y^2$ ) та віссю  $OX$ ,

що вираховується за формулою  $\frac{\pi}{12 * i}$ . Щоб виявити момент, коли явище локального резонансу помітно найкраще, всього було проведено 25 «запусків» частинки з різними кутами ( 25 для того, щоб повністю замкнути коло).

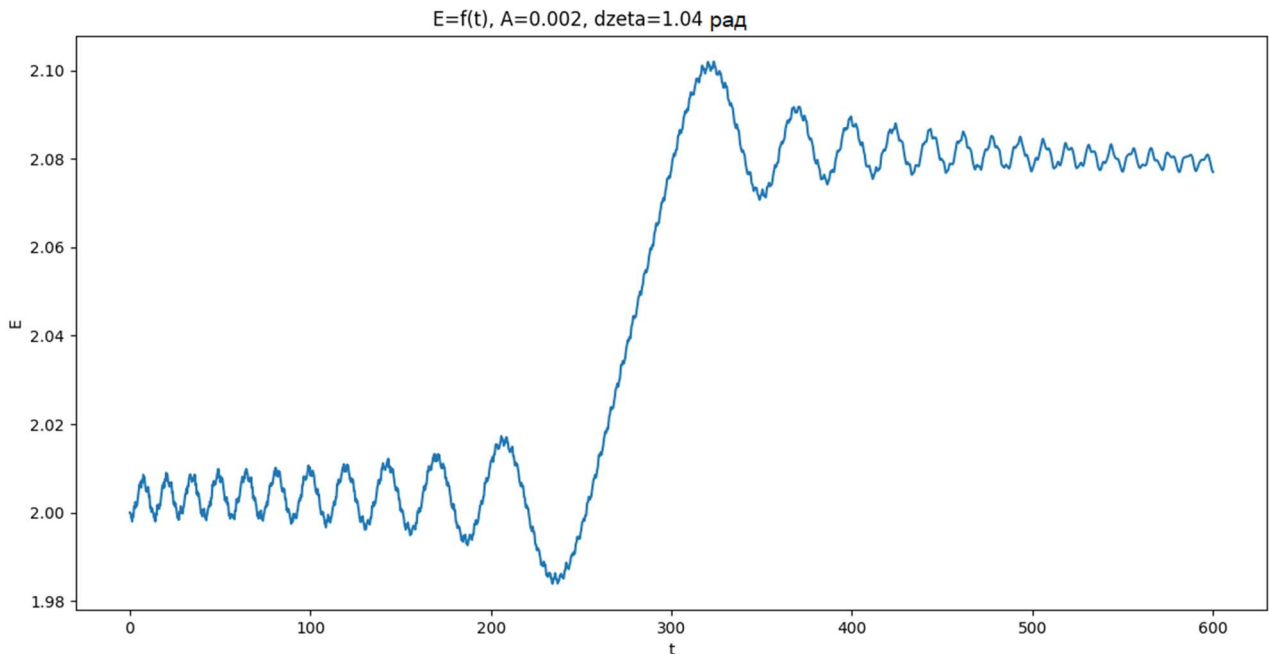


Рис.5.2.3. Явище локального циклотронного резонансу.

На Рис. 5.2.3. можна побачити «стрибок» кінетичної енергії частинки після проходження точки локального циклотронного резонансу. Найкраще воно помітно при  $\zeta=1,04$  рад, що дорівнює  $\frac{\pi}{3}$ . Значення кінетичної енергії частинки різко зростає при проходженні частинкою точки резонансу. Аналогічно і радіус кола, по якому рухається частинка, також збільшується після проходження цієї точки( Рис.5.2.4., Рис.5.2.5.).

Траєкторія частинки на  $OXYZ$ ,  $A=0.002, dzeta=1.04$

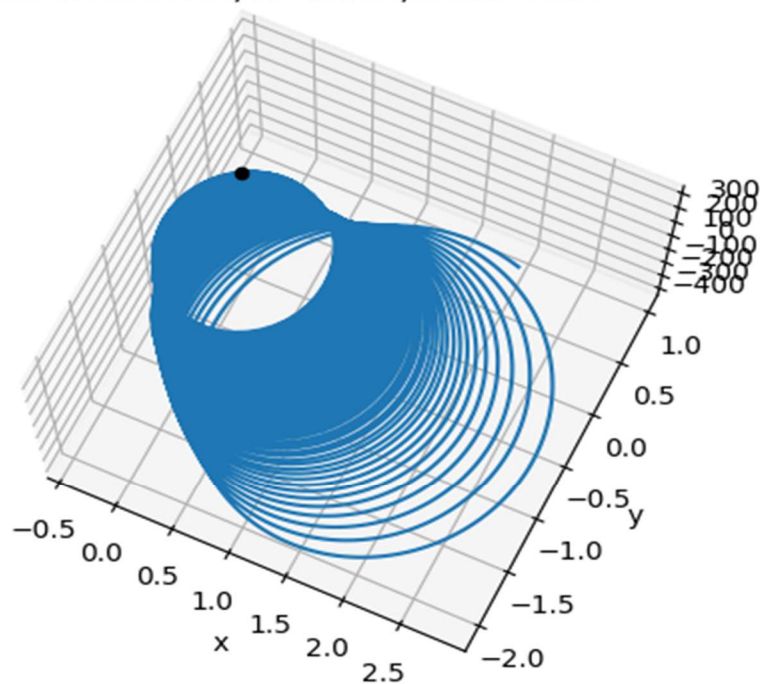


Рис. 5.2.4. Траєкторія частинки у просторі  $OXYZ$ , вигляд зверху.

Траєкторія частинки на  $OXYZ$ ,  $A=0.002, dzeta=1.04$  рад

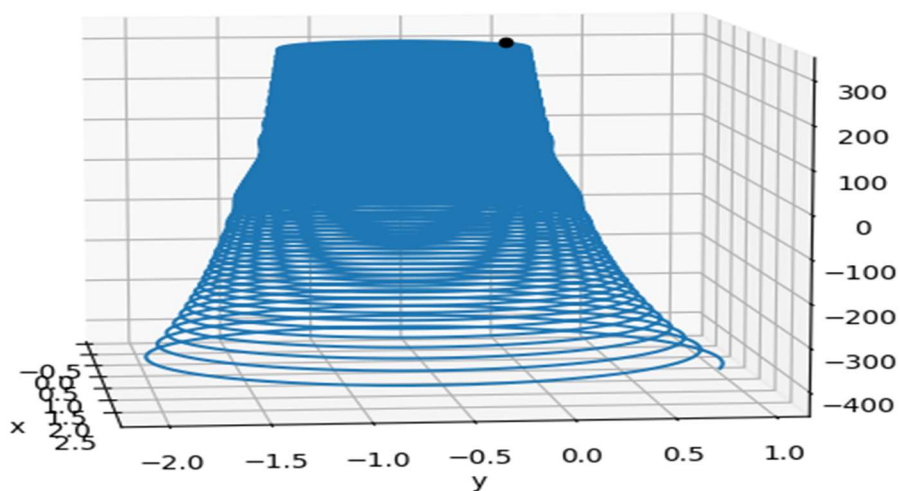


Рис. 5.2.5. Траєкторія частинки у просторі  $OXYZ$ , вигляд збоку.

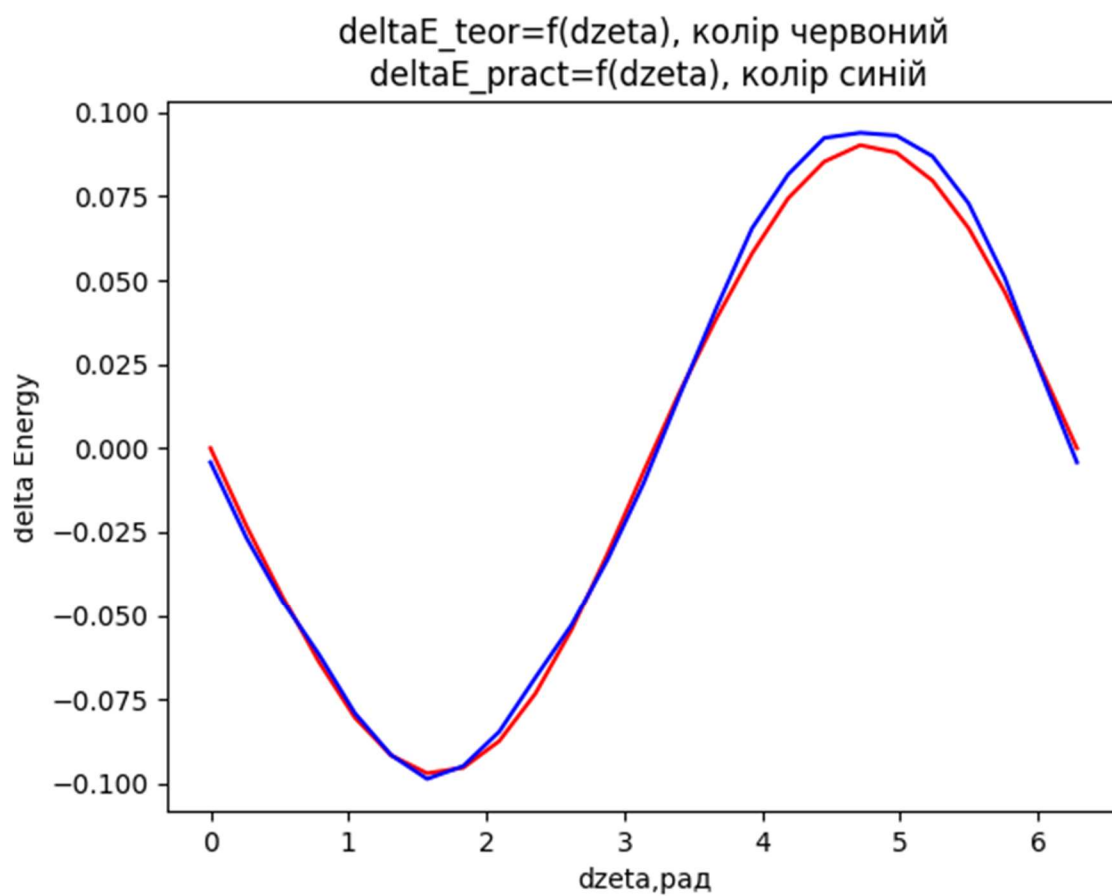


Рис.5.2.6. Порівняння  $\delta E_{teor}$  та  $\delta E_{pract}$ .

З даного графіка видно, що дві формули мають відмінне узгодження. Це свідчить про те, що і аналітична формула, отримана шляхом асимптотичного розкладу, і числове моделювання є придатними для обчислення величини «стрибка» кінетичної енергії.

## ВИСНОВОК

В даній роботі було проведено числове дослідження динаміки руху зарядженої частинки в електромагнітному полі при її проходженні через циклотронний резонанс. Встановлено, що в однорідному магнітному полі за відсутності електростатичної хвилі траєкторію руху частинки є коло ( на площині  $OXY$ ), кінетична енергія є незмінною, а швидкість змінюється гармонічно. Якщо ж ввести електростатичну хвилю з невеликою величиною амплітуди ( наприклад, 0,005) й встановити циклічну частоту  $\omega$  меншою за одиницю, то можна побачити як кінетична енергія частинки буде зазнавати коливань. Проте за циклічної частоти, яка рівна частоті циклотронного резонансу і дорівнює одиниці, частинка буде рухатись по спіралі, а кінетична енергія частинки буде змінюватись лише в одному напрямку. Також було показано квадратичну залежність зміни енергії частинки від амплітуди електростатичної хвилі. Модуль швидкості частинки, в свою чергу, має лінійну залежність від амплітуди.

Окрім того, було досліджено динаміку руху частинки у тривимірному просторі, в однорідному та неоднорідному магнітному полях. Встановлено, що частинка буде рухатись по гвинтовій лінії, а радіус та зміщення «гвинта» в сторону певної осі координатної системи буде залежати від величини компонент магнітного поля  $B$  та його характеру.

Наостанок, було показано, що кінетична енергія частинки зазнає різкої зміни під час проходження нею точки локального циклотронного резонансу. Виявлено узгодження між аналітичною ( за методом стаціонарної фази) формулою для розрахунку величини «стрибка» енергії та формулою, що була отримана емпіричним шляхом, під час проведення практичного досліду.

Оскільки зміна кінетичної енергії частинки під час проходження циклотронного резонансу відіграє важливу роль у методах нагрівання плазми, подальші дослідження в цій тематиці можуть становити вагоме наукове значення.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Whistler waves in space and laboratory plasma. *Journal of geophysical research*. 1999. Vol. 104, a7.
2. Miyamoto K. Fundamentals of plasma physics and controlled fusion. 2000.
3. Ceurstemont S. Plasma jet engines that could take you from the ground to space. *New Scientist*. URL: <https://www.newscientist.com/article/mg23431264-500-plasma-jet-engines-that-could-take-you-from-the-ground-to-space/>
4. Lawrence E. O. The production of high speed light ions without the use of high voltages. *Physical review*. 1932. Volume 40.
5. Ольховский И. И. Курс теоретической механики для физиков: Учебное пособие. 3-тє вид. Москва: Издательство Московск. университета, 1978. 575 с.
6. Попов О. А. Characteristics of electron cyclotron resonance plasma sources. *Journal of vacuum science & technology A: vacuum, surfaces, and films*. 1989. Vol. 7, no. 3. P. 894–898.
7. Kuo S. P. Electron cyclotron harmonic resonances in high-frequency heating of the ionosphere. *Physics of plasmas*. 2013. Vol. 20, no. 9. P. 092124.
8. Ion cyclotron resonance heated conics: theory and observations / G. B. Crew et al. *Journal of geophysical research*. 1990. Vol. 95, a4. P. 3959.
9. The Role of Proton Cyclotron Resonance as a Dissipation Mechanism in Solar Wind Turbulence: A Statistical Study at Ion-kinetic Scales / L. D. Woodham et al. *The Astrophysical Journal*. 2018. Vol. 856, no. 1. P. 49.
10. Sensitive Test for Ion-Cyclotron Resonant Heating in the Solar Wind / J. C. Kasper et al. *Physical Review Letters*. 2013. Vol. 110, no. 9.
11. Cranmer S. R., Field G. B., Kohl J. L. Spectroscopic Constraints on Models of Ion Cyclotron Resonance Heating in the Polar Solar Corona and High-Speed Solar Wind. *The Astrophysical Journal*. 1999. Vol. 518, no. 2. P. 937–947.
12. Prater R. Heating and current drive by electron cyclotron waves. *Physics of plasmas*. 2004. Vol. 11, no. 5. P. 2349–2376.

13. Koch R. The ion cyclotron, lower hybrid, and alfvén wave heating methods. *Fusion science and technology*. 2004. Vol. 45, 2T. P. 203–210.
14. The role of energetic particles in fusion plasmas / S. D. Pinches et al. *Plasma physics and controlled fusion*. 2004. Vol. 46, 12B. P. B187–B200.
15. Chapter 5: physics of energetic ions / A. Fasoli et al. *Nuclear fusion*. 2007. Vol. 47, no. 6. P. S264–S284.
16. Breizman B. N., Sharapov S. E. Major minority: energetic particles in fusion plasmas. *Plasma physics and controlled fusion*. 2011. Vol. 53, no. 5. P. 054001.
17. Central flattening of the fast-ion profile in reversed-shear DIII-D discharges / W. W. Heidbrink et al. *Nuclear fusion*. 2008. Vol. 48, no. 8. P. 084001.
18. Gorelenkov N. N. Energetic particle-driven compressional Alfvén eigenmodes and prospects for ion cyclotron emission studies in fusion plasmas. *New journal of physics*. 2016. Vol. 18, no. 10. P. 105010.
19. Fast particle-driven ion cyclotron emission (ICE) in tokamak plasmas and the case for an ICE diagnostic in ITER / K. G. McClements et al. *Nuclear Fusion*. 2015. Vol. 55, no. 4. P. 043013.
20. Correlation between electron transport and shear alfvén activity in the national spherical torus experiment / D. Stutman et al. *Physical review letters*. 2009. Vol. 102, no. 11.
21. Density perturbation mode structure of high frequency compressional and global Alfvén eigenmodes in the National Spherical Torus Experiment using a novel reflectometer analysis technique / N. A. Crocker et al. *Nuclear fusion*. 2017. Vol. 58, no. 1. P. 016051.
22. Coupling of neutral-beam-driven compressional alfvén eigenmodes to kinetic alfvén waves in NSTX tokamak and energy channeling / E. V. Belova et al. *Physical review letters*. 2015. Vol. 115, no. 1.
23. Compressional Alfvén eigenmode instability in NSTX / N. N. Gorelenkov et al. *Nuclear fusion*. 2002. Vol. 42, no. 8. P. 977–985.

24. Belikov V. S., Kolesnichenko Y. I. Quasilinear theory for a tokamak plasma in the presence of cyclotron resonance. *Plasma physics and controlled fusion*. 1994. Vol. 36, no. 11.
25. Эрдейи А. Асимптотические разложения. М.: Физматлит, 1962. 127 с.
26. Копсон Э. Т. Асимптотические разложения. М.: Мир, 1966. 159 с.
27. Stationary phase approximation - Wikipedia. *Wikipedia, the free encyclopedia*. URL: [https://en.wikipedia.org/wiki/Stationary\\_phase\\_approximation](https://en.wikipedia.org/wiki/Stationary_phase_approximation)

## ДОДАТОК А.НАПИСАННЯ ПРОГРАМИ ДЛЯ ДВОВИМІРНОГО ПРОСТОРУ

Для дослідження двовимірного простору  $(x, V_x, y, V_y)$ , де  $z$  та  $V_z$  дорівнюють нулю, код програми буде виглядати наступним чином.

$B=B_z=const$ ,  $B_x=B_y=0$ , нормалізовані рівняння сили Лоренца матимуть такий вигляд, «тильда» над нормалізованими величинами прибирається.

```

6 # Рівняння у 4-х вимірному фазовому просторі:
7
8 #B=Bz=const, Bx=By=0 , z=Vz=0
9
10 #1)dx/dt=Vx
11 #2)dVx/dt=A*sin(kx-wt)+VyBz-VzBy=A*sin(kx-omega*t)+VyBz
12
13 #3)dy/dt=Vy
14 #4)dVy/dt=VzBx-VxBz=-VxBz

```

Рис.А.1.Система рівнянь у двовимірному просторі.

Для того щоб розв'язати цю систему рівнянь за допомогою модуля *integ.solve\_ivp*, необхідно створити 4-х компонентний вектор  $\mathbf{a}$ , який складається з координат та швидкостей по осям  $Ox$  та  $Oy$  відповідно. Після цього підставляємо даний вектор у отриману систему рівнянь й отримуємо векторне рівняння( Рис.А.2.).

```

16 #Вводимо вектор a[0]=x, a[1]=Vx, a[2]=y, a[3]=Vy
17 #Отримуємо векторне рівняння:
18 #da/dt=f(a), f[0]=a[1], f[1]=A*sin(k*a[0]-omega*t)+a[3]*Bz
19 # f[2]=a[3], f[3]=-a[1]*Bz

```

Рис.А.2.Векторне рівняння. Тут і далі нумерація в певних видах даних буде починатись з 0, адже в мові Python нуль є першим натуральним числом.

Далі слід створити функцію  $f(t, \mathbf{a})$ , яка в кожний момент часу буде отримувати значення компонент вектора  $\mathbf{a}$  й складати їх в новостворений

numpy-масив. Функції векторного рівняння  $f[0]$  та  $f[2]$  є вже означені як компоненти  $a[1]$ ,  $a[3]$  вектора  $\mathbf{a}$ , а ось  $f[1]$  та  $f[3]$  розпишемо у визначенні функції  $f(t, \mathbf{a})$ .

```

27 def f (t, a):
28     f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz
29     f3=-a[1]*Bz
30     return np.array ([a[1],f1,a[3],f3])

```

Рис.А.3.Означення функції  $f(t, \mathbf{a})$ .

Потім слід задати початкові значення координат й швидкості для руху частинки. Відтак отримуємо також початкове значення вектора  $\mathbf{a}$ , яке рівне  $\mathbf{a}_0$ .

```

32 # Початкова умова
33 x0 = 0
34 vx0 = 1
35 y0 = 0
36 vy0 = 1
37 a0 = np.array ([x0,vx0,y0,vy0])

```

Рис.А.4. Задання початкової умови.

Також в програмі задаються константи, такі як: хвильове число  $k$ , циклічна частота  $\omega$ , амплітуда  $A$ , а також компонента  $B_z$  магнітного поля  $B$ . Їхні значення можна змінювати в залежності від досліду.

```

20
21 k=0.01
22 omega=1
23 A=0
24 Bz=1

```

Рис.А.5. Задання значень констант.

Наступним кроком є визначення функції «solution» за допомогою модуля *integ.solve\_ivp* із бібліотеки SciPy, яка обрахує нашу систему лінійних диференціальних рівнянь.

```

39 t_max = 100
40 t_points = np.linspace (0., t_max, 1000)
41 solution = integ.solve_ivp (f, t_span=(0., t_max), y0=a0,
42                             t_eval = t_points)
43 if not solution.success:
44     print ('Помилка')
45 else:
46     t_list = solution.t
47     x_list = solution.y[0,:]
48     Vx_list = solution.y[1,:]
49     y_list = solution.y[2,:]
50     Vy_list = solution.y[3,:]
51
52     E=Vx_list**2+Vy_list**2

```

Рис.А.6.Задання функції «solution».

Тут *t\_span* – це інтервал інтегрування по *t*, який починається з 0 та закінчується на *t\_max*, який рівний 100. Також аргументами функції *solution* є: функція  $f(t, \mathbf{a})$ , що була задана раніше; початковий стан  $y_0$ , який дорівнює початковому значенню  $\mathbf{a}_0$  вектора  $\mathbf{a}$  ( аргумент функції  $y_0$  не має відношення до початкової координати частинки  $y_0$ . Такою є умова задання функцій з допомогою модуля *integ.solve\_ivp* – початкове значення вектора в розв’язуваній системі диференціальних рівнянь має передаватись в аргумент  $y_0$ ); *t\_eval* – значення *t*, в яких мають зберігатись обраховані розв’язки.

Було створено списки з назвами *t\_list*, *x\_list*, *Vx\_list*, *y\_list* та *Vy\_list*, в які складаються значення відповідних змінних із вже розв’язаної системи диференціальних рівнянь. Крім того, шляхом зведення у квадрат та додавання відповідних списків значень швидкостей утворюється список значень кінетичної енергії *E* в кожен момент часу.

Після того, як програма розв’язала систему диференціальних рівнянь, будуються необхідні графіки. Алгоритм побудови є однаковим для кожного графіка. За допомогою модуля *plt.scatter* чорним кольором виділяється точка з початковими значеннями величин.

```

54     plt.style.use('fivethirtyeight')
55
56     plt.plot (t_list, x_list)
57     plt.title(f'x=f(t),A={A}, Vx0={Vx0}, Vy0={Vy0}')
58     plt.xlabel ('t')
59     plt.ylabel ('x')
60     plt.scatter(0,x0, color='black')
61     plt.show()

```

Рис.А.7. Побудова графіка  $x = f(t)$ ( Рис.5.1.1.).

```

96     fig, ax = plt.subplots()
97     plt.plot (x_list, y_list)
98     plt.title(f"Траєкторія руху частинки на OXY,A={A}, omega={omega},Vx0={Vx0},Vy0={Vy0}")
99     plt.xlabel ('x')
100    plt.ylabel ('y')
101    ratio = 1.0
102    x_left, x_right = ax.get_xlim()
103    y_low, y_high = ax.get_ylim()
104    ax.set_aspect(abs((x_right-x_left)/(y_low-y_high))*ratio)
105    plt.scatter(x0,y0,s=100, color='black')
106    plt.show()
107

```

Рис.А.8. Побудова графіка траєкторії руху частинки на  $OXY$ .

Повний текст програми поданий нижче. Якщо нам необхідно продемонструвати лише якусь частину графіків, то інші можна «закоментувати» – виділити їх таким чином, щоб програма сприймала це як коментар, а не текст коду.

```

1.     # Підключення бібліотек
2.     import matplotlib.pyplot as plt
3.     import scipy.integrate as integ
4.     import numpy as np
5.     import math
6.     # Рівняння у двовимірному просторі:
7.
8.     #B=Bz=const, Vx=Vy=0 , z=Vz=0
9.
10.    #1)dx/dt=Vx
11.    #2)dVx/dt=A*sin(kx-wt)+VyBz-VzBy=A*sin(kx-omega*t)+VyBz
12.
13.    #3)dy/dt=Vy
14.    #4)dVy/dt=VzBx-VxBz=-VxBz
15.
16.    #Вводимо вектор a[0]=x, a[1]=Vx, a[2]=y, a[3]=Vy
17.    #Отримуємо векторне рівняння:
18.    #da/dt=f(a), f[0]=a[1], f[1]=A*sin(k*a[0]-omega*t)+a[3]*Bz
19.    #           f[2]=a[3], f[3]=-a[1]*Bz

```

```

20.
21.     k=0.01
22.     omega=1
23.     A=0
24.     Bz=1
25.
26.
27.     def f (t, a):
28.         f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz
29.         f3=-a[1]*Bz
30.         return np.array ([a[1],f1,a[3],f3])
31.
32.     # Початкова умова
33.     x0 = 0
34.     Vx0 = 1
35.     y0 = 0
36.     Vy0 = 1
37.     a0 = np.array ([x0,Vx0,y0,Vy0])
38.
39.     t_max = 100
40.     t_points = np.linspace (0., t_max, 1000)
41.     solution = integ.solve_ivp (f, t_span=(0., t_max), y0=a0,
42.                                 t_eval = t_points)
43.     if not solution.success:
44.         print ('Помилка')
45.     else:
46.         t_list = solution.t
47.         x_list = solution.y[0,:]
48.         Vx_list = solution.y[1,:]
49.         y_list = solution.y[2,:]
50.         Vy_list = solution.y[3,:]
51.
52.         E=Vx_list**2+Vy_list**2
53.
54.         plt.style.use('fivethirtyeight')
55.
56.         plt.plot (t_list, x_list)
57.         plt.title(f'x=f(t),A={A}, Vx0={Vx0}, Vy0={Vy0}')
58.         plt.xlabel ('t')
59.         plt.ylabel ('x')
60.         plt.scatter(0,x0, color='black')
61.         plt.show()
62.
63.         plt.plot (t_list, Vx_list)
64.         plt.title(f'Vx=f(t),A={A},
65.                 omega={omega},Vx0={Vx0},Vy0={Vy0}')
66.         plt.xlabel ('t')
67.         plt.ylabel ('Vx')
68.         plt.scatter(0,Vx0, color='black')
69.         plt.show()

```

```

69.
70.     plt.plot (t_list, y_list)
71.     plt.title(f'y=f(t),A={A}, Vx0={Vx0}, Vy0={Vy0}')
72.     plt.xlabel ('t')
73.     plt.ylabel ('y')
74.     plt.scatter(0,y0, color='black')
75.     plt.show()
76.
77.     plt.plot (t_list, Vy_list)
78.     plt.title(f'Vy=f(t),A={A}, Vx0={Vx0}, Vy0={Vy0}')
79.     plt.xlabel ('t')
80.     plt.ylabel ('Vy')
81.     plt.scatter(0,Vy0, color='black')
82.     plt.show()
83.
84.     fig, ax = plt.subplots()
85.     plt.plot (x_list, Vx_list)
86.     plt.title(f'Vx=f(x),A={A},Vx0={Vx0}, Vy0={Vy0}')
87.     plt.xlabel ('x')
88.     plt.ylabel ('Vx')
89.     ratio = 1.0
90.     x_left, x_right = ax.get_xlim()
91.     y_low, y_high = ax.get_ylim()
92.     ax.set_aspect(abs((x_right-x_left)/(y_low-y_high))*ratio)
93.     plt.scatter(0,1,s=100, color='black')
94.     plt.show()
95.
96.     fig, ax = plt.subplots()
97.     plt.plot (x_list, y_list)
98.     plt.title(f"Траєкторія руху частинки на OXY,A={A},
99.             omega={omega},Vx0={Vx0},Vy0={Vy0}")
100.    plt.xlabel ('x')
101.    plt.ylabel ('y')
102.    ratio = 1.0
103.    x_left, x_right = ax.get_xlim()
104.    y_low, y_high = ax.get_ylim()
105.    ax.set_aspect(abs((x_right-x_left)/(y_low-y_high))*ratio)
106.    plt.scatter(x0,y0,s=100, color='black')
107.    plt.show()
108.
109.    fig, ax = plt.subplots()
110.    plt.plot(t_list,E)
111.    plt.title(f'E=f(t),A={A},Vx0={Vx0}, Vy0={Vy0}')
112.    plt.xlabel('t')
113.    plt.ylabel('E')
114.    plt.scatter(0,2,s=100, color='black')
115.    plt.show()

```

## ДОДАТОК Б. НАПИСАННЯ ПРОГРАМИ ДЛЯ ВИВЧЕННЯ ЗАЛЕЖНОСТІ ПАРАМЕТРІВ РУХУ ЧАСТИНКИ ВІД АМПЛІТУДИ

Для того, щоб дослідити залежність радіуса спіралі, по якій рухається частинка, а також швидкості та зміни енергії від амплітуди електростатичної хвилі, було створено окрему програму. Однак, велика її частина базується на програмі з додатку А.

Однією з головних відмінностей є присутність циклу `for`, що дозволяє повторювати основний код програми для всіх шести значень амплітуди, які є в цьому досліді. Крім того, були створені списки `Ampls`, `E_list`, `V_list` для значень відповідних величин, які необхідно буде виводити на графік.

```

22     Ampls=[]
23     E_list=[]
24     V_list=[]
25     for i in range(6):
26         Ampls.append(0.005*i)
27
28     for A in Ampls:
29         k=0.01
30         omega=1
31         Bz=1
32
33         def f (t, a):
34             f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz
35             f3=-a[1]*Bz
36             return np.array ([a[1],f1,a[3],f3])
37
38         # Початкова умова
39         x0 = 0
40         Vx0 = 0
41         y0 = 0
42         Vy0 = 0
43         a0 = np.array ([x0,Vx0,y0,Vy0])

```

Рис.Б.1. Цикл `for` для різних значень амплітуди.

Для кращої візуалізації квадратичної залежності енергії частинки від амплітуди на графік була нанесена проста парабола  $y = cx^2$ .

```

82     def parabola (x, c):
83         return c * x**2
84
85     fitted_coef, covariance = opt.curve_fit (parabola, Ampls, E_list)
86     print ('fitted coefficient', fitted_coef)
87
88     Ampls_for_fit = np.linspace(Ampls[0], Ampls[-1], 100)
89     E_fit = [parabola(A, *fitted_coef) for A in Ampls_for_fit]
90
91     plt.plot(Ampls,E_list,color='magenta', marker='x')
92     plt.plot(Ampls_for_fit, E_fit, color='green')
93     plt.title('E=f(A),значення енергії - колір "маджента"\nпарабола - зелений колір')
94     plt.xlabel('Amplitude')
95     plt.ylabel('Energy')
96     plt.show()
97
98     plt.plot(Ampls,V_list,color='red')
99     plt.title('V=f(A)')
100    plt.xlabel('Amplitude')
101    plt.ylabel('V')
102    plt.show()
103

```

Рис.Б.2. Графіки залежності зміни енергії та швидкості руху частинки від амплітуди.

```

1.     # Підключення бібліотек
2.     import matplotlib.pyplot as plt
3.     import scipy.integrate as integ
4.     import scipy.optimize as opt
5.     import numpy as np
6.     import math
7.     # Рівняння у 4-х вимірному фазовому просторі:
8.
9.     #B=Bz=const, Vx=Vy=0 , z=Vz=0
10.
11.    #1)dx/dt=Vx
12.    #2)dVx/dt=A*sin(kx-wt)+VyBz-VzBy=A*sin(kx-omega*t)+VyBz
13.
14.    #3)dy/dt=Vy
15.    #4)dVy/dt=VzBx-VxBz=-VxBz
16.
17.    #Вводимо вектор a[0]=x, a[1]=Vx, a[2]=y, a[3]=Vy
18.    #Отримуємо векторне рівняння:
19.    #da/dt=f(a), f[0]=a[1], f[1]=A*sin(k*a[0]-omega*t)+a[3]*Bz
20.    #           f[2]=a[3], f[3]=-a[1]*Bz
21.
22.    Ampls=[]
23.    E_list=[]
24.    V_list=[]
25.    for i in range(6):
26.        Ampls.append(0.005*i)
27.
28.    for A in Ampls:
29.        k=0.01
30.        omega=1
31.        Bz=1
32.
33.        def f (t, a):
34.            f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz
35.            f3=-a[1]*Bz
36.            return np.array ([a[1],f1,a[3],f3])
37.
38.        # Початкова умова
39.        x0 = 0
40.        Vx0 = 0
41.        y0 = 0
42.        Vy0 = 0
43.        a0 = np.array ([x0,Vx0,y0,Vy0])
44.
45.        t_max = 100
46.        t_points = np.linspace (0., t_max, 1000)
47.        solution = integ.solve_ivp (f, t_span=(0., t_max), y0=a0,
48.                                   t_eval = t_points)
49.        if not solution.success:
50.            print ('Помилка')

```

```

51.         else:
52.             t_list = solution.t
53.             x_list = solution.y[0,:]
54.             Vx_list = solution.y[1,:]
55.             y_list = solution.y[2,:]
56.             Vy_list = solution.y[3,:]
57.
58.
59.             E = Vx_list[-1]**2 + Vy_list[-1]**2
60.
61.             E_list.append(E)
62.             V_list.append(math.sqrt(E)) # Модуль швидкості
63.
64.
65.             plt.style.use('fivethirtyeight')
66.
67.             fig, ax = plt.subplots()
68.             plt.plot (x_list, y_list)
69.             plt.title(f"Траєкторія руху частинки на OXY,A={A},
70.             omega={omega},Vx0={Vx0},Vy0={Vy0}")
71.             plt.xlabel ('x')
72.             plt.ylabel ('y')
73.             ratio = 1.0
74.             x_left, x_right = ax.get_xlim()
75.             y_low, y_high = ax.get_ylim()
76.             ax.set_aspect(abs((x_right-x_left)/(y_low-
77.             y_high))*ratio)
78.             plt.scatter(x0,y0,s=100, color='black')
79.             plt.show()
80.
81.             print (Ampls)
82.             print (E_list)
83.
84.             def parabola (x, c):
85.                 return c * x**2
86.
87.             fitted_coef, covariance = opt.curve_fit (parabola, Ampls,
88.             E_list)
89.             print ('fitted coefficient', fitted_coef)
90.
91.             Ampls_for_fit = np.linspace(Ampls[0], Ampls[-1], 100)
92.             E_fit = [parabola(A, *fitted_coef) for A in Ampls_for_fit]
93.
94.             plt.plot(Ampls_for_fit, E_fit, linestyle='dashed',
95.             linewidth='8')
96.             plt.plot(Ampls,E_list,color='black', marker='x',linewidth='4'
97.             )
98.             plt.title('delta E=f(A),зміна енергії - чорна крива\нпарабола
99.             - синя пунктирна крива')
100.            plt.xlabel('Amplitude')

```

```
95. plt.ylabel('delta Energy')
96. plt.show()
97.
98. plt.plot(Ampls,V_list,color='red')
99. plt.title('V=f(A)')
100. plt.xlabel('Amplitude')
101. plt.ylabel('V')
102. plt.show()
```

## ДОДАТОК В. НАПИСАННЯ ПРОГРАМИ ДЛЯ ТРИВИМІРНОГО ПРОСТОРУ З ОДНОРІДНИМ МАГНІТНИМ ПОЛЕМ

Система рівнянь, вектор  $\mathbf{a}$ , векторне рівняння  $\frac{d\mathbf{a}}{dt} = f(\mathbf{a})$  та функція  $f(t, \mathbf{a})$  будуть відповідним чином видозмінені, із врахування координати  $z$  та швидкості  $V_z$ .

```

8 #1)dx/dt=Vx
9 #2)dVx/dt=A*sin(kx-wt)+VyBz-VzBy
10
11 #3)dy/dt=Vy
12 #4)dVy/dt=VzBx-VxBz
13
14 #5)dz/dt=Vz
15 #6)dVz/dt=VxBy-VyBx
16
17 #Вводимо вектор a[0]=x, a[1]=Vx, a[2]=y, a[3]=Vy, a[4]=z, a[5]=Vz
18 #Отримуємо векторне рівняння:
19 #da/dt=f(a), f[0]=a[1], f[1]=A*sin(ka[0]-wt)+a[3]Bz-a[5]By
20 # f[2]=a[3], f[3]=a[5]Bx-a[1]Bz
21 # f[4]=a[5], f[5]=a[1]By-a[3]Bx

```

Рис.В.1. Система рівнянь та векторне рівняння у тривимірному просторі.

```

31 def f (t, a):
32     f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz-a[5]*By
33     f3=a[5]*Bx-a[1]*Bz
34     f5=a[1]*By-a[3]*Bx
35     return np.array ([a[1],f1,a[3],f3,a[5],f5])

```

Рис.В.2. Означення функції  $f(t, \mathbf{a})$ .

Попри те, що у випадку однорідного магнітного поля  $B_x$  та  $B_y$  рівні нулю (відповідні доданки у функціях  $f_1, f_3, f_5$  будуть занулятися), для зручності вони були запрограмовані у функцію  $f(t, \mathbf{a})$ , тому що в наступному досліді буде розглядатися неоднорідне магнітне поле (див. додаток Г). Також координата  $Z$  та швидкість  $V_z$  додаються у початкове значення  $\mathbf{a}_0$  вектора  $\mathbf{a}$ . Функція «solution» програмується аналогічно до двовимірного простору, при цьому слід враховувати те, що тепер додається координата  $z$  та швидкість  $V_z$ .

```

36 #Початкова умова
37 x0 = 0
38 Vx0 = 1
39 y0 = 0
40 Vy0 = 1
41 z0=0
42 Vz0=1
43 a0 = np.array ([x0,Vx0,y0,Vy0,z0,Vz0])
44 t_max = 100
45 t_points = np.linspace (0., t_max, 1000)

```

Рис.В.3. Початкова умова.

Графік руху частинки у просторі  $OXYZ$  будується наступним чином.

```

63 fig = plt.figure()
64 ax = fig.add_subplot(111, projection='3d')
65 ax.plot(x,y,z)
66 plt.title(f'Траекторія руху частинки на OXYZ, A={A}, Vx0={Vx0}, Vy0={Vy0}, Vz0={Vz0}')
67 plt.xlabel('x')
68 plt.ylabel('y')
69 ax.scatter3D(0,0,0, color='black')
70 plt.show()
71 plt.plot(t_list,E)
72 plt.xlabel('t')
73 plt.ylabel('E')
74 plt.show()

```

Рис.Б.4. Побудова графіка на рис.5.2.1..

```

1. # Підключення бібліотек
2. import matplotlib.pyplot as plt
3. import scipy.integrate as integ
4. import numpy as np
5. import math
6. # Рівняння:
7.
8. #1)dx/dt=Vx
9. #2)dVx/dt=A*sin(kx-wt)+VyBz-VzBy
10.
11. #3)dy/dt=Vy
12. #4)dVy/dt=VzBx-VxBz
13.
14. #5)dz/dt=Vz
15. #6)dVz/dt=VxBz-VyBx
16.
17. #Вводимо вектор a[0]=x, a[1]=Vx, a[2]=y, a[3]=Vy, a[4]=z,
    a[5]=Vz
18. #Отримуємо векторне рівняння:
19. #da/dt=f(a), f[0]=a[1], f[1]=A*sin(ka[0]-wt)+a[3]Bz-a[5]By
20. # f[2]=a[3], f[3]=a[5]Bx-a[1]Bz
21. # f[4]=a[5], f[5]=a[1]By-a[3]Bx
22.
23. By=0

```

```

24.     k=0.01
25.     omega=1
26.     A=0.05
27.     Bx=0
28.     Bz=1
29.
30.
31.     def f (t, a):
32.         f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz-a[5]*By
33.         f3=a[5]*Bx-a[1]*Bz
34.         f5=a[1]*By-a[3]*Bx
35.         return np.array ([a[1],f1,a[3],f3,a[5],f5])
36.
37.     #Початкова умова
38.     x0 = 0
39.     vx0 = 1
40.     y0 = 0
41.     vy0 = 1
42.     z0=0
43.     vz0=1
44.     a0 = np.array ([x0,vx0,y0,vy0,z0,vz0])
45.     t_max = 100
46.     t_points = np.linspace (0., t_max, 1000)
47.
48.     solution = integ.solve_ivp (f, t_span=(0.,t_max), y0=a0,
49.                                 t_eval = t_points)
50.     if not solution.success:
51.         print ('Щось пішло не так')
52.     else:
53.         t_list = solution.t
54.         x = solution.y[0,:]
55.         vx0_list = solution.y[1,:]
56.         y = solution.y[2,:]
57.         vy0_list = solution.y[3,:]
58.         z = solution.y[4,:]
59.         vz0_list = solution.y[5,:]
60.
61.         E=vx0_list**2+vy0_list**2+vz0_list**2
62.
63.         fig = plt.figure()
64.         ax = fig.add_subplot(111, projection='3d')
65.         ax.plot(x,y,z)
66.         plt.title(f'Траєкторія руху частинки на OXYZ, A={A},
67.                 Vx0={Vx0}, Vy0={Vy0}, Vz0={Vz0}')
68.         plt.xlabel('x')
69.         plt.ylabel('y')
70.         ax.scatter3D(0,0,0, color='black')
71.         plt.show()
72.         plt.plot(t_list,E)
73.         plt.xlabel('t')

```

```
73.     plt.ylabel('E')  
74.     plt.show()
```

## ДОДАТОК Г.НАПИСАННЯ ПРОГРАМИ ДЛЯ ТРИВИМІРНОГО ПРОСТОРУ З НЕОДНОРІДНИМ МАГНІТНИМ ПОЛЕМ

Оскільки магнітне поле тепер неоднорідне, то слід задавати не лише компоненту  $B_x$  магнітного поля як функцію від  $\mathbf{a}$ , а й компоненту  $B_z$ .

```

23 By=0
24 k=0.01
25 omega=1
26 A=0.002
27 Bz0=1
28 alpha=1/600
29
30 ▾ def Bx(a):
31     Bx=-Bz0*alpha*a[0]
32     return Bx
33
34 ▾ def Bz(a):
35     Bz=Bz0*(1+alpha*a[4])
36     return Bz

```

Рис. Г.1.Визначення компонент магнітного поля  $B_x$ ,  $B_z$  як функцій від  $\mathbf{a}$ , а також задання сталих величин.

Окрім того, ключовою відмінністю цієї програми від попередніх є те, що тепер початкові швидкості  $V_{x0}$ ,  $V_{y0}$  будуть не числами, які можна задати власноруч, а тригонометричними функціями від кута  $\zeta$ . Кут  $\zeta$  в даному досліді (тривимірний простір з неоднорідним магнітним полем) – це кут між напрямом швидкості  $V_{\perp}$  (про яку піде мова нижче) та віссю  $OX$ . Для даного досліді ми візьмемо 25 значень кута в радіанах (25 для того, щоб повністю замкнути коло), а кожен кут буде визначатись за формулою  $\frac{\pi}{12i}$ . Відповідно, в одному досліді буде проведено 25 «запусків» частинки. Це можна зробити, взявши частину програми, в якій задається функція «solution», в цикл for (див. рис. Г.2.). Крім того, для того, щоб отримати значення стрибків енергій ( $\delta E_{pract}$ ) під час практичної частини досліді, слід:

- створити масив значень потенційних енергій  $E$ , де кожне значення  $E$  дорівнює сумі квадратів відповідних значень швидкостей з масивів  $V_{x0\_list}$ ,  $V_{y0\_list}$ ,  $V_{z0\_list}$  (див.рис.Г.3.);

- отримати перші 20 значень потенційної енергії і знайти їхнє середнє;
- аналогічно, знайти середнє значення енергії для останніх двадцяти величин;
- знайти різницю між даними середніми значеннями – це і буде  $\delta E_{pract}$ ;
- додати отримане значення  $\delta E_{pract}$  в список  $\delta E_{pract\_list}$ , де будуть зберігатись ці значення.

Оскільки ця частина запрограмована в циклі `for`, то вона буде виконуватись для всіх 25 значень  $\zeta$ , які ми беремо в цьому досліді.

```

44 zetas=[]
45 for i in range(25):
46     zetas.append(math.pi/12*i)
47
48 deltaE_pract_list=[]
49 deltaE_teor_list=[]
50 arctg_list=[]
51 Vperp_list=[]
52 T_list=[]
53 vz_list=[]
54
55 ▼ for zeta in zetas:
56     # Початкові значення
57     x0 = 0
58     vx0 = math.sin(zeta)
59     y0 = 0
60     vy0 = math.cos(zeta)
61     z0=300
62     vz0=-1
63     a0 = np.array ([x0,vx0,y0,vy0,z0,vz0])
64     t_max = 600
65     t_points = np.linspace (0., t_max, 6000)
66
67     solution = integ.solve_ivp (f, t_span=(0.,t_max), y0=a0,
68                               t_eval = t_points)
69     if not solution.success:

```

Рис. Г.2. Використання циклу `for` для різних «запусків» частинки в залежності від значень кута  $\zeta$ .

```

79
80 E=Vx0_list**2+Vy0_list**2+Vz0_list**2
81
82 E_first20_avg=sum(E[:20])/20
83 E_last20_avg=sum(E[-20:])/20
84 deltaE_pract=(E_first20_avg-E_last20_avg)
85 deltaE_pract_list.append(deltaE_pract)
86

```

Рис. Г.3. Отримання списку  $\delta E_{pract\_list}$ .

Фінальним етапом програмування даного досліді є отримання списку  $\delta E_{teor\_list}$  – тобто списку теоретичних значень, отриманих за формулою 4.2.1. Для того, щоб отримати значення змінних для підстановки у формулу, необхідно (див. рис. Г.4.):

- знайти номер, під яким у масиві  $z$  розв'язків функції «solution» стоїть таке значення  $z$ , квадрат якого є найменшим;
- взяти значення  $V_x, V_y, V_z$ , із списків  $V_{x0\_list}, V_{y0\_list}, V_{z0\_list}$ , які мають однаковий індекс з числом  $z$  з попереднього пункту;
- знайти  $\arctg \frac{V_y}{V_x}$ , а також  $V_{\perp} = \sqrt{V_x^2 + V_y^2}$ ;
- підставити відповідні значення у формулу.

```

87     z_min_index=np.argmin(z**2)
88     vx=Vx0_list[z_min_index]
89     vy=Vy0_list[z_min_index]
90     vz=Vz0_list[z_min_index]
91     vz_list.append(vz)
92     t=t_list[z_min_index]
93     T_list.append(t)
94     arctg=math.atan2(vy,vx)
95     arctg_list.append(arctg)
96     #дзета - це кут між напрямком vперп та віссю OX
97     Vperp=math.sqrt(vx**2+vy**2)
98     Vperp_list.append(Vperp)
99
100    deltaE_teor=A*Vperp*math.sqrt(abs((2*math.pi)/(alpha*vz))*math.sin(arctg+omega*t+math.pi/4))
101    deltaE_teor_list.append(deltaE_teor)

```

Рис.Г.4.Отримання списку  $\delta E\_teor\_list$ .

Врешті-решт, списки значень  $\delta E$ , отриманих теоретичним та практичним виводяться на один графік ( Рис.5.2.6), що дозволяє їх порівняти і визначити, наскільки точним є кожен з цих способів.

```

103    fig = plt.figure()
104    ax = fig.add_subplot(111, projection='3d')
105    ax.plot(x,y,z)
106    plt.title(f"Траєкторія частинки на OXYZ, A={A},dzeta={zeta} рад")
107    plt.xlabel('x')
108    plt.ylabel('y')
109    ax.scatter3D(x0,y0,z0, color='black')
110    plt.show()
111
112    plt.plot(t_list,E)
113    plt.title(f"E=f(t), A={A}, dzeta={zeta} рад")
114    plt.xlabel('t')
115    plt.ylabel('E')
116    plt.show()
117
118    plt.plot(zetas,deltaE_teor_list,color='red')
119    plt.plot(zetas,deltaE_pract_list,color='blue')
120    plt.title('deltaE_teor=f(dzeta), колір червоний\n deltaE_pract=f(dzeta), колір синій')
121    plt.xlabel('dzeta,рад')
122    plt.ylabel('delta Energy')
123    plt.show()

```

Рис.Г.5.Побудова графіків.

```

1.     # Підключення бібліотек
2.     import matplotlib.pyplot as plt
3.     import scipy.integrate as integ
4.     import numpy as np
5.     import math
6.     # Рівняння:
7.
8.     #1)dx/dt=Vx
9.     #2)dVx/dt=A*sin(kx-wt)+VyBz-VzBy
10.
11.    #3)dy/dt=Vy
12.    #4)dVy/dt=VzBx-VxBz
13.
14.    #5)dz/dt=Vz
15.    #6)dVz/dt=VxBz-VyBx
16.
17.    #Вводимо вектор a[0]=x, a[1]=Vx, a[2]=y, a[3]=Vy, a[4]=z,
18.    a[5]=Vz
19.    #Отримуємо векторне рівняння:
20.    #da/dt=f(a), f[0]=a[1], f[1]=A*sin(ka[0]-wt)+a[3]Bz-a[5]By
21.    #           f[2]=a[3], f[3]=a[5]Bx-a[1]Bz
22.    #           f[4]=a[5], f[5]=a[1]By-a[3]Bx
23.
24.    By=0
25.    k=0.01
26.    omega=1
27.    A=0.002
28.    Bz0=1
29.    alpha=1/600
30.
31.    def Bx(a):
32.        Bx=-Bz0*alpha*a[0]
33.        return Bx
34.
35.    def Bz(a):
36.        Bz=Bz0*(1+alpha*a[4])
37.        return Bz
38.
39.    def f (t, a):
40.        f1=A*math.sin(k*a[0]-omega*t)+a[3]*Bz(a)-a[5]*By
41.        f3=a[5]*Bx(a)-a[1]*Bz(a)
42.        f5=a[1]*By-a[3]*Bx(a)
43.        return np.array ([a[1],f1,a[3],f3,a[5],f5])
44.
45.    zetas=[]
46.    for i in range(25):

```

```

46.         zetas.append(math.pi/12*i)
47.
48.     deltaE_pract_list=[]
49.     deltaE_teor_list=[]
50.     arctg_list=[]
51.     Vperp_list=[]
52.     T_list=[]
53.     vz_list=[]
54.
55.     for zeta in zetas:
56.         # Початкові значення
57.         x0 = 0
58.         Vx0 = math.sin(zeta)
59.         y0 = 0
60.         Vy0 = math.cos(zeta)
61.         z0=300
62.         Vz0=-1
63.         a0 = np.array ([x0,Vx0,y0,Vy0,z0,Vz0])
64.         t_max = 600
65.         t_points = np.linspace (0., t_max, 6000)
66.
67.         solution = integ.solve_ivp (f, t_span=(0.,t_max), y0=a0,
68.                                     t_eval = t_points)
69.         if not solution.success:
70.             print ('Щось пішло не так')
71.         else:
72.             t_list = solution.t
73.             x = solution.y[0,:]
74.             Vx0_list = solution.y[1,:]
75.             y = solution.y[2,:]
76.             Vy0_list = solution.y[3,:]
77.             z = solution.y[4,:]
78.             Vz0_list = solution.y[5,:]
79.
80.             E=Vx0_list**2+Vy0_list**2+Vz0_list**2
81.
82.             E_first20_avg=sum(E[:20])/20
83.             E_last20_avg=sum(E[-20:])/20
84.             deltaE_pract=(E_first20_avg-E_last20_avg)
85.             deltaE_pract_list.append(deltaE_pract)
86.
87.             z_min_index=np.argmin(z**2)
88.             vx=Vx0_list[z_min_index]
89.             vy=Vy0_list[z_min_index]
90.             vz=Vz0_list[z_min_index]
91.             vz_list.append(vz)
92.             t=t_list[z_min_index]
93.             T_list.append(t)
94.             arctg=math.atan2(vy,vx)
95.             arctg_list.append(arctg)

```

```

96.         #дзета - це кут між напрямком vперп та віссю OX
97.         Vperp=math.sqrt(vx**2+vy**2)
98.         Vperp_list.append(Vperp)
99.
100.        deltaE_teor=A*Vperp*math.sqrt(abs((2*math.pi)/(alpha*v
        z)))*math.sin(arctg+omega*t+math.pi/4)
101.        deltaE_teor_list.append(deltaE_teor)
102.
103.        fig = plt.figure()
104.        ax = fig.add_subplot(111, projection='3d')
105.        ax.plot(x,y,z)
106.        plt.title(f'Траєкторія частинки на OXYZ,
        A={A},dzeta={zeta} рад')
107.        plt.xlabel('x')
108.        plt.ylabel('y')
109.        ax.scatter3D(x0,y0,z0, color='black')
110.        plt.show()
111.
112.        plt.plot(t_list,E)
113.        plt.title(f»E=f(t), A={A}, dzeta={zeta} рад»)
114.        plt.xlabel('t')
115.        plt.ylabel('E')
116.        plt.show()
117.
118.        plt.plot(zetas,deltaE_teor_list,color='red')
119.        plt.plot(zetas,deltaE_pract_list,color='blue')
120.        plt.title('deltaE_teor=f(dzeta), колір червоний\n
        deltaE_pract=f(dzeta), колір синій')
121.        plt.xlabel('dzeta,рад')
122.        plt.ylabel('delta Energy')
123.        plt.show()

```