

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет **ФИЗИЧЕСКИЙ**

Кафедра **ФИЗИКИ ПЛАЗМЫ**

Направление подготовки **03.03.02 ФИЗИКА**

Образовательная программа: **БАКАЛАВРИАТ**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Эккарт Арины

Тема работы: Влияние нелинейной динамики на равновесный эмиттанс пучка в источнике синхротронного излучения СКИФ

«К защите допущена»

Заведующий кафедрой

Беклемишев А. Д.

к. м.-ф. н.

ИЯФ СО РАН

Беклемишев А. Д./.....
(фамилия И., О.) / (подпись, МП)

«.....».....20...г.

Научный руководитель

Богомяков А.В.

к. м.-ф. н., звание

в. н. с. ИЯФ СО РАН

Богомяков А. В./.....
(фамилия И., О.) / (подпись, МП)

«.....».....20...г.

Дата защиты: «.....».....20...г.

Содержание

1. Введение	3
2. Теоретическая часть	4
2.1. Оптика ускорителя.....	4
2.1.1. Диполь.....	4
2.1.2. Квадруполь	5
2.1.3. Секступоль	6
2.2. Статистическое определение эмиттанса	8
2.3. Функции искажения (Distortion functions)	9
3. Практическая часть.....	14
3.1. Линейная динамика в FODO-ячейке	14
3.2. Нелинейная динамика в FODO-ячейке.....	15
3.3. Программы для подсчета эмиттанса.....	16
3.4. Ячейка СКИФ.....	17
4. Результаты	18
4.1. Сравнение линейной и нелинейной динамики FODO	18
4.2. Нелинейная динамика СКИФ.....	19
5. Обсуждение результатов.....	21
6. Заключение	23
7. Список литературы	24
8. Приложения	24
8.1. Приложение 1	24
8.2. Приложение 2	26
8.3. Приложение 3	27
8.4. Приложение 4.....	31
8.5. Приложение 5.....	37

1. Введение

Объектом исследования этой работы стала нелинейная динамика СКИФ и ее влияние на эмиттанс пучка в синхротроне.

Синхротронное излучение (СИ) испускается движущимися с высокой скоростью заряженными частицами в точках, где их траектория искривляется с помощью магнитных полей (Рис. 1) [1].

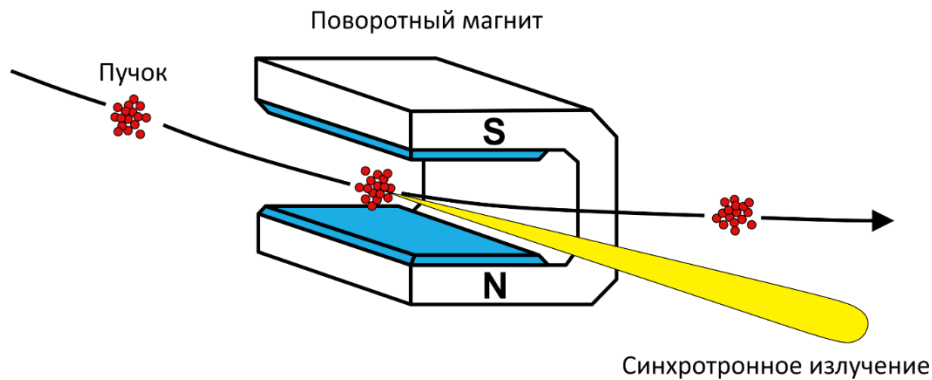


Рис. 1. Возникновение СИ при пролете пучка электронов через поворотный магнит.

Важность СИ в науке связана с его уникальными свойствами. Во-первых, оно имеет очень широкий спектр, включая в себя длины волн от инфракрасного излучения до гамма-лучей ($\sim 10^{-4} - 10^{-22}$ м). Это позволяет выбирать нужный для работы диапазон. Во-вторых, излучение является хорошо сфокусированным, так что узкий луч попадает точно в образец, установленный на большом расстоянии от источника. В-третьих, СИ имеет высокую интенсивность. Например, яркость излучения рентгеновского диапазона СИ по яркости в триллионы раз превосходит традиционные источники — рентгеновские трубки [2]. Эти особенности делают СИ востребованным во многих направлениях науки таких, как физика, химия, биотехнологии, археология, материаловедение, медицина и т.д.

Одним из наиболее важных параметров СИ является его спектральная яркость. Современные синхротроны создаются с целью его повышения. Есть несколько известных способов сделать излучения синхротрона ярче (Рис. 2) [3]. В данной работе был выбран путь, выделенный красным цветом на схеме.

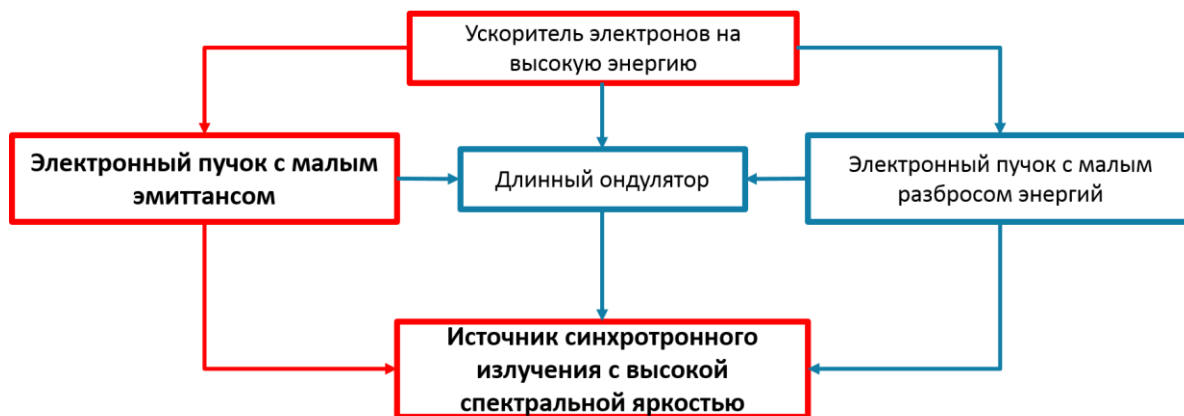


Рис. 2. Пути создания источника СИ с высокой спектральной яркостью.

Прежде чем добиваться малости эмиттанса, было решено сначала исследовать нелинейную динамику в СКИФ и как она влияет на изменение эмиттанса электронного пучка в ходе его движения в синхротроне.

2. Теоретическая часть

2.1. Оптика ускорителя

Замкнутая орбита задаётся дипольными магнитами, а устойчивость движения определяется фокусирующими линзами. Набор дипольных магнитов и квадрупольных линз, обеспечивающих замкнутую и устойчивую орбиту называется оптической структурой (оптикой) ускорителя (Рис. 3) [5].

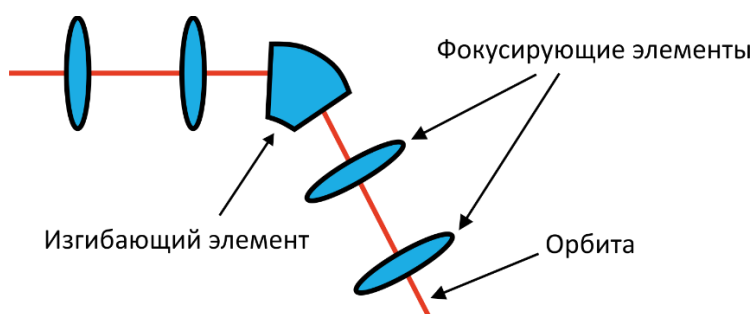


Рис. 3. Оптика пучка.

2.1.1. Диполь

Дипольные магниты используются для изгиба и поддержания пути ускорения пучка частиц [4].

В дипольном магните с однородным магнитным полем B_0 на движущуюся заряженную частицу действует сила Лоренца (Рис. 4) [5]:

$$F = \frac{e}{c} [v \times B_0].$$

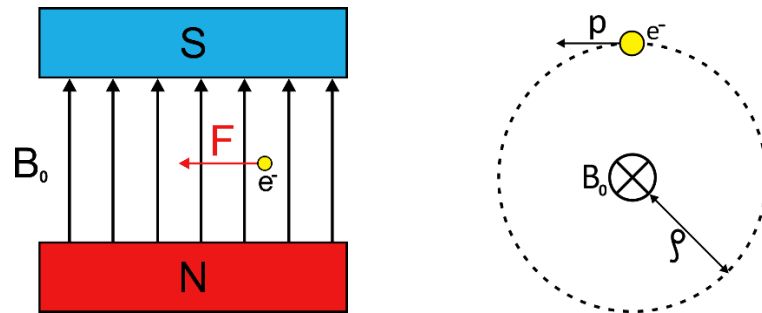


Рис. 4. Изгиб траектории электрона под действием созданного диполем магнитного поля.

На данном рисунке предполагается, что электрон летит «в страницу».

Под действием этой силы частица отклоняется на угол $\varphi = \frac{eB_0L}{pc}$, где L – длина магнита. Радиус поворота частицы: $\rho = \frac{pc}{eB_0}$.

2.1.2. Квадруполь

Квадрупольные магниты создают фокусирующую и дефокусирующую силу, пропорциональную расстоянию от центра. Этот магнит имеет 4 полюса (Рис. 5) [5]:

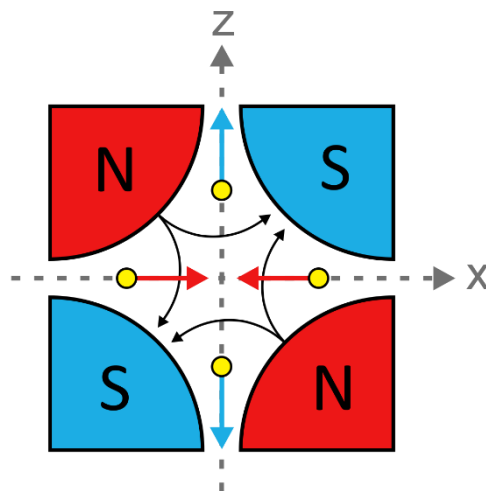


Рис. 5. Фокусировка пучка с помощью квадрупольного магнита.

Рассмотрим электрон, летящий перпендикулярно плоскости XZ «в страницу». Согласно правилу правой руки, силы, действующие на частицы с правой и левой сторон магнита, направлены в центр магнита, а силы, действующие на верхнюю и нижнюю частицы (относительно центра), – от центра. Данный магнит, таким образом, горизонтально фокусирует, а вертикально – расфокусирует. Противоположно направленные силы Лоренца имеют место, когда направление тока, заряд частицы или вектор ее движения обратны [4].

Поле является линейным по отклонению от оси:

$$B_x = -Gz, \quad B_z = -Gx,$$

где G – градиент магнитного поля.

Отношение градиента магнитного поля к импульсу частицы называется силой квадруполя [5]:

$$k_1 [\text{м}^{-2}] = \frac{eG}{pc}.$$

2.1.3. Секступоль

Секступольный магнит состоит из шести магнитных полюсов, расположенных в виде чередующихся северного и южного полюсов (Рис. 6) [5]. Он необходим для коррекции хроматизма квадруполя.

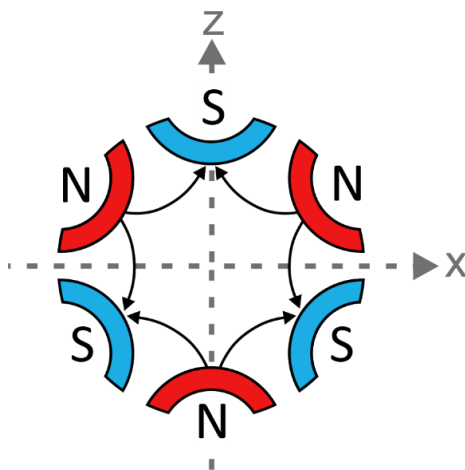


Рис. 6. Упрощенная схема секступольного магнита.

k_1 , сила квадруполя, обратно пропорциональна pc , поэтому фокусирующая сила зависит от энергии частицы (хроматизм). Для его компенсации нужны секступоли.

Рассмотрим квадруполь, фокусирующий пучок электронов в направлении X к фокальной точке. Если бы некоторые из электронов имели более высокие энергии, их траектории были бы меньше изогнуты квадрупольным полем, а их фокальные точки были бы дальше (Рис. 7). То есть, фокусное расстояние линзы отличается для электронов с разными энергиями [6].

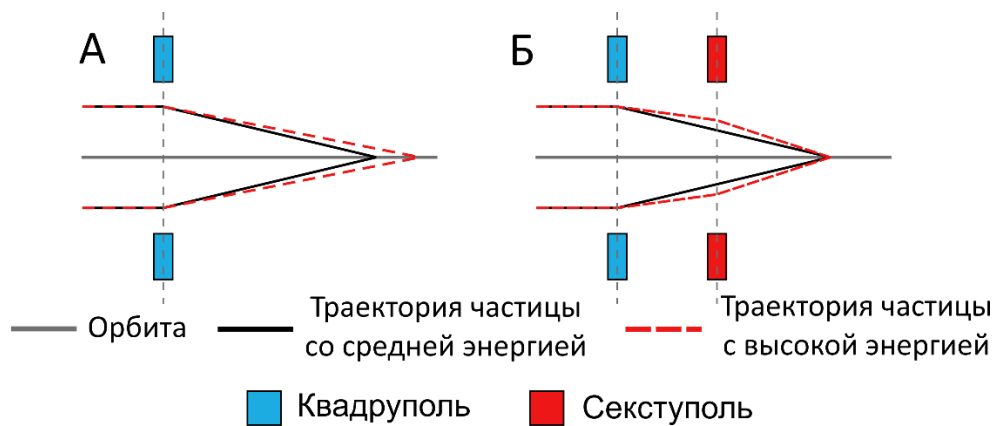


Рис. 7. Хроматизм (А) и исправление хроматизма с помощью секступоля (Б).

Чтобы компенсировать избыточную или недостаточную фокусировку, используют секступольный магнит. Секступоли ставят в места с дисперсионной функцией, чтобы компенсировать хроматизм фокусировки квадруполей. Но секступоли создают дополнительные нелинейные члены, которые возмущают линейное движение [6]:

$$B_x = Sxz, \quad B_z = \frac{1}{2}S(x^2 - z^2).$$

Через коэффициент S выражается сила секступоля [5]:

$$k_2[\text{м}^{-3}] = \frac{eS}{pc}.$$

2.2. Статистическое определение эмиттанса

Эмиттанс (ϵ_x) обычно определяют как площадь эллипса в фазовом пространстве, содержащего около 95% всех частиц внутри. Он же является произведением двух полуосей эллипса. Если эллипс находится в вертикальном положении, его оси совпадают с горизонтальной и вертикальной осями. Две полуоси - половина размера пучка (Δx) и расходимости пучка ($\Delta x'$) соответственно (Рис. 8) [7].

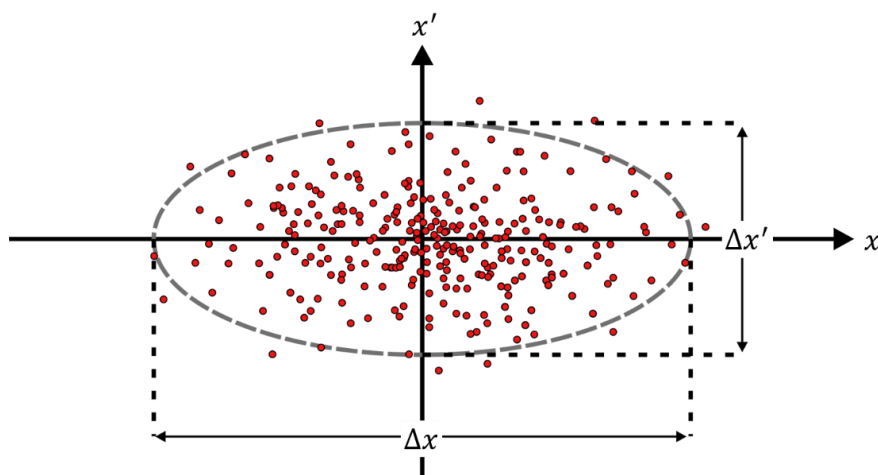


Рис. 8. Распределение электронов пучка в фазовом пространстве.

Рассматривая пучок как статистический набор точек в двумерном фазовом пространстве, эмиттанс пучка является мерой их разброса в этой плоскости. Он аналогичен стандартному отклонению, которое измеряет дисперсию набора точек на линии:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}.$$

Аналогично, эмиттанс пучка должен быть «среднеквадратичной» величиной, т.е. функцией моментов второго порядка распределения точек [7].

Если в начальный момент координаты и импульсы частиц распределены по Гауссу (1) с заданным начальным эмиттансом и параметрами Твисса (Рис.

9), то можно легко сосчитать $\overline{x^2}$, $\overline{p_x^2}$ и $\overline{xp_x}$, которые будут выражаться через эмиттанс и параметры Твисса:

$$f(x, p_x) = \frac{1}{2\pi\varepsilon} e^{-\frac{(\alpha x + \beta p_x)^2 + x^2}{2\beta\varepsilon}}, \quad (1)$$

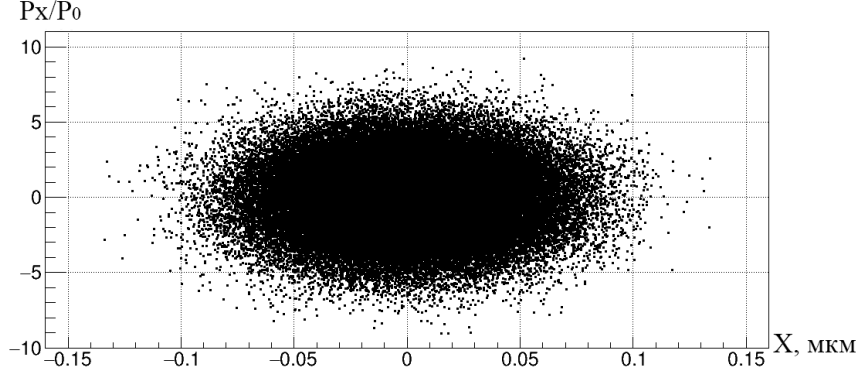


Рис. 9. Начальное распределение частиц в фазовом пространстве.

$$\overline{x^2} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^2 f(x, p_x) dx dp_x = \varepsilon\beta, \quad (2)$$

$$\overline{p_x^2} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} p_x^2 f(x, p_x) dx dp_x = \frac{\varepsilon(1 + \alpha^2)}{\beta}, \quad (3)$$

$$\overline{xp_x} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} xp_x f(x, p_x) dx dp_x = -\alpha\varepsilon. \quad (4)$$

Решая систему уравнений из (2), (3) и (4), можно выразить эмиттанс:

$$\varepsilon = \sqrt{\overline{x^2} \cdot \overline{p_x^2} - (\overline{xp_x})^2}. \quad (5)$$

2.3. Функции искажения (Distortion functions)

Поперечная линейная динамика частиц в фокусирующих магнитных полях (бетатронные колебания) описывается уравнением Хилла:

$$x'' + K(s)x = 0,$$

где $K(s) = K(s + \Pi)$ – периодический коэффициент.

По теореме Флоке, решением этого уравнения является:

$$x(s) = \sqrt{2j_x \beta_x(s)} \cos(\psi_x(s) + \psi_{x0}),$$

где j_x, ψ_{x0} – константы, определяющиеся начальными условиями, а $\beta_x(s)$ и $\psi_x(s) = \int_0^s \frac{ds}{\beta_x(s)}$ – функции, которые определяются магнитной структурой ускорителя.

Для уравнения Хилла существует инвариант Куранта-Снайдера, $[x^2 + (\alpha x + \beta x')^2]/\beta = 2j = \varepsilon$, который является площадью эллипса, поделенной на π , в фазовом пространстве [8].

Причем, j_x также является канонической переменной действия, которая задается интегралом $j_x = \frac{1}{2\pi} \oint x dp_x$.

Гамильтониан частицы в линейном случае записывается как:

$$H(x, p_x) = \frac{p_x^2}{2} + \frac{x^2}{2} (k_0^2 + k_1).$$

Или можно записать его через переменные действия-фаза, которые также являются каноническими координатами:

$$H(j_x, \psi_x) = \frac{\nu_x j_x}{R},$$

где ν_x – бетатронная частота.

В случае нелинейной динамики уравнение Хилла становится нелинейным: $x'' + K(s)x = F(x, s)$, соответственно решение меняется, и действие, заданное инвариантом Куранта-Снайдера, перестает быть инвариантом. Действие и фаза в таком случае терпят биения (Рис. 10).

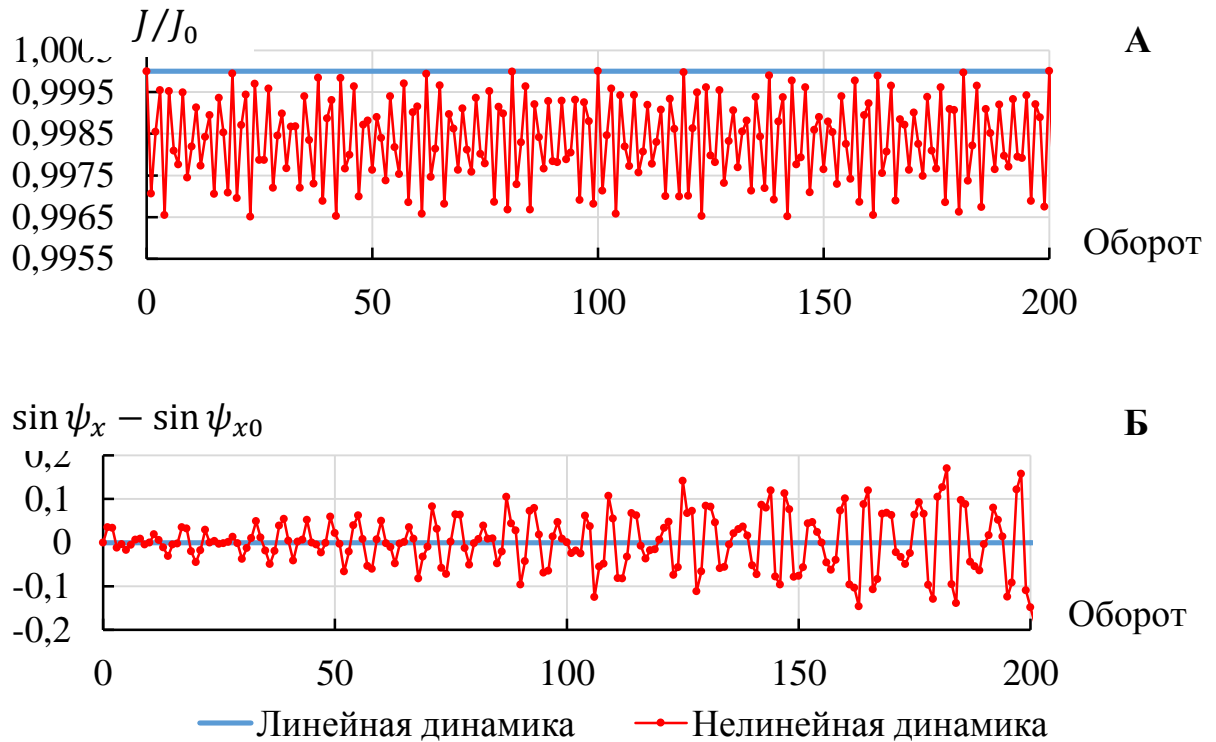


Рис. 10. Биение действия (А) и фазы (Б) от оборота в СКИФ.

Точное решение нелинейного уравнения неизвестно, поэтому его записывают с помощью теории возмущений. В первом порядке решение можно выразить через функции искажения, которые вводятся следующим образом.

Гамильтониан частицы при наличии секступолей в структуре имеет вид:

$$H(x, p_x; s) = \frac{p_x^2}{2} + \frac{x^2}{2} (k_0^2 + k_1) + \frac{x^3}{6} k_2.$$

Чтобы понять, какое влияние нелинейная динамика оказывает на эмиттанс, найдем сначала биение действия и фазы. Запишем гамильтониан частицы в переменных действия и фазы:

$$H(\psi_x, j_x; s) = \frac{v_x j_x}{R} + \frac{k_2}{6} \sqrt{(2j_x \beta_x)^3} \cos^3 \psi_x,$$

$$\Phi_x(s) = \psi_x + \int_0^s \frac{ds}{\beta_x} - \frac{v_x s}{R}.$$

Перейдем к переменной $\theta = \frac{s}{R}$:

$$H(\psi_x, j_x; s) = v_x j_x + \frac{Rk_2}{6} \sqrt{\frac{(j_x \beta_x)^3}{2}} [\cos(3\Phi_x(\theta)) + 3\cos(\Phi_x(\theta))].$$

Решая уравнения движения $j'_x = -\frac{\partial H}{\partial \psi_x}$, $\psi'_x = \frac{\partial H}{\partial j_x}$ в 0-ом (при $k_2 = 0$) и 1-ом (при $k_2 \neq 0$) порядках по силе секступоля, получаем выражения для биений действия и фазы, выраженные через некоторые интегралы:

$$\delta j_x = j_{0x}^{3/2} \{A_3(\theta) \sin 3\Phi_x(\theta) - B_3(\theta) \cos 3\Phi_x(\theta) + A_1(\theta) \sin \Phi_x(\theta) - B_1(\theta) \cos \Phi_x(\theta)\}, \quad (6)$$

$$\delta \psi_x = \frac{j_{0x}^{1/2}}{2} \{B_3(\theta) \sin 3\Phi_x(\theta) + A_3(\theta) \cos 3\Phi_x(\theta) + 3B_1(\theta) \sin \Phi_x(\theta) + 3A_1(\theta) \cos \Phi_x(\theta)\}, \quad (7)$$

где $A_1(\theta)$, $B_1(\theta)$, $A_3(\theta)$, $B_3(\theta)$ – периодические функции, называемые функциями искажения, зависящие только от геометрии ускорителя [9]:

$$A_1(\theta) = \int_{\theta}^{\theta+2\pi} \frac{Rk_2}{2\sqrt{2}} \beta_x^{3/2} \frac{\sin(\varphi_x(\tau) - \varphi_x(\theta) - \pi\nu_x)}{2\sin(\pi\nu_x)} d\tau, \quad (8)$$

$$B_1(\theta) = \int_{\theta}^{\theta+2\pi} \frac{Rk_2}{2\sqrt{2}} \beta_x^{3/2} \frac{\cos(\varphi_x(\tau) - \varphi_x(\theta) - \pi\nu_x)}{2\sin(\pi\nu_x)} d\tau, \quad (9)$$

$$A_3(\theta) = \int_{\theta}^{\theta+2\pi} \frac{Rk_2}{2\sqrt{2}} \beta_x^{3/2} \frac{\sin(3(\varphi_x(\tau) - \varphi_x(\theta) - \pi\nu_x))}{2\sin(3\pi\nu_x)} d\tau, \quad (10)$$

$$B_3(\theta) = \int_{\theta}^{\theta+2\pi} \frac{Rk_2}{2\sqrt{2}} \beta_x^{3/2} \frac{\cos(3(\varphi_x(\tau) - \varphi_x(\theta) - \pi\nu_x))}{2\sin(3\pi\nu_x)} d\tau. \quad (11)$$

После разложения функций координаты, $x = \sqrt{2j_x \beta_x} \cos \psi_x$, и импульса, $p_x = -\sqrt{\frac{2j_x}{\beta_x}} (\alpha_x + \tan \psi_x)$, по переменным действия и фазы и усреднения по

ним, найдем как изменились средние значения \bar{x} , $\overline{p_x}$ из-за влияния нелинейной динамики:

$$\bar{x} = -\sqrt{2\beta_x}\varepsilon_0 B_1, \quad (12)$$

$$\overline{p_x} = \sqrt{\frac{2}{\beta_x}}\varepsilon_0(\alpha B_1 - A_1). \quad (13)$$

Из формул (12, 13) видно, что средние значения координаты и импульса в нелинейной динамике не зануляются, то есть под влиянием секступолей происходит сдвиг замкнутой орбиты.

Вторые моменты $\overline{x^2}$, $\overline{p_x^2}$ и $\overline{xp_x}$ под влиянием нелинейной динамики:

$$\overline{x^2} = \varepsilon_0\beta_x - \frac{1}{4}\varepsilon_0^2\beta_x\{15A_1^2 + 10A_1A_3 + B_1[B_1 + 10B_3]\}, \quad (14)$$

$$\begin{aligned} \overline{p_x^2} &= \frac{\varepsilon_0(1 + \alpha_x^2)}{\beta_x} \\ &\quad - \frac{1}{4}\frac{\varepsilon_0^2}{\beta_x}\{A_1^2[1 + 15\alpha_x^2] \\ &\quad + 2A_1[2(7B_1 + 5B_3)\alpha_x + 5A_3(\alpha_x^2 - 1)] \\ &\quad + B_1[B_1(15 + \alpha_x^2) - 10(B_3 + 2A_3\alpha_x - B_3\alpha_x^2)]\}, \end{aligned} \quad (15)$$

$$\begin{aligned} \overline{xp_x} &= -\varepsilon_0\alpha_x \\ &\quad + \frac{1}{4}\varepsilon_0^2\{15A_1^2\alpha_x + B_1[-10A_3 + (B_1 + 10B_3)\alpha_x] \\ &\quad + 2A_1[7B_1 + 5(B_3 + \alpha_x A_3)]\}. \end{aligned} \quad (16)$$

В выражениях (14-16) зависимость функций искажения от угла θ опущена. Заметим, что первые слагаемые в этих выражениях совпадают со средними значениями, подсчитанными через интегралы от функции распределения (2-4).

Подставляя средние значения координаты и импульса (14-16) в формулу среднестатистического эмиттанса (5) и раскладывая квадратный корень в ряд Тейлора, получаем выражение для эмиттанса через функции искажения:

$$\varepsilon = \varepsilon_0 - 2\varepsilon_0^2[A_1^2(\theta) + B_1^2(\theta)]. \quad (17)$$

Исходя из выражения (17), видно, что под влиянием нелинейной динамики эмиттанс должен уменьшаться, т.к. $A_1(\theta), B_1(\theta) \in \mathbb{R}$.

3. Практическая часть

Перед тем, как перейти к основной задаче работы, исследованию нелинейной динамики СКИФ, было решено начать с более простой структуры ускорителя – FODO-ячейки. Она использовалась в двух целях. Во-первых, научиться пользоваться программой MAD-X, прописывать в ней структуру ячейки и рассчитывать траекторию частицы, движущейся в ней. Во-вторых, чтобы сравнить линейную и нелинейную динамику в FODO-ячейке.

Исходя из этого работа была разделена на 5 этапов:

- 1) Моделирование FODO-ячейки;
- 2) Добавление секступолей в структуру;
- 3) Написание программы для подсчёта среднестатистического эмиттанса;
- 4) Написание программы для вычисления функций искажения и эмиттанса через значения этих функций;
- 5) Замена FODO-ячейки на структуру СКИФ.

3.1. Линейная динамика в FODO-ячейке

Была написана программа на MAD-X (см. Приложение 1) [10], формирующая структуру ячейки (Рис. 11), создающая файл с параметрами Твисса и другими параметрами оптики ускорителя. В ячейке был произведен трекинг пучка электронов, состоящего из 1000 частиц с энергией $E = 3$ ГэВ в течение 1024 оборотов. Начальные координаты и импульсы электронов пучка были распределены по Гауссу (1) с $\varepsilon = 70$ пм. В точке наблюдения: $\beta \approx 0,79$ и $\alpha \approx 0$ (Рис. 12).

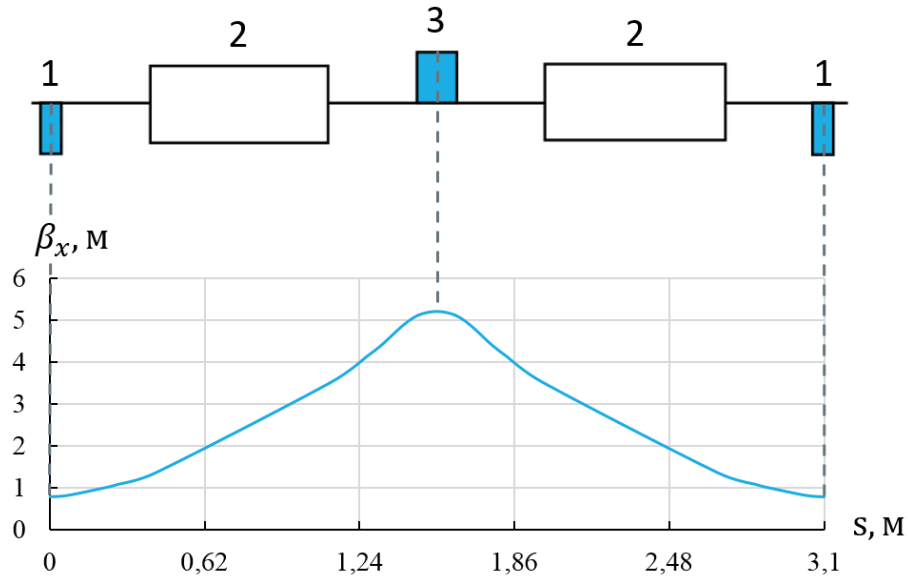


Рис. 11. Схема структуры FODO-ячейки, где 1 – дефокусирующие квадруполь, 2 – диполи и 3 – фокусирующий квадруполь. Ниже представлен график горизонтальной бета-функции от азимута.

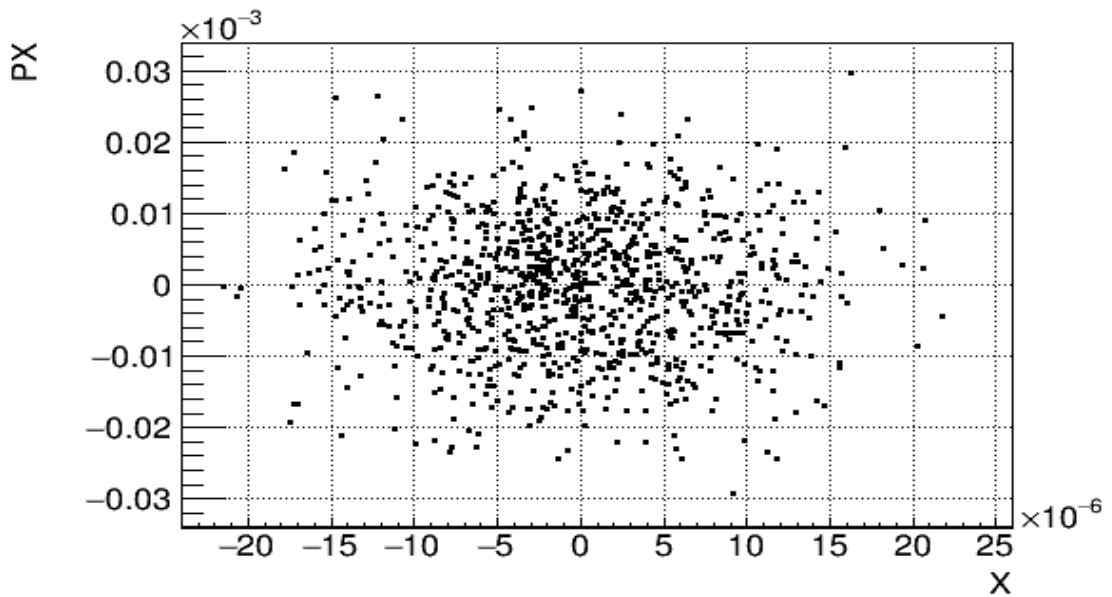


Рис. 12. Начальное распределение пучка из 1000 электронов в фазовом пространстве.

3.2. Нелинейная динамика в FODO-ячейке

В программу, формирующую FODO-ячейку, были добавлены секступоли (Рис. 13) (см. Приложение 2). Был произведен трекинг в течение 1024 оборотов пучка электронов с энергией $E = 3$ ГэВ. Начальное распределение частиц в фазовом пространстве показано на Рис. 12.

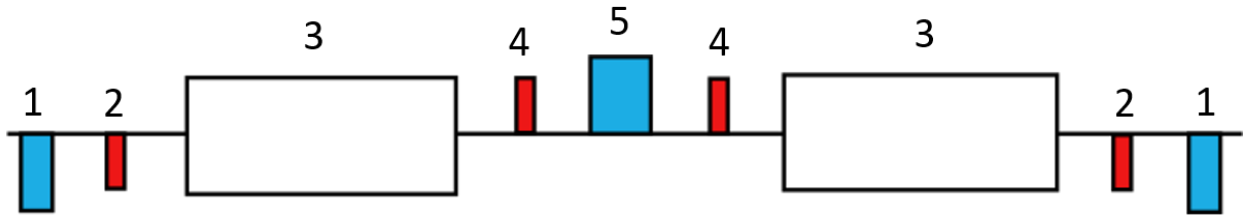


Рис. 13. Схема структуры FODO-ячейки с добавлением секступолей, где 1, 5 – квадруполи, 2, 4 – секступоли, 3 – диполи.

3.3. Программы для подсчета эмиттанса

Было написано 2 программы на языках С и С++ при помощи программы ROOT [11].

Первая программа (см. Приложение 3) считывает файлы трекинга и считает средние значения координат и импульсов на каждом обороте по формулам:

$$\bar{x} = \sum_{i=1}^N \frac{x_i}{N}; \quad \bar{p}_x = \sum_{i=1}^N \frac{p_{xi}}{N};$$

$$\overline{x^2} = \sum_{i=1}^N \frac{x_i^2}{N}; \quad \overline{p_x^2} = \sum_{i=1}^N \frac{p_{xi}^2}{N};$$

$$\overline{xp_x} = \bar{x} \cdot \bar{p}_x.$$

Также она считает ошибки с помощью формулы для несмещенной выборочной дисперсии:

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2},$$

где N – количество частиц.

Далее считается эмиттанс (5) на каждом обороте и строится график эмиттанса от оборота. В отдельный файл выводятся значения на нулевом и последнем обороте.

Вторая программа (см. Приложение 4) считывает созданный программой MAD-X файл twiss.tfs, который содержит всю информацию об оптике ячейки ускорителя. Затем строит графики функций искажения и находит средние значения координаты и импульса (14-16) и эмиттанс по формуле (17) в той же точке наблюдения, что была в трекинге.

3.4. Ячейка СКИФ

В программе MAD-X из 2.1 [10] ячейка FODO была заменена на ячейку СКИФ (Рис. 14) (см. Приложение 5). На этот раз трекинг был проведен с пучком из 100000 электронов с энергией $E=3$ ГэВ. Частицы были распределены по Гауссу (1) с $\varepsilon_0 = 70$ пм. Параметры Твисса в точке наблюдения: $\beta \approx 14,24$ и $\alpha \approx 0$ (Рис. 15).

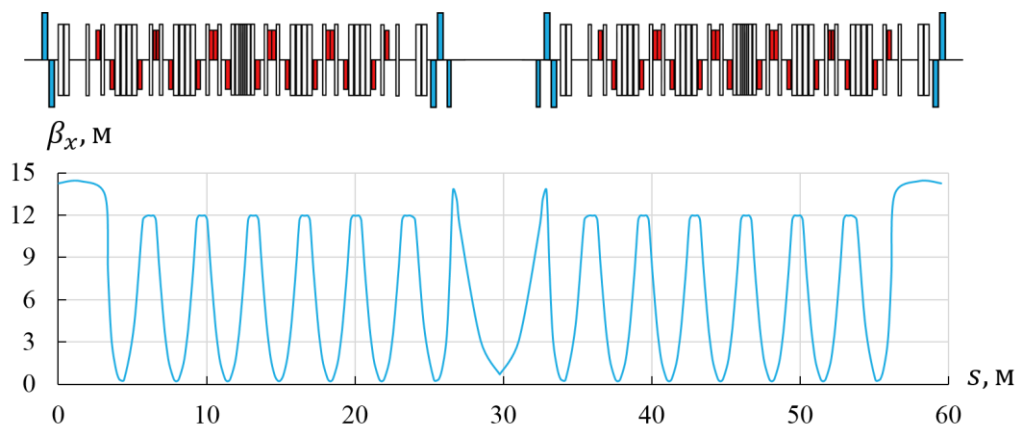


Рис. 14. Схема структуры СКИФ, где белым обозначены диполи, голубым – квадруполь, и красным – секступоль. Ниже приведен график горизонтальной бета-функции от азимута.

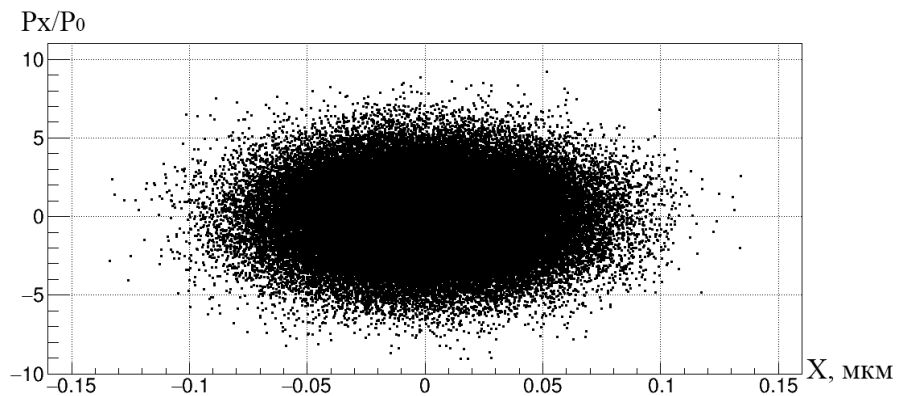


Рис. 15. Начальное распределение пучка из 100000 электронов в фазовом пространстве.

4. Результаты

4.1. Сравнение линейной и нелинейной динамики FODO

Сначала было интересно сравнить линейную и нелинейную динамику в FODO-ячейке. Результаты трекинга пучка из 1000 электронов приведены в таблице ниже:

Таблица. Сравнение эмиттансов на 0 и 1024 обороте

Оборот	ϵ , пм (линейная динамика)	ϵ , пм (нелинейная динамика)
0	$68,22 \pm 1,04$	$68,22 \pm 1,04$
1024	$68,22 \pm 1,07$	$68,21 \pm 1,07$

Из таблицы видно, что после 1024 оборотов в FODO-ячейке в линейном случае эмиттанс не изменился, а в нелинейном – незначительно уменьшился. Однако, это уменьшение недостоверно, т.к. происходит в пределах ошибки. То есть, нелинейная динамика не оказывает значительного влияния на эмиттанс пучка в структуре FODO-ячейки (Рис. 16, Рис. 17).

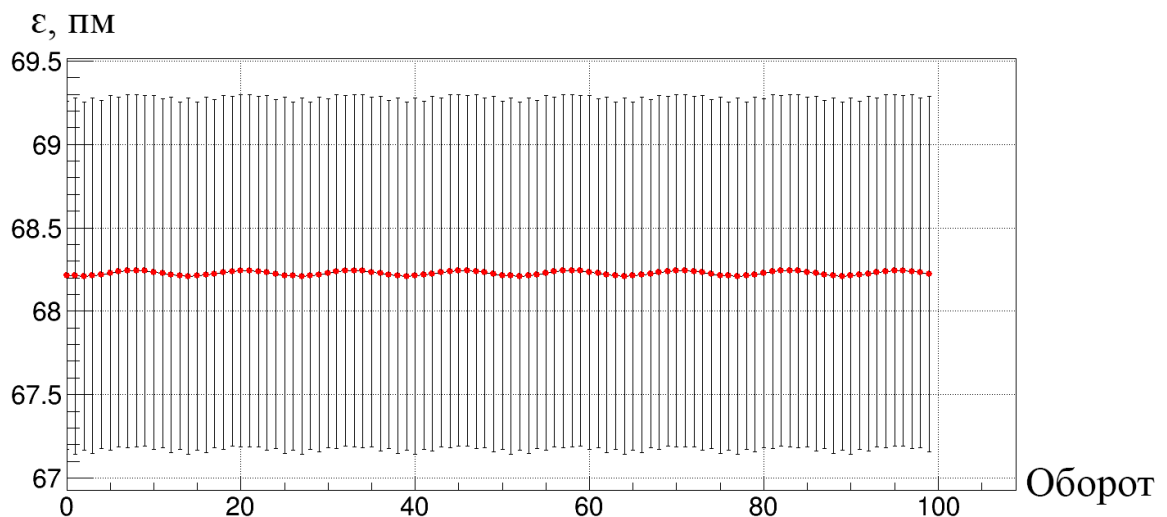


Рис. 16. График зависимости эмиттанса от оборота в линейной динамике FODO-ячейки.

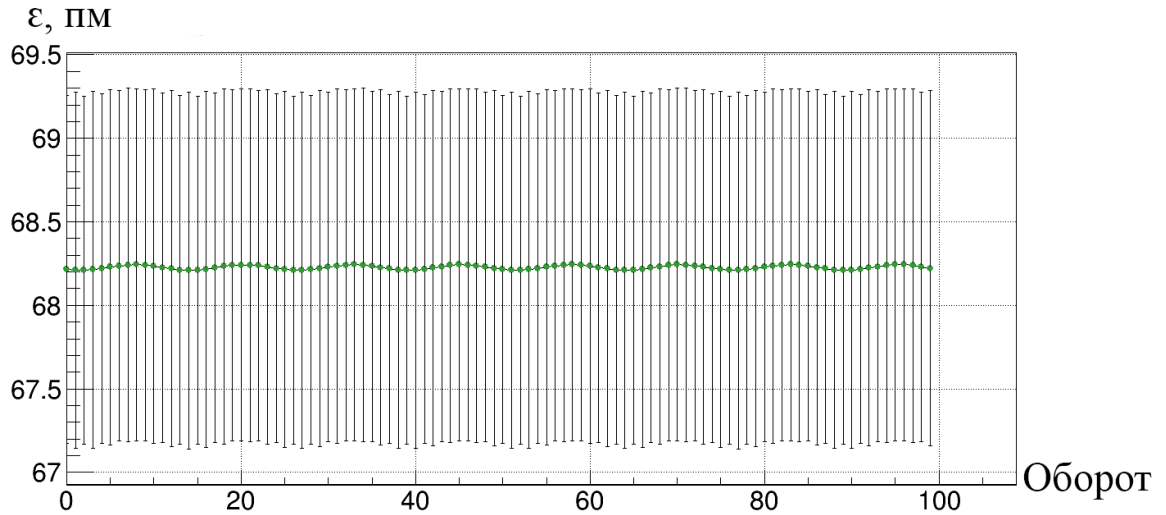


Рис. 17. График зависимости эмиттанса от оборота в нелинейной динамике FODO-ячейки.

Возможно, колебания на графиках связаны с численными ошибками программ расчётов (Рис. 16, Рис. 17).

Также были посчитаны функции искажения для нелинейной динамики структуры FODO (Рис. 18).

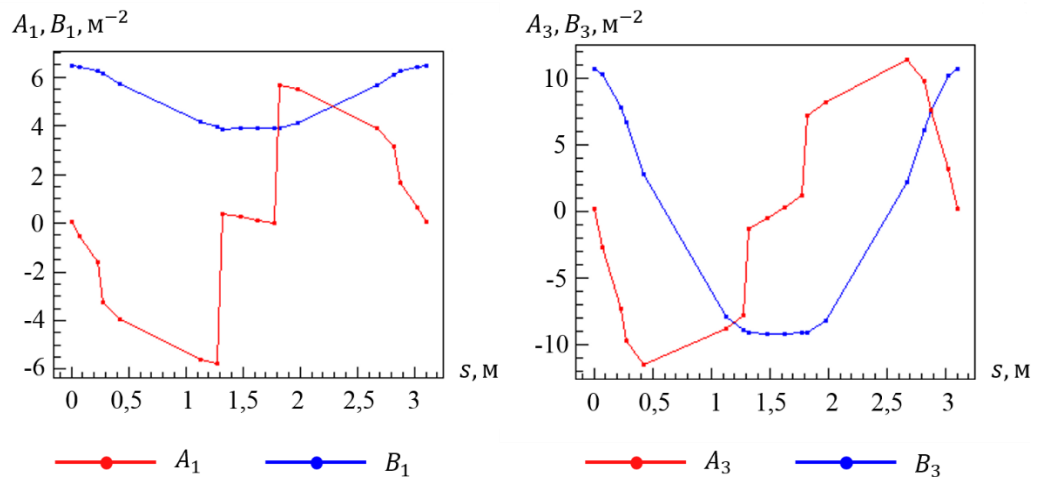


Рис. 18. Функции искажения структуры FODO.

4.2. Нелинейная динамика СКИФ

Результаты трекинга пучка и 100000 электронов в СКИФ показали, что эмиттанс не изменился после 1024 оборотов:

$$\varepsilon_0 = 70,25 \pm 0,16 \text{ пм,}$$

$$\varepsilon_{1024} = 70,25 \pm 0,16 \text{ пм.}$$

Графики распределения частиц (Рис. 19) и зависимости эмиттанса от оборота (Рис. 20) также показывают постоянность эмиттанса в ходе его движения по орбите.

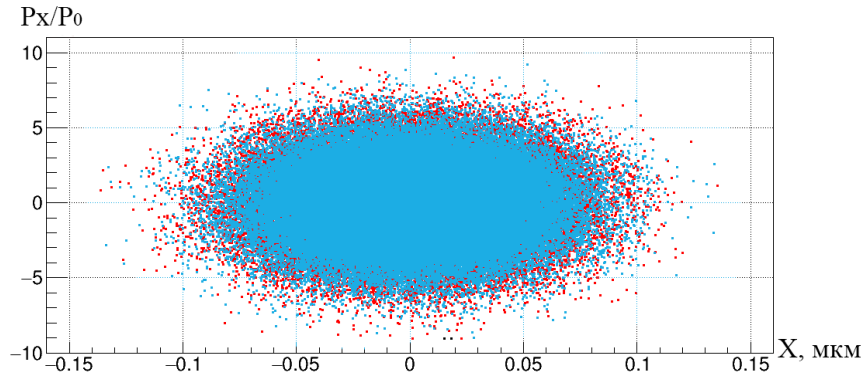


Рис. 19. Распределение электронов в фазовом пространстве, где голубым обозначено распределение на 0-ом обороте, а красным – на 1024-ом.
 ϵ , ПМ

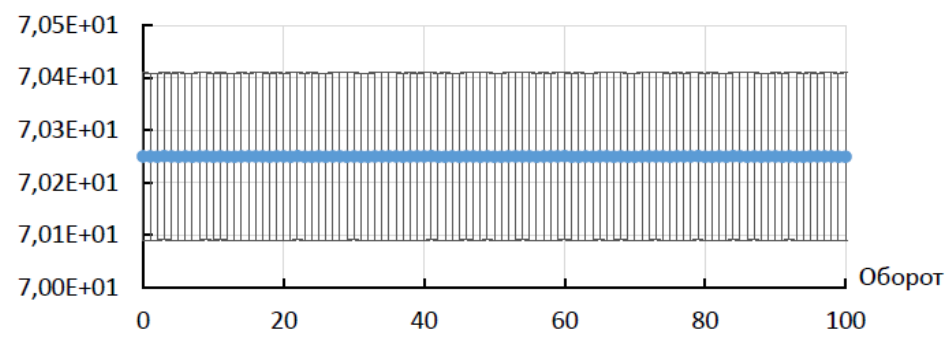


Рис. 20. График зависимости эмиттанса от оборота в структуре СКИФ.

На следующем этапе были посчитаны и нарисованы графики функций искажения A_1, B_1, A_3, B_3 по всей длине ячейки (Рис. 21).

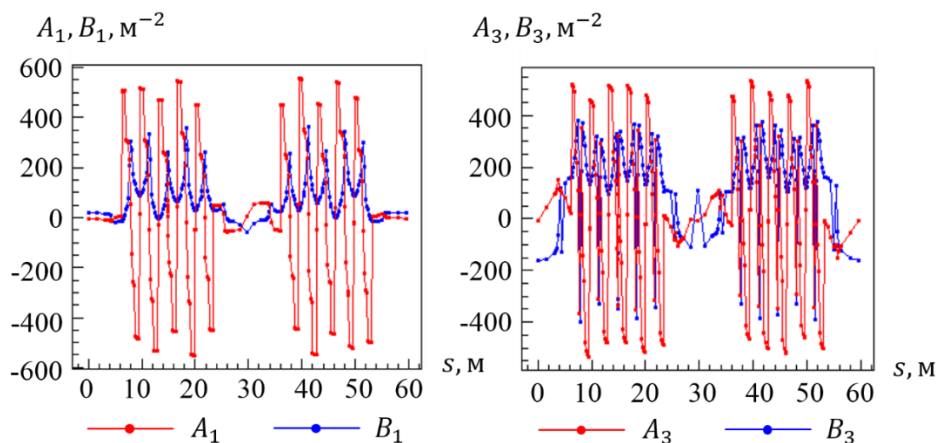


Рис. 21. Функции искажения структуры СКИФ.

Были записаны их значения в точке наблюдения:

$$A_1(0) = -4.23594, \quad B_1(0) = 17.1491,$$

$$A_3(0) = -6.12631, \quad B_3(0) = -161.236,$$

и через них посчитан ожидаемый эмиттанс (17):

$$\varepsilon \approx 69,999997 \text{ пм.}$$

5. Обсуждение результатов

Исходя из результатов трекинга в ячейке СКИФ, нелинейная динамика не повлияла на эмиттанс пучка. Это связано с маленькими значениями функций искажения. Программа, считающая эмиттанс через функции искажения (см. Приложение 4), показала, что он должен уменьшиться в шестом знаке после запятой.

Стало интересно, какой должна быть сумма квадратов функций искажения, $A_1^2 + B_1^2$, чтобы эмиттанс стал равным 63 пм, то есть уменьшился на 10%. Для этого второе слагаемое в формуле (17) должно составлять 10% от начального эмиттанса пучка. На основе проделанной оценки:

$$2\varepsilon_0(A_1^2 + B_1^2) = 10\%,$$

при $\varepsilon_0 = 70$ пм сумма квадратов должна быть 9-ого порядка, $A_1^2 + B_1^2 \sim 10^9$.

Для достижения такого значения варьировались силы секступолей, k_2 , и бетатронная частота, ν_x (8, 9).

Оптимальные значения, с которыми проходил трекинг частиц:

$$k_{2f}L_0 \approx 34,345, \quad k_{2d}L_0 \approx -59,878, \quad \nu_{x0} = 6,78996,$$

где $k_{2f}L, k_{2d}L$ – силы фокусирующего и дефокусирующего секступолей соответственно, помноженные на длину магнитов.

Если не менять бетатронную частоту, а изменить только силу секступолей, то для достижения нужного значения суммы квадратов функций

искажения нужно сделать силы фокусирующего и дефокусирующего секступолей равными:

$$k_{2f}L \approx 50000,$$

$$k_{2d}L \approx -87500.$$

То есть для того, чтобы эмиттанс уменьшился на 10%, нужно увеличить силы секступолей на 3 порядка. После подставления этих значений в интегралы для нахождения функций искажения (8, 9), а затем в выражение для эмиттанса (17), было получено:

$$\varepsilon = 63,59 \text{ пм.}$$

Для подбора нужной бетатронной частоты были построены графики зависимости суммы квадратов функций искажения от бетатронной частоты для ячейки СКИФ (Рис. 22). На графиках (Рис. 22, Рис. 23) красной линией обозначено значение $A_1^2 + B_1^2$, которое мы пытаемся достичь.

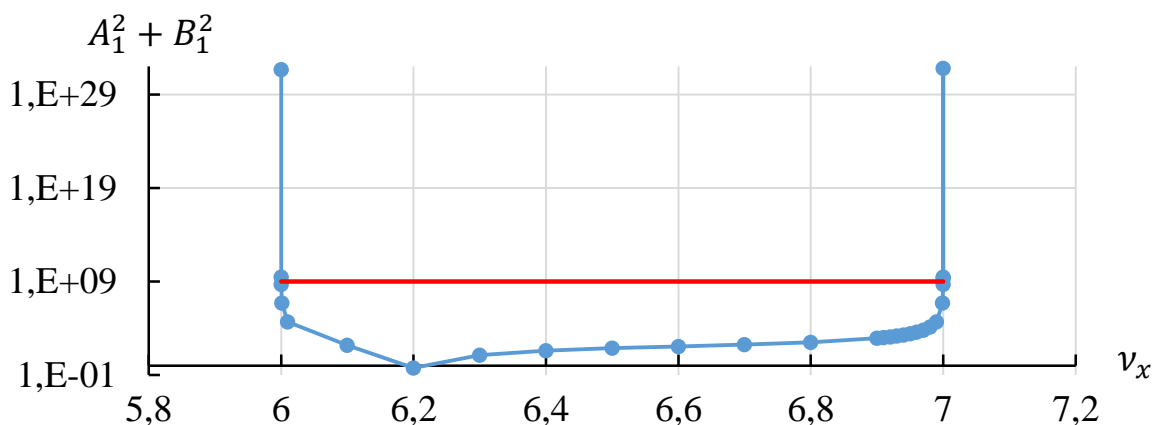


Рис. 22. Зависимость суммы квадратов функций искажения в точке наблюдения от бетатронной частоты.

Исходя из этой зависимости, был сделан вывод, что эмиттанс будет тем меньше, чем ближе значение бетатронной частоты к целому значению. Чтобы понять, в каком знаке должно быть отличие от целого значения, зависимость была перестроена в логарифмической шкале, где абсцисса – это разница бетатронной частоты и ближайшего целого значения (Рис. 22).

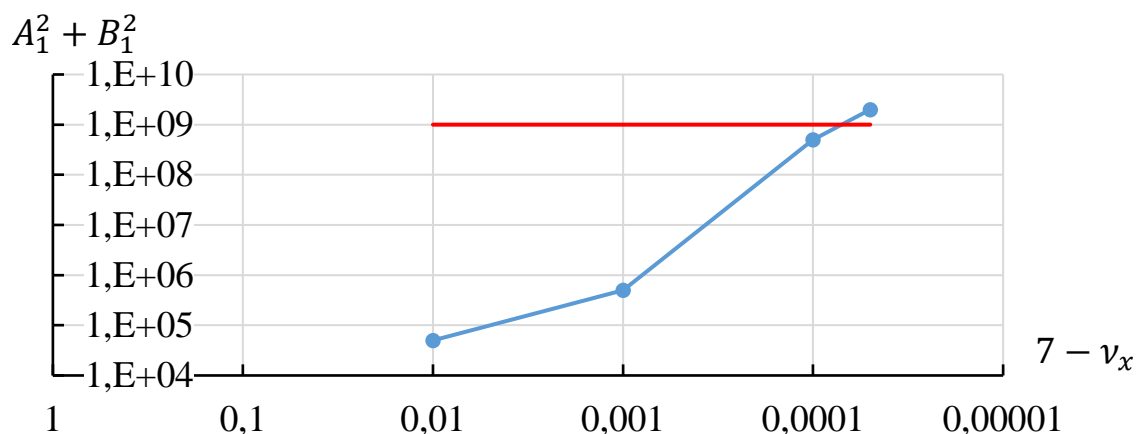


Рис. 23. Зависимость суммы квадратов функций искажения в точке наблюдения вблизи целого значения бетатронной частоты в логарифмической шкале.

На графике видно, что отличие должно быть в 4-ом знаке после запятой.

При $\nu_x = 6,99987$ и начальных значения сил фокусирующего и дефокусирующего секступолей эмиттанс равен (17):

$$\varepsilon = 63,11 \text{ пм.}$$

6. Заключение

В ходе работы была освоена теория функций искажения – структурных функций, описывающих возмущения движения частицы в первом порядке теории возмущения. Написана программа, рассчитывающая функции возмущения и визуализирующая их.

Результаты работы показали, что нелинейная динамика СКИФ не меняет эмиттанс.

Исходя из теоретических расчетов (17), было предложено два способа уменьшения эмиттанса на 10%: приблизить бетатронную частоту к целому числу или увеличить силу секступолей на 3 порядка. Однако, это практически невозможно. Возможно поставить специальную пару секступолей таким образом, чтобы функции возмущений испытали локальное сильное увеличение внутри пары, а снаружи бы не изменились. Этот вопрос может стать объектом следующих исследований.

7. Список литературы

- [1] Фетисов, Г. Синхротронное излучение. Методы исследования структуры веществ. / Под редакцией Л. Асланова. – М.: ФИЗМАТЛИТ, 2007. – 672 с.
- [2] Winick H., Doniach S. Synchrotron radiation research. – Springer Science & Business Media, 2012.
- [3] Кулипанов Г. Н. Синхротронное излучение, история и применение- Новосибирск: Ин-т ядерной физики им. ГИ Будкера СО РАН. – 2012.
- [4] Holmes J. et al. Magnetic Fields and Magnet Design //USPAS, Vanderbilt. – 2009.
- [5] Rossbach J., Schmueser P. Basic course on accelerator optics //Conf. Proc. – 1992. – Т. 9209071. – С. 17-88.
- [6] Hock K. Sextupole magnet //transfer. – 2010. – Т. 5. – №. y2. – С. 14.
- [7] Buon J. Beam phase space and emittance. – Paris-11 Univ., 1992. – №. LAL-RT--92-03.
- [8] Левичев Е. Лекции по нелинейной динамике частиц в циклическом ускорителе: учеб. пособие / Е.Б.Левичев. – Новосибирск: Изд-во НГТУ, 2009. – 224 с.
- [9] T. Collins, Proceedings of the 1984 Summer Study on the Design and Utilization of the Superconducting Super Collider, Snowmass, Colorado, 1984.
- [10] Сайт MAD: MAD - Methodical Accelerator Design, CERN - BE/ABP Accelerator Beam Physics Group. – Режим доступа: <https://mad.web.cern.ch/mad/>
- [11] Brun R. et al. The ROOT Users Guide //CERN, – 2003. – Режим доступа: <http://root.cern.ch>.

8. Приложения

8.1. Приложение 1

`option, -echo, -info, warn, -no_fatal_stop;`


```

TITLE, 'Example 1: FODO.MADX';

KQF = 4.69;
KQD = -4.69;

D: DRIFT, L=0.15;
D2: DRIFT, L=0.05;
QF: QUADRUPOLE, L=0.15, K1= KQF;
QD: QUADRUPOLE, L=0.075, K2= KQD;
B: SBEND, L=0.7, ANGLE=TWOPI/60;

FODO: LINE=(QD,D,D2,D,B,D,D2,D,QF,D,D2,D,B,D,D2,D,QD);

Nrj=3;           //Beam energy, GeV

beam,particle=electron,energy=Nrj,radiate=false;

set,format="22.15g";

select,flag=twiss,clear;

select,flag=twiss,COLUMN=S,NAME,keyword,L,BETX,BETY,ALFX,ALFY,MUX,MUY,DX,DPX,
DY,DPY,DDX,DDPX,WX,PHIX,WY,PHIY,ANGLE,E1,E2,TILT,K1L,K2L,K3L;

USE,period=FODO;

TWISS,file="twiss.tfs";

PLOT,TABLE=TWISS,TITLE="FODO",HAXIS=S,VAXIS1=BETX,BETY,VAXIS2=DX,COLOUR=100,I
NTERPOLATE,-noverion;

delete,table=tracksumm;

i=0;
EM=70e-12;
betx=0.793730100900356;

    use,sequence=FODO;
    ptc_create_universe;
    ptc_create_layout, time=true, model=2, exact=false, method=6, nst=10;
        while (i<1000) {
            ptc_start,x=sqrt(EM*betx)*GAUSS(),px=sqrt(EM/betx)*GAUSS(),
            y=0, py=0,pt=0, T=0;
            i=i+1;
        };
    ptc_track,  icase=4, closed_orbit, dump, maxaper={100,100,100,100,100,100},
    turns=1025, ffile=1;
    ptc_track_end;
    ptc_end;

write,table=tracksumm,file="ptc_table.tfs";

Value, TABLE (SUMM, Q1);
Value, TABLE (SUMM, Q2);

return;

```

8.2. Приложение 2

```

option, -echo, -info, warn, -no_fatal_stop;
TITLE, 'Example 1: FODO.MADX';
KQF = 4.69;
KQD = -4.69;
KSF = 44.2;
KSD = -44.2;
D: DRIFT, L=0.15;
QF: QUADRUPOLE, L=0.15, K1:=KQF;
QD: QUADRUPOLE, L=0.075, K2:=KQD;
SF: SEXTUPOLE, L=0.05, K2:= KSF;
SD: SEXTUPOLE, L=0.05, K2:= KSD;
B: SBEND, L=0.7, ANGLE=TWOPI/60;
FODO: LINE=(QD, D, SD, D, B, D, SF, D, QF, D, SF, D, B, D, SD, D, QD);
Nrj=3;           //Beam energy, GeV
beam, particle=electron, energy=Nrj, radiate=false;
USE, PERIOD=FODO;
MATCH, SEQUENCE = FODO;
CONSTRAINT, SEQUENCE=FODO, RANGE=#E, MUX=0.27, MUY=0.27;
VARY, NAME=KQF, STEP=1E-6;
VARY, NAME=KQD, STEP=1E-6;
jacobian, calls=30, tolerance=1e-15, bisec=jac_bisec, strategy=1;
ENDMATCH;
USE, period=FODO;
match, sequence=FODO, chrom;
    vary, name=KSF, step=0.0001;
    vary, name=KSD, step=0.0001;
    global, sequence=FODO, dq1=0, dq2=0;
    jacobian, calls=30, tolerance=1e-15, bisec=jac_bisec, strategy=1;
endmatch;
set, format="22.15g";
select, flag=twiss, clear;
select, flag=twiss, COLUMN=S, NAME, keyword, L, BETX, BETY, ALFX, ALFY, MUX, MUY, DX, DPX,
DY, DPY, DDX, DDPX, WX, PHIX, WY, PHIY, ANGLE, E1, E2, TILT, K1L, K2L, K3L;
USE, period=FODO;
TWISS, file="twiss.tfs";
PLOT, TABLE=TWISS, TITLE="FODO", HAXIS=S, VAXIS1=BETX, BETY, VAXIS2=DX, COLOUR=100, I
NTERPOLATE, -noverision;

```

```

delete, table=tracksumm;

i=0;
EM=70e-12;
betx=0.793730100900356;

    use, sequence=FODO;

    ptc_create_universe;

    ptc_create_layout, time=true, model=2, exact=false, method=6, nst=10;
        while (i<1000) {
            ptc_start, x=sqrt (EM*betx) *GAUSS (), px=sqrt (EM/betx) *GAUSS (),
                y=0, py=0, pt=0, T=0;
            i=i+1;
        };

    ptc_track, icense=4, closed_orbit, dump, maxaper={100,100,100,100,100,100},
        turns=1025, ffile=1;

    ptc_track_end;

    ptc_end;

write, table=tracksumm, file="ptc_table.tfs";

Value, TABLE (SUMM, Q1);
Value, TABLE (SUMM, Q2);

return;

```

8.3. Приложение 3

```

#include <fstream>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <iostream>
#include "TROOT.h"
#include "TFile.h"
#include "TTree.h"
#include "TGraph.h"
#include "TGraphErrors.h"
#include "TCanvas.h"
#include "TMath.h"

struct track_t{
    Double_t X,PX,Y,PY,T,PT,S,E;
    UInt_t part,turn,obs;
};

```

```

UInt_t Stable_turns;

Int_t not;

Int_t treer(UInt_t Stability_turns,UInt_t obs, UInt_t part, UInt_t num_obs,
UInt_t num_part){

    Stable_turns=Stability_turns;

    char file[10000];

    std::ifstream from;

    track_t entry;

    string buffer;

    UInt_t i_obs=0;

    UInt_t i_part=0;

    TFile *out=0;

        if( (out=(TFile*)gROOT->FindObject("out")) ) out->Close();

        out=new TFile("track.root","RECREATE");

    TTree *tree=0;

        if( (tree=(TTree*)gDirectory->Get("tree")) ) delete tree;

        tree=new TTree("tree","tracking data");

        tree -> Branch ("track", &entry.X, "X/D: PX: Y: PY: T: PT: S: E:
part/i:turn:obs");

        TEntryList *list=0;

        if( (list=(TEntryList*)gDirectory->Get("list" )) ) delete list;

        list=new TEntryList("list" ,"stable particles" ,tree);

        do{

            do{

                sprintf(file,"../../track.obs%04u.p%04u",
obs+i_obs,part+i_part);

                from.open(file);

                if

(!from){ std::cout<< "Error opening file " << file << std::endl; }

                {

                    while( from && (from.peek()=='*' || from.peek()=='$' ||
from.peek()=='@') ){

                        getline(from,buffer);

                    }while (from>>entry.part>>entry.turn>>entry.X>>entry.PX>>entry.Y>>entry.P
Y>>entry.T>>entry.PT>>entry.S>>entry.E) {

                        if(from.fail() || from.bad()){ std::cout<<"Error
reading file "<<file<<std::endl; }

                        entry.obs=obs+i_obs;

                        tree->Fill();

```

```

        PHIX_old=PHIX;
        PHIY_old=PHIY;
    }
    if(entry.turn==Stability_turns){
entry.part),    "entrylist",    tree->Draw(">>+    list",    Form("part==%u",
(Stability_turns + 1)); }    Stability_turns+1,    tree    ->    GetEntries()    -
    }
    from.close();
    ++i_part;
}while(i_part<num_part);
i_part=0;
++i_obs;
}while(i_obs<num_obs);
out->Write();
out->Close();
return 0;
}

void mean(UInt_t Stability_turns,UInt_t obs, UInt_t part, UInt_t num_obs,
UInt_t num_part, Double_t* a,Double_t* b,Double_t* c,Double_t* d, Double_t*
e,Double_t* ef,Double_t* g,Double_t* h,Int_t* p, Double_t* J){
    TFile *out=0;
    if( (out=(TFile*)gROOT->FindObject("out")) ) out->Close();
    out=new TFile("track.root","read");
    TTree *tree = (TTree*)out->Get("tree");
    tree->SetEstimate(-1);
    track_t entry;
    tree->SetBranchAddress("track",&entry.X);
    Int_t i=0;
    Int_t j=0;
    Int_t nentries=tree->GetEntries();
    for(j =0; j<nentries; j++){
        tree->GetEntry(j);
        p[entry.turn]+=entry.part;
        a[entry.turn]+=entry.X;
        b[entry.turn]+=entry.PX;
        c[entry.turn]+=pow(entry.X,2);
        d[entry.turn]+=pow(entry.PX,2);
    }
}

```

```

    for(i=0;i<=Stable_turns;i++){
        if(p[i]==0){ cout<<"All particles were lost at turn number
"<<i<<endl; not=i-1; break;}
        else{not=Stable_turns;}
        a[i]=a[i]/(Double_t)p[i];
        b[i]=b[i]/(Double_t)p[i];
        c[i]=c[i]/(Double_t)p[i];
        d[i]=d[i]/(Double_t)p[i];
    }
    for(j =0; j<=1024; j++){
        e[j]+=pow((entry.X-a[entry.turn]),2)/(p[j]-1);
        ef[j]+=pow((entry.PX-b[entry.turn]),2)/(p[j]-1);
        g[j]+=pow((entry.X*entry.X-c[entry.turn]),2)/(p[j]-1);
        h[j]+=pow((entry.PX*entry.PX-d[entry.turn]),2)/(p[j]-1);
    }
    for(j =0; j<=1024; j++){
        e[j]=sqrt(e[j]);
        ef[j]=sqrt(ef[j]);
        g[j]=sqrt(g[j]);
        h[j]=sqrt(h[j]);
    }
}

void emittance(UInt_t Stability_turns,UInt_t obs, UInt_t part, UInt_t num_obs,
UInt_t num_part){
    const int n=1025;
    Int_t partic[n]={0};
    Double_t EX[n]={0},EPX[n]={0}, EX2[n]={0}, EPX2[n]={0}, EXPX[n]={0},
emit[n]={0}, turn1[n], dt[n]={0},DX[n]={0}, DPX[n]={0},DX2[n]={0},DPX2[n]={0},
DXPX[n]={0}, Demit[n]={0};
    treer(Stability_turns,obs, part, num_obs, num_part);
    mean(Stability_turns,obs, part, num_obs, num_part, EX, EPX, EX2,EPX2,
DX,DPX,DX2,DPX2, partic, JX0);
    for(int i=0;i<1025;i++){
        EXPX[i]=EX[i]*EPX[i];
        DXPX[i]=(EPX[i]*DX[i]+EX[i]*DPX[i]);
        turn1[i]=(Double_t)i;
    }
    for(int i=0;i<1025;i++){
        emit[i]=sqrt(EX2[i]*EPX2[i]-EXPX[i]*EXPX[i]);
    }
}

```

```

        Demit[i]=0.5*(DX2[i]*EPX2[i]+DPX2[i]*EX2[i]/(sqrt(EX2[i]*EPX2[i]-
EXPX[i]*EXPX[i]))-2*(EX[i]*EPX[i]*EPX[i]*DX[i]+DPX[i]*EPX[i]*EX[i]*EX[i]));
    }
    ofstream f;
    f.open("Data.txt");
    f << "На 0 обороте:" << endl;
    f<< "Среднее X:\t" <<EX[0]<< "+-" <<DX[0]<<endl;
    f<< "Среднее PX:\t" <<EPX[0]<< "+-" <<DPX[0]<<endl;
    f<< "Среднеквадратичное X:\t" <<EX2[0]<< "+-" <<DX2[0]<<endl;
    f<< "Среднеквадратичное PX:\t" <<EPX2[0]<< "+-" <<DPX2[0]<<endl;
    f<< "Среднее XPX:\t" <<EXPX[0]<< "+-" <<DXPX[0]<<endl;
    f<< "Следовательно, эмиттанс равен:\t" <<emit[0]<< "+-" <<Demit[0]<<endl;
    f<< "На последнем стабильном обороте обороте:" <<endl;
    f<< "Среднее X:\t" <<EX[not]<< "+-" <<DX[not]<<endl;
    f<< "Среднее PX:\t" <<EPX[not]<< "+-" <<DPX[not]<<endl;
    f<< "Среднеквадратичное X:\t" <<EX2[not]<< "+-" <<DX2[not]<<endl;
    f<< "Среднеквадратичное PX:\t" <<EPX2[not]<< "+-" <<DPX2[not]<<endl;
    f<< "Среднее XPX:\t" <<EXPX[not]<< "+-" <<DXPX[not]<<endl;
    f << "Следовательно, эмиттанс равен:\t" <<emit[not]<< "+-" << Demit[not];
    f.close();
    TGraphErrors *gr1 = new TGraphErrors(1025, turn1, emit,dt,Demit);
    TCanvas *c1 = new TCanvas( "c1", "Graph Draw Options",200,10,600,400);
    gr1->Draw("ALP");
}

```

8.4. Приложение 4

```

#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <iomanip>
#include "TROOT.h"
#include "TDirectory.h"
#include "TFile.h"
#include "TTree.h"
#include "TGraph.h"
#include "TMultiGraph.h"
#include "TPad.h"

```

```

#include "TCanvas.h"
#include "TPad.h"
#include "TLegend.h"
#include "TString.h"
#include "TObjString.h"
#include "TCut.h"
#include "TH1F.h"
#include "TStyle.h"
#include "TMath.h"

Double_t Energy, Length, Alfa, Qx, Qy, R;      // eV, m

struct optic_t{
    Double_t S, L, BETX, BETY, ALFX, ALFY, MUX, MUY, DX, DPX, DY, DPY,
    DDX, DDPX, WX, PHIX, WY, PHIY, ANGLE, E1, E2, TILT, K1L, K2L, K3L, PHIA;
};

int treer(char* file){
    std::ifstream from;

    optic_t entry;
    entry.PHIA=0.0;
    std::string buffer;
    std::istringstream buffer_stream;
    TFile *out=0;

    if( (out=(TFile*)gROOT->GetListOfFiles()->FindObject("out")) )
delete out;

    out=new TFile("cern.root","RECREATE");

    TTree *tree=0;
    if( (tree=(TTree*)gDirectory->Get("tree")) ) delete tree;
    tree=new TTree("tree","optics data");

    tree-> Branch("optics", &entry.S, "S/D: L: BETX: BETY: ALFX: ALFY:
MUX: MUY: DX: DPX: DY: DPY: DDX: DDPX: WX: PHIX: WY: PHIY: ANGLE: E1: E2: TILT:
K1L: K2L: K3L: PHIA");

    std::cout<<"Reading "<<file<<std::endl;
    from.open(file);

    if(!from){ std::cout<<"Error opening file"<<endl; from.close(); exit(1);}
    else{
        while( from && (from.peek()!='*' || from.peek()!='$' ||
from.peek()=='@') ){
            getline(from,buffer);
        }

        while(from>>entry.S>>name_str>>type_str>>entry.L>>entry.BETX>>entry.BETY
>>entry.ALFX>>entry.ALFY>>entry.MUX>>entry.MUY>>entry.DX>>entry.DPX>>entry.DY

```



```

>>entry.DPY>>entry.DDX>>entry.DDPX>>entry.WX>>entry.PHIX>>entry.WY>>entry.PHI
Y>>entry.ANGLE>>entry.E1>>entry.E2>>entry.TILT>>entry.K1L>>entry.K2L>>entry.K
3L)
{
    if(from.fail() || from.bad()){ std::cout<<"Error
reading\n"; return 1;}
    name_str.ReplaceAll("\", "");
    name.SetString(name_str.Data());
    type_str.ReplaceAll("\", "");
    type.SetString(type_str.Data());
    entry.PHIA+=entry.ANGLE;
}
tree->Fill();
}
}
tree->Write();
out->Close();
return 0;
}
void plotdistortion(){
    TFile *out=0;
    if((out=(TFile*)gROOT->GetListOfFiles()->FindObject("out")) delete out;
    out=new TFile("cern.root", "READ");
    TTree *tree = (TTree*)out->Get("tree");
    Long64_t nentries=tree->GetEntries();
    optic_t entry;
    tree->SetBranchAddress("optics", &entry.S);
    Int_t i=0, j=0;
    Double_t *s=new Double_t[nentries];
    Double_t *betx=new Double_t[nentries];
    Double_t *bety=new Double_t[nentries];
    Double_t *phix=new Double_t[nentries];
    Double_t *phiy=new Double_t[nentries];
    Double_t *k2l =new Double_t[nentries];
    Double_t *b1=new Double_t[nentries];
    Double_t *a1=new Double_t[nentries];
    Double_t *b3=new Double_t[nentries];
    Double_t *a3=new Double_t[nentries];
}

```

```

for(i=0; i<nentries; i++){
    tree->GetEntry(i);
    s[i]=entry.S;
    betx[i]=entry.BETX;
    bety[i]=entry.BETY;
    phix[i]=TMath::TwoPi()*entry.MUX;
    phiy[i]=TMath::TwoPi()*entry.MUY;
    k2l[i]=entry.K2L;
    b1[i]=0.0;
    a1[i]=0.0;
    b3[i]=0.0;
    a3[i]=0.0;
}

for(i=0; i<nentries; i++){
    for(j=i+1; j<nentries; j++){
        b1[i]+=k2l[j]*pow(betx[j],1.5)*cos(phix[j]-phix[i]-
TMath::Pi()*Qx);
        b3[i]+=k2l[j]*pow(betx[j],1.5)*cos(3*(phix[j]-phix[i]-
TMath::Pi()*Qx));
        a1[i]+=k2l[j]*pow(betx[j],1.5)*sin(phix[j]-phix[i]-
TMath::Pi()*Qx );
        a3[i]+=k2l[j]*pow(betx[j],1.5)*sin(3*(phix[j]-phix[i]-
TMath::Pi()*Qx));
    }
    for(j=0; j<=i; j++){
        b1[i]+=k2l[j]*pow(betx[j],1.5)*cos(phix[j]-
phix[i]+TMath::Pi()*Qx );
        b3[i]+=k2l[j]*pow(betx[j],1.5)*cos(3*(phix[j]-
phix[i]+TMath::Pi()*Qx));
        a1[i]+=k2l[j]*pow(betx[j],1.5)*sin(phix[j]-
phix[i]+TMath::Pi()*Qx);
        a3[i]+=k2l[j]*pow(betx[j],1.5)*sin(3*(phix[j]-
phix[i]+TMath::Pi()*Qx));
    }
}

for(i=0; i<nentries; i++){
    b1[i]/=2.*sqrt(2.)*2.*sin(TMath::Pi()*Qx);
    a1[i]/=2.*sqrt(2.)*2.*sin(TMath::Pi()*Qx);
    b3[i]/=2.*sqrt(2.)*2.*sin(3*TMath::Pi()*Qx);
    a3[i]/=2.*sqrt(2.)*2.*sin(3*TMath::Pi()*Qx);
}

```

```

}

Double_t EX, EX2, DX2, EPX, EPX2, DPX2, EXPX, EM, EM0;

EM0=70e-12;

EX=(-1)*sqrt(2*betx[0])*EM0*b1[0];

EX2=EM0*betx[0]-EM0*EM0*betx[0]/4*5*(3*a1[0]*a1[0]+2*a1[0]*a3[0]-
3*b1[0]*b1[0]+2*b1[0]*b3[0]);

DX2=EM0*betx[0]-1/4 *EM0*EM0*betx[0]* (15*a1[0]*a1[0] + 10*a1[0]*a3[0] +
b1[0]*b1[0]+10*b1[0]*b3[0]);

EPX=sqrt(2/betx[0])*EM0*(b1[0]*alphax[0]-a1[0]);

EPX2=EM0*(1+pow(alphax[0],2))/betx[0]-5/4*EM0*EM0/betx[0]*((1-
pow(alphax[0],2))*((-3)*a1[0]*a1[0]+3*b1[0]*b1[0]-2*a1[0]*a3[0]-
2*b1[0]*b3[0])+alphax[0]*(6*a1[0]*b1[0]+4*a1[0]*b3[0]-4*b1[0]*a3[0]));

DPX2=EM0*(1+pow(alphax[0],2))/betx[0]-EM0*EM0/4/betx[0]*((1 + 15 * pow
(alphax[0],2))*a1[0]*a1[0]+4*alphax[0]*a1[0]*(7*b1[0]+5*b3[0])+10*a1[0]*a3[0]
*(pow(alphax[0],2)-1)+b1[0]*b1[0]*(15+pow(alphax[0],2))-
10*b1[0]*(b3[0]+2*alphax[0]*a3[0]-b3[0]*pow(alphax[0],2)));

EXPX=(-1)*EM0*alphax[0]+1/4*EM0*EM0*(15*a1[0]*a1[0]*alphax[0]+b1[0]*((-
10)*a3[0]+(b1[0]+10*b3[0])*alphax[0])+2*a1[0]*(7*b1[0]+5*(b3[0]+alphax[0]*a3[
0])));

EM=EM0-2*EM0*EM0*(a1[0]*a1[0]+b1[0]*b1[0]);

ofstream f;

f.open("Data.txt");

f<<"Среднее X:\t"<<EX<<endl;

f<<"Среднее PX:\t"<<EPX<<endl;

f<<"Среднеквадратичное X:\t"<<DX2<<endl;

f<<"Среднеквадратичное PX:\t"<<DPX2<<endl;

f<<"Среднее XPX:\t"<<EXPX<<endl;

f<<"Следовательно, эмиттанс равен:\t"<<EM<<endl;

f.close();

}

TGraph *gr_B1=0;

if( (gr_B1=(TGraph*)gDirectory->Get("gr_B1")) ) delete gr_B1;

gr_B1=new TGraph(nentries,s,b1);

gr_B1->SetTitle("B_{1}");

gr_B1->SetMarkerColor(kBlue);

gr_B1->SetLineColor(kBlue);

gr_B1->SetFillColor(kWhite);

gr_B1->SetMarkerStyle(7);

TGraph *gr_A1=0;

if( (gr_A1=(TGraph*)gDirectory->Get("gr_A1")) ) delete gr_A1;

gr_A1=new TGraph(nentries,s,a1);

```

```

    gr_A1->SetTitle("A_{1}");
    gr_A1->SetMarkerColor(kRed);
    gr_A1->SetLineColor(kRed);
    gr_A1->SetFillColor(kWhite);
    gr_A1->SetMarkerStyle(7);
    TMultiGraph *mg_dist1 = 0;
mg_dist1;    if( (mg_dist1=(TMultiGraph*)gDirectory->Get("mg_dist1")) ) delete
mg_dist1;
    mg_dist1= new TMultiGraph();
    mg_dist1->Add(gr_B1,"lp");
    mg_dist1->Add(gr_A1,"lp");
    TGraph *gr_B3=0;
    if( (gr_B3=(TGraph*)gDirectory->Get("gr_B3")) ) delete gr_B3;
    gr_B3=new TGraph(nentries,s,b3);
    gr_B3->SetTitle("B_{3}");
    gr_B3->SetMarkerColor(kBlue);
    gr_B3->SetLineColor(kBlue);
    gr_B3->SetFillColor(kWhite);
    gr_B3->SetMarkerStyle(7);
    TGraph *gr_A3=0;
    if( (gr_A3=(TGraph*)gDirectory->Get("gr_A3")) ) delete gr_A3;
    gr_A3=new TGraph(nentries,s,a3);
    gr_A3->SetTitle("A_{3}");
    gr_A3->SetMarkerColor(kRed);
    gr_A3->SetLineColor(kRed);
    gr_A3->SetFillColor(kWhite);
    gr_A3->SetMarkerStyle(7);
    TMultiGraph *mg_dist3 = 0;
mg_dist3;    if( (mg_dist3=(TMultiGraph*)gDirectory->Get("mg_dist3")) ) delete
mg_dist3;
    mg_dist3= new TMultiGraph();
    mg_dist3->Add(gr_B3,"lp");
    mg_dist3->Add(gr_A3,"lp");
    gStyle->SetLabelSize(0.05, "XY");
    gStyle->SetTitleSize(0.05, "XY");
    gStyle->SetTitleXOffset(0.9);
    gStyle->SetTitleYOffset(1);
    TCanvas *c_dist=0;

```

```

        if( (c_dist=(TCanvas*)gROOT->FindObject("c_dist")) ) delete c_dist;
        c_dist = new TCanvas("c_dist", "Plot", 0, 0, 1200, 680);
c_dist->Divide(1, 2);
c_dist->cd(1);
        mg_dist1->Draw("A");
        TLegend *leg1=gPad->BuildLegend(0.92, 0.8, 0.975, 0.9);
        leg1->SetFillColor(kWhite);
        mg_dist1->GetXaxis()->SetTitle("S, m");
        mg_dist1->GetYaxis()->SetTitle("B_{1}, A_{1}, m^{-2}");
        gPad->Modified();

c_dist->cd(2);
        mg_dist3->Draw("A");
        TLegend *leg3=gPad->BuildLegend(0.92, 0.8, 0.975, 0.9);
        leg3->SetFillColor(kWhite);
        mg_dist3->GetXaxis()->SetTitle("S, m");
        mg_dist3->GetYaxis()->SetTitle("B_{3}, A_{3}, m^{-2}");
        gPad->Modified();

c_dist->Update();

delete[] s;
delete[] betx;
delete[] bety;
delete[] phix;
delete[] phiy;
delete[] k2l;
delete[] b1;
delete[] a1;
delete[] b3;
delete[] a3;
}

void plot_madx(){
    char* file = "twiss.tfs";
    treer(file);
    plotdistortion();
}

```

8.5. Приложение 5

option, -echo, -info, warn, -no_fatal_stop;

TITLE, 'Example 2: SKIF.MADX';

none = 0; k1_qqm03 = 4.69871387056912; k1_qqm02 = -4.69946236750035; tbf2 = 0.0351384608523084; k1_bf2 = 0; tbd2 = 0; k1_bd2 = 4.72012908178626; k2_sd = 228.970158709681; tbd = -0.00455185444994243; k1_bd = 5.07698699415851; k2_sf = -239.511390984404; tbf = 0.0735881408987063; k1_bf = -0.756386861479606; tbdi = -0.00460736742632279; k1_bdi = 5.11946706262366; k2_sfi = -239.511390984404; tbfi = 0.0294214084257335; atme = 0.0644844319988214; tbi = 0.01485635; k1_bfi = -1.06556062300676; tbd2t = 0; k1_bd2t = 4.72012908178626; tbf2t = 0.0351384608523084; k1_bf2t = 0; k1_qqm02t = -5.12561224158368; k1_qqm03t = 6.10862805949363; k1_qqm04t = -0.29713472925691; iid2length = 59.5176094700135; ilengthl = 6.22740236750338; dds1 = 3; dds2 = 0.121702367503376; dds3 = 0.2228; dds4 = 1.0371; dds5 = 0.4458; e1length = 3.4608; dd2 = 0.2815; dd1 = 0.6489; e2length = 3.4608; dd2i = 0.2805; dd1i = 0.6749; dd0i = 0.119627738095238; idlengthr = 6.22740236750338; dds0t = 2.5768; dds1t = 0.2232; dds2t = 0.121702367503376; dds3t = 0.2228; dds4t = 1.0371; dds5t = 0.4458; idlengthl = 6.22740236750338; ilengthr = 6.22740236750338;

KSF = -239.511390984404; KSD = 228.970158709681; qf1 = 4.69871387056912; qd1 = -4.69946236750035; qf2 = 6.10862805949363; qd2 = -5.12561224158368; qd3 = -0.29713472925691;

a1.start: **marker**;

ill.start: **marker**;

qqm03: **quadrupole**, l= 0.3, k1:=qf1;

qqm02: **quadrupole**, l= 0.3, k1:=qd1;

bf2l: **sbend**, l= 0.325, angle= 0.0175692304261542, k1= 0, e1= 0.0175692304261542;

bf2r: **sbend**, l= 0.325, angle= 0.0175692304261542, k1= 0, e2= 0.0175692304261542;

bd2: **sbend**, l= 0.15, angle= 0, k1= 4.72012908178626, e1= 0, e2= 0;

ill.end: **marker**;

c1.start: **marker**;

sd: **sextupole**, l= 0.15, k2=KSD;

bd: **sbend**, l= 0.15, angle= -0.00455185444994243, k1= 5.07698699415851, e1= -0.00227592722497121, e2= -0.00227592722497121;

sf: **sextupole**, l= 0.25, k2=KSF;

bf1: **sbend**, l= 0.325, angle= 0.0183970352246766, k1= -0.756386861479606, e1= 0.0367940704493532;

bf: **sbend**, l= 0.325, angle= 0.0183970352246766, k1= -0.756386861479606;

c1.cma: **marker**;

bfr: **sbend**, l= 0.325, angle= 0.0183970352246766, k1= -0.756386861479606, e2= 0.0367940704493532;

c1.end: **marker**;

s1.start: **marker**;

bdi: **sbend**, l= 0.15, angle= -0.00460736742632279, k1= 5.11946706262366, e1= -0.00230368371316139, e2= -0.00230368371316139;

```

sfi: sextupole,l= 0.25,k2=KSF;

bf1l: sbend,l= 0.235,angle= 0.0147107042128668,k1= -1.06556062300676,e1=
0.0147107042128668;

bf1r: sbend,l= 0.235,angle= 0.0147107042128668,k1= -1.06556062300676,e2=
0.0147107042128668;

bil: sbend,l= 0.017686130952381,angle= 0.0037140875,e1= 0.007428175;

bi: sbend,l= 0.017686130952381,angle= 0.0037140875;

s1.sma: marker;

bir: sbend,l= 0.017686130952381,angle= 0.0037140875,e2= 0.007428175;

s1.end: marker;

idr1.start: marker;

bd2t: sbend,l= 0.15,angle= 0,k1= 4.72012908178626,e1= 0,e2= 0;

bf2lt: sbend,l= 0.325,angle= 0.0175692304261542,k1= 0,e1= 0.0175692304261542;
bf2rt: sbend,l= 0.325,angle= 0.0175692304261542,k1= 0,e2= 0.0175692304261542;

qqm02t: quadrupole,l= 0.3,k1:= qd2;
qqm03t: quadrupole,l= 0.3,k1:=qf2;
qqm04t: quadrupole,l= 0.2,k1:=qd3;

idr1.end: marker;

a1.cent: marker;

idl1.start: marker;

idl1.end: marker;

ir1.start: marker;

ir1.end: marker;

a1.end: marker;

rf1: rfcavity,l= 1e-005,volt= 0.75,lag= 0.5,harmon= 567;

RF3: RFCAVITY, L=0.1, HARMON=1701, VOLT=0.0, LAG=0.5;

kick1:HKICKER,L=0.25, KICK=0;
kick2:HKICKER,L=0.25, KICK=0;
kick3:HKICKER,L=0.25, KICK=0;
kick4:HKICKER,L=0.25, KICK=0;

ring: sequence, l = 8 * iid2length ;

a1.start, at = 0; a1.start, at = 0; ill.start, at = -1.33226762955019e-015;
rf1, at = 0.050005; qqm03, at = 3.15; qqm02, at = 3.57170236750337; bf2l, at =
4.10700236750337; bf2r, at = 4.43200236750337; bd2, at = 5.70660236750337;
ill.end, at = 6.22740236750337; c1.start, at = 6.22740236750338; sd, at =
6.30240236750338; bd, at = 6.58390236750338; sf, at = 7.11280236750338; bf1, at
= 7.47030236750338; bf, at = 7.79530236750338; c1.cma, at = 7.95780236750338;

```

bf, at = 8.12030236750338; bfr, at = 8.44530236750338; sf, at = 8.80280236750338; bd, at = 9.33170236750338; sd, at = 9.61320236750338; c1.end, at = 9.68820236750338; c1.start, at = 9.68820236750338; sd, at = 9.76320236750338; bd, at = 10.0447023675034; sf, at = 10.5736023675034; bfl, at = 10.9311023675034; bf, at = 11.2561023675034; c1.cma, at = 11.4186023675034; bf, at = 11.5811023675034; bfr, at = 11.9061023675034; sf, at = 12.2636023675034; bd, at = 12.7925023675034; sd, at = 13.0740023675034; c1.end, at = 13.1490023675034; s1.start, at = 13.1490023675034; sd, at = 13.2240023675034; bdi, at = 13.5045023675034; sfi, at = 14.0094023675034; bfil, at = 14.3719023675034; bfir, at = 14.6069023675034; bil, at = 14.8528731710748; bi, at = 14.8705593020272; s1.sma, at = 14.8794023675034; bi, at = 14.8882454329796; bir, at = 14.9059315639319; bfil, at = 15.1519023675034; bfir, at = 15.3869023675034; sfi, at = 15.7494023675034; bdi, at = 16.2543023675034; sd, at = 16.5348023675034; s1.end, at = 16.6098023675034; c1.start, at = 16.6098023675034; sd, at = 16.6848023675034; bd, at = 16.9663023675034; sf, at = 17.4952023675034; bfl, at = 17.8527023675034; bf, at = 18.1777023675034; c1.cma, at = 18.3402023675034; bf, at = 18.5027023675034; bfr, at = 18.8277023675034; sf, at = 19.1852023675034; bd, at = 19.7141023675034; sd, at = 19.9956023675034; c1.end, at = 20.0706023675034; c1.start, at = 20.0706023675034; sd, at = 20.1456023675034; bd, at = 20.4271023675034; sf, at = 20.9560023675034; bfl, at = 21.3135023675034; bf, at = 21.6385023675034; c1.cma, at = 21.8010023675034; bf, at = 21.9635023675034; bfr, at = 22.2885023675034; sf, at = 22.6460023675034; bd, at = 23.1749023675034; sd, at = 23.4564023675034; c1.end, at = 23.5314023675034; idr1.start, at = 23.5314023675034; bd2t, at = 24.0522023675034; bf2lt, at = 25.3268023675034; bf2rt, at = 25.6518023675034; qqm02t, at = 26.1871023675034; qqm03t, at = 26.6088047350068; qqm04t, at = 27.0820047350068; idr1.end, at = 29.7588047350067; a1.cent, at = 29.7588047350067; idl1.start, at = 29.7588047350068; qqm04t, at = 32.4356047350068; qqm03t, at = 32.9088047350068; qqm02t, at = 33.3305071025101; bf2lt, at = 33.8658071025101; bf2rt, at = 34.1908071025101; bd2t, at = 35.4654071025101; idl1.end, at = 35.9862071025101; c1.start, at = 35.9862071025101; sd, at = 36.0612071025101; bd, at = 36.3427071025101; sf, at = 36.8716071025101; bfl, at = 37.2291071025101; bf, at = 37.5541071025101; c1.cma, at = 37.7166071025101; bf, at = 37.8791071025101; bfr, at = 38.2041071025101; sf, at = 38.5616071025101; bd, at = 39.0905071025101; sd, at = 39.3720071025101; c1.end, at = 39.4470071025101; c1.start, at = 39.4470071025101; sd, at = 39.5220071025101; bd, at = 39.8035071025101; sf, at = 40.3324071025101; bfl, at = 40.6899071025101; bf, at = 41.0149071025101; c1.cma, at = 41.1774071025101; bf, at = 41.3399071025101; bfr, at = 41.6649071025101; sf, at = 42.0224071025101; bd, at = 42.5513071025101; sd, at = 42.8328071025101; c1.end, at = 42.9078071025101; c1.start, at = 42.9078071025101; sd, at = 42.9828071025101; bd, at = 43.2643071025101; sf, at = 43.7932071025101; bfl, at = 44.1507071025101; bf, at = 44.4757071025101; c1.cma, at = 44.6382071025101; bf, at = 44.8007071025101; bfr, at = 45.1257071025101; sf, at = 45.4832071025101; bd, at = 46.0121071025101; sd, at = 46.2936071025101; c1.end, at = 46.3686071025101; c1.start, at = 46.3686071025101; sd, at = 46.4436071025101; bd, at = 46.7251071025101; sf, at = 47.2540071025101; bfl, at = 47.6115071025101; bf, at = 47.9365071025101; c1.cma, at = 48.0990071025101; bf, at = 48.2615071025101; bfr, at = 48.5865071025101; sf, at = 48.9440071025101; bd, at = 49.4729071025101; sd, at = 49.7544071025101; c1.end, at = 49.8294071025101; c1.start, at = 49.8294071025101; sd, at = 49.9044071025101; bd, at = 50.1859071025101; sf, at = 50.7148071025101; bfl, at = 51.0723071025101; bf, at = 51.3973071025101; c1.cma, at = 51.5598071025101; bf, at = 51.7223071025101; bfr, at = 52.0473071025101; sf, at = 52.4048071025101; bd, at =


```
52.9337071025101; sd, at = 53.2152071025101; kick1, at = 53.513107102510105;
bd2, at = 53.8110071025101; bf2l, at = 55.0856071025101; bf2r, at =
55.4106071025101;qqm02, at = 55.9459071025101; qqm03, at = 56.3676094700135;
kick2, at = 56.7426094700135; RF3, at = 56.9176094700135;
```

...Здесь я опускаю дальнейшее описание структуры, т.к. она периодична. Все нужные нам элементы перечислены выше...

```
endsequence;

seqedit, sequence=ring;

install, element=A2.START ,class=MARKER, at = QQM03->L/2+3,from=QQM03[4];
install, element=A2.MIDDLE,class=MARKER, at = QQM03->L/2+3,from=QQM03[6];
install, element=A2.END ,class=MARKER, at = QQM03->L/2+3,from=QQM03[8];
seqedit, sequence=ring;

    flatten;

    extract, sequence=ring, from=A2.START, to=A2.MIDDLE, newname=one_supercell;
endedit;

Nrj=3;           //Beam energy, GeV

beam,particle=electron,energy=Nrj, radiate=false;

set, format="22.15g";

select, flag=twiss, clear;

select, flag=twiss, COLUMN=S, NAME, keyword, L, BETX, BETY, ALFX, ALFY, MUX, MUY, DX, DPX,
DY, DPY, DDX, DDPX, WX, PHIX, WY, PHIY, ANGLE, E1, E2, TILT, K1L, K2L, K3L;

USE, period=FODO;

TWISS, file="twiss.tfs";

PLOT, TABLE=TWISS, TITLE="FODO", HAXIS=S, VAXIS1=BETX, BETY, VAXIS2=DX, COLOUR=100, I
NTERPOLATE, -noverision;

delete, table=tracksumm;

i=0;

EM=70e-12;

betx= 14.2439966206815;

    use, sequence=FODO;

    ptc_create_universe;

    ptc_create_layout, time=true, model=2, exact=false, method=6, nst=10;

        while (i<100000) {

            ptc_start, x=sqrt (EM*betx) *GAUSS (), px=sqrt (EM/betx) *GAUSS (),
y=0, py=0, pt=0, T=0;

            i=i+1;

        };
```

```
ptc_track, icas=4, closed_orbit, dump, maxaper={100,100,100,100,100,100},
turns=1025, ffile=1;

ptc_track_end;

ptc_end;

write, table=tracksumm, file="ptc_table.tfs";

Value, TABLE (SUMM, Q1);

Value, TABLE (SUMM, Q2);

return;
```