

Санкт-Петербургский государственный университет

**Кафедра компьютерного моделирования и
многопроцессорных систем**

Низова Диана Александровна

Выпускная квалификационная работа бакалавра

**Реализация алгоритма обработки
информации при испытании магнитов**

Направление 010400

Прикладная математика и информатика

Научный руководитель,
доктор физ.-мат. наук
профессор
Андреанов С.Н.

Санкт-Петербург

2016

Содержание

Введение	3
Постановка задачи	5
1 Глава 1. Формализация задачи	6
1.1. Терминология теоретической части	6
1.2. Терминология для практической части	8
2 Глава 2. Теоретическая часть	11
2.1. Принцип работы нуклотрона и бустера	11
2.2. Процесс получения экспериментальных данных магнитов . .	13
2.3. Алгоритм обработки данных	16
3 Глава 3. Разработка программы	18
3.1. Выбор языка программирования	18
3.2. Реализация алгоритма	20
3.3. Результаты работы программы	22
Вывод	26
Список литературы	27
Приложение	29

Введение

Современное технологическое развитие позволило добиться многих результатов. Наука в ускоренном режиме отвечает на огромное количество интересующих нас вопросов. Одним из действительно сложных вопросов является интерес к развитию вселенной. С чего все начиналось? На данный момент пусть мы и далеки от точного ответа, но предпринимается все для того, чтобы хотя бы приблизиться к нему. Открываются новые, неизвестные до этого частицы, проводятся прогрессивные эксперименты, о которых мы не задумывались еще десять-двадцать лет назад. Так, в ЦЕРНе [1] (англ. CERN — Европейская организация по ядерным исследованиям, находящая вблизи Женевы) был запущен большой адронный коллайдер, работа на котором дала возможность сделать массу потрясающих открытий, например, открытие Бозона Хиггса [2] и других новых, теоретически предсказанных частиц. Но такие масштабные проекты реализуются не только за рубежом. В Объединенном Институте Ядерных Исследований (ОИЯИ) [3] города Дубна производится создание на основе сверхпроводящего протонного синхротрона Нуклотрон нового ускорительного комплекса Nuclotron-based Ion Collider fAcility NICA. [4]

Основной целью проекта является проведение в ближайшие 5-7 лет экспериментов по изучению сильного взаимодействия в горячей и плотной кварк-глюонной материи и поиск возможного образования так называемой «смешанной фазы» такой материи. На пучках ионов, ускоренных в Бустере NICA, который входит в состав комплекса, будет выполняться программа прикладных и фундаментальных исследований [5].

Структурными элементами синхротрона Бустер NICA и коллайдера NICA являются сверхпроводящие (СП) магниты с полем, формируемым железным ярмом. В ЛФВЭ (лаборатория физики высоких энергий) ОИЯИ происходят сборка и испытания данных магнитов. Программа тестов включает измерения параметров магнитного поля при температуре окружающей среды и при рабочей температуре магнита — 4,5 К. Проводятся испытания дипольного магнита, в том числе теплые (при комнатной тем-

пературе) и холодные (при температуре жидкого гелия) магнитные измерения. Целью магнитных измерений является раннее обнаружение дефектов в магните, проверка качества его изготовления и сборки.

Постановка задачи

Целью данной работы является улучшение процесса обработки информации, получаемой при испытании магнитов для проекта NICA. Для того, чтобы добиться поставленной цели, необходимо решить задачу разработки программы, которая реализует алгоритм обработки информации. Решение задачи разделено на несколько этапов:

1. Изучить состав будущего комплекса ускорителей NICA и принципы работы важных для поставленной цели компонентов, в частности, бустера и коллайдера.
2. Рассмотреть алгоритм, с помощью которого будет реализована программа.
3. Выбрать язык программирования для реализации и удобную для него среду программирования.
4. Реализовать программу на выбранном языке.
5. Проверить корректность выполнения программы на экспериментальных данных.

Поставленная задача решается на основе тестовых данных, предоставленных лабораторией магнитных испытаний Объединенного Института Ядерных Исследований города Дубна.

1 Глава 1. Формализация задачи

В этой главе будет подробнее рассмотрена поставленная задача, введены необходимые термины для объяснения используемых физических процессов, определены основные понятия для практической реализации проекта.

1.1. Терминология теоретической части

- В 2008 году была образована Лаборатория физики высоких энергий (ЛФВЭ), объединившая коллективы Лаборатории высоких энергий и Лаборатории физики частиц для концентрации усилий на создание проекта NICA [6].

С помощью NICA ученые будут изучать особое состояние материи, в котором пребывала наша Вселенная вскоре после Большого взрыва — кварк-глюонную плазму.

Поясим несколько необходимых терминов, которые будут использованы в дальнейшем.

Нуклотрон [7] - базовая установка ОИЯИ, предназначенная для получения многозарядных ионов (ядер) с энергией до 6 ГэВ на нуклон, пучков протонов, а также поляризованных дейтронов.

Ускоритель Нуклотрон был сооружен в течение 1987-92 годов. Он создан на основе уникальной технологии сверхпроводящих магнитов, предложенной и развитой в Лаборатории высоких энергий, которая в настоящее время носит имя академиков В.И.Векслера и А.М.Балдина. Конструкторские разработки, испытания и монтаж элементов Нуклотрона целиком выполнены силами коллектива Лаборатории. Производство структурных элементов осуществлено в Центральных экспериментальных мастерских ОИЯИ.

Инжектор [8] - ускоритель (обычно линейный) для ввода заряженных частиц в основной ускоритель. Энергия, сообщаемая частицам в инжекторе, должна превышать минимальную, необходимую для начала работы

основного ускорителя.

Сверхпроводящий магнит — электромагнит, в котором ток, создающий магнитное поле, протекает в основном по сверхпроводнику, вследствие чего омические потери в обмотке сверхпроводящего магнита весьма малы. Сверхпроводящий магнит приобретает свои сверхпроводящие свойства только при низких температурах, поэтому его помещают в сосуд, заполненный жидким гелием, который в свою очередь помещен в сосуд с жидким азотом (чтобы минимизировать испарение жидкого гелия). На Рис.1 показан сверхпроводящий магнит [9].

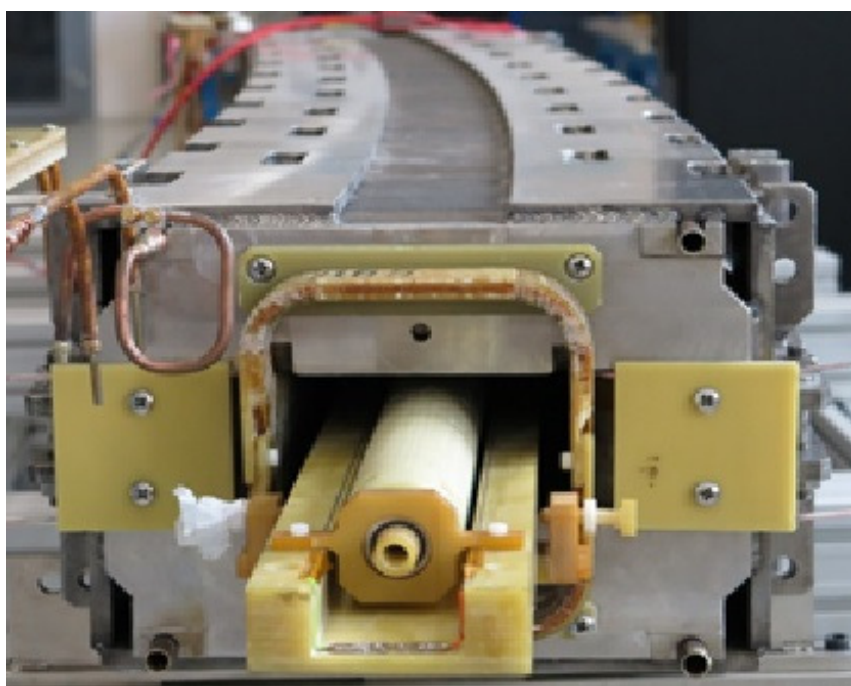


Рис. 1: Сверхпроводящий магнит.

Квадрупольная линза [10] — устройство для фокусировки пучков заряженных частиц с помощью магнитного или реж*е электрического поля квадрупольной конфигурации.

Электродвижущая сила (ЭДС) [11] — скалярная физическая величина, характеризующая работу сторонних сил, то есть любых сил не электрического происхождения, действующих в цепях постоянного или переменного тока.

Аналого-цифровой преобразователь (АЦП) — устройство, преобразующее входной аналоговый сигнал в дискретный код.

1.2. Терминология для практической части

Рассмотрим терминологию, связанную с практической реализацией.

Среда программирования [12] — это набор инструментов, которые используются для преобразования символов в выполнимые вычисления. Компоненты среды программирования:

1. Редактор — это средство для создания и изменения исходных файлов, которые содержат написанную на языке программирования программу. Условно редакторы делятся на два типа. Первый тип работает с последовательностью символов в текстовых файлах и обеспечивают расширенную функциональность — подсветку синтаксиса, сортировку строк, конвертацию кодировок, показ кодов символов и т.п. Часто такие редакторы называют редакторами кода, поскольку основное их назначение — это написание исходных кодов компьютерных программ.
2. Компилятор — транслирует символы из исходного файла в объектный модуль, который содержит команды в машинном коде для конкретного компьютера.
3. компоновщик или редактор связей — собирает объектные файлы отдельных компонентов программы и разрешает внешние ссылки от одного компонента к другому, формируя исполняемый файл.
4. Загрузчик — копирует исполняемый файл с диска в память и инициализирует компьютер перед выполнением программы.
5. Отладчик — это средство, которое дает возможность программисту управлять выполнением программы на уровне отдельных операторов для диагностики ошибок. Позволяет выполнять пошаговую трассировку (пошаговое выполнение программы с остановками на каждой

команде или строке), отслеживать, устанавливать или изменять значения переменных в процессе выполнения программы, устанавливать и удалять контрольные точки или условия останова и т.д.

6. Средства тестирования – автоматизирует процесс тестирования программ, создавая и выполняя тесты и анализируя результаты тестирования.
7. Интерпретатор – выполняет исходный код программы в отличие от компилятора, переводящего исходный файл в объектный.

Библиотека в программировании — сборник подпрограмм или объектов, используемых для разработки программного обеспечения (ПО). В зависимости от назначения различают:

- библиотеки, используемые одной программой и содержащие критические для работы программы функции;
- библиотеки, используемые одной программой и содержащие дополнительные функции. Например, библиотеки плагинов используются для расширения функциональности программы;
- библиотеки общего пользования. Содержат функции, используемые несколькими программами. Могут загружаться в адресное пространство ОС для экономии памяти: одна копия библиотеки будет использоваться несколькими процессами.

При написании программы достаточно указать транслятору (компилятору или интерпретатору) путь к библиотеке и имя функции.

Метод трапеций — метод численного интегрирования функции одной переменной, заключающийся в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, то есть линейную функцию. Площадь под графиком функции аппроксимируется прямоугольными трапециями (Рис.2). Если отрезок $[a, b]$ является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по формуле:

$$\int_a^b f(x)dx \approx \frac{f(a)+f(b)}{2}(b-a) + E(f)$$

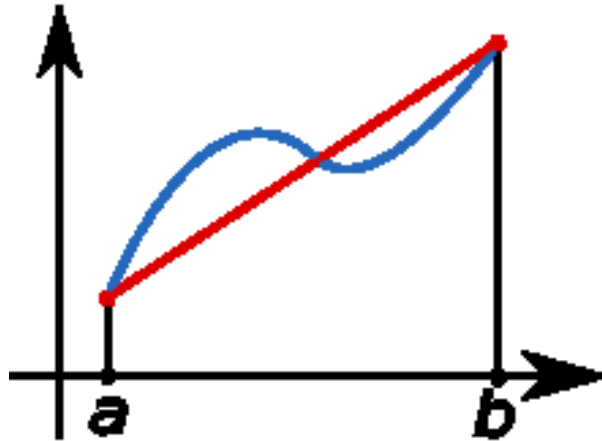


Рис. 2: Метод трапеций.

Ряд Фурье [13]— представление произвольной функции f с периодом τ в виде ряда:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} A_k \cos(2\pi \frac{k}{\tau} x + \theta_k)$$

2 Глава 2. Теоретическая часть

В данной главе будет исследована предметная область, процессы, происходящие в ускорителях, а также разбор нескольких этапов получения необходимой информации для дальнейшей её обработки.

2.1. Принцип работы нуклотрона и бустера

К 2020 году планируется построить установку, которая будет содержать несколько взаимосвязанных ускорителей. Состав ускорительного комплекса NICA (Рис.3) [14]:

- инжекционный комплекс;
- сверхпроводящий синхротрон;
- Бустер;
- существующий сверхпроводящий синхротрон Нуклотрон;
- коллайдер, состоящий из двух сверхпроводящих колец с двумя точками встречи пучков;
- каналы транспортировки пучков.

Наибольший интерес для нас представляют бустер и Нуклотрон NICA. Начнем с рассмотрения генератора частиц.

Структура инжекционного комплекса состоит из следующих элементов:

- криогенный источник тяжёлых ионов;
- лазерный источник;
- дуоплазматрон;
- источник поляризованных протонов и дейтронов;

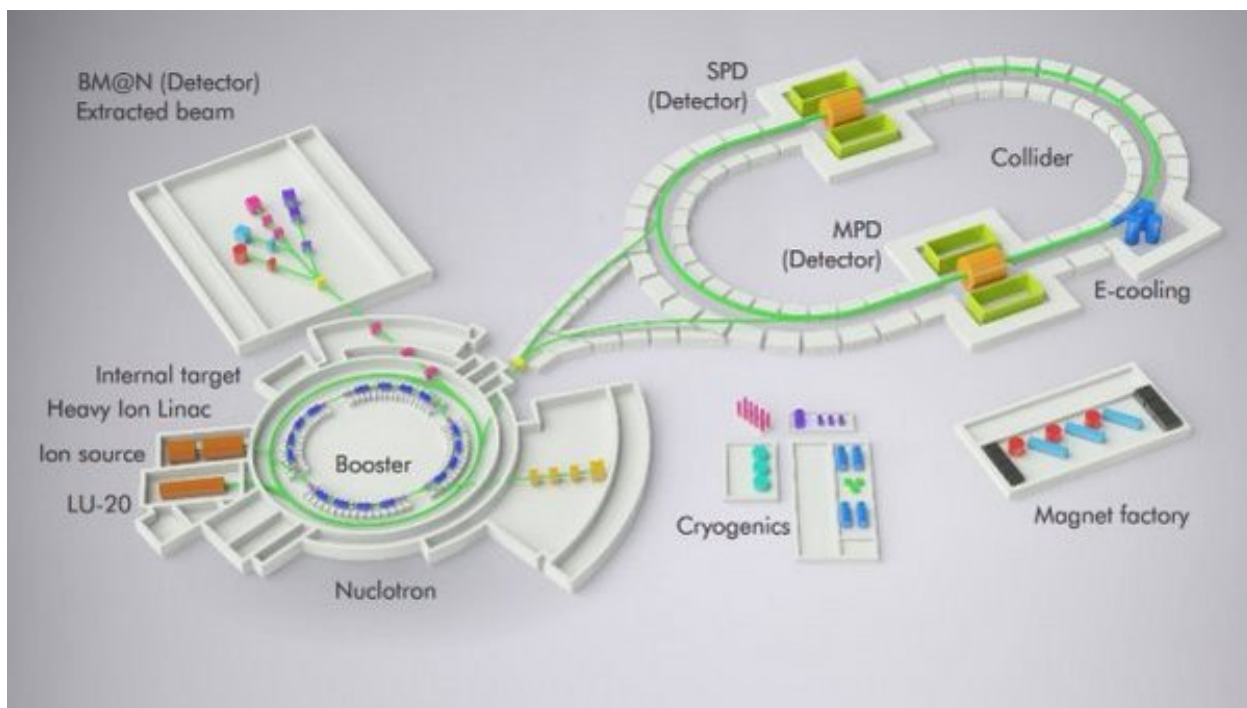


Рис. 3: Расположение бустера и Нуклотрона проекта NICA.

- модернизированный линейный ускоритель ЛУ-20;
- новый тяжёлоионный линейный ускоритель (HILAC);
- каналы транспортировки пучков.

В инжекторе генерируются частицы, в данном случае, атомы золота. Также частицы проходят так называемую «обдирку» с помощью двух сеток разных полярностей и на выходе получают частицы Au^{78} .

Следующий шаг — направление частиц в Бустер. Канал транспортировки пучка от инжектора до Бустера собирается с использованием дипольных и квадрупольных магнитов ускорителя Amps, перевезённых в ОИЯИ. Основными задачами Бустера являются:

- ускорение тяжёлых ионов до нужной энергии;
- формирование требуемого фазового объёма пучка с использованием системы электронного охлаждения;
- быстрый вывод ускоренного пучка в Нуклотрон;

- накопление $2 * 10^9$ ионов Au.

Далее частицы линейно ускоряются в бустере, состоящем из сверхпроводящих магнитов и квадрупольных линз. При достижении нужного ускорения частиц, их необходимо направить из бустера в Нуклотрон.

Сверхпроводящий синхротрон Нуклотрон предназначен для работы в трёх основных режимах:

1. Ускорение тяжёлых ионов для инъекции в Коллайдер.
2. Ускорение пучков поляризованных протонов и дейтронов для инъекции в Коллайдер.
3. Ускорение протонов, дейтронов (поляризованных и неполяризованных) и тяжёлых ионов для экспериментов на внутренней мишени или медленного вывода для экспериментов на фиксированных мишенях.

Таким образом, данные элементы проекта NICA несомненно имеют огромную важность, и их выход из строя понесет за собой большой временной простой всей системы и потерю огромных денежных средств, необходимо, чтобы магниты, из которых состоят эти элементы, были надлежащим образом проверены.

2.2. Процесс получения экспериментальных данных магнитов

Существует два типа измерений для магнитов: теплые (при комнатной температуре) и холодные (при температуре жидкого гелия, т.е. при 4,5 К). По последовательности действий эти виды измерений почти не отличаются друг от друга. Главное отличие — число точек видов величин в холодных измерениях примерно в 25 раз больше, чем в теплых. Основные этапы проведения измерений:

- включение источника и его выход в рабочий режим;

- проверка всех соединений питания обмоток магнита для устранения возможных коротких замыканий и нарушения изоляции;
- подключение на коммутационной панели нужного вида цикла. Циклов существует три вида: измерение электродвижущей силы (ЭДС) датчика в основном поле без компенсации; измерение ЭДС датчика в основном поле с компенсацией; измерение ЭДС в реперном поле;
- в программе управления шаговым двигателем вводятся необходимые характеристики;
- в основной программе выставляются пределы, количество шагов двигателя и имя файла куда будет записываться цикл. Запуск программы;
- ЭДС индукции с датчиков оцифровывается каналами АЦП с одиночных катушек (некомпенсированные сигналы), либо с катушек включенных навстречу друг другу, т.е. в режиме компенсации. Синхронно с измерением ЭДС оцифровывается сигнал с датчика тока $I(t)$. Сервомотор поворачивает датчик на угол 9 градусов пока не сделает полный оборот и не придет в стартовую точку;
- запись файла заканчивается, сервомотор останавливается, система готова к следующему циклу.

Для режима компенсации, когда катушки включаются встречно, необходимо всего лишь переключить контакты на коммутационной панели, поменять пределы графиков в программе и повторить измерения.

Для измерения реперного поля необходимо переключить клеммы источника с основной обмотки магнита на реперную – соблюдая полярность. Так же поменять пределы в программе.

После того как программа записала все три цикла, набор этих циклов можно назвать одним полным измерением. Все измерения записываются в одну папку с датой измерения. Формат данных- TDMS.

Файл формата TDMS (technical data management streamer) структурирован с использованием трех уровней иерархии - файла, группы и канала (Рис.4) [15]. Уровень файла может содержать неограниченное количество групп, и каждая группа может содержать неограниченное количество каналов. Каждый объект TDMS однозначно идентифицируется по пути. Каждый файл TDMS содержит два типа информации: мета данные и исходные данные [16]. Мета данные - это описывающая информация, которая хранится в объектах или свойствах. Массивы данных, подключенные к каналу объектов, называются исходными данными. TDMS файлы содержат исходные данные для нескольких каналов в одном непрерывном блоке. Для того, чтобы иметь возможность извлечь исходные данные из этого блока, TDMS файлы используют индекс, содержащий информацию о блоке данных. В ее число входит канал, соответствующий этим данным, сумма значений, которые блок содержит для этого канала, и порядок, в котором данные были сохранены.

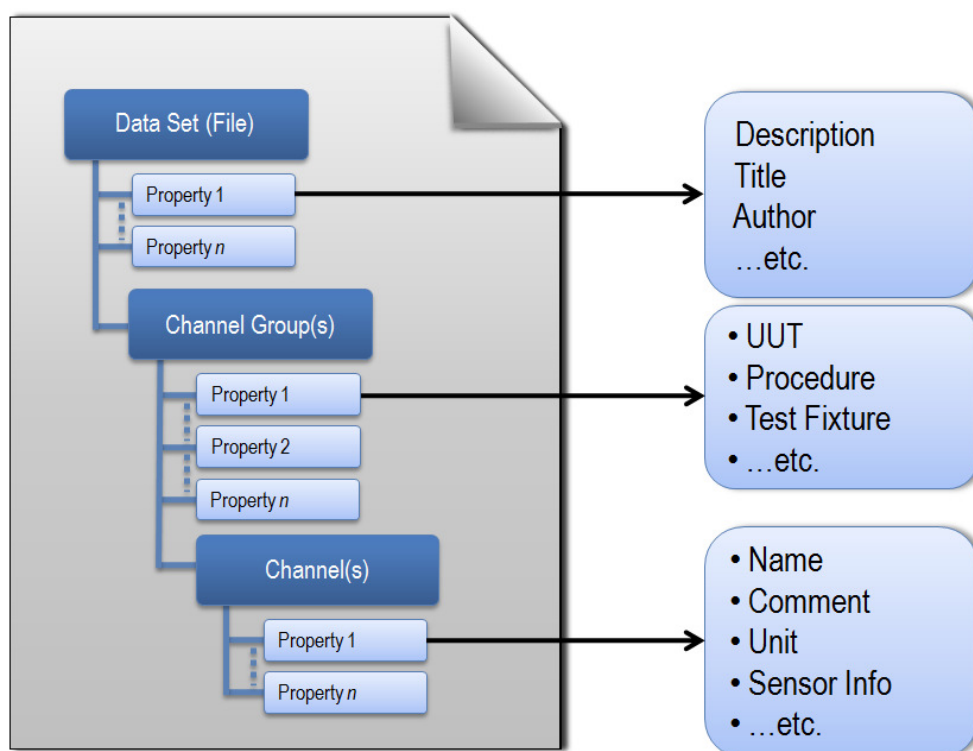


Рис. 4: Структура файла формата TDMS.

Далее производится анализ и обработка данных.

2.3. Алгоритм обработки данных

Основные этапы работы программы:

1. Оцифрованные сигналы с каждой катушки интегрируются численно.
2. Вычисляются отношения.
3. Для M значений функций $F_i(\theta)$ вычисляются коэффициенты Фурье.
4. Из полученных коэффициентов Фурье вычисляются гармоники магнитного поля.
5. Вычисляются начальные угловые положения катушек в реперном поле.
6. Вычисляются начальные угловые положения катушек в основном магнитном поле.
7. Вычисляются начальные угловые положения катушек относительно ярма магнита.
8. Вычисляются интегральные гармоники.

Рассмотрим алгоритм обработки информации подробнее. Сначала считываем с сохраненных файлов экспериментальные данные, такие как значения тока и напряжение. Именно эту информацию в дальнейшем необходимо преобразовать. Вводим следующие переменные:

- число отсчетов АЦП на цикле тока(в нашем случае 14800);
- число шагов датчиков за один оборот (в нашем случае 40);
- счетчик времени;
- время дискретизации;

- коэффициент передачи трансформатора тока.

Считываем с файла силу току в магните, умножая её на коэффициент передачи трансформатора тока, и напряжение с датчика. Производим нормировку магнитного потока на ток. Находим интеграл напряжения по времени:

$$\int U_m dt \approx \text{delta}T * \left(\frac{U_{0,m} + U_{N_2,m}}{2} + \sum_{n=1}^{N_2} U_{n,m} \right)$$

Полученные значения необходимо разложить в ряд Фурье с помощью следующих формул:

$$\begin{aligned} A_0 &= \frac{1}{M} \sum_i \int U_i dt; \\ B_j &= \frac{2}{M} \sum_i (\int U_i dt * \cos(j * x_i)); \\ A_j &= \frac{2}{M} \sum_i (\int U_i dt * \sin(j * x_i)); \end{aligned}$$

После вычисления, вводим такие данные как радиус приведения и параметры катушки.

Далее вычисляем гармоники магнитного поля.

Результаты записываем в файл.

Данный цикл необходимо провести трижды: в реперном поле, в основном поле с компенсацией, в основном поле без компенсации.

После, используя полученные результаты, рассчитываем следующие величины L_{eff} , α и a_n, b_n . Где:

- L_{eff} (Рис.5) считается по формуле:

$$L_{eff} = \frac{\int_{-\infty}^{+\infty} B_y ds}{B_y(0)};$$

- α - угол поворота магнитного поля;
- a_n и b_n -гармоники магнитного поля.

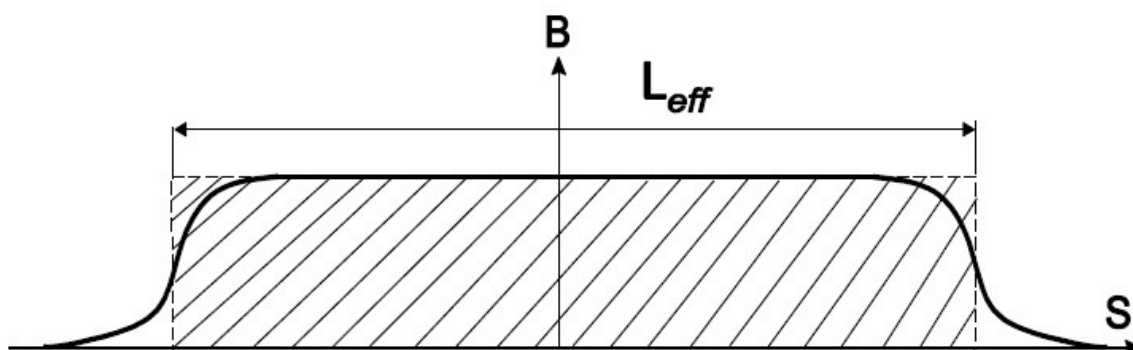


Рис. 5: Графическое представление L_{eff} .

3 Глава 3. Разработка программы

На основании информации, описанной в предыдущей главе, разработаем программу, обрабатывающую данные, полученные в процессе испытания магнитов. В этой главе рассматриваются основные этапы реализации и результаты работы программы.

3.1. Выбор языка программирования

На данный момент существует огромное количество языков программирования. Многие из них созданы для решения определенных узкоспециализированных задач. При выборе языка программирования для данной задачи необходимо было обратить внимание на достаточную распространенность языка, содержание необходимых средств для реализации идей, достаточная быстрота выполнения написанного кода и возможность обработки действительно больших объемов данных. Также нужно выбрать такой язык программирования, для которого существует удобная среда программирования для компиляции и поиска ошибок в процессе написания кода.

В результате наиболее подходящим по данным параметрам оказался

язык C++. C++ - это компилируемый, статически типизированный язык программирования общего назначения. Он является одним из самых популярных и распространенных языков программирования. Широко применяется при разработке программного обеспечения.

Язык создал сотрудник фирмы Bell Labs Бьерн Страуструп в начале 1980-х годов. На основе языка C с помощью добавления возможности работы с классами и объектами появился язык C++. Международный стандарт этого языка появился только в 1998 году. На данный момент язык программирования C++ является свободным, т.е. не существует компании или человека, владеющего правами на него.

Также C++ имеет неоспоримое преимущество перед другими языками в том, что в дальнейшем при обработке огромных объемов информации есть возможность использовать параллельные процессы для ускорения работы программы. Например, использовать библиотеку OpenMP [17], которая позволяет запускать параллельно определенные участки кода на многопроцессорных системах.

Для языка программирования C++ существует огромное множество различных сред программирования. Так как нет необходимости использовать узконаправленные среды для определенных задач, то прежде всего обратить внимание следует на удобство использования и легкость выявления ошибок. Для работы с C++ можно обратить внимание на среду программирования компании Microsoft Visual Studio [18]. Visual Studio имеет интуитивно понятный интерфейс, возможность подключать и использовать сторонние библиотеки, встраиваемые дополнения (плагины) (Рис. 6). В полной версии данный продукт платный, но существует урезанная бесплатная версия Visual C++ Express, которой достаточно для разработки программы под наши задачи.

Таким образом язык C++ в среде программирования Visual Studio является достойным выбором для реализации программы обработки информации, обладающим всеми важными свойствами.

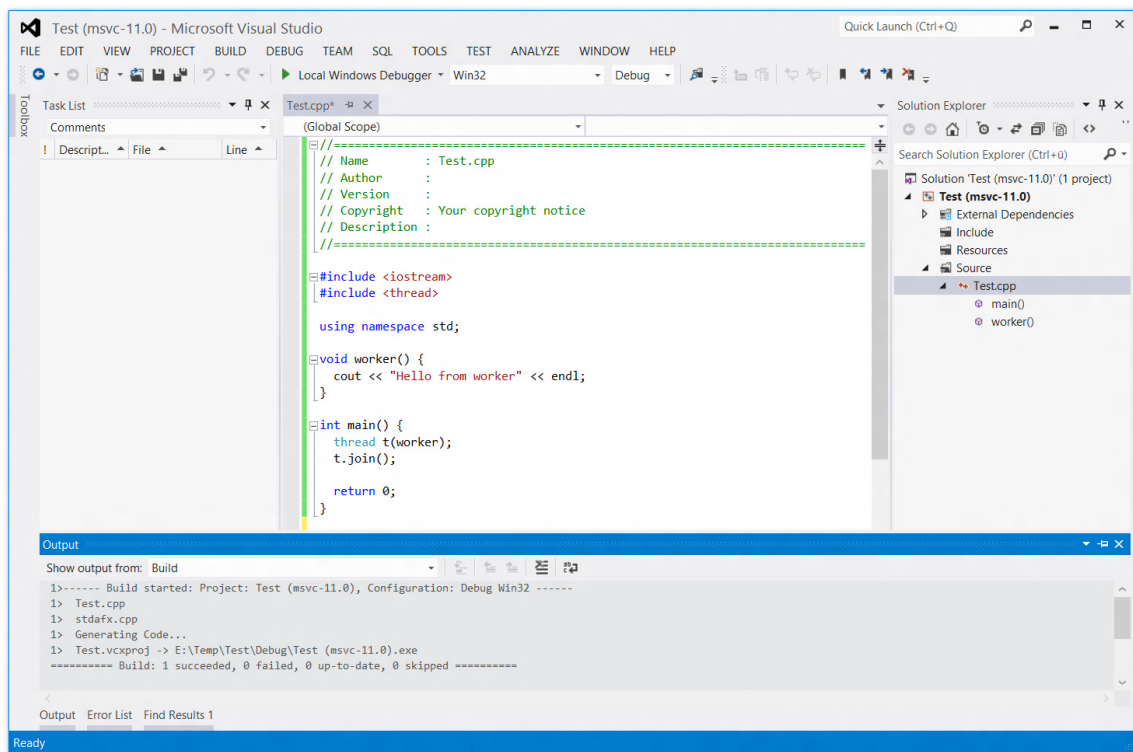


Рис. 6: Пример интерфейса Visual Studio.

3.2. Реализация алгоритма

Реализация алгоритма разделилась на 2 части:

1. Перевод файлов формата TDMS в вид, удобный для работы с ним в основной программе.
2. Написание основного кода программы, обрабатывающего полученную информацию.

Рассмотрим первую часть реализации. Перевод файлов формата TDMS в удобно читаемую форму возможен несколькими способами :

1. Подключить библиотеку для работы с TDMS файлами из пакета LabWindows компании National Instrument [19].
2. Перевести TDMS файл с помощью сторонней программы в бинарный вид и далее использовать стандартные функции для чтения файлов языка программирования C++.

Была предпринята попытка использовать первый способ, но при подключении библиотеки с помощью дополнительных средств возникали неожиданные ошибки. Библиотека была подключена, но не читалась, использование функций из нее было невозможно. Была проверена совместимость версий, произведен поиск решения в сети Internet, но в итоге эта попытка не увенчалась успехом. Тогда был предложен второй способ, с помощью пакета LabView была создана программа для перевода файла TDMS формата в бинарный файл (Рис.6), так как пакет LabView содержит обширную библиотеку для работы с файлами данного формата. Далее бинарный файл сохраняется и его чтение производится с помощью стандартных функций языка C++.

Была решена первая часть вопроса, и появилась возможность работать с измерениями напрямую.

Следующим шагом стала реализация самого алгоритма, приведенного в предыдущей главе. Рассмотрим ход программы на примере обработки файла с данными, полученными в режиме без компенсации.

Необходимо проинтегрировать данные из файла методом трапеций. Считаем интеграл напряжения по времени, то есть магнитный поток проходящий через катушку.

```
double *Integral_U = new double[41];
for (int m = 0; m < 41; ++m) {
    I[m][N2] = I[m][N2] * K_tr;
    for (int k = 0; k < N; ++k){
        I[m][k] = I[m][k] * K_tr;
        U[m][k] = U[m][k] / I[m][N2]; }
    Integral_U[m] = (U[m][0] + U[m][N2])/2;
    for (int k = 1; k < N2; ++k)
        Integral_U[m] = Integral_U[m] + U[m][k];
    Integral_U[m] = Integral_U[m] * deltaT * 10000; }
```

Полученные результаты раскладываем в ряд Фурье.

```
for (int j = 1; j <= M2; ++j) {
```

```

A[j] = 0;
B[j] = 0;
for (int i = 0; i <= M - 1; ++i) {
A[j] = A[j] + Integral_U[i] * sin(j * 2 * M_PI * i / M);
B[j] = B[j] + Integral_U[i] * cos(j * 2 * M_PI * i / M); }
A[j] = A[j] * 2 / M;
B[j] = B[j] * 2 / M;

```

Далее считаем гармоники магнитного поля (a,b), угол магнитного поля (theta) и эффективную длину (d):

```

double R_ref = 0.03, L_coil = 0.4291;
double R2 = 0.0233125, R1 = 0.0094375;
double w = 400;

```

```

for (int n = 1; n < M2; ++n){
a[n] = (-A[n] * n) / (L_coil * w * R_ref * (pow(R2 / R_ref, n)-
- pow(R1 / R_ref, n)));
b[n] = (B[n] * n) / (L_coil * w * R_ref * (pow(R2 / R_ref, n)-
- pow(R1 / R_ref, n)));
d[n] = sqrt(a[n] * a[n] + b[n] * b[n]);
theta[n] = atan2(a[n], b[n]) / M_PI * 180; }

```

3.3. Результаты работы программы

Результаты рассмотрим также на примере обработки файла с данными, полученными в режиме без компенсации. Численные результаты приведены ниже в виде таблиц.

Результаты интегрирования (Табл.1).

Результаты разложения в Ряд Фурье (Табл.2).

Результаты выполнения алгоритма (Табл.3), где a_n, b_n -гармоники магнитного поля, θ_n -угол магнитного поля, d_n -эффективная длина. Все результаты были проверены, данные обработаны верно.

Таблица 1: Результат интегрирования.

m	Integral _U [m]	m	Integral _U [m]
0	2.37932	21	-2.36494
1	2.36302	22	-2.29132
2	2.28983	23	-2.1616
3	2.16067	24	-1.97862
4	1.97774	25	-1.74679
5	1.74707	26	-1.47156
6	1.4739	27	-1.16121
7	1.16366	28	-0.82137
8	0.82432	29	-0.46118
9	0.465854	30	-0.0894182
10	0.0949246	31	0.283848
11	-0.277695	32	0.650157
12	-0.644502	33	1.00083
13	-0.994475	34	1.32596
14	-1.32084	35	1.61921
15	-1.61471	36	1.8721
16	-1.86845	37	2.07865
17	-2.07665	38	2.23463
18	-2.23326	39	2.33521
19	-2.33519	40	2.37845
20	-2.37939		

Таблица 2: Разложение в ряд Фурье.

j	A[j]	j	B[j]
0	0	0	0
1	0.0922451	1	2.37876
2	-0.00105136	2	-0.00153625
3	-0.000193666	3	9.57376e-005
4	-2.62923e-005	4	-4.12119e-005
5	-3.4644e-005	5	0.00017039
6	-8.3282e-005	6	3.95495e-005
7	7.93939e-005	7	7.15972e-005
8	2.81544e-005	8	6.37866e-005
9	-1.54012e-005	9	3.84408e-005
10	-3.96648e-005	10	-7.70623e-005
11	-4.11136e-005	11	6.95329e-005
12	-0.000129219	12	0.000108684
13	1.4066e-005	13	3.93823e-005
14	1.71024e-005	14	5.08384e-005
15	7.13005e-005	15	4.47683e-005
16	9.11315e-005	16	1.92807e-005
17	-4.31057e-005	17	8.24777e-005
18	-3.08355e-005	18	0.000127872
19	7.86902e-005	19	-1.01947e-005
20	4.57615e-015	20	2.7593e-005

Таблица 3: Разложение в ряд Фурье.

n	a_n	b_n	d_n	θ_n
0	0	0	0	0
1	-0.0387339	0.998845	0.999596	-2.22074
2	0.000808798	-0.00118182	0.00143208	145.613
3	0.00025754	0.000127314	0.00028729	63.6948
4	5.75574e-005	-9.02186e-005	0.000107015	147.463
5	0.000120024	0.000590316	0.000602394	11.4928
6	0.000442663	0.000210214	0.000490041	64.5976
7	-0.000631898	0.000569844	0.000850892	-47.9559
8	-0.000329207	0.000745851	0.000815274	-23.8159
9	0.000260602	0.00065045	0.000700713	21.8334
10	0.00095949	-0.00186414	0.00209657	152.765
11	0.00140772	0.00238079	0.00276583	30.5951
12	0.00621103	0.00522403	0.00811587	49.9332
13	-0.00094254	0.00263894	0.00280221	-19.655
14	-0.00158818	0.00472101	0.00498099	-18.5933
15	-0.00912915	0.00573203	0.0107795	-57.876
16	-0.0160165	0.00338861	0.016371	-78.0541
17	0.0103584	0.0198197	0.0223633	27.5931
18	0.0100964	0.0418689	0.043069	13.5577
19	-0.0349986	-0.00453425	0.0352911	-97.3819

Вывод

В результате выполнения данной работы был рассмотрен проект комплекса ускорителей NICA. Были изучены процессы, которые будут реализованы в этом проекте. Выявлены важность разработки комплекса ускорителей и необходимость точных расчетов.

Рассмотрен алгоритм, по которому была реализована программа.

Выбран наиболее удобный и подходящий для достижения цели работы язык программирования.

Алгоритм реализован на языке программирования C++ в среде программирования Visual Studio.

Программа проверена на экспериментальных данных, получены и предоставлены корректные результаты.

В дальнейшем планируется создать графический интерфейс, ввести параллельный алгоритм, который позволит с достаточной скоростью обрабатывать информацию в больших объемах, усложнить алгоритм для повышения точности вычислений.

Таким образом, были получены следующие результаты:

1. Изучены принципы работы важных для решения задачи элементов проекта NICA -бустера и Нуклотрона.
2. Реализован описанный в предыдущих главах алгоритм.
3. Получены корректные результаты.

Список литературы

1. Information about CERN. <http://home.cern/about>
2. Higgs boson. <http://press.cern/press-releases/2012/07/cern-experiments-observe-particle-consistent-long-sought-higgs-boson>
3. Информация об Объединенном Институте Ядерных Исследований. <http://www.jinr.ru/about/>
4. Технический проект NICA. <http://nucloweb.jinr.ru/nica/TDR/partaI.pdf>
5. Magnetic measurement system for the NICA booster magnets. <http://eraper.kek.jp/IPAC2014/papers/wepri088.pdf>
6. Ускорительный комплекс NICA. http://nica.jinr.ru/files/NICA_booklet.pdf
7. Нуклотрон. <http://nucloserv.jinr.ru/>
8. Информация об инжекторе. <http://megabook.ru/article/injector>
9. Определение сверхпроводящего магнита. <http://elementy.ru/lib/431987>
10. Quadrupole Magnets. <http://uspas.fnal.gov/materials/09VU/Lecture2.pdf>
11. Сивухин Д. В. Общий курс физики. М.: Физматлит, 2004. 656 с.
12. Среда программирования. <http://www.maksakov-sa.ru/ProgrProd/KompProgrProd/index.html>
13. Жук В.В., Натансон Г.И. Тригонометрические ряды Фурье и элементы теории аппроксимации. Л.: Изд-во Ленингр. ун-та, 1983. 188 с.
14. Состав ускорительного комплекса. <http://nucloweb.jinr.ru/nica/TDR/2015/TDR Volume1.pdf>
15. Structure of TDMS. <http://www.ni.com/white-paper/3727/en/>

16. TDMS file. <http://www.ni.com/white-paper/5696/en/>
17. OpenMP. <http://astronu.jinr.ru/wiki/index.php/OpenMP>
18. About Visual Studio. <https://www.visualstudio.com/>
19. National Instrument. <http://www.ni.com/>

Приложение

Источники

Рекомендуемые ресурсы:

1. <http://www.jinr.ru/>
2. <http://home.cern/>

Пример TDMS файла

The screenshot shows the TDMS File Viewer interface. The left pane displays a tree view of the file structure under 'uncomp.tdms', with 'unnamedTask<537>' selected. The main pane shows a table of values for this task, with columns for different slots (ai0, ai1, ai2, ai3) and rows for time points. The data is presented in scientific notation.

Time	unnamedTask<537> PXI1Slot3/ai0	unnamedTask<537> PXI1Slot3/ai1	unnamedTask<537> PXI1Slot3/ai2	unnamedTask<537> PXI1Slot3/ai3	unnamedTask<537> PXI1Slot4/ai0	unnamedTask<537> PXI1Slot4/ai1	unnamedTask<537> PXI1Slot4/ai2	unnamedTask<537> PXI1Slot4/ai3
0,011907	0,000517	0,001131	0,001190	0,001053	0,000235	2,980232E-5	4,768372E-6	
0,011954	0,000840	0,001415	0,001520	0,001432	0,000355	-2,145767E-5	-4,768372E-6	
0,011945	0,000660	0,001276	0,001336	0,001290	0,000353	9,536743E-6	-7,152557E-6	
0,012066	0,000863	0,001351	0,001448	0,001397	0,000426	-2,145767E-5	2,622604E-5	
0,012120	0,000801	0,001305	0,001408	0,001312	0,000690	1,192093E-5	1,072884E-5	
0,012223	0,000908	0,001442	0,001483	0,001469	0,000761	-7,152557E-6	-4,172325E-5	
0,012066	0,000767	0,001280	0,001363	0,001240	0,001038	-2,145767E-5	-1,907349E-5	
0,012070	0,000772	0,001527	0,001621	0,001420	0,000640	-3,695488E-5	5,125999E-5	
0,011895	-0,000325	0,000654	0,000676	0,000300	0,001086	9,059906E-5	6,437302E-5	
0,012078	0,002836	0,002580	0,002846	0,003262	0,003128	-0,000142	-7,033348E-5	
0,012020	-0,000236	0,001404	0,001311	0,000892	-0,000683	7,033348E-5	5,722046E-5	
0,012035	0,001589	0,002060	0,002238	0,002213	0,000241	-3,337860E-5	1,192093E-5	
0,012103	0,000933	0,001793	0,001888	0,001798	-0,000191	2,503395E-5	4,529953E-5	
0,011992	0,000910	0,001591	0,001689	0,001603	0,000480	-5,960464E-6	4,768372E-6	
0,011976	0,001360	0,001931	0,002021	0,002068	0,001146	-1,788139E-5	-4,529953E-5	
0,011806	0,000671	0,001514	0,001607	0,001380	0,001048	3,814697E-5	5,841255E-5	
0,011951	0,001440	0,002028	0,002133	0,002079	0,001476	-4,847581E-5	1,668930E-5	
0,011965	0,000831	0,001785	0,001833	0,001633	0,001119	5,960464E-6	5,125999E-5	
0,012046	0,001320	0,002099	0,002179	0,002106	0,001196	-2,384186E-5	1,192093E-5	
0,012099	0,000881	0,001882	0,001962	0,001849	0,000806	2,026558E-5	2,503395E-5	
0,011959	0,001352	0,002246	0,002316	0,002280	0,000910	-1,907349E-5	2,145767E-5	
0,011983	0,001075	0,002041	0,002093	0,002025	0,000789	-1,907349E-5	8,344650E-6	
0,011910	0,001392	0,002217	0,002365	0,002313	0,001023	-2,384186E-5	4,649162E-5	
0,011986	0,001209	0,002105	0,002201	0,002069	0,001091	-7,152557E-6	-1,192093E-5	
0,012145	0,001503	0,002317	0,002407	0,002378	0,001369	-7,152557E-6	8,344650E-6	
0,012081	0,001186	0,002106	0,002217	0,002061	0,001386	1,192093E-6	9,536743E-6	
0,012304	0,001388	0,002197	0,002283	0,002263	0,001357	-2,026558E-5	-2,384186E-5	
0,011953	0,000947	0,002036	0,002048	0,001881	0,001150	3,099442E-5	6,079674E-5	
0,012027	0,001506	0,002366	0,002469	0,002427	0,001135	-2,861023E-5	1,192093E-5	
0,011970	0,000749	0,001956	0,001994	0,001742	0,000867	3,695488E-5	3,695488E-5	
0,012184	0,002900	0,003133	0,003368	0,003649	0,001371	-0,000108	-4,410744E-5	
0,011994	2,384186E-6	0,001875	0,001720	0,001203	-0,001400	0,000104	5,245209E-5	

Рис. 7: Пример файла TDMS формата.