



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 44 за 2018 г.



ISSN 2071-2898 (Print)  
ISSN 2071-2901 (Online)

Галанин М. П., Лукин В.В.,  
Родин А.С., Сорокин Д.Л.

Применение программной  
платформы Теметос для  
разработки среды  
моделирования  
электромагнитного  
ускорителя

**Рекомендуемая форма библиографической ссылки:** Применение программной платформы Теметос для разработки среды моделирования электромагнитного ускорителя / М.П.Галанин [и др.] // Препринты ИПМ им. М.В.Келдыша. 2018. № 44. 32 с. doi:[10.20948/prepr-2018-44](https://doi.org/10.20948/prepr-2018-44)  
URL: <http://library.keldysh.ru/preprint.asp?id=2018-44>

**Ордена Ленина  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
им. М.В. Келдыша  
Российской академии наук**

**М.П. Галанин, В.В. Лукин, А.С. Родин, Д.Л. Сорокин**

**Применение программной платформы Теметос  
для разработки среды моделирования  
электромагнитного ускорителя**

Москва — 2018

**Галанин М.П., Лукин В.В., Родин А.С., Сорокин Д.Л.**

Применение программной платформы Теметос для разработки среды моделирования электромагнитного ускорителя

Представлено развитие архитектуры программной платформы сопровождения вычислительного эксперимента Теметос. Описан базовый модуль платформы и система предметных библиотек. Приведены основные принципы разработки платформы, обеспечивающие ее безболезненный рост и гибкость при адаптации к различным задачам математического моделирования. Описан опыт применения платформы к разработке программной среды моделирования электромагнитного ускорителя. Приведен пример расчета, подготовленного с использованием разработанной программной среды.

**Ключевые слова:** программная платформа Теметос, математическое моделирование, вычислительный эксперимент, электродинамический ускоритель.

**Michail Pavlovich Galanin, Vladimir Vladimirovich Lukin, Alexander Sergeevich Rodin, Dmitry Leonidovich Sorokin**

Application of the Temetos software platform to electromagnetic accelerator simulation environment development

The advance in the architecture of software platform Temetos for computing experiment supply is presented. The basic module of the platform and the system of subject libraries are described. The main principles of the platform development providing its painless growth and flexibility in adapting to various problems of mathematical modeling are presented. The experience of using the platform to develop a software environment for simulating an electromagnetic accelerator is described. An example of a calculation prepared using the developed software environment is given.

**Key words:** software platform Temetos, mathematical modeling, computational experiment, electrodynamic accelerator.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 18-01-00252.

## 1. Введение

Программная платформа Теметос [1, 2] предназначена для сопровождения вычислительного эксперимента на всех стадиях анализа и исследования математических моделей физических процессов и технических объектов. Основная цель платформы Теметос заключается в создании программного окружения, позволяющего с помощью ряда открытых интерфейсов передачи данных и управления расчетом оснастить авторский проблемно-ориентированный решатель инструментами подготовки, проведения и анализа результатов вычислительного эксперимента. Данная цель является значимой в ситуации, когда задача математического моделирования является сложной и требует разработки нестандартного вычислительного алгоритма, поскольку в этом случае ряд вспомогательных, но оттого не менее важных, задач оказывается не в фокусе внимания. Так, платформа и программные комплексы, разработанные на ее основе, предоставляют возможности для подготовки геометрической и физической моделей исследуемой конструкции или физического процесса к расчету, дают инструменты настройки внешних или встроенных проблемно-ориентированных модулей, позволяют осуществлять запуск, контроль прохождения и анализ результатов расчетов.

Платформа Теметос разрабатывается так, чтобы на ее основе могли быть созданы пакеты прикладных программ для математического моделирования в самых различных физических приложениях. Для этого в структуру и интерфейсы платформы должны быть заложены инструменты работы с самыми общими объектами и элементами математических моделей. К ним относятся геометрические модели, сетки, физические таблицы и зависимости для свойств сред и материалов, начальные и граничные условия, настройки численных методов и расчетных программ-решателей. В отличие от широко известных программных библиотек и фреймворков (OpenFOAM [3], code Aster [4], MARPLE3D [5]), при применении платформы предполагается высокая степень автономности решателя, в то время как на платформу возлагаются задачи подготовки входных данных, запуска расчета и визуализации полученных результатов. При этом архитектура платформы должна позволить работать с насколько возможно широким спектром форматов представления данных, а также расширять функциональность системы за счет подключения внешних вспомогательных модулей (например, построения сетки).

В данной работе представлены архитектурные решения, призванные сделать платформу Теметос гибкой для адаптации ко внешним решателям, и в то же время как можно более универсальной с точки зрения решаемых задач. Также представлен реализованный на основе платформы программный комплекс для численного моделирования разгона тел в ускорителе рельсотронного типа. На примере данного комплекса показана методика конфигурации базы знаний программной платформы и кратко описаны разработанные служебные модули комплекса.

## 2. Архитектура платформы Теметос

### 2.1. Основные принципы

В центре вычислительного эксперимента находится математическая модель исследуемого явления и соответствующий ей методический, алгоритмический и программный инструментарий [6, 7]. Математическая модель определяется набором используемых предположений и упрощений, имеющим результатом систему уравнений, описывающих явление. В этом отношении основное внимание при разработке программной реализации того или иного численного алгоритма всегда отводится непосредственно программе-решателю, в то время как вопросы построения сеток, визуализации результатов и ряд других оказываются на периферии и реализуются за счет сторонних, часто трудно приспособляемых программных решений.

В то же время ряд стадий вычислительного эксперимента, хоть и являются, условно говоря, „сервисными“ (внешними непосредственно для расчета), но могут и должны быть реализованы в виде, адаптированном к общей схеме моделирования в конкретной задаче. К таким обязательным действиям, конкретный набор и способ реализации которых определяется математической моделью, как правило, относятся следующие [2].

- 1) Геометрическое моделирование: формирование набора геометрических областей, описывающих (в принятой пространственной постановке) исследуемый объект, задание расчетной области.
- 2) Дискретизация геометрической модели: задание в расчетной области и на ее поверхности сеток.
- 3) Выбор физических моделей свойств рассматриваемых сред и материалов.
- 4) Задание начальных и граничных условий для рассматриваемой модели.
- 5) Настройка численных методов и расчетной программы, используемых для решения уравнений модели.
- 6) Анализ и визуализация результатов и работа с ними.

В зависимости от решаемой задачи набор программных модулей, обеспечивающих реализацию указанных стадий, может меняться. С другой стороны, каждый из таких этапов оперирует со своим типом элементов математической модели, которые являются взаимосвязанными и взаимозависимыми (как, например, сетка и расчетная область), но должны описываться каждый в своей логике и с учетом содержательной „морфологии“ элемента. Например, расчетная область описывается в терминах геометрических примитивов и операций над ними, а сетка в ней, учитывающая размеры отдельных примитивов в целях, например, сгущения, — в терминах узлов, ячеек и связей между

ними. Для построения целостной программной системы необходимо задать ряд стандартов представления элементов математической модели и обеспечить возможность использования этих элементов на разном уровне доступа в разных модулях системы: расчетная область строится в геометрическом редакторе, но используется (на достаточно глубоком уровне, превышающем потребности редактора) и при подготовке сетки.

При этом набор программных модулей, реализующих работу с выбранными стандартами представления данных, может (и должен) подбираться (или дорабатываться) под потребности конкретной математической модели. Для обеспечения безболезненного роста программной платформы в этом случае необходимо различать и по-разному описывать и реализовывать следующие понятия и элементы:

- 1) стандартизованный формат хранения элемента модели (например, расчетная область, сетка) в базе данных платформы — например, в виде XML-файла;
- 2) программное представление элемента модели в виде структуры данных и набора процедур для работы с этой структурой (система классов языка C++), обеспечивающих максимальную функциональность элемента как в ходе его редактирования, так и при использовании на других стадиях моделирования;
- 3) визуальный программный модуль редактирования элемента модели, реализованный в виде независимой программной библиотеки, подключаемой к базовому модулю платформы.

Таким образом, программная структура платформы Теметос состоит из базового модуля, осуществляющего хранение, упорядочивание, выборку и предоставление пользователю элементов математических моделей в системе предметных библиотек, набора управляющих структур (классов) для работы с такими элементами в соответствующих им форматах и служебных модулей-редакторов (надстроечных модулей), реализующих функции по разработке, расчету и анализу математической модели для конкретного процесса или конструкции.

## **2.2. Система предметных библиотек и базовая платформа**

В основу информационной системы платформы Теметос положена система предметных библиотек, описывающих различные, но взаимосвязанные, элементы математической модели [2]. Набор библиотек и связи между ними определяются решаемой задачей и выбранной математической моделью. Можно указать примерный набор, характерный для решения задач математической физики сеточными методами:

- 1) геометрическая библиотека элементарных областей (БЭО), содержащая векторное описание используемых при моделировании геометрических областей;
- 2) библиотека сеток (БС) в элементарных областях;
- 3) библиотека моделей физических свойств (БМФС) в виде таблиц, функций, программных библиотек;
- 4) библиотека расчетных областей (БРО) или конструкций, которые формируются из элементарных областей с учетом используемой сетки и свойств материала;
- 5) библиотека сценариев моделирования (БСМ), содержащая наборы входных данных, начальных и граничных условий;
- 6) библиотека конфигураций расчетов (БКР), содержащая наборы настроек модуля-решателя;
- 7) библиотека результатов расчетов (БРР), содержащая полный набор выходных файлов решателя при проведении расчета из БКР.

Элементы библиотек связаны друг с другом принципом подчинения. В набор возможностей базового модуля платформы Теметос добавлены функции, позволяющие на каждой стадии подготовки вычислительного эксперимента получать полную информацию обо всех составляющих элементах модели.

Содержимое библиотек структурируется с помощью реляционной базы данных формата SQL и хранится в файлах форматов, поддерживаемых платформой Теметос. Разделение структуры хранения данных и самих данных обеспечивает безразличность базового модуля системы к используемым форматам хранения данных. Поэтому набор форматов может быть легко расширен путем разработки классов, реализующих операции с данными в нужном формате, добавления соответствующих функций в служебные модули и строк в таблицы базы данных системы. Наполнение „форматного“ класса определяется как представлением соответствующего элемента модели в редактирующем его модуле, так и потребностями модулей, использующих данный элемент в качестве материала для последующих стадий моделирования.

Для непосредственного хранения данных, как правило, используется язык разметки XML как один из наиболее эффективных способов представления данных. Это делает структуру соответствующих библиотечных файлов прозрачной и позволяет осуществлять их чтение и редактирование и вне платформы Теметос. В то же время логическая разметка хранимых данных в этом случае может быть хорошо адаптирована к используемому формату.

Хранение данных осуществляется в виде структурированной файловой системы, метаданные и описатели которой хранятся в базе данных стандарта SQLite. База содержит следующие таблицы.

- 1) Таблица **formats**, формально описывающая форматы хранения данных различных библиотек платформы. Задача таблицы — декларировать форматы хранения и привязать их к предметным библиотекам. Столбцы таблицы следующие:
  - а) **id (integer)** — идентификатор формата, первичный ключ;
  - б) **name (text)** — имя формата;
  - в) **type (text)** — тип предметной библиотеки (например, **TGeometry** — БЭО, **TMesh** — БС и другие).
- 2) Таблица **formatfiletypes** включает в себя метаописание типов файлов, содержащих информацию, предусмотренную тем или иным форматом. Предполагается, что один формат может содержать более одного файла. Минимальное количество файлов — один, поскольку информация содержится не в самой базе данных, а в соответствующей файловой системе платформы Теметос. Столбцы таблицы следующие:
  - а) **id (integer)** — идентификатор типа файла, первичный ключ;
  - б) **fid (integer)** — идентификатор формата данных;
  - в) **name (text)** — имя типа файлов (например, **XML** или **NetgenGeo**);
- 3) Таблица **libitems** содержит метаданные элементов предметных библиотек без учета их привязки к форматам хранения. Такой подход позволяет хранить один и тот же элемент в нескольких форматах. Например, хранить элементарную геометрическую область во внутреннем XML-формате пакета и во внешнем CAD-формате, используемом для импорта или экспорта геометрических моделей. Столбцы таблицы следующие:
  - а) **id (integer)** — идентификатор элемента библиотеки, первичный ключ;
  - б) **name (text)** — имя элемента библиотеки;
  - в) **creation (date)** — дата и время создания элемента;
  - г) **modification (date)** — дата и время последней модификации элемента;
  - д) **libraryid (text)** — тип предметной библиотеки, к которой принадлежит элемент.
  - е) **description (text)** — текстовое описание элемента (пояснение, аннотация);
- 4) Таблица **libitemsfmtrealizations** содержит в себе данные о хранящихся в платформе реализациях того или иного элемента библиотеки в одном из форматов из таблицы **formats**. Столбцы таблицы следующие:
  - а) **id (integer)** — идентификатор реализации, первичный ключ;



- б) `lid (integer)` — идентификатор элемента библиотеки;
  - в) `fmtid(integer)` — идентификатор формата хранения данных.
- 5) Таблица `libfiles` сопоставляет элементам библиотеки имена хранящихся в файловой системе пакета файлов. Столбцы таблицы следующие:
- а) `id (integer)` — идентификатор файла, первичный ключ;
  - б) `lid (integer)` — идентификатор элемента библиотеки, которому принадлежит файл;
  - в) `fftid (integer)` — идентификатор формата данных, в котором хранится информация в файле;
  - г) `path (text)` — путь к файлу (имя файла в операционной системе);
  - д) `fftname (text)` — имя типа файла данного формата (важно в случае, когда один формат предусматривает хранение данных в нескольких файлах разных типов).
- 6) Таблица `libitemsrelation` хранит сведения об иерархическом подчинении элементов одних библиотек элементам других. Так, например, элементы библиотеки сеток однозначно связаны с теми элементами библиотеки расчетных областей, по которым они построены. Столбцы таблицы следующие:
- а) `id (integer)` — идентификатор связи, первичный ключ;
  - б) `childlid (integer)` — идентификатор дочернего элемента;
  - в) `parentlid (integer)` — идентификатор родительского элемента.

Работу с базой данных осуществляет только базовый модуль, служебные модули получают данные из библиотеки в виде ссылок на файлы выбранного формата.

### 2.3. Программная реализация платформы: кроссплатформенность и система модулей

На современном этапе развития вычислительных средств программная платформа математического моделирования должна быть приспособлена к работе в рамках операционных систем (ОС) различных архитектур (быть кроссплатформенной). Это связано, в частности, с требованием масштабируемости пакетов расчетных программ, когда задание расчетной области и условий моделирования, а также подготовительные расчеты на сравнительно небольшой сетке выполняются на рабочем месте прикладного специалиста, а „серьезные“ расчеты, использующие сетки в десятки и сотни миллионов ячеек, проводятся на многоузловых кластерных системах. Первые, как правило, работают под управлением ОС семейства Windows, а для вторых средой исполнения программ являются различные реализации ОС Linux. Кроме

того, для графического моделирования и подготовки сеток высокого разрешения (до миллиарда ячеек) зачастую используются персональные многоядерные высокопроизводительные системы, также, как правило, оснащенные ОС Linux. Подобная ситуация требует использования кроссплатформенных программных решений при разработке как графической оболочки моделирующего программного пакета, так и при создании чисто вычислительных модулей.

Дополнительные сложности создает необходимость чтения исходных данных, подготовленных в одной ОС, программой, работающей (в многопоточном режиме) на другой архитектуре. В этом случае также необходимо применять стандарты хранения данных, для оперирования с которыми созданы кроссплатформенные программные библиотеки. К таким стандартам относятся, например, открытый стандарт разметки данных XML, а также различные реализации систем управления реляционными базами данных стандарта SQL.

Другая серия требований к платформе Теметос возникает из концепции безболезненного роста программы [8], которая обеспечивается, с одной стороны, модульным подходом к разработке исходных кодов, а с другой — широким использованием механизма подключаемых библиотек при реализации различных функций и инструментов создаваемой среды моделирования. В первую очередь эти требования относятся к графическому инструментарию подготовки расчета.

Платформа Теметос обеспечивает безболезненное наращивание функций и возможностей среды моделирования за счет добавления новых модулей при фиксированных интерфейсах взаимодействия этих модулей с базовым модулем системы, осуществляющим хранение, обработку и представление для доступа данных из предметных библиотек. При этом весь функционал системы по взаимодействию с оператором возложен на служебные модули — от модулей геометрического моделирования и дискретизации модели до подсистемы запуска и отслеживания прохождения моделирующего расчета в рамках модуля-решателя.

Возможность безболезненного развития пакета достигается за счет использования механизмов объектно-ориентированного программирования, реализованных в языке C++, а также архитектуры пакета в виде базового модуля, берущего на себя задачу выборки и предоставления данных из базы знаний системы, и ряда модулей надстроек, подключаемых через интерфейсы виртуальных классов (рис. 1). Каждый из служебных модулей платформы Теметос в этом случае представляет собой реализацию виртуального класса модуля-надстройки и обеспечивает прием на входе и выдачу после обработки структурированных записей из базы знаний платформы. Основной функцией базового модуля является осуществление функций системы управления базой данных SQLite.

Служебные модули системы, имеющие графический интерфейс, разработаны с применением кроссплатформенного инструментария разработки ПО

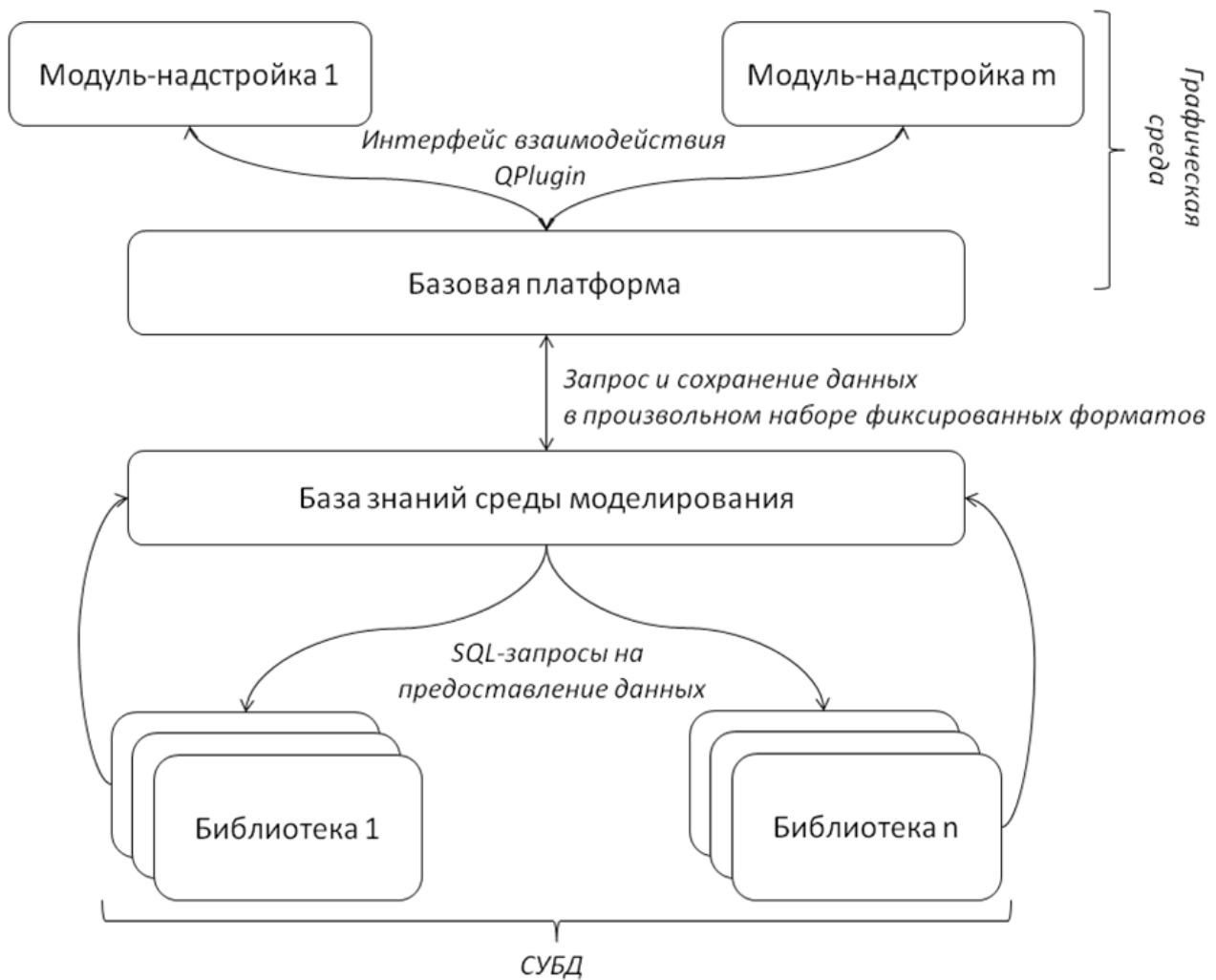


Рис. 1. Программная архитектура платформы Теметос

на языке программирования C++ Qt [9]. Фреймворк Qt позволяет запускать написанное с его помощью ПО в большинстве современных ОС путём простой перекомпиляции программы без изменения исходного кода. Фреймворк включает в себя ряд классов, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных (включая собственную реализацию реляционных баз данных SQLite) и файлами в формате XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Подключение служебных модулей к базовому осуществляется с использованием системы классов QPlugin, входящих во фреймворк Qt, что позволяет оформлять модули в виде динамических библиотек (DLL-библиотеки среды Windows и shared-библиотеки среды Linux) и осуществлять динамическое связывание запрашиваемых базовым пакетом подпрограмм с их реализацией в подключаемых модулях. Объявление виртуального класса, обеспечивающего подключение служебных модулей к базовому, приведено в листинге на рис. 2.

Платформа Теметос разрабатывается как база для создания программ-

```

class TPlugin
{
public:
    QString Name;
    QString Vendor;
    QString Version;
    QString Description;
    QString Path;
    QString libraryId;
    QStringList formats;
    QString parentLibId;
    QStringList parentFormats;

    QPluginLoader * Loader;

    QMap<QString, QWidget *> * tPluginControls;
    QWidget * tPluginPage;
    QToolBar * tPluginToolBar;
    QString * basePathPrefix;

    TLibrary * libReader;

    virtual ~TPlugin() {}
    virtual void getPluginInfo () = 0;
    virtual void initPlugin (QString pathPrefix, TLibrary * library) = 0;
    virtual void loadTLibItem(TLibraryItem * inputLI) = 0;
    virtual TLibraryItem * getCurrentTLibItem() = 0;
    virtual TLibraryItem * saveTLibItem() = 0;
    virtual QString getTLibItemId() = 0;
    virtual QString getTLibItemName() = 0;
    virtual void setTLibItemName(QString name) = 0;
    virtual QWidget * getPluginPage() = 0;
    virtual QToolBar * getPluginToolBar() = 0;
    virtual QMap<QString, QWidget *> * getPluginControls() = 0;
};

Q_DECLARE_INTERFACE(TPlugin, "Temetos.TPlugin")

```

Рис. 2. Листинг интерфейсного класса подключения служебных модулей

ных комплексов математического моделирования. Она включает в себя ряд программных средств, библиотек исходных кодов и интерфейсов, задача которых состоит в упрощении разработки визуальной среды математического моделирования в конкретной задаче. Для построения такой среды „под задачу“ необходимо

- 1) определить необходимый набор предметных библиотек, описывающих выбранную математическую модель;
- 2) разработать или выбрать из имеющихся в платформе стандарты представления данных предметных библиотек;
- 3) разработать или адаптировать имеющиеся служебные модули платформы, опираясь на возможности библиотеки Qt.

Подключение нового решателя всегда требует разработки модуля, осуществляющего его запуск. Остальные служебные модули могут быть унифицированы с другими средами, построенными на базе платформы. Отметим еще раз, что конкретный список и иерархия предметных библиотек и соответствующих служебных модулей являются с точки зрения базовой платформы произвольными.

### **3. Применение платформы Теметос для моделирования ускорителя рельсотронного типа**

Архитектура платформы Теметос позволяет использовать её для решения широкого круга задач. Это достигается прежде всего благодаря модульности платформы. Рассмотрим пример построенного на базе платформы программного пакета, разработанного для моделирования процесса ускорения макротел в канале электродинамического ускорителя рельсотронного типа.

#### **3.1. Математическая модель электромагнитного ускорителя**

Схема простейшего ускорителя типа рельсотрон представлена на рис. 3. К токоподводящим рельсам подключается источник питания. Между рельсами находится подвижный якорь, который замыкает цепь, вследствие чего по рельсам и якорю протекает электрический ток. Вокруг рельсов возникает магнитное поле. Взаимодействие этого поля и электрического тока в якоре приводит к возникновению силы Лоренца (Ампера), выталкивающей якорь из канала ускорителя [10, 11].

Конструкция ускорителя может быть усложнена введением рельсов подмагничивания, усложнением формы якоря, объединением нескольких простейших ускорителей в один многовитковый (рис. 4) и т. п.

Наиболее важные для исследования явления в электродинамических ускорителях происходят в окрестности якоря, причём длина якоря сопоставима

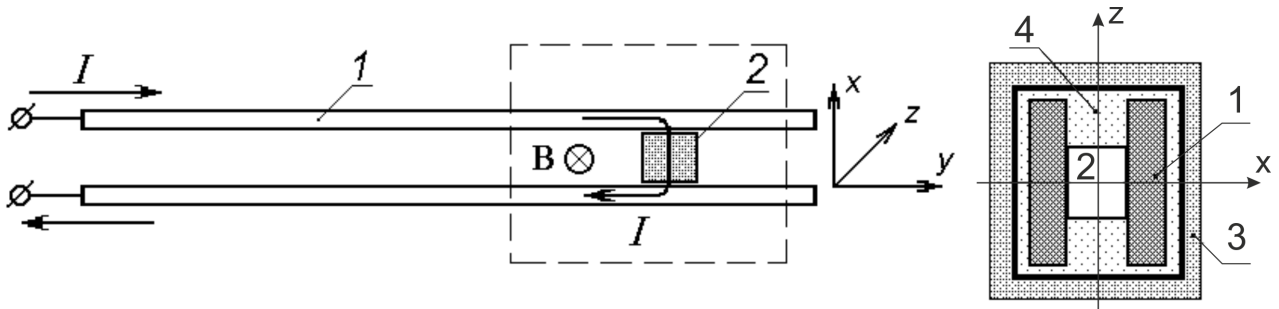


Рис. 3. Принципиальная схема рельсотрона: 1 — направляющий и токоподводящий рельс, 2 — ускоряемое тело (якорь или иная токовая арматура), 3 — силовой бандаж канала, 4 — изолятор

с поперечным размером канала ускорителя и много меньше его длины, поэтому моделирование целесообразно проводить в расчётной области, жёстко связанной с якорем и движущейся вместе с ним. Также заметим, что ускоритель имеет две плоскости симметрии, этот факт необходимо учесть при моделировании (для уменьшения объема вычислений).

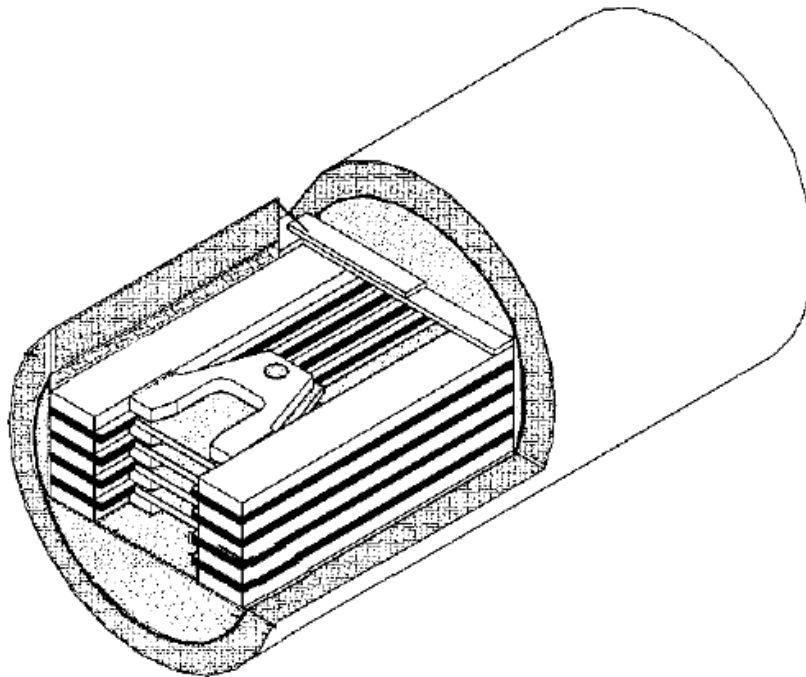


Рис. 4. Схема ускорителя RBM-5-2 [12]

С математической точки зрения процесс ускорения макротела в канале рельсотрона можно описать с использованием системы уравнений Максвелла в квазистационарном приближении, закона Ома, уравнения энергии с учетом фазовых переходов и уравнения движения якоря [13]:

$$\begin{aligned}
\operatorname{rot} \mathbf{H} &= 4\pi\sigma\mathbf{E}, \\
\operatorname{rot}(\mathbf{E} - [\mathbf{u} \times \mathbf{H}]) &= -\mu \frac{\partial \mathbf{H}}{\partial t}, \\
\operatorname{div} \mu \mathbf{H} &= 0, \quad \mathbf{j} = \sigma \mathbf{E}, \\
\rho \frac{D\varepsilon}{Dt} + \rho(\mathbf{w}, \nabla)\varepsilon &= (\mathbf{j}, \mathbf{E}) + \operatorname{div}(\kappa \operatorname{grad} T), \\
m \frac{d\mathbf{v}}{dt} &= \int_{arm} [\mathbf{j} \times \mathbf{H}]_y dV.
\end{aligned}$$

Здесь и далее  $\mathbf{E}$  — вектор напряженности электрического поля в системе координат, в которой вещество покоится, причём  $\mathbf{E} = \mathbf{E}^* + [\mathbf{u} \times \mathbf{H}]$ ,  $\mathbf{E}^*$  — вектор напряженности электрического поля в неподвижной (лабораторной) системе координат,  $\mathbf{H}$  — вектор напряженности магнитного поля,  $\mathbf{j}$  — вектор плотности тока,  $\mathbf{u}$  — вектор скорости движения вещества,  $\mathbf{w} = \mathbf{u} - \mathbf{v}$  — вектор относительной скорости вещества,  $\mathbf{v}$  — скорость движения точек пространственной области (в нашем случае  $\mathbf{v}$  — скорость движения якоря как целого, не зависящая от координат пространственной точки),  $\sigma$  — удельная проводимость,  $\mu$  — магнитная проницаемость,  $\mathbf{r} = (x, y, z)$  — радиус-вектор,  $t$  — время,  $\rho$  — плотность вещества,  $\varepsilon = \int_0^T c_v dT$  — удельная внутренняя энергия,  $c_v$  — удельная теплоёмкость,  $\kappa$  — коэффициент теплопроводности,  $T$  — температура,  $m$  — масса якоря.

Для решения системы уравнений Максвелла введем векторный потенциал  $\mathbf{A}$ :

$$\begin{aligned}
\mathbf{H} &= \frac{1}{\mu} \operatorname{rot} \mathbf{A}, \\
\mathbf{E} &= [\mathbf{u} \times \operatorname{rot} \mathbf{A}] - \frac{D\mathbf{A}}{Dt} + (\mathbf{v} \cdot \nabla) \mathbf{A}.
\end{aligned}$$

С учётом граничных и начальных условий получим систему [14, 16]:

$$4\pi\sigma \left\{ [\mathbf{u} \times \operatorname{rot} \mathbf{A}] - \frac{D\mathbf{A}}{Dt} + (\mathbf{v} \cdot \nabla) \mathbf{A} \right\} = \operatorname{rot} \frac{1}{\mu} \operatorname{rot} \mathbf{A} - \theta(\sigma) \operatorname{grad} \frac{1}{\mu} \operatorname{div} \mathbf{A}, \quad (1)$$

$$\mathbf{A}|_{t=0, \mathbf{r} \in G_1} = 0, \quad (2)$$

$$(\operatorname{rot} \mathbf{A})_\tau|_{\mathbf{r} \in \Gamma_2} = \Psi_\tau(\mathbf{r}, t), \quad \mathbf{A}_\tau|_{\mathbf{r} \in \Gamma_1} = 0, \quad (3)$$

$$\operatorname{div} \mathbf{A}|_{\mathbf{r} \in \gamma_{12}} = 0, \quad \mathbf{A}_n|_{\mathbf{r} \in \gamma_{22}} = 0. \quad (4)$$

Здесь  $G$  — рассматриваемая пространственная область ( $G = G_1 \cup G_2$ ),  $G_1 = \{\mathbf{r} \in G : \sigma > 0\}$ ,  $G_2 = \{\mathbf{r} \in G : \sigma = 0\}$ ,  $\partial G_1$  и  $\partial G_2$  — границы  $G_1$  и  $G_2$  соответственно,  $\partial G_{12} = \partial G_1 \cap \partial G_2$ ,  $\Gamma_1$  — часть общей границы  $\partial G$ , на которой задано условие для  $\mathbf{E}_\tau^*$ , что эквивалентно условию для  $\mathbf{A}_\tau$ ,  $\Gamma_2$  — часть

$\partial G$ , на которой задано условие для  $\mathbf{H}_\tau$  ( $\Psi_\tau$  — известная вектор-функция),  $\partial G = \Gamma_1 \cup \Gamma_2$ ,  $\Gamma_{12} = \Gamma_1 \cap \partial G_2$ ,  $\Gamma_{22} = \Gamma_2 \cap \partial G_2$ ,  $\gamma_{12} = \partial G_{12} \cup \Gamma_{12}$ . В записи (1)–(4) использованы смешанные эйлерово-лагранжевые (СЭЛ) переменные:  $D/Dt = \partial/\partial t + (\mathbf{v}, \nabla)$ , где  $\partial/\partial t$  — производная при фиксированных эйлеровых переменных,  $D/Dt$  — при фиксированных СЭЛ-переменных. Индекс  $n$  указывает на нормальную по отношению к границе составляющую вектора,  $\tau$  — тангенциальную. В рассматриваемых задачах в декартовой системе координат движение якоря происходит в положительном направлении оси  $Oy$ . Кроме того, в (1)–(4) учтена неоднородность задачи по пространству:  $\theta(\sigma) = 0$  в  $G_1$  и  $\theta(\sigma) = 1$  в  $G_2$ .

Для получения решения в трёхмерной расчётной области строим регулярную прямоугольную сетку со сгущениями. Для решения системы уравнений (1)–(4) используем метод конечных разностей, применяя метод опорных операторов [13, 14]. При решении задачи о фазовом переходе используем однородный метод сквозного счета [13–15]. Более подробно численный метод решения задачи и его программная реализация описаны в [16].

Энергопитание ускорителя осуществляется от внешнего электрического источника. В качестве источника тока, например, могут использоваться индуктивный накопитель или объединение нескольких RLC-цепочек.

### 3.2. Набор предметных библиотек

Многообразие конструкций моделируемых ускорителей, используемых сеток с большим числом ячеек, применяемых моделей материалов и большой объем получаемых в результате моделирования массивов данных требует создания и использования иерархически и реляционно упорядоченной системы хранения получаемых на каждом этапе моделирования объектов. Для этого использована система предметных библиотек платформы Теметос.

Моделирование процесса разгона тела в канале электромагнитного ускорителя состоит из следующих стадий.

- 1) Геометрическое моделирование: задание пространственной трехмерной модели исследуемого участка ускорителя, включая основные геометрические параметры направляющих рельсов, ускоряемого тела, рельсов подмагничивания, внешнего контура (кожуха) расчетной области, заполненной диэлектриком.
- 2) Дискретизация геометрической модели: задание прямоугольных структурированных неравномерных сеток.
- 3) Задание физических параметров используемых материалов (электрофизических и термофизических параметров, параметров фазовых переходов) с их привязкой к конкретным элементам конструкции ускорителя.
- 4) Задание начальных и граничных условий для рассматриваемой модели, а также определение настроек применяемых методов.



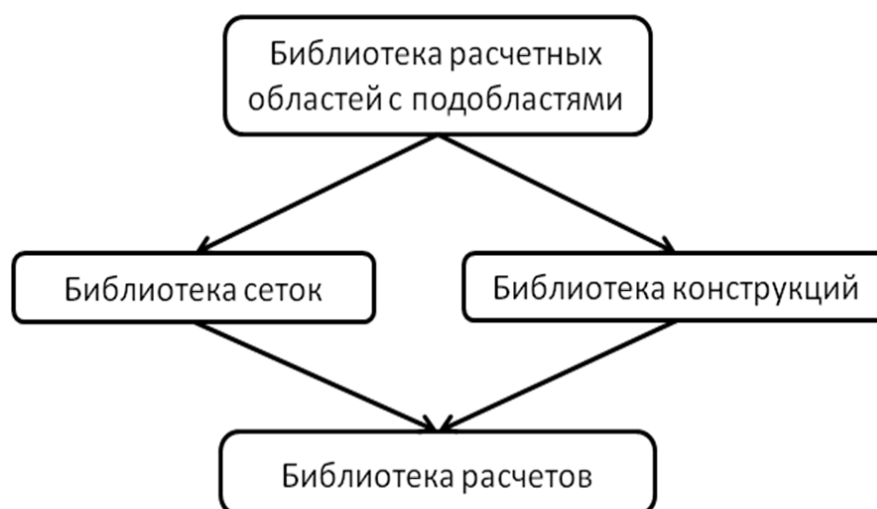


Рис. 5. Иерархия предметных библиотек в задаче о рельсотроне

- 5) Собственно численное моделирование — расчет представляющих интерес полей внутри расчетной области в рамках вычислительного модуля системы.
- 6) Визуализация результатов проведенных вычислений, позволяющая проводить анализ характеристик процесса разгона тела в ускорителе.

В соответствии с выделенными этапами реализованы следующие предметные библиотеки.

- 1) Геометрическая библиотека расчетных областей с подобластями (БРОП).
- 2) Библиотека сеток (БС).
- 3) Библиотека конструкций (БК), сопоставляющая каждой подобласти элемента БРОП набор физических параметров материала или среды, заполняющих данную подобласть.
- 4) Библиотека расчетов (БР), содержащая наборы настроек вычислительного модуля-решателя, включая исходные и начальные условия для моделирования конструкции (масса метаемого тела, его начальная скорость и пр.) и настройки численных методов и алгоритмов (в том числе параллельных), применяемых к расчету объекта из БК, а также результаты расчетов.

Иерархические связи предметных библиотек показаны на рис. 5.

### 3.3. Служебные модули программной среды

Опишем кратко разработанные модули-редакторы предметных библиотек. Инструментарий трехмерного геометрического задания расчетных областей, а также модули построения и визуализации сетки и графической обработки

расчетных данных построены с использованием открытой кроссплатформенной библиотеки для трёхмерного моделирования, обработки изображений и прикладной визуализации VTK [17]. Библиотека написана на языке C++ и представляет собой набор классов для обработки данных, представления их в виде трехмерных графических объектов и рендеринга сцен со множеством подобных объектов. Библиотека VTK вместе со своими расширениями содержит набор графических примитивов и булевых операций над ними.

### 3.3.1. Модуль задания геометрии электромагнитного ускорителя Geometrix

Задачей модуля Geometrix является трехмерное визуальное построение расчетных областей, моделирующих рассматриваемый участок ствола электромагнитного ускорителя (рис. 6). Расчетная область может содержать подобласти, каждая из которых отвечает тем или иным конструктивным элементам ствола ускорителя, таким как направляющие рельсы, метаемое тело, рельсы подмагничивания. Внешние границы расчетной области предполагаются моделирующими сверхпроводящий защитный кожух и геометрически представляют собой грани параллелепипеда с ребрами, параллельными осям координат. Подобласти могут быть заданы существенно более сложными, соответствующими геометрии моделируемой конструкции.

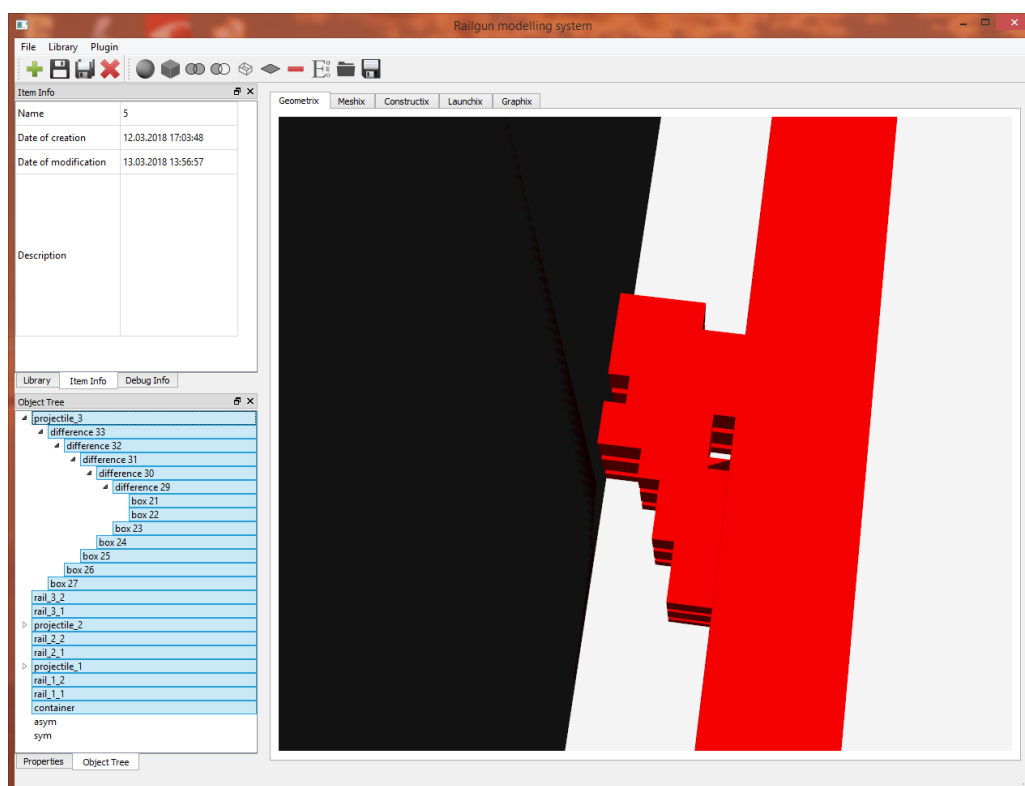


Рис. 6. Редактор геометрии Geometrix

Модуль Geometrix представляет собой векторный редактор, позволяющий задавать расчетную область и ее подобласти с помощью набора графических примитивов (сфера, параллелепипед и др.) и поддерживаемых буле-

вых операций (объединение, вычитание и др.) над ними. Выбранный подход позволяет добавлять примитивы, специфичные для рассматриваемых конструкций и принятой математической модели. В частности, к таким примитивам относятся „контейнер“, задающий внешнюю границу расчетной области — сверхпроводящий кожух, и „плоскости симметрии“, задание которых позволяет учитывать условия симметрии в модели.

```
class Primitive
{
public:
    QString Name;
    QPluginLoader * Loader;
    QString * basePathPrefix;

    virtual ~Primitive() {}
    virtual bool isOperation() = 0;
    virtual bool isContainer() = 0;
    virtual bool usedInMeshing() = 0;
    virtual vtkSmartPointer<vtkPolyData> getPrimitivePolyData(
        QMap<QString,QVariant> properties = QMap<QString,QVariant> (),
        QVector< vtkSmartPointer<vtkPolyData> > childsPolyDataList =
            QVector< vtkSmartPointer<vtkPolyData> >()
    ) = 0;
    virtual QMap<QString,QVariant> getDefaultProperties() = 0;
    virtual QStringList getPrimitiveNetgenGeo(
        QMap<QString,QVariant> properties,
        QString selfId,
        QStringList childsIds
    ) = 0;
    virtual bool isPointInside(
        QMap<QString,QVariant> properties,
        QVector<bool> childsPointInsideInfo,
        double * point
    ) = 0;
};
```

Рис. 7. Листинг виртуального класса Primitive

При разработке геометрической подсистемы пакета моделирования ускорителя применен тот же подход, что и при создании платформы Теметос в целом. Примитивы и операции над ними программно оформляются как подключаемые по интерфейсу QPlugin внешние динамические библиотеки. Унифицированный программный интерфейс позволяет во внутреннем представлении геометрической модели в модуле Geometrix не делать различия между примитивами и операциями над ними. И те, и другие являются узлами дерева, с той только разницей, что примитивы не имеют дочерних вершин, а операции обязаны их иметь. При этом в дочерних вершинах могут фи-

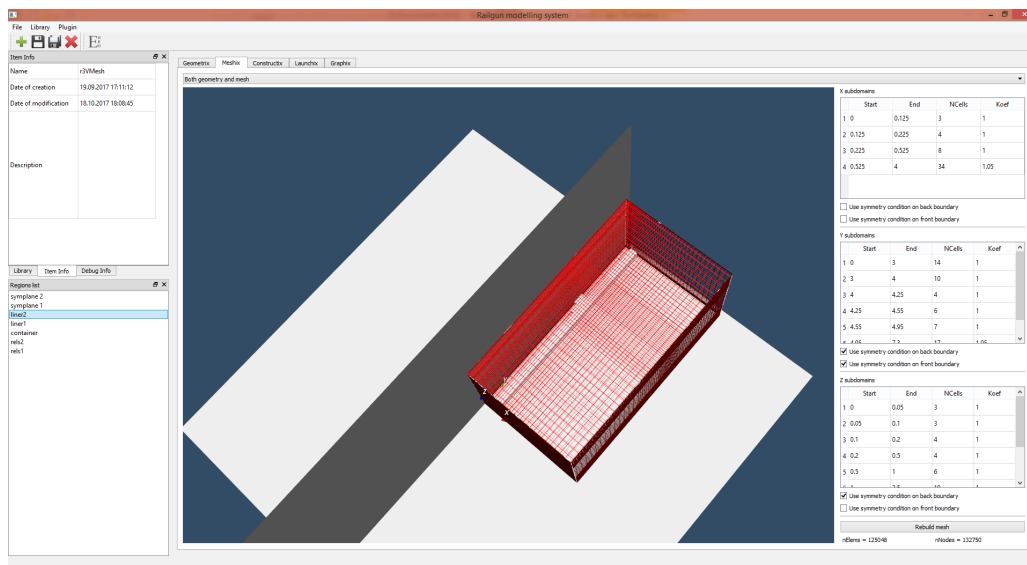


Рис. 8. Рабочее пространство модуля Meshix, осуществляющего построение сеток в расчетной области

гурировать примитивы или другие операции. Специальные примитивы „контейнер“ и „плоскость симметрии“ не могут быть дочерними для операций.

Интерфейс виртуального класса **Primitive**, обеспечивающего подключение примитивов и операций к модулю Geometrix, приведен в листинге на рис. 7. Класс включает в себя функции, возвращающие геометрические объекты библиотеки VTK, которая используется для отображения расчетной области на экране. Сигнатура функции `getPrimitivePolyData` позволяет единообразно описывать примитивы и операции над ними.

Описанный гибкий подход позволяет, например, создавать примитивы, существенно более сложные, чем сфера, куб или цилиндр. Таким „примитивом“, например, может быть полученное во внешней CAD системе описание лайнера или рельса, экспортированное в формат дискретного описания геометрических объектов STL.

### 3.3.2. Модуль задания сетки Meshix

Редактор сеток позволяет строить прямоугольные сетки в трехмерной области с подобластями (рис. 8). Элементы библиотеки TMesh, редактирование которой обеспечивает данный модуль, являются дочерними по отношению к элементам библиотеки расчетных областей TGeometry.

За счет векторного задания подобластей возникает возможность автоматического поиска референсных точек подобластей, к которым относятся границы примитивов, угловые точки и линии и др. Такой поиск осуществляется независимо по каждой из координат в процессе загрузки геометрической модели области, в результате чего при начале построения сетки пользователь получает набор крупных отрезков разбиения, для каждого из которых могут быть заданы свои шаги сетки, а также коэффициент уменьшения шага.

Описанный подход позволяет сгущать сетку в окрестностях граничных точек подобластей и в местах расчетной области, представляющих наибольший интерес, что позволяет создавать пространственную сетку, адаптивную к геометрии устройства. Также интерфейс модуля Geometrix позволяет сохранять сетку в формате входных файлов открытого свободнораспространяемого сеткопостроителя Netgen, производящего построение пространственной неструктурированной сетки из тетраэдров. Такие сетки позволяют еще более точно и подробно учесть геометрию устройства, особенно в окрестностях острых кромок и угловых точек.

### 3.3.3. Модуль задания физических моделей материалов Constructix

Предметная библиотека конструкций сопоставляет каждой подобласти набор физических параметров материалов или сред, заполняющих данную подобласть, включая электрофизические и теплофизические характеристики. При работе с программой пользователь может либо загрузить уже существующий вариант конструкции с заданными параметрами материалов, либо заново задать эти параметры, заполняя соответствующие поля на вкладке Constructix (рис. 9).

В программном комплексе предусмотрено описание как однофазных материалов (не имеющих фазовых переходов), так и многофазных (фазовые переходы для которых включены в математическую модель). Каждый из материалов может быть проводником или диэлектриком, что влияет на его описание в ходе расчета.

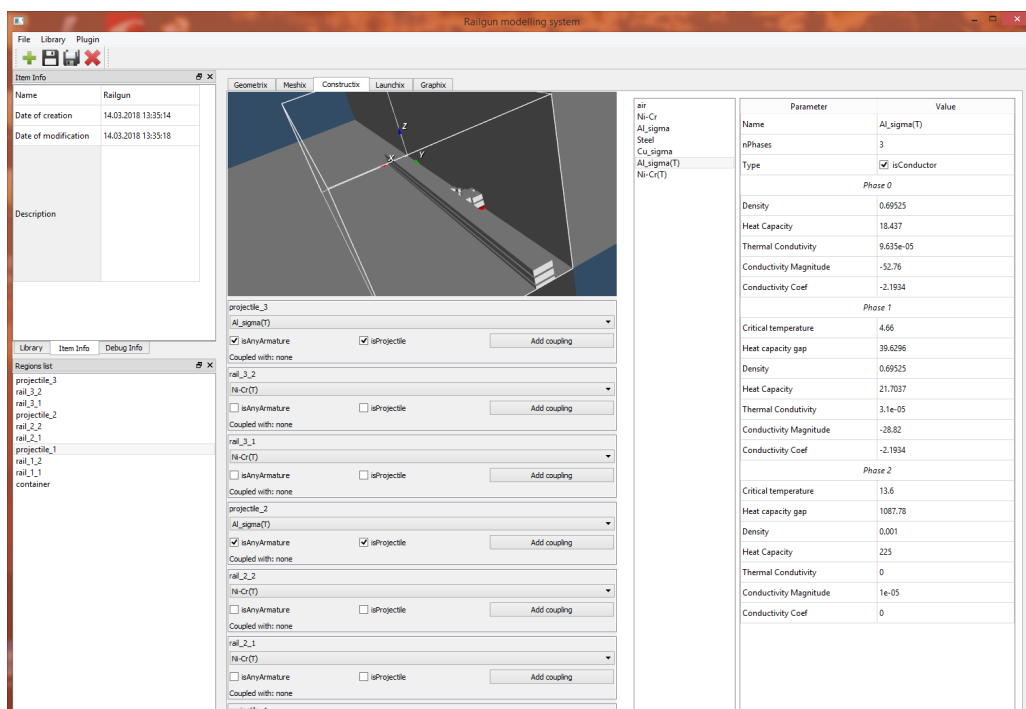


Рис. 9. Интерфейс модуля Constructix

Однофазный материал характеризуется следующими физическими константами:  $\rho$  — плотность материала;  $c_v$  — теплоёмкость материала; коэффициенты  $\sigma_{\max}$  и  $\alpha$  в модели электропроводности материала

$$\sigma(T) = \sigma_{\max}/(1 + \alpha T);$$

$\kappa$  — теплопроводность материала.

В случае, если материал „многофазный“, он имеет дополнительные характеристики:  $N$  — количество фаз;  $T_{\text{крит}}$  — температура фазового перехода;  $a_{\text{крит}}$  — скачок теплоёмкости при фазовом переходе. Количество задаваемых температур и скачков равно количеству фазовых переходов  $N - 1$ . В стандартном случае необходимо задать четыре величины: температуру плавления, скачок теплоёмкости при плавлении, температуру кипения и скачок теплоёмкости при кипении. Все эти данные сохраняются в отдельной базе данных материалов. Модуль Constructix поддерживает задание конструктивных элементов, состоящих из нескольких материалов. Для этого введена система объединения (coupling) подобластей исходной расчетной области.

### 3.3.4. Модули запуска расчета и графической обработки результатов Launchix и Graphix

Для подготовки запуска вычислительного модуля программного пакета реализованы модуль Launchix (рис. 10) и предметная библиотека (библиотека расчетов), содержащая наборы настроек вычислительного модуля-решателя, включая исходные данные и начальные условия для моделирования (масса метаемого тела, его начальная скорость, начальные поля температур и пр.) и начальные условия для токов и напряжений во внешних электрических цепях.

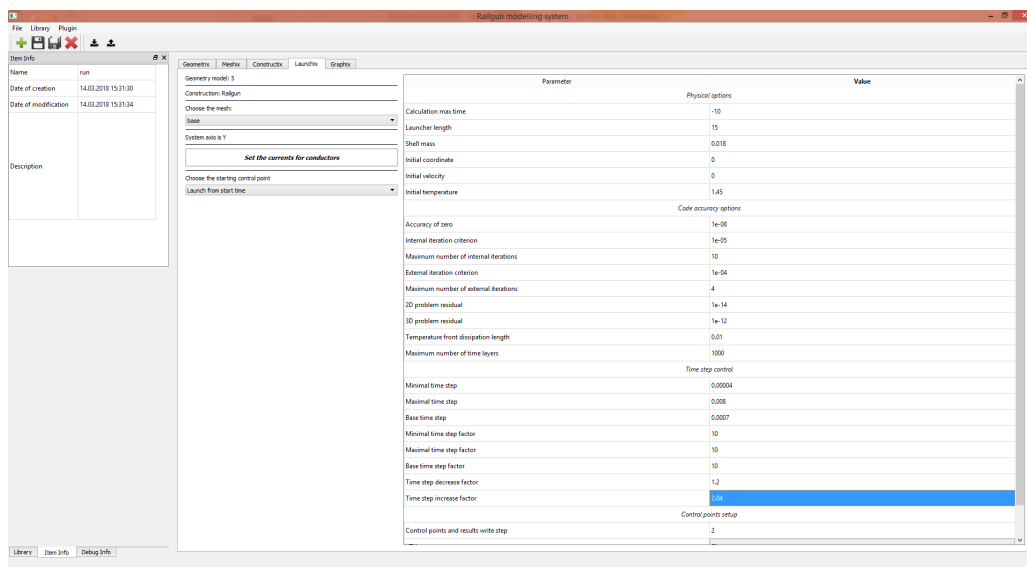


Рис. 10. Интерфейс модуля Launchix

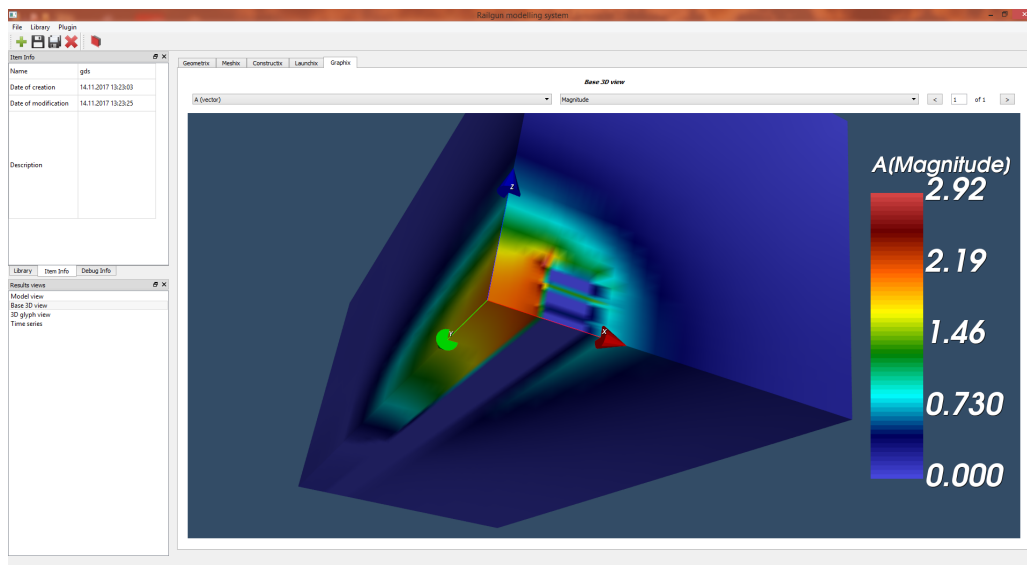


Рис. 11. Пример отображения результата в графическом модуле Graphix

Задачей модуля Launchix является подготовка полного набора входных файлов для решателя, который далее может быть запущен как на локальной рабочей станции, так и на удаленной суперЭВМ. Поскольку расчет на достаточно больших сетках является вычислительно сложным, именно использование кластерных вычислительных систем является приоритетным, и модуль позволяет как формировать входные и настроечные файлы решателя, так и загружать в систему результаты проведенных расчетов, однозначно ассоциируемые с конкретной конфигурацией запуска. Иными словами, результаты расчетов являются в данной конфигурации платформы Теметос элементом библиотеки расчетов.

Использование возможностей фреймворка Qt и библиотеки VTK позволили построить служебный модуль, осуществляющий визуализацию рассчитанных массивов данных — модуль Graphix (рис. 11). Модуль принимает на вход результаты расчетов в виде файлов стандарта VTK и позволяет проводить отображение пространственных распределений расчетных величин задачи: температуры, векторного потенциала, вектора напряженности магнитного поля и т.д. Векторные величины отображаются как в виде распределения модуля вектора, так и покомпонентно. Доступно как объемное отображение результатов, так и построение распределений параметров задачи в двумерных сечениях расчетной области, а также отображение зависимости сосредоточенных параметров системы от времени (текущий шаг по времени, величина входящего и выходящего токов, скорость якоря, сила Лоренца и др.).

## 4. Пример проведения полного расчёта

### 4.1. Постановка задачи

Продemonстрируем работу программного комплекса на примере расчёта пятивиткового ускорителя с С-образным якорем, представленного на рис. 12.

Описание задания геометрии в редакторе Geometrix (рис. 6), построения сетки с помощью модуля Meshix (рис. 8), а также задания свойств каждой подобласти расчётной области в модуле Constructix (рис. 9) приведено ранее.

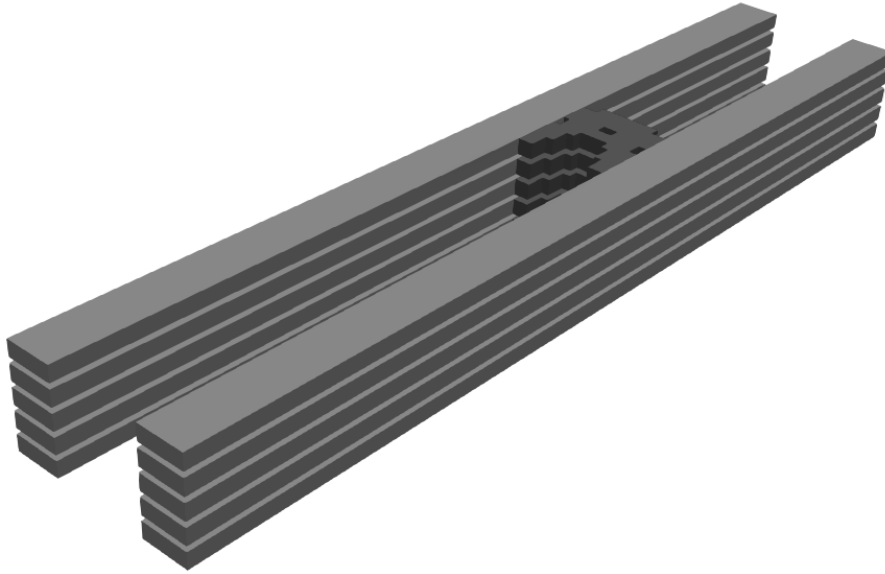


Рис. 12. Модель пятивиткового ускорителя

Габаритные размеры одного якоря: длина (по  $y$ ) — 0.84, ширина (по  $x$ ) — 0.72, высота (по  $z$ ) — 0.24. Все якоря скреплены между собой и движутся с единой скоростью.

Для моделирования процесса ускорения выбрана расчётная область, представленная на рис. 6. Её размеры: длина (по  $y$ ) — 7.5, ширина (по  $x$ ) — 3.0, высота (по  $z$ ) — 2.88.

В расчётной области с использованием модуля Meshix введена регулярная прямоугольная сетка, состоящая из 34440 точек и 31320 ячеек.

В модуле Launchix заданы длина ускорителя  $l = 15$ , масса якоря  $m = 0.018$ , начальная скорость  $v_0 = 0$ , начальная температура  $T_0 = 1.45$ . Отметим, что все величины приведены в безразмерном виде.

Будем считать, что рельсы соединены последовательно, т.е. полный ток во всех рельсах одинаковый. Зададим ток в цепи в соответствии графиком, изображенным на рис. 13. Проведём полный расчёт процесса разгона макротела.



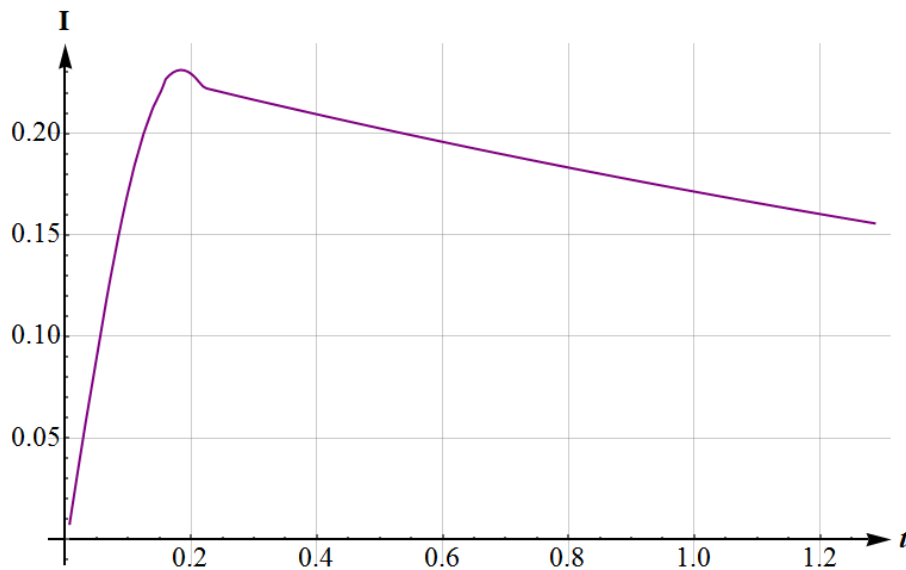


Рис. 13. Зависимость силы тока в цепи от времени

## 4.2. Результаты расчёта

Результатом работы вычислительного модуля программного комплекса являются распределения характеристик электромагнитного и теплового полей в расчётной области в различные моменты времени, а также графики зависимостей сосредоточенных характеристик ускорения (таких как координата и скорость якоря, сила, действующая на якорь, погонная индуктивность, максимальная температура в области) от времени (рис. 14 — 16).

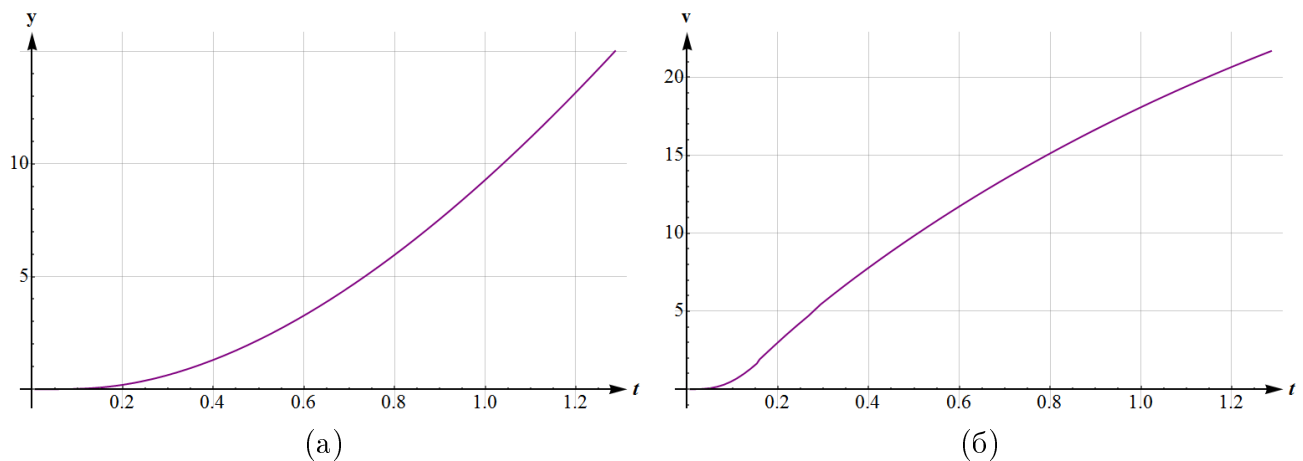


Рис. 14. Зависимости координаты (а) и скорости (б) якоря от времени при расчёте ускорителя, представленного на рис. 12

Согласно графикам 14 можно сделать вывод, что скорость вылета якоря из канала ускорителя равна 21,7.

Одной из основных характеристик рельсотрона, зависящей от его геомет-

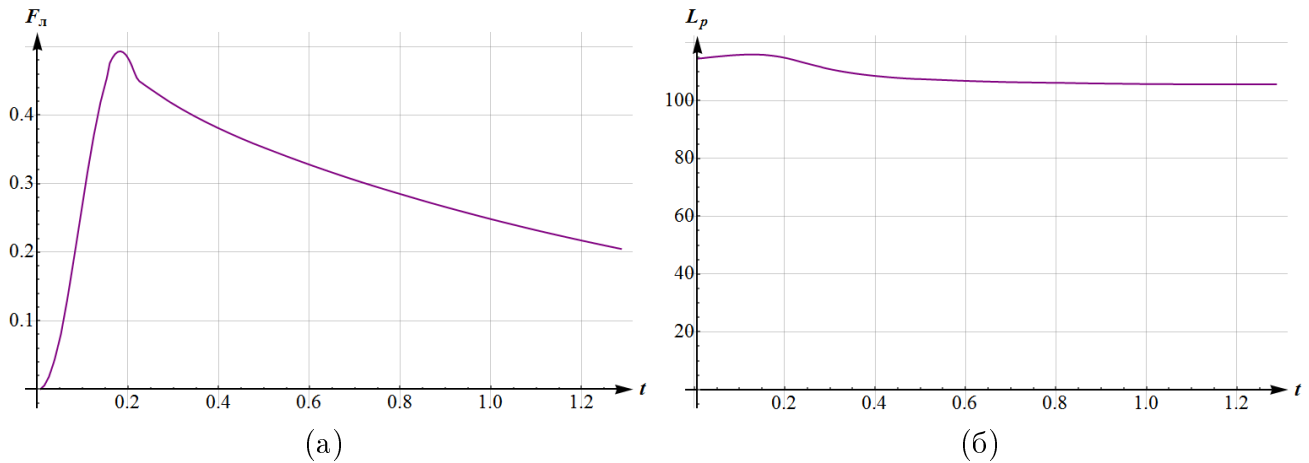


Рис. 15. Зависимости силы Лоренца (а) и погонной индуктивности (б) от времени при расчёте ускорителя, представленного на рис. 12

рии, является погонная индуктивность [18], определяемая соотношением

$$F_L = \frac{L_p I^2}{2}.$$

В процессе ускорения величина погонной индуктивности ускорителя может незначительно меняться в зависимости от скорости изменения тока. В силу того, что в процессе расчёта большую часть времени сила тока в цепи равномерно убывает, то и погонная индуктивность с высокой точностью постоянна.

По графику зависимости максимальной температуры в области от времени (рис. 16) можно определить моменты начала плавления (0.43) и кипения якоря (0.84). Фиксировать эти значения необходимо, т.к. плавление и кипение якоря приводят к ухудшению контакта между рельсом и якорем. Также эти процессы приводят к разрушению якоря, поэтому важно отслеживать не только моменты начала этих процессов, но и их динамику. На рис. 16 приведена зависимость количества расплавленного материала якоря от времени.

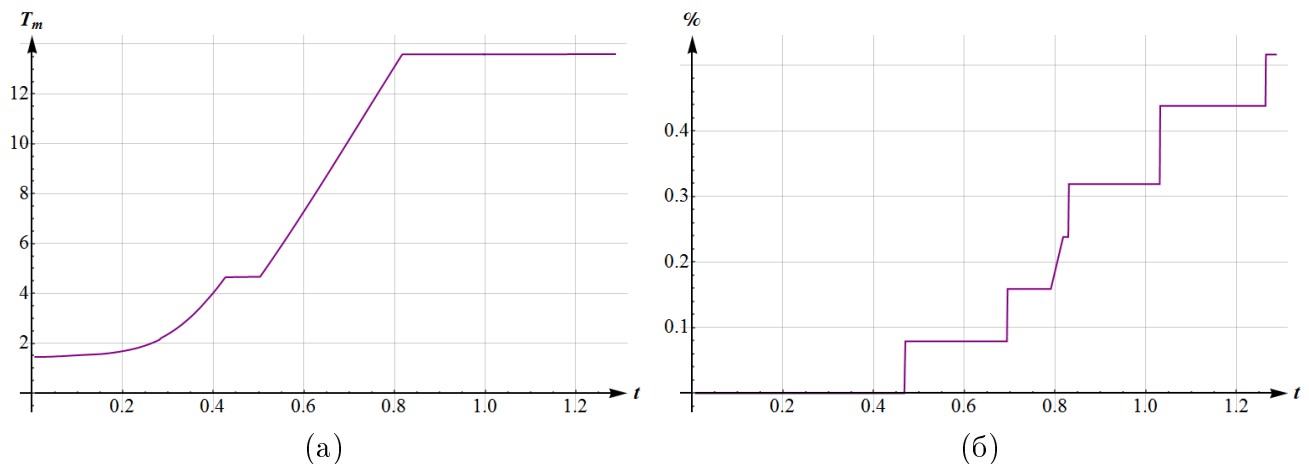


Рис. 16. Зависимости максимальной температуры в области (а) и количества расплавленного материала якоря (б) от времени при расчёте ускорителя, представленного на рис. 12

В процессе ускорения важно, чтобы электрический ток от рельса к якору протекал распределённо по всей поверхности между рельсами и якорями. В этом случае негативное воздействие джоулева нагрева на якорь минимально. Приведём графики распределения плотности тока в якоре в различные моменты времени (рис. 17).

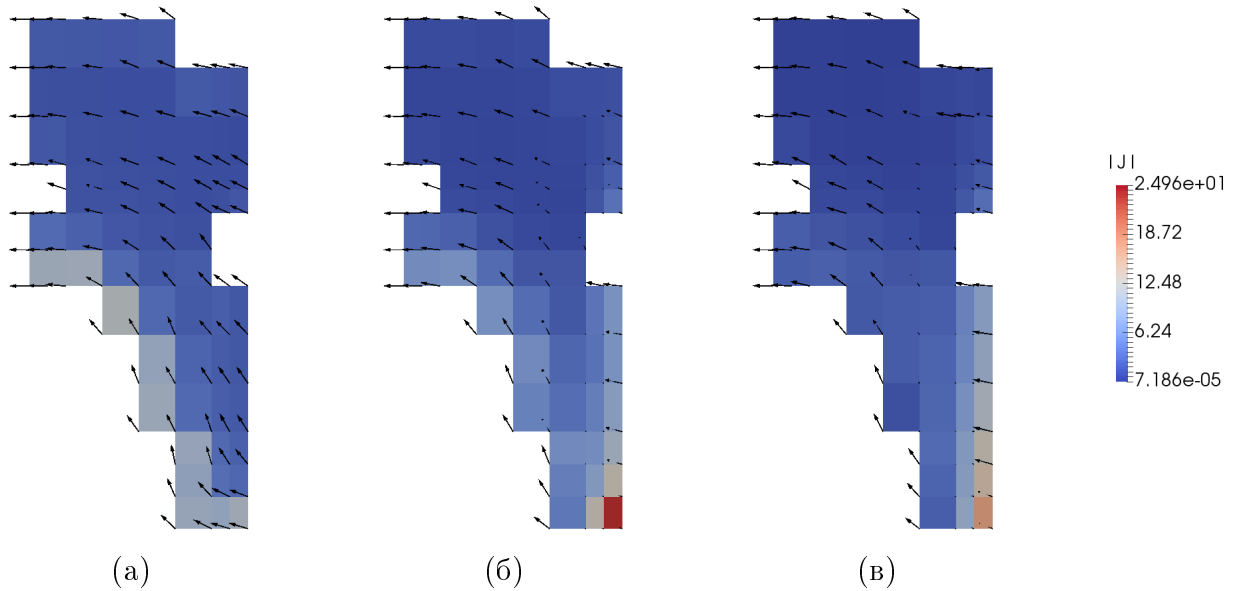


Рис. 17. Распределение плотности тока в якоре (в сечении  $z = 0.2$ ) в моменты времени (а) 0.12 (б) 0.39, (в) 1.29. Цветом показана величина плотности тока, стрелками — направление тока

С целью ускорения расчёта и обеспечения условия устойчивости разностной схемы при условии изменяющейся скорости якоря реализован алгоритм автоматического выбора шага по времени. На рис. 18 видно, что в начале расчёта имеется возможность увеличить шаг. Затем наступает момент, когда для выполнения критерия сходимости требуется совершить большее, чем допустимое настройками программы, число итераций, происходит уменьшение шага. В момент времени 0.23 скорость движения якоря становится настолько большой, что для выполнения критерия устойчивости разностной схемы требуется уменьшать шаг по времени (и он начинает уменьшаться обратно пропорционально росту скорости якоря).

Для проверки корректности работы программы реализованы модули внутреннего тестирования. При переходе с одного временного слоя на другой вычисляются значения полного протекающего тока в выбранных сечениях и норма дивергенции векторного потенциала в диэлектрике.

По значениям тока в разных сечениях можно судить, насколько численная модель соответствует математической, а по значениям дивергенции векторного потенциала в диэлектрике — убедиться, что отсутствуют паразитные источники тока. В ходе проведения расчёта величина дивергенции векторного потенциала не превышала значения  $10^{-11}$ , а полный ток в цепи сохранялся с точностью 0.02%, что свидетельствует о корректности расчёта.

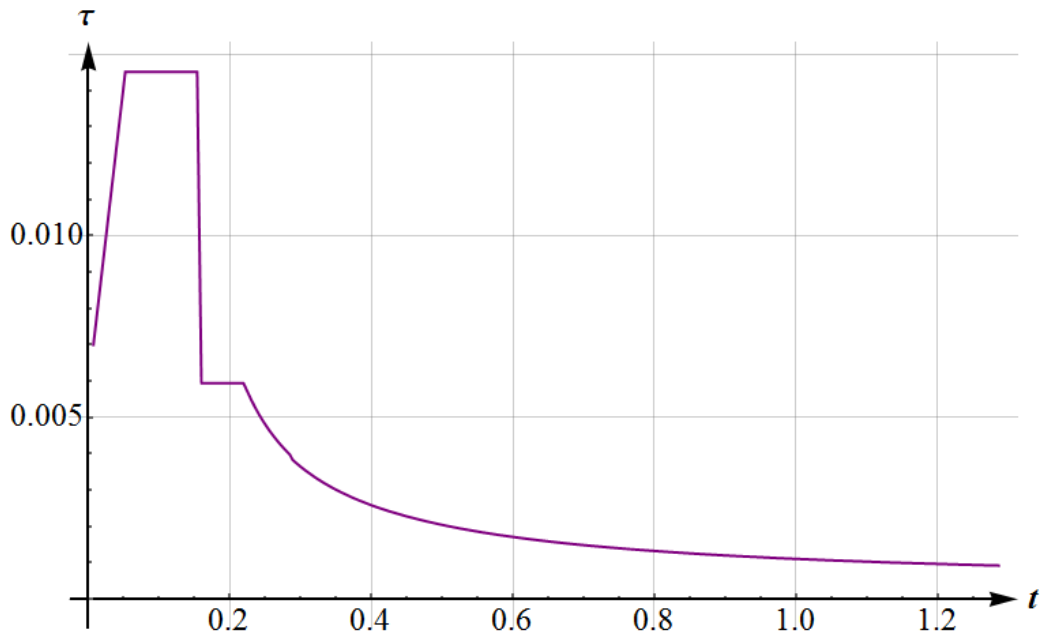


Рис. 18. Изменение шага по времени в процессе расчёта

### 4.3. Использование высокопроизводительных вычислений

Основная вычислительная сложность алгоритма моделирования процесса разгона макротела состоит в решении возникающих вследствие применения численного метода систем линейных алгебраических уравнений (СЛАУ). В случае большого числа неизвестных (порядка миллиона) имеет смысл решать СЛАУ, используя вычислительные кластеры.

Для решения разреженной СЛАУ на ЭВМ с распределенной памятью необходимо распределить пары уравнений — неизвестных по процессам согласно заранее определенной структуре. Это распределение произведём с помощью разделения графа матрицы (рис. 19). Предположим, что каждый субдомен (подграф) относится к отдельному процессу.

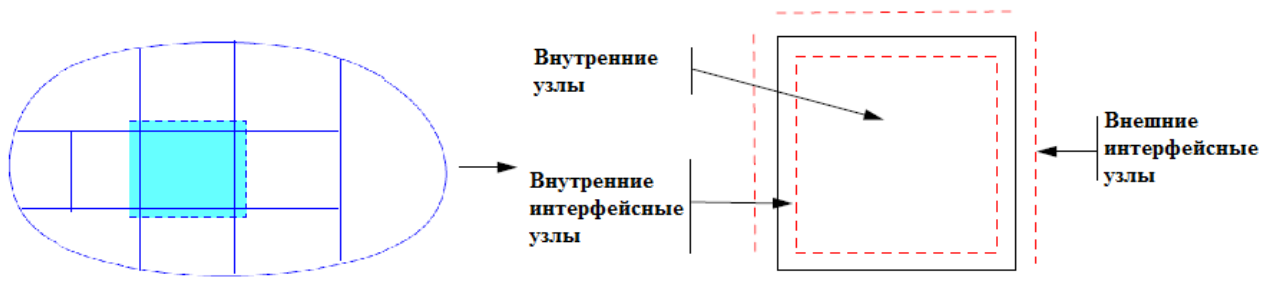


Рис. 19. Схема декомпозиции структуры данных

Локальную структуру данных будем хранить на каждом процессе, так как это позволит выполнять матрично-векторное умножение и предобуславливание эффективнее. Также на каждом процессе следует хранить список процессов, с которыми текущий должен обмениваться данными (назовём их „со-

седними“ процессами, хотя физически они могут располагаться далеко друг от друга), и список локальных узлов, которые взаимодействуют с внешними узлами. Такие узлы назовём локальными интерфейсными узлами.

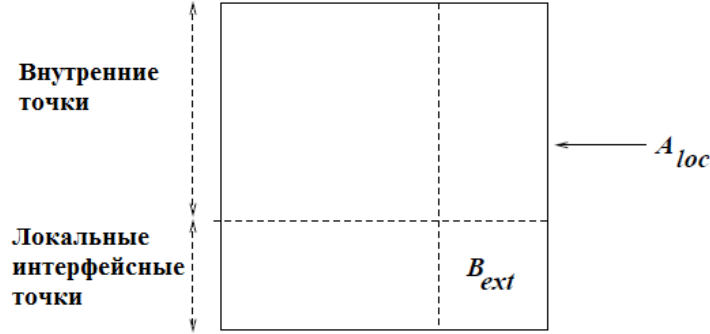


Рис. 20. Структура локальных матриц, относящаяся к каждому процессу

Опишем подробно выполнение операции умножения матрицы  $A$  на вектор  $x$ . Пусть  $A_{loc}$  — локальная матрица, состоящая из части строк (закреплённых за процессом) и всех столбцов матрицы  $A$ . Назовем диагональным блоком  $A_{loc}$  ту часть матрицы, в которой номера столбцов те же, что и номера локальных переменных. Также обозначим недиагональный блок через  $B_{ext}$  (подматрицу матрицы  $A_{loc}$ , чьи ненулевые элементы  $a_{ij}$  таковы, что  $j$  не является локальной переменной этого процесса).

Операция векторно-матричного умножения начинается с перемножения диагонального блока  $A_{loc}$  на локальные переменные. Затем перемножаются матрица  $B_{ext}$  и внешние переменные. Запишем это в виде алгоритма.

- 1) Обменяться интерфейсными данными (раздать данные соседним процессам и собрать данные с соседних процессов).
- 2) Выполнить локальное матрично-векторное умножение  $y = A_{loc}x_{loc}$ .
- 3) Выполнить внешнее матрично-векторное умножение  $y = y + B_{ext}x_{ext}$ .

Здесь  $x_{loc}$  — вектор переменных, которые локальны для данного процесса,  $x_{ext}$  — внешние интерфейсные точки. Матрица  $B_{ext}$  работает с внешними переменными  $x_{ext}$ , отдавая скорректированные данные, которые должны быть добавлены в вектор  $A_{loc}x_{loc}$  для получения желаемого вектора  $y = (Ax)_{loc}$ . Отметим, что второй и третий шаги могут быть выполнены в параллельном режиме.

Вычислительный модуль программного комплекса имеет две реализации: последовательную и параллельную. Для переноса расчёта на многопроцессорный кластер необходимо лишь добавить в папку с входными данными файл с настройками параллельного решателя СЛАУ. Форматы всех остальных входных и выходных данных этих двух реализаций идентичны и обрабатываются с помощью платформы Теметос.

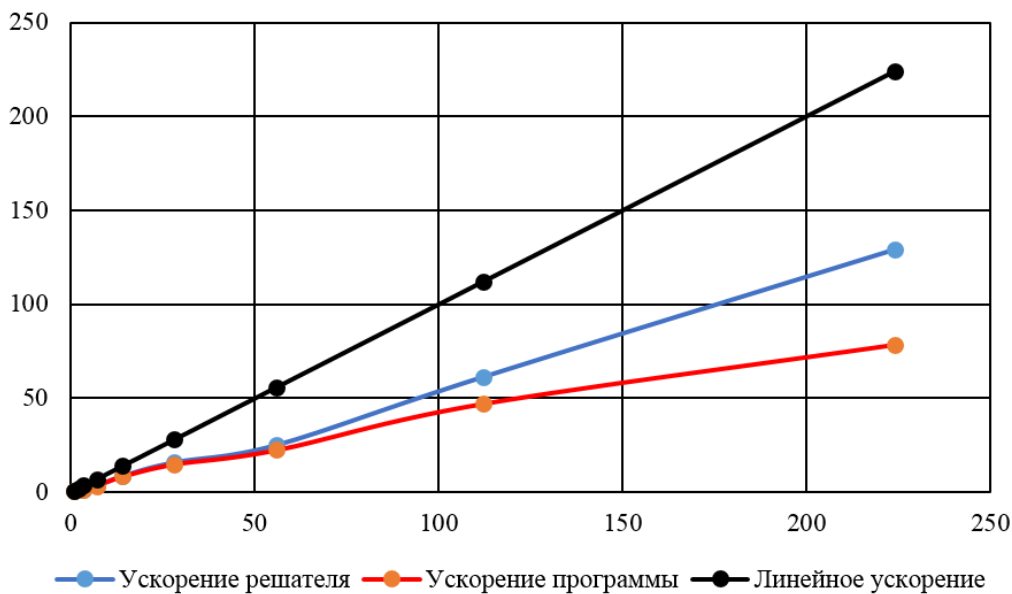


Рис. 21. Зависимость времени расчёта от числа задействованных в расчёте ядер

Исследование ускорения вычислений на кластерной суперЭВМ проведено с использованием вычислительного кластера К-60 ИПМ им. М.В. Келдыша РАН. Для исследования масштабируемости вычислительного модуля задана сетка, содержащая порядка 2 млн. неизвестных и проведены расчёты с использованием разного количества вычислительных ядер.

Отметим, что провести вычисления с использованием одного вычислительного ядра невозможно, поэтому уменьшение времени расчёта изучалось относительно запуска программы на двух вычислительных ядрах. Зависимость уменьшения времени расчёта от количества использованных ядер приведена на рис. 21 (на графике по оси абсцисс отложено относительное число вычислительных ядер).

График на рис. 21 показывает, что разработанный программный комплекс хорошо масштабируется на сотни вычислительных ядер. В то же время алгоритм решения полной задачи содержит участки последовательного кода, что снижает эффективность распараллеливания (по сравнению с теоретической).

## 5. Заключение

Рассмотрена архитектура и основные принципы разработки платформы Теметос. Платформа предназначена для сопровождения вычислительного эксперимента и создания на ее основе программных сред для математического моделирования физических процессов и технических систем. Архитектура платформы, включающая в себя набор предметных библиотек и программные средства для работы с ними, а также ряд стандартов представления данных и интерфейсов их обработки через служебные модули позволяет оснастить авторский проблемноориентированный решатель средствами визуаль-

ной подготовки, проведения и анализа результатов расчетов.

Описано применение платформы для создания программной среды моделирования ускорения тела в электромагнитном ускорителе рельсотронного типа. Особенности математической модели определяют конкретный набор используемых предметных библиотек и разрабатываемых служебных модулей. Продемонстрированы возможности построенной программной среды по визуальной подготовке вычислительного эксперимента. Приведен пример расчета рельсотронного ускорителя, выполненного с использованием платформы Теметос, включая анализ ускорения расчетов при использовании кластерной суперЭВМ.

## Список литературы

- [1] Галанин М.П., Горбунов-Посадов М.М., Ермаков А.В., Лукин В.В., Родин А.С., Шаповалов К.Л. Прототип интегрированной программной платформы для сопровождения вычислительного эксперимента в комплексных задачах математического моделирования // Труды ИСП РАН. 2014. Т. 26, вып. 3. С. 51–68.
- [2] Галанин М.П., Горбунов-Посадов М.М., Ермаков А.В., Лукин В.В., Родин А.С., Шаповалов К.Л. Архитектура программной платформы сопровождения вычислительного эксперимента Теметос // Препринты ИПМ им. М.В. Келдыша. 2013. № 99. 23 с.
- [3] OpenFOAM // The OpenFOAM Foundation. Режим доступа: <http://www.openfoam.org> (дата обращения: 22.03.2018).
- [4] Code\_Aster. Режим доступа: <https://code-aster.org> (дата обращения: 22.03.2018).
- [5] Пакет прикладных программ MARPLE3D для моделирования на высокопроизводительных ЭВМ импульсной магнитоускоренной плазмы / В.А.Гасилов [и др.] // Препринты ИПМ им. М.В.Келдыша. 2011. № 20. 36 с.
- [6] Самарский А.А., Михайлов А.П. Математическое моделирование: Идеи. Методы. Примеры. М.: Физматлит, 2002. 320 с.
- [7] Галанин М.П., Савенков Е.Б. Методы численного анализа математических моделей. М.: Изд-во МГТУ им. Н. Э. Баумана, 2010. 591 с.
- [8] Горбунов-Посадов М.М. Как растет программа // ИПМ им. М.В. Келдыша РАН. Режим доступа: <http://www.keldysh.ru/gorbunov/grow.htm> (дата обращения: 21.03.2018).
- [9] Qt // Digia. Режим доступа: <http://qt.digia.com/> (дата обращения: 22.03.2018).

- [10] Галанин М.П., Лотоцкий А.П., Попов Ю.П., Храмцовский С.С. Численное моделирование пространственно трехмерных явлений при электромагнитном ускорении проводящих макротел // Математическое моделирование. 1999. Т. 11, № 8. С. 3–22.
- [11] Галанин М.П., Лотоцкий А.П., Уразов С.С., Халимуллин Ю.А. Математическое моделирование эрозии металлических контактов в рельсотронном ускорителе // Препринты ИПМ им. М.В. Келдыша. 2003. № 79. 28 с.
- [12] Poltanov A.E., Kondratenko A.K., Glinov A.P., Ryndin V.N. Multi-Turn Railguns: Concept Analysis and Experimental Results // IEEE Transactions On Magnetics. 2001. Vol. 37, № 1. P. 457–461.
- [13] Галанин М.П., Попов Ю.П. Квазистационарные электромагнитные поля в неоднородных средах: математическое моделирование. М.: Наука. Физматлит, 1995. 320 с.
- [14] Уразов С.С. Математическое моделирование многомерных квазистационарных электромагнитных полей в канале электродинамического ускорителя: дис. ... канд. физ.-мат. наук. ИПМ им. М.В. Келдыша РАН, Москва, 2007.
- [15] М.П. Галанин, Н.Н. Прошунин, А.С. Родин, Д.Л. Сорокин. Решение трехмерного нестационарного уравнения теплопроводности методом конечных элементов с учетом фазовых переходов // Препринты ИПМ им. М.В. Келдыша. 2016. № 66. 27 с.
- [16] Галанин М.П., Сорокин Д.Л. Расчёт квазистационарных электромагнитных полей в областях, содержащих несвязные проводящие подобласти // Препринты ИПМ им. М.В. Келдыша. 2017. № 19. 24 с.
- [17] The Visualization Toolkit // Kitware. Режим доступа: <http://www.vtk.org/> (дата обращения: 22.03.2018).
- [18] Калантаров П.Л., Цейтлин Л.А. Расчет индуктивностей. Лен-д: Энергоатомиздат, 1986. 488 с.



# Содержание

<b>1. Введение</b>	<b>3</b>
<b>2. Архитектура платформы Теметос</b>	<b>4</b>
2.1. Основные принципы . . . . .	4
2.2. Система предметных библиотек и базовая платформа . . . . .	5
2.3. Программная реализация платформы: кроссплатформенность и система модулей . . . . .	8
<b>3. Применение платформы Теметос для моделирования ускорителя рельсотронного типа</b>	<b>12</b>
3.1. Математическая модель электромагнитного ускорителя . . . . .	12
3.2. Набор предметных библиотек . . . . .	15
3.3. Служебные модули программной среды . . . . .	16
3.3.1. Модуль задания геометрии электромагнитного ускорителя Geometrix . . . . .	17
3.3.2. Модуль задания сетки Meshix . . . . .	19
3.3.3. Модуль задания физических моделей материалов Constructix . . . . .	20
3.3.4. Модули запуска расчета и графической обработки результатов Launchix и Graphix . . . . .	21
<b>4. Пример проведения полного расчёта</b>	<b>23</b>
4.1. Постановка задачи . . . . .	23
4.2. Результаты расчёта . . . . .	24
4.3. Использование высокопроизводительных вычислений . . . . .	27
<b>5. Заключение</b>	<b>29</b>