



Гасилов В.А., Дьяченко С.В.,
Болдарев А.С., Ольховская О.Г.,
Карташева Е.Л., Багдасаров Г.А.

Пакет прикладных программ
MARPLE3D для
моделирования на
высокопроизводительных
ЭВМ импульсной
магнитоускоренной плазмы

Рекомендуемая форма библиографической ссылки: 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. URL: <http://library.keldysh.ru/preprint.asp?id=2011-20>

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

**В.А.Гасилов, С.В.Дьяченко, А.С.Болдарев,
О.Г.Ольховская, Е.Л.Карташева, Г.А.Багдасаров**

**Пакет прикладных программ
MARPLE3D
для моделирования
на высокопроизводительных ЭВМ
импульсной магнитоускоренной плазмы**

Москва — 2011

**В.А.Гасилов, С.В.Дьяченко, А.С.Болдарев, О.Г.Ольховская,
Е.Л.Карташева, Г.А.Багдасаров**

Пакет прикладных программ MARPLE3D для моделирования на высокопроизводительных ЭВМ импульсной магнитоускоренной плазмы

На основе численных методик с применением неструктурированных расчетных сеток и технологий коллективной разработки прикладных программ в ИПМ им. М.В.Келдыша РАН создан пакет прикладных программ MARPLE3D, предметной областью которого являются задачи радиационной плазмодинамики. Пакет MARPLE3D сконструирован как программное обеспечение для высокопроизводительных параллельных вычислительных систем. В статье дано описание инфраструктуры MARPLE3D и краткий обзор средств разработки, использованные для ее создания. MARPLE3D основан на модели магнитной гидродинамики с учетом диссипативных процессов в веществе и лучистым переносом энергии. Представлены примеры расчетов из предметной области пакета – задачи о сжатии токонесущей плазмы собственным магнитным полем.

Ключевые слова: высокопроизводительные ЭВМ, объектно-ориентированное программирование, магнитная газовая динамика, перенос излучения, магнитоускоренная плазма

**V.A.Gasilov, S.V.Dyachenko, A.S.Boldarev, O.G.Olkhovskaya, E.L.Kartasheva,
G.A.Bagdasarov**

Program package MARPLE3D for simulation of pulsed magnetically driven plasma using high performance computing

The paper presents the new application software (code) MARPLE 3D designed in KIAM RAS by means of progressive numerical techniques using unstructured meshes and modern technologies of a workgroup program development. The application scope of the code is radiative gasdynamics and dynamics of plasmas driven by a magnetic field. The code is developed with the aim of using the high-performance computing systems to large-scale 3D simulations of problems appearing in experimental Pulsed Power Energetic (PPE) and High Energy Density Plasmas (HEDP). The base model is plasma multiphysics including magnetogasdynamics, dissipative processes and radiative transfer. This is an object-oriented, parallel code designed for scientific simulations using high-performance computing (HPC). The paper provides the description of the code infrastructure as well as a survey of tools used for its design and development. To illustrate the code application we present the results of multiparametric studies a Z-pinch flow, i.e. the rapid matter implosion by magnetic field in a current-carrying plasma.

Key words: high-performance computing, object-oriented programming, magnetogasdynamics, radiative transfer, magnetically-driven plasma

Введение

Высокопроизводительные вычисления являются перспективной технологией исследования нелинейных комплексных физических моделей самосжатых электрических разрядов – пинчей, возникающих вследствие электрического пробоя в газоструйных и пенных оболочках, сборках металлических проволочек и в других конструкциях, и прохождения по образующейся ионизированной среде мощного токового импульса [1]. Суперкомпьютеры открывают возможность использования усовершенствованных моделей пинчей и детализации плазменного течения на микронных масштабах, благодаря чему можно будет делать прогнозные расчеты для экспериментов на устройствах с проектируемой мощностью 1 ГВт [1, 2].

Одной из разработок, ориентированных на использование суперкомпьютеров, является объектно-ориентированный программный комплекс (код), созданный в ИММ РАН на основе современных технологий программирования. Код развивается в сотрудничестве с физиками из ФГУП ГНЦ РФ ТРИНИТИ, ИТЭФ, МФТИ, а также французского исследовательского центра CEA (Centre des Etudes de Gramat).

Программное обеспечение (ПО), предназначенное для расчетно-теоретических работ по сопровождению физических экспериментов должно быть универсальным в такой мере, чтобы его можно было быстро адаптировать к изменившимся условиям. Тем самым оно должно иметь в своем составе средства, упрощающие работу с моделями, алгоритмами и данными различной структуры. С другой стороны, создание "параллельного" кода - весьма трудоемкий процесс, затраты на который будут оправданы при условии его эксплуатации пользователями вне команды разработчиков. Поэтому качество современного ПО для научных исследований должно быть на уровне "индустриальных" кодов, так же как и поддержка труда пользователей и сопровождение программ. Перечисленные требования заставляют обратить особое внимание на архитектуру кода и технологии программирования.

В данной работе описывается статус разработки прикладного кода MARPLE3D [3], выполняемой в настоящее время в ИПМ им. М.В.Келдыша РАН. Работы по проекту объектно-ориентированного МГД-кода начались в

Работа поддержана Отделением математических наук РАН (проект 3-ОМН), Российским фондом фундаментальных исследований (проекты 09-01-12109-офи_м, 09-02-01532-а, 10-02-00063-а, 10-02-00449-а), а также исследовательским центром г. Грама (CEA Gramat, Франция) при участии его сотрудников из группы экспериментального комплекса SPHINX. Расчеты выполнены на компьютерах СКИФ МГУ и МВС-100К МСЦ РАН.

2003 году в ИММ РАН и завершились созданием в 2006 г. кода MARPLE2D для моделирования на основе двумерных постановок задач. Данный код использовался в исследованиях систем магнитоплазменной компрессии [4], а также пинчей различных конструкций [5-7]. В исследованиях магнитной имплозии конусныхборок [5] были получены результаты, важные для развития систем квазитрехмерного сжатия токонесущей плазмы – перспективного направления в области импульсной плазменной энергетики. С 2008 года выполняется проект создания кода для трехмерного моделирования процессов в импульсной высокотемпературной плазме. Разработка кода выполняется в соответствии с современными представлениями о технологиях и моделях разработки.

Первый раздел статьи содержит общие сведения о пакете MARPLE. Во втором разделе представлена реализованная в пакете математическая модель плазмотоники. Принципы разработки кода и используемые для этого средства описаны в третьем разделе. Здесь обсуждаются, в частности, вопросы выбора технологий проектирования и программирования, а также использования средств командной разработки программного обеспечения. Четвертый раздел содержит описание основных характеристик созданного кода. В пятом разделе представлен пример вычислительных экспериментов, выполненных средствами кода MARPLE, с целью анализа свойств одной перспективной пинчевой конструкции.

1. Общие сведения о коде MARPLE

В предметную область кода MARPLE в первую очередь входят задачи, связанные с сильноточными импульсными разрядами. Существующая в настоящее время версия кода может применяться для трехмерного моделирования комплекса взаимосвязанных процессов магнитной гидродинамики (МГД) и лучистого теплообмена. Решается система уравнений магнитной гидродинамики в полной трехмерной постановке задач, в декартовой системе координат. В рамках кода имеется возможность выполнять расчеты задач в одномерной и двумерной постановках с использованием аппарата периодических граничных условий. Возможно также решение одномерных и двумерных задач в цилиндрической системе ($R-Z$ геометрия), что реализуется посредством периодических граничных условий и вычислений с модифицированными метрическими коэффициентами.

Код представляет собой совокупность солверов систем уравнений базовых физико-математических моделей и вычислительной среды (инфраструктуры), поддерживающей выполнение расчетов начально-краевых задач математической физики на массивно-параллельных системах с распределенной памятью.

Код включает следующие основные солверные модули:

- 1) Подготовка данных. Этот модуль выполняет построение геометрической модели физической расчетной области, генерацию расчетной сетки, а также усвоение начальных и граничных условий;
- 2) Модуль решения трехмерных идеальных (т.е. в приближении отсутствия диссипации) уравнений магнитной гидродинамики для расчетов в декартовой координатной системе;
- 3) Модуль расчета физических свойств плазмы (термодинамическое состояние вещества, коэффициенты переноса, коэффициенты поглощения и излучения);
- 4) Модуль расчета диссипативных процессов;
- 5) Модуль расчета лучистого теплообмена;
- 6) Модуль управления расчетом, включая определение параметров расчетной методики, процедуры управления итерационными процессами и продвижением решения по времени, диагностику, а также обработку промежуточных результатов расчета в режиме реального времени.

Кроме того, в коде содержится большое число вспомогательных модулей, обеспечивающих полноценную работу физической и вычислительной инфраструктур MARPLE 3D. Следует заметить, что суммарный объем исходного кода вспомогательных (сервисных) модулей существенно превышает объем исходного кода основных солверных модулей.

Расчеты выполняются для скорректированных компьютерных моделей, обеспечивающих однородность вычислений во всем объеме расчетной области. Наряду с общепринятым требованием "автоматического" определения позиций скачкообразных изменений решения, в нашей ситуации однородность подразумевает еще и то, что коррекция основной системы уравнений позволяет использовать приближение МГД по всей расчетной области, включая подобласти сильно разреженного вещества.

Код написан на языке C++. Параллельные вычисления поддерживаются средствами библиотеки MPI.

2. Модели, методы аппроксимации, алгоритмы

В основу кода MARPLE положена модель плазмы, включающая трехмерное описание гидродинамики плазмы в магнитном поле [8]. Решаемая численно система уравнений включает ряд важных эффектов, необходимых для описания высокочастотных плазменных процессов (эффект Холла, возникновение термоэлектродвижущих сил и т.д. [9]). Энергетический баланс плазмы учитывает эффект "двух температур", т.е. энергетическую неравновесность электронов и ионов. МГД-модель дополнена описанием лучистого обмена энергией, играющего важную роль в высокотемпературной плазме [10].

Для успешного решения задач о пинчах на основе многопроволочных сборок весьма важен вопрос описания процесса генерации плазмы при испарении материала проволочной сборки под действием импульса тока. Наша компьютерная модель включает описание протяженного во времени плазмообразования в соответствии с результатами работы [11]. В модель включено электротехническое уравнение, описывающее ток в полной электрической цепи (в типовых задачах импульсной энергетики полная цепь включает генератор, подающие линии и разрядную камеру с плазмой). Теплофизические свойства вещества и характеристики взаимодействия с излучением (излучательная способность и коэффициенты поглощения) используются в форме функциональных или табличных зависимостей. Для большей части расчетов использовались данные, полученные в ИПМ РАН по методам [12]. Для расчетов электровзрывов индивидуальных проволочек использовались как табличные данные на основе методов [12], так и широкодиапазонные уравнения состояния вещества [13].

Течение плазмы рассматривается в одножидкостной МГД-модели. Аппроксимируемая система МГД-уравнений записана в декартовой системе координат. Энергетический баланс записан в двухтемпературной модели, описывающей электрон-ионный энергообмен. Соответственно, полное термическое давление P является суммой электронной (P_e) и ионной (P_i) компонент.

Основная система уравнений имеет вид:

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \vec{w}) = 0$$

$$\frac{\partial}{\partial t} \rho \vec{w} + \text{DIV} \Pi = 0$$

$$\Pi_{ik} = \rho w_i w_k + P \delta_{ik} - \frac{1}{4\pi} \left(B_i B_k - \frac{1}{2} B^2 \delta_{ik} \right)$$

$$\frac{\partial}{\partial t} \vec{B} - \nabla \times (\vec{w} \times \vec{B}) = 0$$

$$\frac{\partial}{\partial t} \left(\rho \varepsilon + \frac{1}{2} \rho \vec{w}^2 + \frac{\vec{B}^2}{8\pi} \right) + \nabla \vec{q} = 0$$

$$\vec{q} = \left(\rho \varepsilon + \frac{1}{2} \rho w^2 + P \right) \vec{w} + \frac{1}{4\pi} \vec{B} \times (\vec{w} \times \vec{B})$$

$$P = P(\rho, \varepsilon)$$

$$\vec{E} = \frac{\vec{j}_{\parallel}}{\sigma_{\parallel}} + \frac{\vec{j}_{\perp}}{\sigma_{\perp}}$$

$$\begin{aligned}
\text{rot}\vec{B} &= \frac{4\pi}{c}\vec{j} + \frac{1}{c}\frac{\partial\vec{E}}{\partial t} \\
\frac{1}{c}\frac{\partial\vec{B}}{\partial t} &= -\text{rot}\vec{E} \\
\frac{\partial(\rho\varepsilon_e)}{\partial t} &= -\text{div}(\hat{\kappa}_e \text{grad}T_e) + Q_{ei} + G_J + G_R \\
\frac{\partial(\rho\varepsilon_i)}{\partial t} &= -\text{div}(\hat{\kappa}_i \text{grad}T_i) - Q_{ei} \\
\varepsilon &= \varepsilon_i + \varepsilon_e, \quad P = P_i + P_e
\end{aligned}$$

Система уравнений РМГД записана в общепринятых обозначениях. компоненты векторов скорости \vec{w} , индукции магнитного поля \vec{B} и напряженности электрического поля \vec{E} . Обозначения интенсивностей: электрон-ионного энергообмена - Q_{ei} , джоулевой диссипации - G_J , выделения лучистой энергии - G_R . Учтена анизотропия транспортных коэффициентов для плазмы в магнитном поле.

Область плотной плазмы пинча обычно бывает окружена слоем разреженной плазмы (периферийная плазма, или "корона" пинча), которая также может быть токонесущей, и для ряда приложений учет процессов в этой области может оказаться существенным. Основную МГД модель для расчетов процессов на периферии пинча необходимо расширить, что в нашем случае сделано за счет использования закона Ома в обобщенной форме. В случае одного сорта ионов это закон имеет вид [9]:

$$\begin{aligned}
\vec{E} &= -\frac{\nabla P_e}{en_e} - \frac{1}{c}\vec{V} \times \vec{B} + \frac{\vec{j}_{\parallel}}{\sigma_{\parallel}} + \frac{\vec{j}_{\perp}}{\sigma_{\perp}} + \sigma_{\Lambda}\vec{B} \times \vec{j} \\
&+ \frac{R}{en_e c}\vec{j} \times \vec{B} - \alpha_{\parallel}\nabla_{\parallel}T_e - \alpha_{\perp}\nabla_{\perp}T_e - \alpha_{\Lambda}(\vec{B} \times \nabla T_e)
\end{aligned}$$

Первый член в правой части обусловлен силой теплового давления, далее идут индуктивное и омические слагаемые, шестое слагаемое определяет эффект Холла, а слагаемые с седьмого по девятое описывают вклад термоэлектрических сил (термо-эдс), причем последнее из них обычно определяют как "эффект Нернста". Все коэффициенты в уравнении закона Ома, т.е. $(\sigma_{\parallel, \perp}; R \sim I; \alpha_{\parallel, \perp, \Lambda})$ зависят от параметров плазмы, таких как: n_e , Z (средний заряд ионов) и T_e , а также, дополнительно, зависят от параметра замагниченности $\chi_e = \omega_{Be}/\nu_{ei}$, где ω_{Be} есть электронная гирочастота, а ν_{ei} есть характерная частота электрон-ионных столкновений. Это уравнение записано в предположении, что инерцией электронов можно пренебречь, что исключает из

рассмотрения высокочастотные плазменные процессы. Здесь не учитываются также эффект вязкости электронного газа, которая принимается ничтожно малой.

Аппроксимации уравнений МГД-модели

Процедура численного интегрирования основной МГД-системы, применяемая в коде, имеет начало в принципе аддитивной аппроксимации [14]. При этом фундаментальным требованием к технике аппроксимации является консервативность результирующей компьютерной модели [14]. В этом отношении используемые нами подходы традиционны, они были развиты в научной школе академика А.А.Самарского для вычислительных экспериментов с комплексными математическими моделями, характерными для физики плазмы.

Идея отдельного учета физических процессов выдвигалась в ряде работ по теории разностных схем. Весьма популярны, например, методы расщепления [23]. При отдельном учете физических процессов почти очевидным образом формируется алгоритм, обеспечивающий работу с разнообразными физическими моделями и возможность их модификаций. Данный принцип делает возможным построение алгоритма продвижения по времени, допускающего независимый расчет всех уравнений, включенных в основную систему. Расчет шага по времени выполняется с использованием явных или неявных по времени аппроксимаций в зависимости от типа уравнения, представляющего определенный физический процесс. Таким образом, компоненты, соответствующие гиперболической подсистеме из общей системы уравнений, представлены в терминах явной аппроксимации, в то время как компоненты, соответствующие параболической и эллиптической подсистемам представлены в терминах неявной аппроксимации.

Суммарная аппроксимация подразумевает не только "расщепление" и последовательный учет вклада физических процессов, но и более тесную связь отдельных разностных уравнений. Эта связь выражается в согласовании аппроксимаций отдельных уравнений системы, с целью обеспечения ряда важных свойств компьютерной модели, от которых зависит ее адекватность исходной физико-математической постановки задачи.

Реализация рассматриваемых математических моделей осуществляется с использованием современных численных методик, обеспечивающих высокий уровень разрешения структуры течения.

Используя принцип суммарной аппроксимации в полной системе уравнений плазмодинамики выделяется подсистема идеальной (бездиссипативной) МГД. Для ее аппроксимации использовано семейство явных дивергентных схем, в которых потоковые члены вычисляются путем специальных реконструкций решения на поверхности контрольного объема. Аппроксимирующие схемы относятся к типу ENO (аппроксимации с существенно пониженным уровнем осцилляций, см., напр., [15]). В отличие от

традиционно рассматриваемых схем на треугольных (тетраэдральных) сетках с узловым хранением переменных ([16-18] и др.), построенные нами схемы используют средние значения величин по ячейкам. Такой способ представления искомых величин позволяет отказаться от построения трехмерных контрольных объемов сложной формы вокруг узлов сетки. В применяемых нами аппроксимациях в качестве контрольных объемов, или ячеек консервативности, выступают непосредственно ячейки расчетной сетки. Сравнительно с конструкциями в виде сеток приузловых контрольных объемов аппроксимации на основе средних по ячейкам имеют и другие преимущества, существенные в случае сеток, состоящих из разнородных элементов. Их подробный анализ будет предметом отдельной публикации.

Для подсистемы, описывающей диссипативные процессы, применяются неявные конечно-элементные схемы на основе метода Галеркина с разрывными базисными функциями (см. напр. [19]). Их использование позволяет построить схемы с такими же ячейками консервативности, как и в схемах идеальной МГД, тем самым существенно облегчая обеспечение консервативности численной модели в целом.

Алгоритм расчета переноса лучистой энергии построен на общепринятом описании поля излучения посредством спектральной интенсивности излучения I_ν , которая характеризует плотность потока энергии фотонов, находящихся в спектральном интервале $[\nu, \nu+d\nu]$, движущихся в элементе телесного угла $d\Omega$ около направления Ω и расположенных в момент времени t в элементе объема dV вблизи точки r [20]

$$I_\nu(\mathbf{r}, \Omega, \nu, t) d\nu d\Omega = h\nu c \cdot f(\mathbf{r}, \Omega, \nu, t) d\nu d\Omega$$

Через I_ν определяются спектральная плотность лучистой энергии U_ν и плотность спектрального потока лучистой энергии W_ν (в дальнейшем – поток лучистой энергии).

$$U_\nu(\mathbf{r}, \nu, t) = h\nu \cdot \int f(\mathbf{r}, \Omega, \nu, t) d\Omega = 1/c \cdot \int I_\nu d\Omega,$$

$$W_\nu(\mathbf{r}, \nu, t) d\nu d\Omega = h\nu c \cdot \int f(\mathbf{r}, \Omega, \nu, t) \Omega d\Omega = \int I_\nu \Omega d\Omega$$

Здесь Ω – единичный вектор телесного угла. Интегрирование выполняется по полному телесному углу.

Вклад переноса лучистой энергии в уравнении баланса энергии учитывается с использованием интегральных величин: полной интенсивности лучистой энергии $I = \int I_\nu d\nu$, полной лучистой энергии $U = \int U_\nu d\nu$, полного потока лучистой энергии $W = \int W_\nu d\nu$.

Интегрирование здесь выполняется по всему спектру частот $0 < \nu < \infty$.

Источник в уравнении энергии вычисляется следующим образом:

$$G_R = - \operatorname{div} W.$$

Для объектов, которые предполагается моделировать кодом MARPLE, можно применять квазистационарное описание поля излучения. Это означает, что состояние поля излучения практически мгновенно подстраивается под текущее состояние вещества и не зависит от предыдущих состояний. В каждый

момент времени распространение фотонов в среде в направлении вектора Ω описывается уравнением переноса [20]:

$$\partial I_\nu / \partial s = \Omega \text{ grad } I_\nu = \kappa_\nu (J_\nu - I_\nu),$$

Где через $\partial(\dots)/\partial s$ обозначена пространственная производная вдоль направления пучка (вектора Ω). Уменьшение энергии в пучке описывается коэффициентом поглощения κ . Производство лучистой энергии веществом описывается излучательной способностью J . Оптические свойства вещества сильно зависят от его температуры $T(\mathbf{r})$ и плотности $\rho(\mathbf{r})$, а также от частоты фотонов ν .

Свойства вещества учитываются посредством многогруппового спектрального приближения. Поскольку уравнение переноса лучистой энергии не содержит производных по спектральной частоте ν , то изучаемый диапазон частот с требуемой степенью подробности делится на k групп: $\nu_1=0 < \nu_2 < \dots < \nu_i < \dots < \nu_k < \infty$. Для каждой группы вычисляются осредненные значения коэффициентов поглощения и излучательной способности, которые и используются для расчета лучистого переноса. Соответственно получаются средние значения интенсивности и энергии для спектральной группы i :

$$I_{[i]} = \int_{\nu_i}^{\nu_{i+1}} I_\nu d\nu, \quad U_{[i]} = \int_{\nu_i}^{\nu_{i+1}} U_\nu d\nu \quad (i = 1, \dots, k).$$

Полная интенсивность лучистой энергии вычисляется суммированием по спектральным группам: $I(\mathbf{r}, \Omega, t) = \int_0^\infty I_\nu d\nu = \sum_{i=1}^k I_{[i]}$.

Средства кода MARPLE позволяют выполнять расчет лучистого теплообмена в диффузионном приближении [5, 18] с коррекцией диффузионных потоков, ограничивающей их в областях больших градиентов плотности излучения с целью повышения точности описания энергообмена между веществом и полем излучения. Для аппроксимации диффузионного уравнения также используется неявная конечно-элементная схема на основе метода Галеркина с разрывными базисными функциями. Это приближение дает физически верные результаты. Оно обеспечивает точный баланс энергии и позволяет правильно оценить общий выход лучистой энергии, но непригодно для диагностики с учетом направленности.

Другой алгоритм, также реализованный в коде, имеет в своей основе Sn – метод [21]. Уравнение переноса аппроксимируется консервативной сеточно-характеристической схемой со специальным (симметризованным) выбором характеристических направлений [22]. Метод характеристик более точен в случае сильной анизотропии поля лучистой энергии, например, вблизи границы раздела плазма-вакуум. В оптически плотной среде он должен адекватно воспроизводить диффузионный предел.

Общей схемой интегрирования по времени служит двухступенчатая схема предиктор-корректор второго порядка аппроксимации. Шаг по времени

ограничен критерием Куранта, кроме того, выбор шага по времени регулируется по количеству итераций в солверах, реализующих неявные расчетные методики.

Вычислительные сетки блочной, нерегулярной или смешанной структуры используются для отображения геометрических условий моделируемых экспериментов, а также составляют основу вычислительных технологий для расчетов пространственно-разномасштабных физических процессов. Допустимы сетки, содержащие разнородные (тетраэдры, пирамиды, гексаэдры и треугольные призмы) и криволинейные элементы. Для описания таких сеток используется геометрическая модель, представленная далее.

Применительно к таким сеткам разработаны оригинальные методики численного решения уравнений в частных производных.

3. Принципы разработки кода

Понятие «мультифизика» подразумевает работу с комплексными математическими моделями, охватывающими описание большого числа взаимодействующих физических процессов. Такая модель может быть скомпонована в виде объединения набора начально-краевых задач математической физики. Так, например, в задачах радиационной плазмодинамики можно выделить задачи для систем уравнений идеальной МГД, задачи кондуктивного и лучистого переноса энергии, диссипации магнитного поля и т.д.

Принципиальное отличие исследовательского кода от специализированного прикладного, служащего, например, для промышленных приложений (авиация, машиностроение, энергетика) состоит в том, что исследовательский код должен быть удобен для поддержания работы по изменениям компьютерных моделей, подходов к дискретизации, и внедрения новых алгоритмов. В то же время коды, создаваемые для трехмерного моделирования, с применением высокопроизводительной параллельной техники, поддерживающие работу с многими моделями, должны по качеству приближаться к промышленным кодам, поскольку их создание и эксплуатация – трудоемкий процесс. Для ввода данных, соответствующих трехмерным моделям, необходимо использование средств компьютерного моделирования геометрических объектов, графических интерфейсов и т.п., развитие которых – отдельная дисциплина программирования, с которой разработчики исследовательских кодов, как правило, не знакомы.

Поэтому исследовательский код для высокопроизводительных вычислений должен обладать следующими качествами:

- робастность, т.е. способность производить расчет в достаточно широком классе входных данных без специальной настройки алгоритма (другими словами, без вмешательства разработчика);

- совместимость с САD-системами, т.е. наличие в коде структур данных, развитых в такой мере, чтобы обеспечить усвоение геометрической информации, создаваемой САD-системой;
- отчуждаемость, т.е. код должен быть подготовлен разработчиками для передачи другим пользователям. Как правило, код, передаваемый от разработчиков к специалистам-прикладникам, снабжается развитым графическим интерфейсом пользователя или гибкими альтернативными средствами конфигурирования (которые в некоторых вариантах использования обеспечивают большую эффективность, чем традиционный графический интерфейс пользователя).

Основные идеи, заложенные при создании MARPLE (как абстрактной вычислительной среды, так и солверных компонентов) это:

- реализация большого числа сервисных функций (ввод-вывод данных, операции с расчетными сетками, поддержка параллельных вычислений, кубатуры и квадратуры, ортогональные полиномиальные базисы и т.д.) на уровне вычислительной инфраструктуры (среды);
- динамическая работа с вычислительными объектами на уровне вычислительной среды, что позволяет создавать объекты физических солверов, аппроксимаций, граничных условий и любые иные вычислительные объекты с минимальными затратами ресурсов и обеспечивает унификацию их интерфейсов, а также автоматическое управление выделяемой памятью;
- автоматическое управление динамическим созданием и уничтожением вычислительных и вспомогательных объектов разных типов, таких как солверы уровней вычислительного домена, физической подобласти и элементарные солверы, аппроксимации, граничные условия.

Составление плана разработки кода обычно начинается с анализа предметной области и выяснение вопроса о том, каким образом должен быть построен код. Здесь принимаются во внимание различные обстоятельства, включая квалификацию разработчиков, трудозатраты на создание кода, характер эксплуатации кода и расходы на эксплуатацию, и т.д.

Разбить большую сложную задачу на несколько более простых подзадач можно различными путями: например, это может быть алгоритмическое разбиение, основанное на порядке происходящих событий, или объектное разбиение, когда выделяются объекты и субъекты некоторого действия. Существуют и иные достаточно распространенные подходы к декомпозиции задач. Практика показывает, что начинать разработку полезнее с объектной декомпозиции [25]. Выделение объектов в предметной области является неформализованной задачей. Особенность нашего решения заключается в том, что в качестве таких объектов были выбраны не элементы физической модели, а сущности, относящиеся к вычислительному процессу – геометрические модели, сетки, аппроксимации, солверы и др.

Объектной декомпозиции задачи соответствует два первых шага разработки программной системы – объектно-ориентированный (ОО) анализ и проектирование. В основе ОО-проектирования лежит представление о том, что программную систему необходимо проектировать как совокупность взаимодействующих друг с другом объектов, рассматривая каждый объект как экземпляр определенного класса, причем классы образуют иерархию. Для наглядного представления и проектирования взаимодействия объектов и иерархий классов в нашем проекте используется унифицированный язык моделирования (Unified Modeling Language – UML).

Соответственно объектной модели язык написания исходного кода программы должен обеспечить поддержку объектов, т.е. абстракции данных, имеющих интерфейс в виде именованных операций и собственные данные, с ограничением доступа к ним. Объекты характеризуются принадлежностью к соответствующим типам (классам), а типы могут наследовать атрибуты супертипов (суперклассов).

Вычислительный процесс представляет собой достаточно неоднородный периодический процесс обмена сообщениями между различными, вообще говоря, динамически создаваемыми, конфигурируемыми и уничтожаемыми программными сущностями (солверами различных уровней, граничными условиями, аппроксимациями, вычислителями потоков и т.д.) физической и вычислительной инфраструктур MARPLE, связи между которыми создаются и уничтожаются также динамически (разумеется, с определенными искусственными ограничениями, введенными с целью оптимизации производительности кода за счет преобразования некоторых таких связей в статические).

В данном проекте наиболее подходящей технологией представляется ОО программирование на языке C++. Опыт использования языка C++ показал, что аппарат производных классов и виртуальных функций, а также методика реализации обобщенного программирования с помощью шаблонов C++ являются весьма эффективным средством для решения прикладных вычислительных задач с использованием техники нерегулярных расчетных сеток.

Объектно-ориентированное и обобщенное программирование на базе языка C++ в совокупности с используемыми нами технологическими решениями обеспечивают:

- Развитые базовые средства, удобные для создания инструментария и унифицированного программного обеспечения геометрического моделирования, включая средства генерации неструктурированных сеток различных типов (блочных, элементно-неоднородных, и т.п).
- Унифицированное описание компонент программного комплекса (структуры данных, их отношения и обрабатывающие процедуры).
- Иерархический доступ к указанным компонентам и специфические процедуры доступа на каждом уровне.

- Коллективное выполнение проекта, возможность замены и/или включения новых участников.
- Увеличенное время активной фазы жизненного цикла разработки и сопровождения проекта (что является особенно актуальным именно для научно-исследовательского кода);
- Пошаговую и комплексную верификацию, тестирование и апробацию созданного программного обеспечения.
- Возможность дальнейшего развития, расширения и изменения отдельных элементов и целых модулей программного комплекса.
- Переносимость программного обеспечения (операционные системы: MS Windows, GNU/Linux, Unix, Mac OS и т.д.; платформы: ПК, высокопроизводительные рабочие станции, многопользовательские вычислительные комплексы, в т.ч. допускающие распределенные вычисления).

Технологии разработки ПО и программирования

При коллективном создании кода силами исследовательской группы используются современные технологии разработки ПО и программирования. Они опираются на частично формализованные модели разработки - смешанный итеративно-инкрементный цикл разработки, система контроля версий исходного кода как составной элемент системы управления конфигурациями ПО, единая система отслеживания ошибок и прогресса исполнения проектных заданий, стандарты проектирования и программирования и т.д., повышающие качество исходного кода. Код пишется на стандартном Си++ (ISO/IEC 14882:2003) с использованием парадигм объектно-ориентированного и обобщенного программирования (шаблоны). Такой подход позволяет получить многофункциональный, надежный, расширяемый код, пригодный к повторному использованию в других проектах.

Система управления версиями

Системы управления версиями (VCS - Version Control System) предназначены для ведения истории развития электронных документов. Они сохраняют ряд версий одного и того же документа, работать с этими версиями получать доступ к этим версиям и находить отличия между версиями, определять авторство сделанных изменений.

VCS отслеживают конфликты, возникающие при изменении одного файла несколькими разработчиками, и предлагают средства их решения. Большинство систем может автоматически разрешить конфликты при условии, что изменялись непересекающиеся части файла. Если автоматическое объединение выполнить не удалось, система может предложить решить проблему вручную.

Ветвление и слияние процесса разработки открывает еще больше возможностей как для разработчиков, так и для пользователей. Разработчики могут создавать ветки кода для экспериментирования (например, для

реализации новой возможности или рефакторинга кода), не мешая другим разработчикам. Современные VCS поддерживают слияние отдельных изменений между ветками и веток целиком.

Итерационная модель разработки подразумевает выпуск рабочей версии программы в конце каждой итерации. Это означает отказ от внедрения новых возможностей в код и начала этапа тестирования кода в текущем состоянии. Часто бывает, что некоторые модули уже протестированы к этому моменту, соответственно, их разработчики в период тестирования не должны вносить изменения в свои модули, чтобы случайно ничего не нарушить. В этом случае формируется новая ветка кода для отладки, в то время как в основной ветке продолжается разработка следующей версии. После завершения тестирования, ветка отладки фиксируется и выпускается новая версия продукта. Пользователи могут работать с протестированной стабильной версией программы, а разработчики переключаются на дальнейшее усовершенствование кода.

В качестве основной системы контроля исходного кода проекта MARPLE была выбрана Subversion - современная и при этом стабильная система, обладающая многими полезными свойствами:

- стандартные функции VCS;
- контроль версий каталогов;
- атомарность транзакций;
- создание ветвей является операцией малозатратной по процессорному времени (это – практически постоянная величина) и экономной (дополнительное место выделяется только для хранения изменений);
- отслеживание слияний;
- блокировка файлов;
- вывод, удобный для анализа компьютерными системами;
- интерактивное разрешение конфликтов;
- эффективная обработка двоичных файлов;
- группы изменяемых файлов и др.

Репозиторий проекта MARPLE включает стандартную иерархию каталогов для хранения исходных файлов и файлов проекта:

- trunk - основная ветка проекта (общая версия, основная для разработчиков);
- branches - альтернативные ветви проекта (специальные рабочие версии, которые планируется в дальнейшем объединить с основной веткой проекта);
- tags - неизменяемые, официально выпущенные и протестированные сборки MARPLE, которые можно считать релиз-версиями.

Система сборки MARPLE

Описание проекта в общем случае представляет собой список исходных файлов и опций, необходимых компилятору и линковщику для сборки исполняемого файла. Для крупного проекта таких опций может потребоваться не один десяток (оптимизирующие опции, подключаемые библиотеки и

другие). При коллективном создании ПО разработчиками могут использоваться различные аппаратные и программные платформы, а также инструменты для сборки. Все это порождает огромное разнообразие в необходимых опциях и даже в самих компиляторах и линковщиках. Поддержка проектных файлов для всех используемых конфигураций в актуальном состоянии не представляется возможным.

Кроссплатформенные системы сборки, такие как GNU Autotools, SCons/waf, CMake и другие, решают (по крайней мере частично, т.к. о полном решении задачи управления конфигурациями с помощью систем сборки речь, конечно, не идет) перечисленные выше проблемы, предоставляя унифицированное описание проекта.

Стандартной системой для сборки проекта MARPLE была выбрана система CMake. CMake - это кроссплатформенная система автоматизации сборки программного обеспечения из исходного кода с поддержкой определенного функционала управления конфигурациями проектов. Файлы проекта CMake представляет собой простые текстовые файлы с инструкциями на специальном достаточно простом языке макросов. Cmake поставляется с большим количеством поисковых макросов, сильно упрощающих запись зависимостей проекта от внешних библиотек и компонентов. Имеется графический интерфейс, предоставляющий наглядный доступ ко всевозможным опциям проекта.

CMake, в отличие от других систем, не занимается непосредственно сборкой, а лишь генерирует файлы управления сборкой для различных целевых платформ и платформ/сред разработки. Таким образом, разработчики и пользователи могут использовать привычные им средства разработки (IDE, компиляторы и другие) для работы с файлами проекта независимо друг от друга.

Система управления проектом

Система управления проектом - это комплекс программ, включающий приложения для планирования задач, составления расписания, распределения ресурсов, совместной работы, общения, быстрого управления, документирования и администрирования системы. Различают несколько типов таких систем: desktop- и web-ориентированные, персональные, одно- и многопользовательские, а также интегрированные. На рынке ПО представлено огромное число реализаций каждого типа систем управления проектом, коммерческих и имеющих в свободном доступе. Выбор конкретного инструмента зависит от многих факторов: численности и распределения коллектива разработчиков, требований к разрабатываемой системе, имеющихся ресурсов.

В качестве централизованной системы управления проектом MARPLE была выбрана система "Trac", ориентированная на разработку ПО небольшим коллективом. К важным для нас преимуществам этой системы можно отнести

простоту установки, конфигурирования, администрирования, гибкость, наличие большого числа специальных плагинов, интеграцию с любыми современными системами контроля версий, простоту освоения и работы, а также наличие удобного Веб-интерфейса, не требующего использования разработчиками специализированного ПО для работы с Trac.

Для организации работы выбранных нами средств (Subversion, Trac) был создан специальный GNU/Linux-сервер, предоставляющий доступ к хранилищу исходных кодов (репозиторию) MARPLE и системе Trac, а также дополнительным сервисам (например, WebDAV-интерфейса к серверному хранилищу файлов). Наиболее используемыми возможностями Trac для нашего коллектива стали интеграция с Subversion, позволяющая быстро просматривать историю кода и конкретные изменения, совместное редактирование документов (wiki) для оформления отчетов и управление проектными заданиями / отслеживанием ошибок (система так называемых билетов). Последнее для нас оказалось чрезвычайно эффективным средством, помогая в разрешении проблем, требующих участия нескольких разработчиков.

Документация

Для нашего кода, как и для большинства исследовательских кодов, характерно большое число настроечных параметров, заданных во внешних файлах конфигурирования времени выполнения. Наличие подробного руководства для запуска, с примерами, является необходимым условием для распространения кода, как за пределами команды разработчиков, так и для внутреннего пользования. Описание различных программных решений является необходимым условием для коллективного владения кодом разработчиками. Это помогает при отладке и локализации ошибок, рефакторинге, ускоряет обучение новых разработчиков. Документирование программных решений можно включать непосредственно в сам код в виде комментариев, оформленных по жесткой формальной схеме. При следовании определенным правилам оформления комментариев в исходном коде становится возможным автоматически извлечь из них информацию и сформировать полное описание программного кода в распространенных форматах (HTML, RTF и другие).

Наш проект выполняется с использованием широко распространенного средства автоматического создания документации по исходному коду - кросс-платформенной системы Doxygen. Doxygen генерирует документацию на основе набора исходных текстов, содержащих специально оформленные комментарии, а также может быть настроен для извлечения структуры программы из недокументированных исходных кодов. Возможно составление графов зависимостей программных объектов, диаграмм классов и исходных кодов с гиперссылками. Doxygen- консольная программа в стиле утилит классической Unix. Она работает подобно языковому транслятору, анализируя исходные тексты и создавая документацию. Для упрощения манипуляций с

конфигурационным файлом существует несколько утилит с графическим интерфейсом.

Doxygen имеет встроенную поддержку генерации документации в формате HTML, LaTeX, man, RTF и XML. Вывод может быть конвертирован в CHM, PostScript и PDF. Когда объем документации MARPLE в формате PDF превысил 1500 страниц формата А4, для ее подготовки преимущественно стал использоваться формат HTML с активным (динамическим) содержимым, более подходящий в качестве контейнера для подобной документации, включающей множество перекрестных ссылок, диаграмм, списков и т.д.

4. Основные характеристики кода MARPLE

Архитектура

Учитывая исследовательский характер кода MARPLE, разработчики прилагали усилия к тому, чтобы его архитектура была настолько общей (расширяемой), насколько это возможно. Архитектура кода спроектирована так, чтобы любые операции модификации исходного кода были «прозрачными», эффективными по затратам и безопасными в плане внесения новых ошибок в ПО.

В соответствии с отмеченными идеями и принципами ОО-проектирования реализация нашего проекта была разделена на разработку проектных модулей: аппроксимация, работа с сетками, солверы, граничные условия и уравнения состояний.

Блочное построение кода существенно для поддержки коллективной работы на всех этапах выполнения проекта (проектирование, программирование, верификация и апробация, моделирование, сопровождение и развитие кода)

Адаптируемая к конкретным условиям архитектура ПО, базирующаяся на программировании средствами C++, предоставляет возможность быстрого обновления кода и делает его удобочитаемым.

Ядро программы составляют:

- Структурированная база данных, включающая всю информацию о постановке задачи (описание расчетной области, свойства среды и материалов, настроечные параметры алгоритмов) и о текущем состоянии процесса (вычисляемые физические величины и вспомогательные функции).
- Набор функциональных модулей, взаимодействующих с этой базой данных (в т.ч. препроцессор, вычислительные блоки, средства первичной обработки и анализа результатов).

Взаимодействие между этими структурами организуется управляющим модулем, получающим информацию из управляющих файлов. Открытые структуры допускают дополнения и изменения, в том числе (с необходимыми

ограничениями) квалифицированным пользователем. Иерархия классов в модулях, а также отношения между ними, представлены на рис. 1. Здесь показаны только существенные классы и отношения, рамками выделены модули.

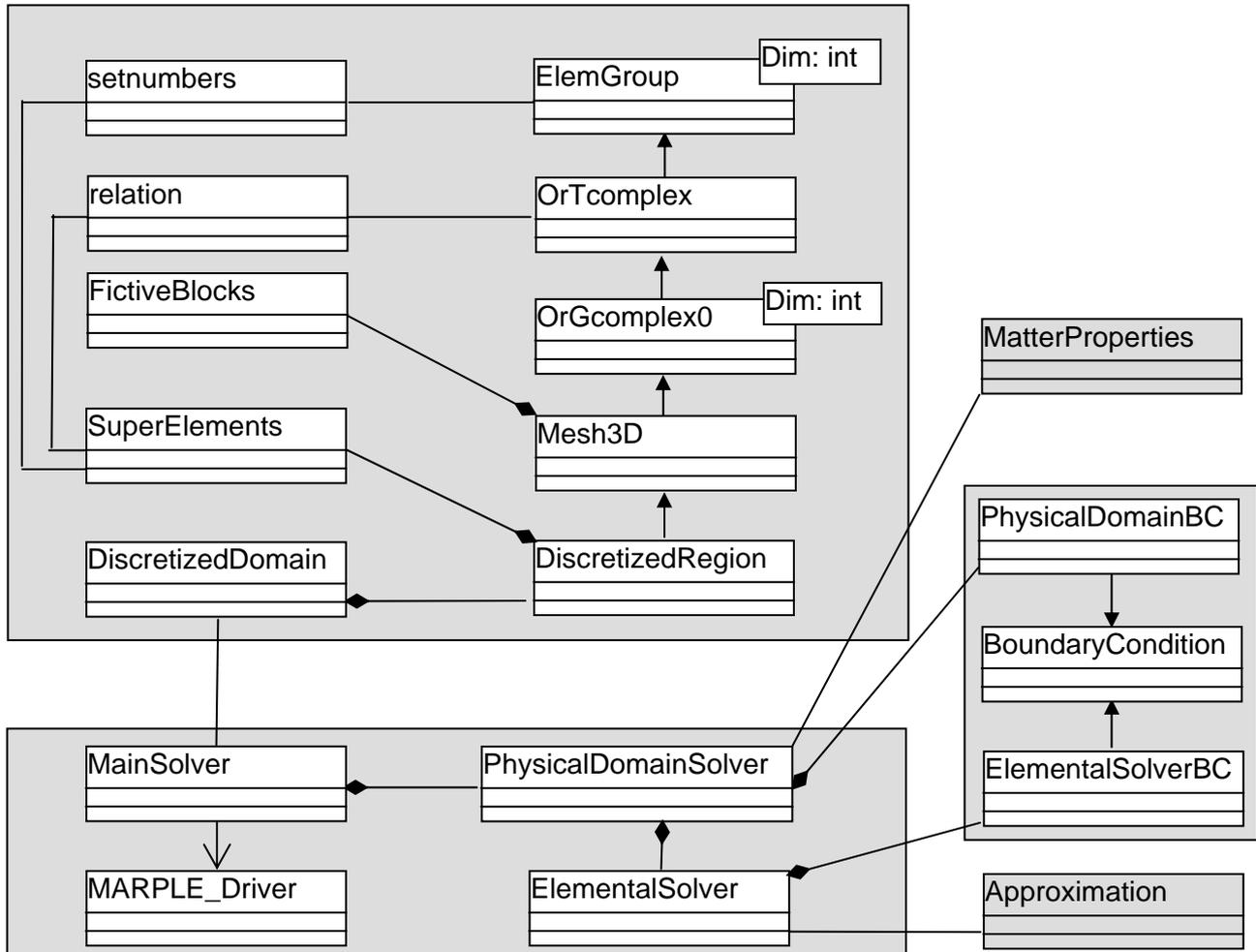


Рис. 1. Диаграмма классов UML: архитектура MARPLE.

- главные солверы (MainSolver) – совокупность физических солверов, выполняют операции верхнего уровня с физическими подобластями, реализуют схему суммарной аппроксимации и схемы аппроксимации по времени;
- физические солверы (PhysicalDomainSolver) – совокупность элементарных солверов, решают систему уравнений в одной физической подобласти;
- элементарные солверы (ElementalSolver), отвечают за решения какого-либо уравнения или системы уравнений;
- элементарные граничные условия (ElementalSolverBC), свои для каждого элементарного солвера, учитывают специфику решаемых элементарным солвером уравнений на границе;
- сложные граничные условия (PhysicalDomainBC) уровня физической подобласти, используются только в том случае, когда не удастся разложить

заданное граничное условие на множество элементарных граничных условий, находящихся ниже в иерархии (ElementalSolverBC);

- аппроксимации (Approximation), предоставляют унифицированный доступ ко всем необходимым для работы элементарных солверов аппроксимациям;
- свойства вещества (MatterProperties), предоставляют унифицированный доступ к уравнениям состояния для каждой физической подобласти.

Выделение аппроксимаций в отдельный модуль с унифицированным интерфейсом позволяет различным вычислительным объектам (солверам и самим аппроксимациям) пользоваться одними и теми же объектами для схожих операций.

В коде используются смешанные сетки нерегулярной структуры. Многоуровневая иерархия классов, представленная на рис. 1 (вверху слева), основана на аппарате топологических комплексов [26] для описания таких сеток. Обобщенная реализация классов позволяет применять их для описания дискретизаций областей любой размерности.

Структуры данных

Код MARPLE разрабатывается как унифицированное программное обеспечение, базирующееся на специализированных структурах данных для поддержки геометрической информации, используемой как в препроцессоре, так и в солвере.

Как показывает опыт, использование индексно-координатных сеток для дискретизации областей сложной геометрической формы во многих случаях не является рациональным. Это относится к моделированию разномасштабных процессов, когда имеют место заметные различия физических параметров в пределах расчетной области. Индексно-координатное описание затрудняет точное воспроизведение процессов вблизи криволинейных границ. Поэтому необходимо использовать комбинированные сетки нерегулярной структуры. Для работы с такими сетками нужен набор геометрических операций: отслеживание структуры сетки и различных отношений между ее элементами, вычисление геометрических характеристик элементов сетки (т. е. длины отрезка, углов, площадей ячеек и т. д.), установление связей различных объектов с сеткой. Кроме того, должна быть обеспечена возможность изменения сетки во время вычислений и использования при вычислениях различных типов инцидентностей (узел – ребро, узел – ячейка и др.).

В состав кода MARPLE входит специализированная геометрическая система. Этот модуль служит для подготовки и предварительной обработки данных, включая создание геометрической модели области физического моделирования, построение сетки, ее оптимизацию и адаптацию, а также подготовку начальных и граничных условий.

Главной задачей подготовки данных для компьютерной модели, основанной на методе сеток, является предоставление дискретного описания

расчетной области с назначением граничных условий и распределением начальных значений.

Используемая нами модель на основе неявных комплексов (Implicit Complexes – IC) [26] обеспечивает унифицированное описание геометрии, топологии и атрибутов гетерогенных объектов. Объект может быть определен произвольным представлением (функциональным или конструктивным представлением геометрии), который обеспечивает корректную дискретизацию. В описание объекта включены его негеометрические атрибуты, а также отношения между его элементами. Основными отношениями, характеризующими расположение элементов объекта, являются отношения двух типов - "граница" и "включение". Негеометрические атрибуты описываются независимо с помощью функциональной или клеточной модели и ассоциируются с ячейками неявного комплекса через отношение атрибутов. Представление на базе IC содержит всю информацию, необходимую для численного моделирования.

В общем случае модель IC включает также описание пересекающихся объектов одной размерности, или представление объекта в виде набора перекрывающихся ячеек. Это могут быть фрагменты расчетной области, выделяемые при описании их физических свойств, при создании полей обмена данными для параллельных вычислений, и другие. Вычислительные технологии с перекрывающимися ячейками и сетками в настоящее время активно развиваются для решения задач механики сплошных сред. Частными случаями неявных комплексов являются симплициальные, полиэдральные и клеточные комплексы, состоящие из неперекрывающихся ячеек. Симплициальные комплексы, например, сетки нерегулярной структуры, состоят, как следует из названия, из симплексов - отрезков, треугольников, тетраэдров. Полиэдральные комплексы формируются из различных многогранников, а клеточные комплексы - из криволинейных полиэдров. Комплексы первых двух типов, как правило, используются для описания неструктурированных расчетных сеток. Клеточные комплексы преимущественно применяются для описания кусочно-аналитических кривых и поверхностей, а также для построения граничных моделей 3D объектов. Клеточные комплексы применяются также и для задания сеток, состоящих из нелинейных элементов.

Негеометрические свойства (атрибуты) объекта и его геометрия описываются независимо.

Технология объектно-ориентированного программирования, применяемая нами, дает возможность независимой реализации главных компонентов неоднородной модели, основанной на неявном комплексе (топологии, геометрии и негеометрических атрибутов неявного комплекса).

Система отношений дает унифицированное описание различных связей между отдельными ячейками, коллекциями ячеек и целыми комплексами. Отношения используются для операций по назначению атрибутов. Использование операций над отношениями дает возможность менять

структуры данных динамически, адаптируя их для специфических приложений. Программная реализация IC включает механизм динамического создания и удаления отношений, эффективный для проектирования многокомпонентных моделей, для работы с клеточными комплексами и сетками.

Предполагается, что комплексы можно менять во время вычислений. Данные, связанные с ячейками IC, не предопределены заранее и могут меняться от одного приложения к другому. Применяемые нами методики обработки событий позволяют поддерживать непрерывную нумерацию ячеек и сохранять связи между ячейками, а также связи между ячейками и данными, ассоциированными с ними.

Структуры данных кода MARPLE предназначены для генерации трехмерных расчетных сеток, построения на них разностных схем и поддержки распределенных вычислений. На этапе генерации сетки предусмотрена ее адаптация к границам расчетной области и измельчение в указанных пользователем областях. Желаемое распределение в пространстве размеров сеточных элементов может быть предварительно определено пользователем, если он располагает априорной информацией об особенностях протекания моделируемых физических процессов. Для задания плотности сетки мы используем функциональное описание.

Распределенная подготовка данных и поддержка параллельного выполнения кода

Для выполнения расчетов на массивно-параллельных системах с распределенной памятью применяется известный подход геометрического параллелизма, основанный на декомпозиции расчетной области (разбиении ее на вычислительные домены, каждый из которых обрабатывается отдельным счетным ядром).

Поддержка геометрического параллелизма встроена в дискретную модель в виде блоков фиктивных элементов (FictiveBlocks и Mesh3D на рис. 1) и средств эффективного обновления расчетных величин в этих блоках (актуализации). Фиктивный блок каждого вычислительного домена содержит сеточные элементы из соседних доменов со специальной маркировкой, позволяющей вычислительным объектам отличать "свои" элементы от "чужих".

Фиктивные блоки также применяются нами для описания различных симметрий в расчетных областях, например, периодических граничных условий.

Наличие нескольких физических подобластей в расчетной области отражено в иерархии классов дискретной модели (DiscretizedRegion и DiscretizedDomain на рис. 1). Объект класса DiscretizedRegion описывает дискретизацию одной физической подобласти на одном вычислительном домене. Для постановки граничных условий между различными физическими подобластями, а также для выделения границ расчетной области применяются

т.н. "суперэлементы", представляющие собой списки сеточных элементов, принадлежащих границам (SuperElements на рис. 1).

Баланс загрузки вычислительных ядер обеспечивается размещением на каждом из них фрагментов всех имеющихся в расчетной области физических подобластей. Для этого расчетная сетка из M физических подобластей при расчете на N вычислительных ядрах разбивается на $(M \cdot N)$ фрагментов (каждая физическая подобласть разбивается на N фрагментов). Список дискретизаций физических подобластей содержится в объекте класса DiscretizedDomain (см. рис. 1) и обрабатывается в главном солвере (объекте класса MainSolver).

Код может запускаться в качестве параллельного приложения MPI на компьютерах с общей или распределенной памятью. Минимальная поддерживаемая версия MPI 1.2.

Совместимость с различными аппаратно-программными платформами

Разрабатываемое ПО совместимо с различными операционными системами (UNIX, MS Windows, GNU/Linux, BSD, Mac OS и т.д.), работать на различных платформах (персональные компьютеры, рабочие станции, ЭВМ коллективного пользования, включая параллельный процессинг), для которых имеются компиляторы C++ стандарта 2003 года. Код не использует каких-либо средств, специфичных для некоторой аппаратно-программной платформы. Код написан на стандартном Си++ (ISO/IEC 14882:2003), используются стандартные возможности MPI 1.2, а для сборки применяется кросс-платформенная система сборки CMake.

Расширяемая архитектура ПО. Модифицируемость исходного кода

Код разрабатывается как мультимодельный инструмент научных исследований, пригодный для расчетов разнообразных по постановкам задач, а для этого требуется так спроектировать архитектуру кода, чтобы обеспечить возможность расширения его предметной области, или адаптации к новым постановкам задач, без изменения базовых средств обработки данных и солверов. Архитектура кода отражает принцип суммарной аппроксимации (раздельной аппроксимации различных физических процессов, представленных уравнениями различных типов). В ней зафиксировано отделение физической составляющей (решаемое уравнение) от вычислительной составляющей (вид аппроксимации по пространству и времени). Такой подход позволяет независимо оперировать такими понятиями, как решаемое уравнение и используемая аппроксимация. Технически данное отделение реализуется через использование по возможности независимых друг от друга и легко заменяемых «вычислительных» (аппроксимации) и «модельных» (уравнения) систем классов, т.е. вычислительной и физической инфраструктур (компонент). Для любых изменений в математической модели или вычислительных методах требуются соответствующие корректировки только на соответствующем

иерархическом уровне MARPLE, без необходимости перестройки других уровней.

Таким образом, код состоит из двух больших компонентов: вычислительная инфраструктура (независимая от физических моделей) и физическая инфраструктура (которая может содержать любое число одновременно поддерживаемых физических моделей). Могут выполняться вычисления одновременно в нескольких физических областях разных типов с привлечением сложных алгоритмов обмена данными между этими областями. Солверы организуются в иерархии (основной управляющий модуль, солверы вычислительных доменов, солверы физических областей, элементарные солверы – см. рис. 1), что делает систему адаптируемой к разнообразным задачам.

Поддержка неструктурированных расчетных сеток со смешанными сеточными элементами

Могут выполняться вычисления на неструктурированных сетках, состоящих из 3D элементов разных типов: шестигранник, тетраэдр, призма с треугольным основанием, пирамида с четырехугольным основанием. Грани указанных 3D элементов могут быть неплоскими (они считаются билинейными). Поддерживаемые 2D элементы: треугольник, четырехугольник.

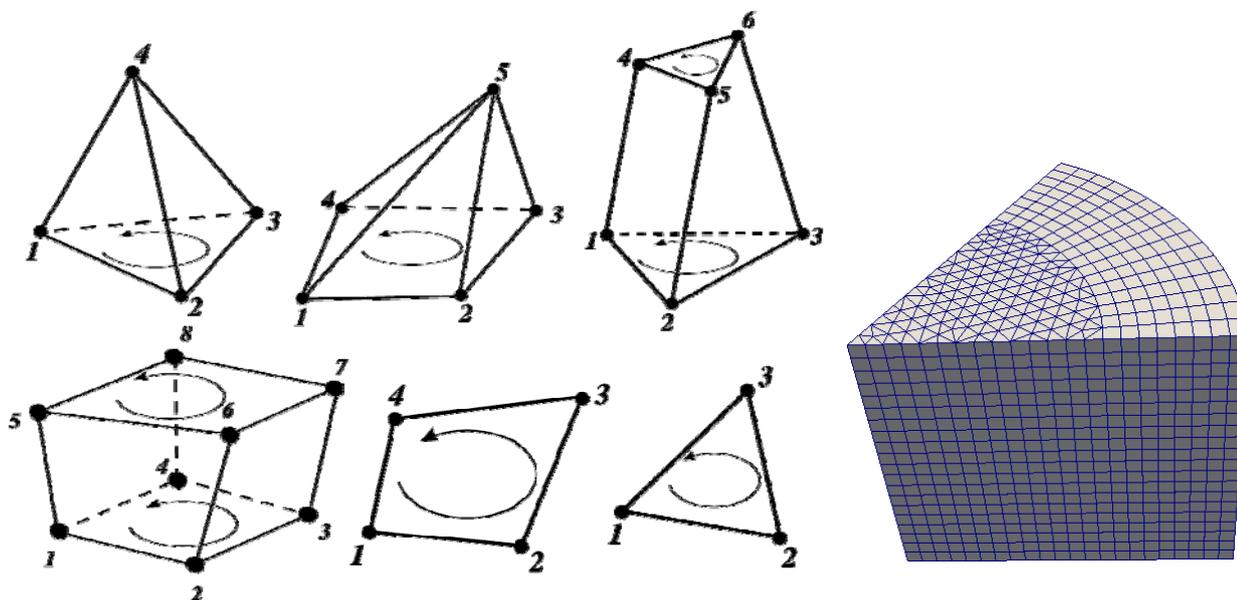


Рис. 2. Виды сеточных элементов и пример расчетной сетки блочно- нерегулярной структуры.

Встроенная поддержка реализации численных схем высокого порядка аппроксимации

Вычислительная среда содержит кубатуры и квадратуры (формулы численного интегрирования) высоких порядков (до 9) для всех

поддерживаемых 3D и 2D элементов, а также иерархические ортогональные полиномиальные базисы для всех указанных типов элементов. Полностью поддерживаются 3D элементы с неплоскими гранями и неплоские 2D элементы. Кубатуры и квадратуры автоматически настраиваются на расчетную сетку (автоматическая самодиагностика, выбор степени формул и т.д.). Кроме того, вычислительная среда содержит модули функций формы, вычисления якобианов, вычисления сеточных метрик и т.д. Все эти инструменты в сильной степени облегчают реализацию высокоточных численных схем на неструктурированных смешанных сетках в рамках инфраструктуры кода.

Динамическое (во время выполнения) автоматическое управление вычислительными объектами

Инфраструктура SAB (Shared Approximations Base) автоматически управляет созданием, инициализацией, конфигурированием и уничтожением вычислительных объектов на основе запросов от высокоуровневых модулей (солверы и т.д.) и любых других вычислительных объектов в режиме времени выполнения. Почти все вычислительные объекты управляются именно этим способом: солверы, граничные условия, аппроксимации, кубатуры и квадратуры и т.д. Любой объект, управляемый в рамках этой инфраструктуры, размещается в памяти только тогда, когда он действительно нужен (зарегистрированы подписчики на него). Любой объект удаляется из памяти, когда он становится не нужен (все подписчики отменяют подписку). Этот механизм предотвращает излишние затраты памяти и утечки памяти, упрощает доступ высокоуровневых модулей и вычислительных объектов к управляемым инфраструктурой SAB объектам, позволяет выполнять замену вычислительных модулей динамически (во время выполнения) и унифицирует процедуру сборки кода (в большинстве случаев неудобные для пользователей препроцессорные манипуляции становятся ненужными, т.к. связи между программными сущностями устанавливаются динамически во время выполнения кода).

Конфигурирование во время выполнения

В большинстве случаев для изменения поведения кода не нужно выполнять повторную сборку. Конфигурирование осуществляется с помощью набора входных конфигурационных файлов, которые считываются при запуске расчета и могут быть загружены повторно во время выполнения кода с целью реконфигурирования программных сущностей (солверов, аппроксимаций, граничных условий и т.д.). Специальный аппарат псевдонимов позволяет подставить любой солвер любого уровня иерархии, граничное условие с любыми параметрами и т.д. вместо другого солвера, граничного условия и т.д. Выключение чего-либо (солвера и т.п.) выполняется тоже с помощью аппарата псевдонимов. Механизм псевдонимов MARPLE позволяют добиться унификации работы пользователя с разнородными программными объектами.

Множество других полезных настроек также могут выполняться с помощью редактирования конфигурационных файлов без повторной сборки кода.

Журналирование событий

Система журналирования событий является конфигурируемым в широких пределах, гибким механизмом, поддерживающим различные типы локальных и распределенных событий (ошибки, предупреждения, уведомления, отладочные сообщения), базовые функции фильтрации сообщений, разные типы выходных потоков и т.д. Развитая система журналирования (протоколирования) событий, возникающих в процессе выполнения кода, является абсолютно необходимой при организации параллельных расчетов сложных задач с помощью научно-исследовательского кода, когда часто приходится прибегать к диагностике и отладке неожиданных ситуаций, что особенно трудно делать именно при параллельном выполнении кода.

Сторонние программные средства.

В проекте MARPLE предусмотрено использование ряда имеющихся в свободном доступе программных средств. Для декомпозиции смешанных сеток нерегулярной структуры использовался один из самых известных и широко применяемых пакетов такого рода – METIS (ParMETIS). Важной задачей является решение в параллельном режиме больших систем линейных алгебраических уравнений, полученных при построении неявных схем. Мы использовали пакет Aztec. Aztec основан на итерационном решении больших линейных систем, которые могут быть плохо обусловлены или получены при моделировании нестационарных процессов. Визуализация результатов параллельных расчетов выполняется посредством системы ParaView. Базой для разработки препроцессора (командный редактор с контролем ввода данных задачи, плюс некоторое графическое сопровождение) является кроссплатформенная система Qt. Для сокращения размера внешних файлов (файлы с сеткой, выходные файлы для визуализации, файлы для сохранения/восстановления расчета) используется библиотека ZLib. Библиотека написана на языке C, и работа с ней из C++-кода "непрозрачна" и подвержена ошибкам. Специальные классы-обертки (zipstream, gzstream) предоставляют стандартный для C++ способ ввода/вывода через потоки.

5. Пример численного моделирования: Z–пинч на основе многопроволочной сборки

В данном разделе рассматривается пример вычислительного эксперимента с задачей из предметной области MARPLE, которая представляет собой типичный пример плазменной "мультифизики". Другими словами, для решения такой задачи необходима поддержка мультидисциплинарных вычислений.

Постановка задачи, выбранной для вычислительных экспериментов, взята из работы [27], в которой представлены данные измерений выхода лучистой энергии и результаты двумерных расчетов применительно к эксперименту № 637 на установке SPHINX французского центра исследований CEA Gramat. Рассматривалась одиночная проволочная сборка цилиндрической формы, состоящая из 240 алюминиевых проволок, имеющих диаметр 10.4 $\mu\text{м}$ и высоту 50 мм. Диаметр сборки 140 мм, диаметр разрядной камеры 150 мм. В качестве граничного условия задавался ток генератора, и вычислялась напряженность магнитного поля на внешней стенке камеры. Плотность фона 10^{-7}г/см^3 . Расчет выполнен при заданном токе генератора (амплитуда 6 МА, время нарастания тока 700 нс). Уравнения состояния, транспортные коэффициенты и оптические свойства использованы в форме таблиц, созданных по методикам [12].

Под действием токового импульса вещество проволок нагревается, испаряется и переходит в плазменное состояние. Возникающее азимутальное магнитное поле сжимает плазму к оси сборки, где в момент максимального сжатия происходит быстрый переход кинетической энергии в энергию излучения. В настоящее время разработка модели так называемого "холодного старта", т.е. расчета электроразряда начиная с состояния проволок как твердых металлических проводников, не завершена по причине недостаточно изученных свойств металлов (главным образом, проводимости) при фазовых переходах, при наличии двухфазных состояний в процессе объемного нагрева электротоком, и т.д. Поэтому расчеты выполнены при упрощенном описании стадии плазмообразования.

Самая простая и наиболее старая модель описания начального состояния пинча соответствует представлениям о мгновенном (очень коротком по сравнению с временем сжатия пинча) взрыве проволочной сборки. Такая модель подразумевает формирование в начальный момент времени сплошной плазменной оболочки при достаточно большом числе проволок в сборке - от нескольких десятков и более. Для расчетов с таким стартовым состоянием начальная конфигурация проволочной сборки принималась в форме плазменной оболочки плотности $1.5 \cdot 10^{-4}\text{г/см}^3$ и температуры 2 эВ. Оболочка была однородной или стохастически возмущенной по плотности.

Более реалистичная модель электрорыва сборки описывает генерацию плазмы как длительный процесс, возникающий в результате нагрева и абляции материала проволок, и сопоставимый по времени с токовым импульсом [11].

Такой процесс приводит к возникновению структуры "предпинча", наблюдаемой в экспериментах. Соответственно модели длительного плазмообразования проволочная сборка в расчете заменяется цилиндрической поверхностью, на которой действует источник плазмы. Неоднородность потока плазмы моделируется стохастическим возмущением (до 5%) темпа появления плазмы.

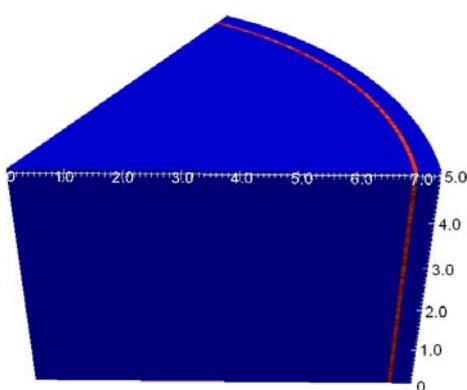
Наконец, третьей моделью, практически реализуемой при условии построения достаточно подробной сетки, может служить модель начального состояния как плазменного, только с учетом дискретности структуры проволочной сборки. В этой модели начальное состояние сборки задано в виде набора индивидуальных проволочек, взорванных токовым импульсом, и перешедших в состояние низкотемпературной, и, вообще говоря, неидеальной плазмы. Параметры вещества проволочек (плотность, температура, диаметр) выбирались в соответствии с экспериментальными данными [28].

Для производства расчетов мы использовали сетку смешанных элементов (шестигранники по периферии цилиндрической разрядной камеры для значений радиуса $R > 2$ см, и треугольные призмы в центральной области, $0 < R \leq 2$ см), построенную в цилиндрическом секторе 60° и состоящую из приблизительно 4 млн. ячеек. Средний линейный размер ячейки $h_\phi \approx 600 \div 400 \mu\text{м}$, $h_r \approx 375 \mu\text{м}$ и $h_z = 400 \mu\text{м}$. (125 ячеек вдоль оси OZ, 120 ячеек вдоль дуги 60° и 200 ячеек вдоль радиуса). Начальная плазменная оболочка имеет толщину 1 ячейки.

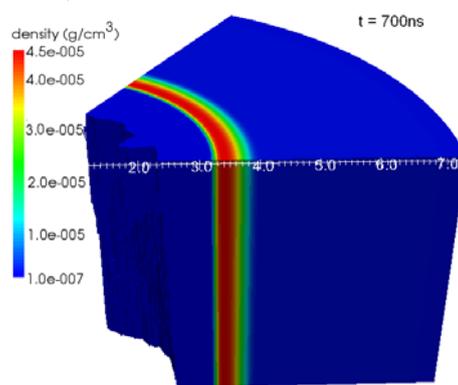
На вертикальных плоских гранях сектора были заданы периодические граничные условия, что эквивалентно полномасштабному 3D моделированию в цилиндре и, в то же время позволяет существенно экономить вычислительные ресурсы при выполнении многопараметрических вычислительных экспериментов. Результаты расчетов по всем трем приведенным моделям начального состояния и генерации плазмы из материала проволочек приведены ниже.

Расчет в модели с начальной плазменной оболочкой:

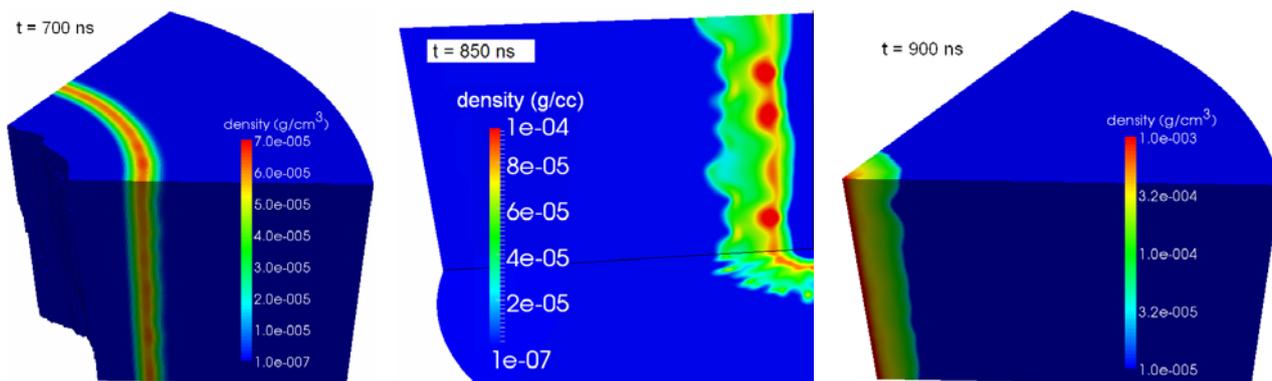
толщина оболочки- $375 \mu\text{м}$, $\rho_0 = 1.5 \cdot 10^{-4} \text{г/см}^3$, $T_0 = 2 \text{ эВ}$



Начальное положение оболочки.

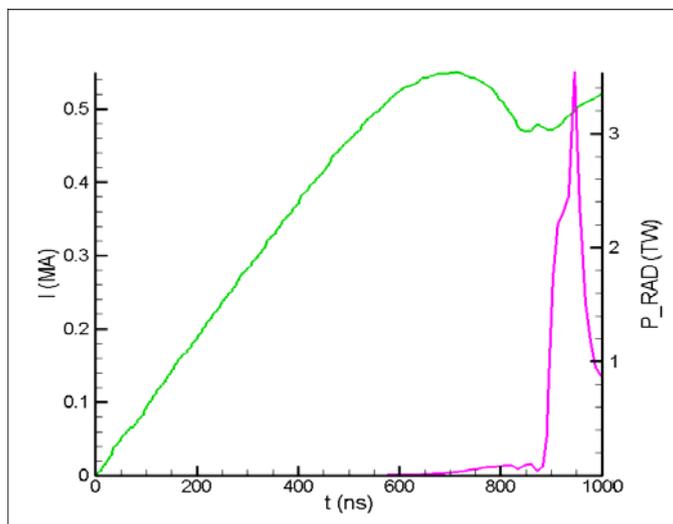


Сжатие однородной оболочки.



Сжатие оболочки со стохастическими возмущениями начальной плотности 6%. Показано состояние на моменты времени 700, 850 900 нс соответственно.

Видно преимущественное развитие возмущений (релей-тейлоровского типа) на внешней стороне плазменной оболочки.



На данном рисунке представлен временной профиль мощности выхода лучистой энергии (фиолетовая линия – выходная мощность, зеленая линия – ток генератора). Максимальная мощность достигла 3.6 ТВт, что весьма близко к экспериментальным результатам.

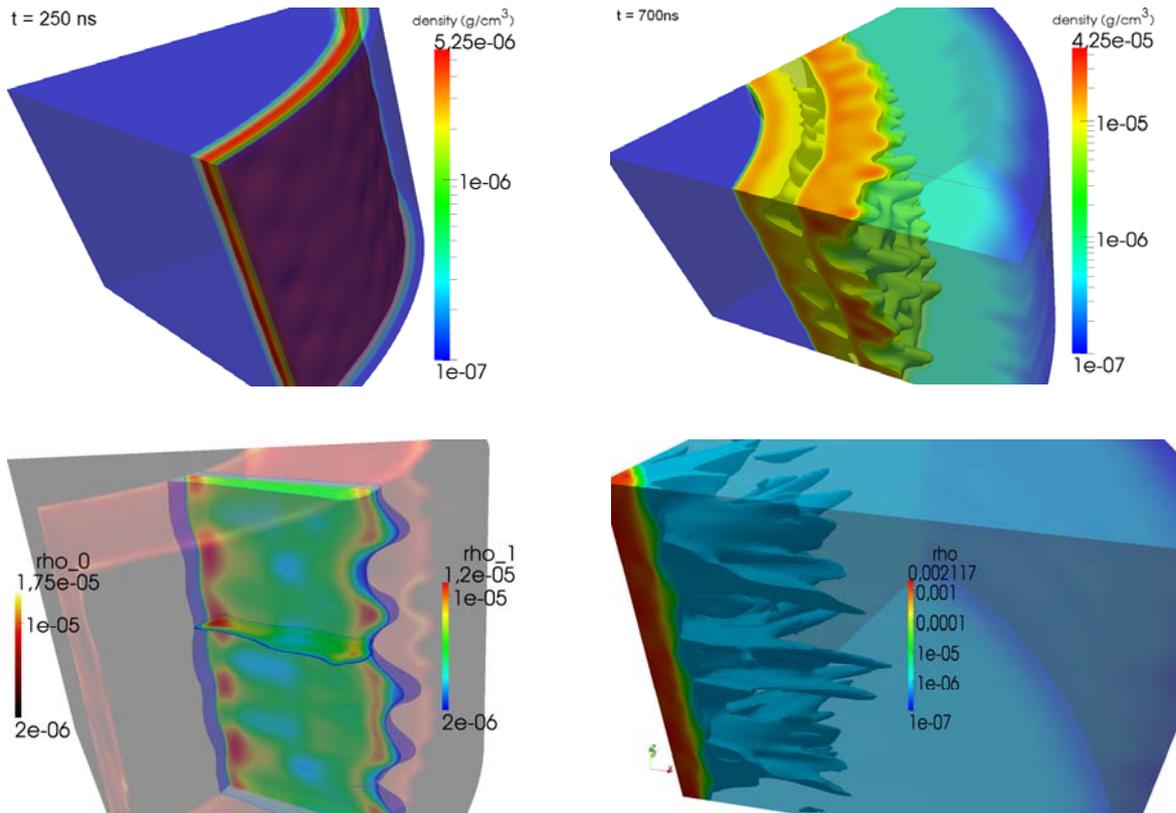
Расчет в модели "источника плазмы"

В данном расчете использована модель протяженной по времени генерации потоков плазмы при испарении импульсом тока материала проволоочной сборки[4]. Модель определяет интенсивность генерации плазмы с единицы поверхности источника представляющего собой неподвижную цилиндрическую поверхность внутри расчетной области:

$$\dot{m}(t) = \begin{cases} kB(t)^2, & t < t_b, \\ \frac{kB(t_b)^2}{M_0(1-\alpha)}(M_0 - m(t)), & t \geq t_b. \end{cases}$$

Здесь $B(t)$ - локальная магнитная индукция, t_b определяется из условия $m(t_b) = \alpha M_0$, M_0 - локальная полная масса, При $\alpha > 1$ режим затухания интенсивность испарения не используется. В данном расчете $k = 0.063$, $M_0 = 2.74 \cdot 10^{-3}$ г.

Неоднородности плазмы, обычно возникающие при электровзрыве проволок, в данном случае моделируются стохастическими возмущениями интенсивности испарения с амплитудой 5%.



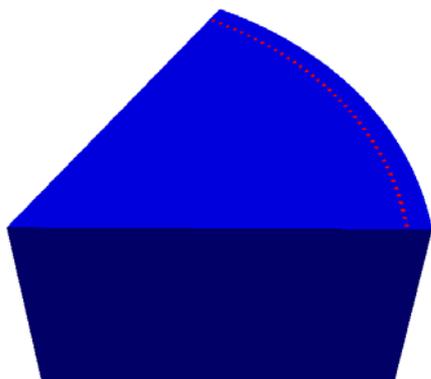
Слева внизу показан мониторинг эволюции плазмы, испаренной в секторе 3° (соответствует 2 проволокам), к моменту $t = 600$ нс.

Справа показано трехмерное распределение плотности для $t = 950$ наносекунд при активизации в программе-визуализаторе выборочной прозрачности вещества в областях с низкоплотной плазмой.

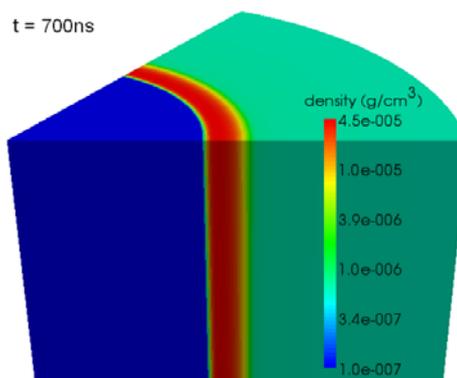
Расчет в модели "взорванной проволоочки"

В завершение приведем результаты расчета, полученные при описании начального состояния вещества проволоочки после электровзрыва согласно данным, полученным в работе [27]. В этом варианте сборка представлена набором индивидуальных проволоочек. Для начала расчета принимается, что вещество проволок однородно, и соответствует состоянию неидеальной низкотемпературной плазмы, при этом параметры проволоочек таковы: диаметр $\varnothing = 400 \mu\text{м}$, плотность $\rho_0 = 4.5 \cdot 10^{-4}$ г/см³, температура $T_0 = 2$ эВ.

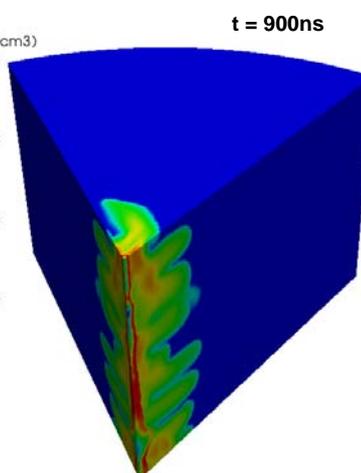
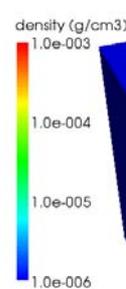
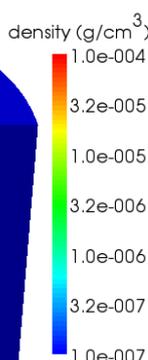
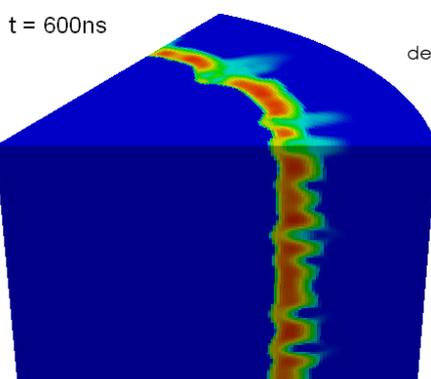
Результаты расчета иллюстрируются на следующих рисунках.



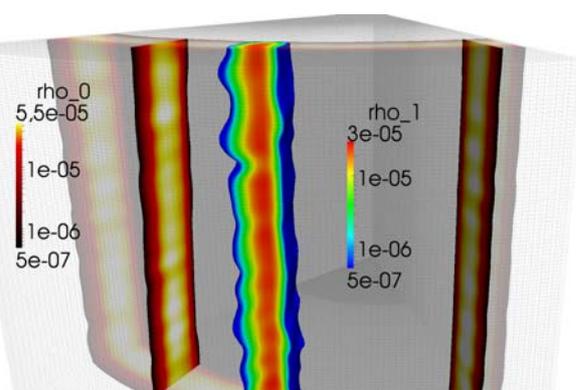
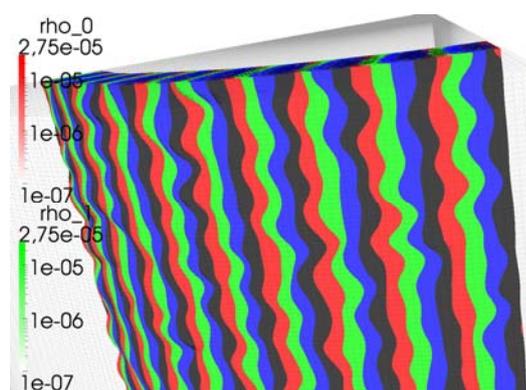
Начальное положение проволок.



Сжатие сборки с невозмущенным начальным состоянием.



Сжатие сборки с стохастическими возмущениями начальной температуры 30%.



На данных рисунках показан мониторинг движения вещества отдельной проволоочки (состояние на время $t = 600$ нс).

Комментарии к расчетам

На эффективность сжатия по отношению к такой важнейшей характеристике пинча, как мощность излучения рентгеновского диапазона,

существенное влияние оказывает асимметрия течения, появляющаяся вследствие гидродинамической неустойчивости сжимающейся плазмы. Возмущения формы оболочки могут возникать как на внешней (в процессе ускорения), так и на внутренней (при торможении около оси) сторонах оболочки. Как показывают выполненные расчеты, для многопроволочной сборки преимущественное развитие получают возмущения периферийных слоев оболочки, в то время как "сердцевина" возмущена слабо. Этот результат соответствует экспериментальным данным и теоретическим оценкам. Для малопроволочных сборок ситуация может быть другой.

Гидродинамическая неустойчивость, возникающая при сжатии пинчей, может оказать заметное влияние на достижимые параметры. Исследование таких неустойчивостей – одна из важных задач 3D-моделирования, т.к. неустойчивость плазмы, ускоряемой электродинамически, является существенно трехмерным эффектом.

В данном примере на периферии пинча можно видеть развитую стадию неустойчивости плазмы. Существенные возмущения формы оболочки в виде отклонений от цилиндрически симметричного движения вещества могут замедлить образование финальной структуры, снизить температуру и плотность пинчевой плазмы. С другой стороны, турбулизация течения плазмы вызывает повышенную диссипацию электрической энергии токового импульса, вследствие чего растет температура вещества. Таким образом, расчет показывает результат конкуренции различных, притом, нелинейных, процессов. Анализ расчетных данных показал, что в результате моделирования были получены физически адекватные результаты, совпадающие с экспериментальными данными по таким показателям, как мощность излучения, время сжатия, размер, плотность и температура пинча. Хорошее соответствие с результатами экспериментов показали также расчеты пинчей при других начальных и краевых условиях. Полученные при апробации кода данные свидетельствуют в пользу того, что построенная трехмерная компьютерная модель Z-пинча является состоятельной для данного класса задач и может быть использована для предсказания результатов будущих экспериментов.

Расчеты с различными типами описания начального состояния показали, что временная развертка мощности излучения хорошо согласуется с экспериментом. Например, в варианте с начальным состоянием в виде отдельных проволочек отклонение расчетного максимума мощности от измеренного экспериментально не превосходит 3.5 процента. Близкие результаты получены в других вариантах описания начального состояния. Отклонение по времени максимума мощности расходится с экспериментом несколько больше – до 10 процентов. Устранение данного расхождения обусловлено внедрением в расчет более точной модели начального состояния пинча, работа над которой в настоящее время ведется. Кроме того, представленные численные результаты могут быть уточнены за счет использования более подробных вычислительных сеток, уточнения

корректировок свойств вещества в диапазоне малых плотностей (имеется в виду модификация УРС и транспортных коэффициентов при плотности ниже пороговой для обеспечения однородности расчета в рамках приближения магнитной гидродинамики), а также за счет дополнительных настроек параметров вычислительного процесса.

Заключение

Практическим результатом разработки кода MARPLE явилось создание программного обеспечения, дающего возможность исследования новых трехмерных постановок задач из области импульсной плазменной энергетики посредством вычислительных экспериментов на высокопроизводительной вычислительной технике. Не менее важно то обстоятельство, что создание кода принесло также опыт использования современных технологий в разработке программного обеспечения высокопроизводительных вычислительных систем, предназначенного для исследований так называемых "больших", т.е. высокозатратных по вычислительным ресурсам комплексных задач механики сплошных сред. К таким задачам относятся, в частности, задачи высокотемпературной газодинамики и магнитной радиационной газодинамики, работа над которыми определила содержание проекта MARPLE. Трехмерные постановки задач механики сплошных сред необходимы для полноценного анализа многих физических явлений, приводящих к появлению сложных геометрических структур в потоках вещества (например, из-за влияния гидродинамической неустойчивости, возникновения турбулентности, и т.д.) Помимо этого трехмерные постановки зачастую рассматриваются с целью полноценного учета геометрических условий, в которых происходят изучаемые явления. При этом речь может идти как о прикладной, так и о фундаментальной, "академической" задаче. Поэтому переход от кодов для решения двумерных задач к кодам, решающим трехмерные задачи, означает усиление значимости и увеличение доли программных средств геометрического моделирования в соответствующем программном обеспечении.

Использование объектно-ориентированного подхода к проектированию и программированию, обобщенного программирования с использованием развитых средств языка C++, а также определенных элементов современных технологий командной разработки программного обеспечения и программирования значительно способствует повышению эффективности разработки и сопровождения долгосрочных коллективных проектов исследовательских кодов. Объектно-ориентированный подход, в частности, технологии множественного наследования, виртуальных функций, RTTI (runtime type information) предоставляют средства для работы с нетривиальными геометрическими объектами (каковой может оказаться исходная расчетная область), сетками нерегулярной структуры, а также для

описания граничных условий, замыкающих условий и свойств среды из обширного, и, в общем случае, открытого списка.

К настоящему времени средствами кода выполнен ряд методических и прикладных расчетов, подтвердивших на практике эффективность идей, заложенных в основание кода. План развития кода предусматривает включение в него новых моделей импульсной плазмы, а также оптимизацию процессов вычислений и обработки данных. В перспективе предусматривается подготовка кода к использованию на вычислительных системах сверхвысокой (пета- и эксафлопной) производительности, которые в предстоящее десятилетие будут внедряться в научные и технологические исследования.

Авторы признательны чл.-корр. РАН Б.Н. Четверушкину за неизменное внимание и многолетнюю поддержку работы, в том числе, в части разработки алгоритмической базы и организации параллельных вычислений. Авторы выражают благодарность российским коллегам, сотрудничество с которыми способствует развитию кода - С.И. Ткаченко, И.В. Гасиловой, В.А. Шмырову, С.Н. Болдыреву, В.В. Александрову, Е.В. Грабовскому, П.В. Сасорову, Ю.Г. Калинину, а также французским коллегам из исследовательского центра CEA Gramat Ж. Грюненвальду и Т. Майяру.

Авторы благодарны В.Г. Новикову, И.В. Ломоносову и Е.А. Апфельбауму за предоставленные данные по свойствам веществ.

Литература

1. Ядерный синтез с инерционным удержанием. Современное состояние и перспективы для энергетики. (сборник статей под ред. чл.-корр. РАН Б.Ю.Шаркова). М.: ФИЗМАТЛИТ, 2005.
2. В.Е.Фортов. Экстремальные состояния вещества. М.:ФИЗМАТЛИТ, 2009.
3. Vladimir Gasilov, Sergei D'yachenko, Olga Olkhovskaya, Alexei Boldarev, Elena Kartasheva, and Sergei Boldyrev. Object-Oriented Programming and Parallel Computing in Radiative Magnetohydrodynamics Simulations. Proceedings ECCOMAS, Venetia, 2008. Section: Parallel Computing: Architectures, Algorithms and Applications. // IOS Press: Advances in Parallel Computing, 2008. Vol. 15, pp. 475-482.
4. В.А.Гасилов, С.В.Дьяченко, А.С.Чуватин, О.Г.Ольховская, А.С.Болдарев, Е.Л.Карташева, Г.А.Багдасаров. Анализ эффективности компрессии магнитного потока плазменным лайнером. – Математическое моделирование, 2009, Т.21, № 11, с. 57-73.
5. Е.В.Грабовский, В.В.Александров, Г.С.Волков, В.А.Гасилов, А.Н.Грибов, А.Н.Грицук, С.В.Дьяченко, В.И.Зайцев, С.Ф.Медовщиков, К.И.Митрофанов, Я.Н.Лаухин, Г.М.Олейник, О.Г.Ольховская, А.А.Самохин, П.В.Сасоров, В.П.Смирнов, И.Н.Фролов. Использование

конусных проволочныхборок для моделирования трехмерных эффектов МГД-сжатия. – Физика плазмы, 2008, т. 34, № 10, с. 885-900.

6. O.Olkhovskaya, E.Grabovskiy, V.Gasilov, V.Alexandrov, A.Boldarev, S.D'yatchenko, I.Frolov, A.Gribov, A.Gritsuk, E.Kartasheva, Ya.Laukhin, S.Medovschikov, K.Mitrofanov, G.Oleynik, A.Samokhin, P.Sasorov, V.Smirnov, G.Volkov, and V.Zaytsev. Numerical Modeling of Plasma Ablation in the Multiwire Arrays Experiments. – Труды 15-го Международного симпозиума по сильноточной электронике, Россия, Томск, 2008. Томск, изд-во ИСЭ СО РАН, с. 259-262.
7. O.G.Olkhovskaya, A.S.Boldarev, V.A.Gasilov, S.V.Dyachenko, E.L.Kartasheva, G.A.Bagdasarov, V.G.Novikov, A.D.Solomyannaya, D.A.Kim, I.Yu.Vichev, V.V.Alexandrov, G.S.Volkov, V.I.Zaytsev, and I.A.Barykov. Point-Type Soft X-Rays Source Formed by Supersonic Gas Jet. – Труды 15-го Международного симпозиума по сильноточной электронике, Россия, Томск, 2008. Томск, изд-во ИСЭ СО РАН, с. 217-220.
8. Имшенник В.С., Боброва Н.А., Динамика столкновительной плазмы Энергоатомиздат, Москва, (1997).
9. Боброва Н.А., Буланов С.В., Сасоров П.В., Высокоэнергетичная плазмодинамика, глава 11, «Энциклопедия низкотемпературной плазмы». Гл. ред. серии В.Е.Фортов. Серия Б. «Справочные приложения, базы и банки данных». Тематический том IX -2. (Отв. ред. А.С.Кингсеп. Москва, ЯНУС-К, 2007, с. 250-277).
10. Я.Б.Зельдович, Ю.П.Райзер. Физика ударных волн и высокотемпературных гидродинамических явлений. М., ФИЗМАТЛИТ, 2008.
11. В.В.Александров, А.В.Браницкий, Г.С.Волков, Е.В.Грабовский, М.В.Зурин, С.Л.Недосеев, Г.М.Олейник, А.А.Самохин, П.В.Сасоров, В.П.Смирнов, М.В.Федулов, И.Н.Фролов. Динамика гетерогенного лайнера с затянутым плазмообразованием. - ФИЗИКА ПЛАЗМЫ, т. 27, № 2, 2001, стр. 99-120.
12. Никифоров А.Ф., Новиков В.Г., Уваров В.Б. Квантово-статистические модели высокотемпературной плазмы. М.: ФИЗМАТЛИТ, 2000.
13. Ломоносов И.В., Фортов В.Е. и др. Метастабильные состояния жидкого металла при электрическом взрыве. ТВТ, т.39, №5, с.728-742, 2001.
14. Самарский А.А. Теория разностных схем. – М., Наука, 1989.
15. Куликовский А.Г., Погорелов Н.В., Семенов А.Ю. Математические вопросы численного решения гиперболических систем уравнений // М.: ФИЗМАТЛИТ, 2001.
16. V.A.Gasilov and S.V.D'yachenko. Quasimonotonous 2D MHD scheme for unstructured meshes. Mathematical Modeling: modern methods and applications. Moscow, Janus-K, 2004, pp.108-125.
17. М.П.Галанин, В.В.Лукин. Разностная схема для решения двумерных задач идеальной МГД на неструктурированных сетках. Препринт ИПМ им. М.В. Келдыша РАН, № 50, 2007.

18. K.Muravski, T.Tanaka. Modern numerical schemes for solving magnetohydrodynamic equations. *Astrophysics and Space Science* 254: 187-210, 1997.
19. Cockburn B., Shu C.W. The local DG method for time-dependent convection-diffusion systems // *SIAM Journal of Numerical Analysis*, 35 (6): pp.2440-2463, 1998.
20. Б.Н. Четверушкин. Математическое моделирование задач динамики излучающего газа. Москва, Наука, 1985.
21. B.G.Carlson, in *Methods in Computational Physics* , p.1, (Academic Press, New York and London, 1963).
22. D. Balsara. Fast and accurate discrete ordinates methods for multidimensional radiative transfer. Part I, basic methods. *Journal of Quantitative Spectroscopy & Radiative Transfer* 69 (2001) 671}707.
23. Г.И.Марчук. Методы вычислительной математики. М., Наука, 1989.
24. Бьерн Страуструп - Язык программирования C++. 3 издание. Издательства: Бином, Невский Диалект, 2004 г.
25. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ // 2-е изд., СПб.: "Невский диалект", 2001.
26. Kartasheva E., et al. An Implicit Complexes Framework for Heterogeneous Objects Modelling, in *Heterogeneous Objects Modelling and Applications*. // *Lecture Notes in Computer Science*, vol. 4889, Springer-Verlag, pp.1-41, 2008.
27. J. Grunenwald, H. Calamy, F.Lassalle, F.Zucchini, A.Loyen, P.Maury, A.Georges, A.Morell, JP. Bedoch. 2D R-Z MHD simulations for Sphinx experiments. Numerical study of single wire array Z-pinches with multi-microsecond current prepulse. 35th IEEE International Conference on Plasma Science June 15-19, 2008, Karlsruhe, Germany
28. С.И.Ткаченко, А.Р.Мингалеев, В.М.Романова, А.Е.Тер-Оганесьян, Т.А.Шелковенко, С.А.Пикуз. Распределение вещества в токопроводящей плазме и плотном керне в канале разряда при взрыве проволочек. *Физика плазмы*, 2009, том 35, № 9, с. 798-818.