

На правах рукописи

ПОПОВ Александр Сергеевич

**АЛГОРИТМЫ И КОМПЛЕКС ПРОГРАММ
ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ПРИ
МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ
КРИТИЧНЫХ ПО ВРЕМЕНИ ПРОЦЕССОВ**

Специальность 05.13.18 – Математическое моделирование,
численные методы и комплексы программ

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Тамбов 2012

Работа выполнена на кафедре «Системы автоматизированного проектирования» федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Тамбовский государственный технический университет» (ФГБОУ ВПО «ТГТУ»).

Научный руководитель

доктор технических наук, профессор,
профессор кафедры «Системы автоматизированного проектирования» ФГБОУ ВПО «ТГТУ»

Литовка Юрий Владимирович

Официальные оппоненты:

доктор технических наук, профессор,
профессор кафедры «Системы искусственного интеллекта» ФГБОУ ВПО «СГТУ им. Гагарина Ю.А.»

Большаков Александр Афанасьевич

доктор технических наук, профессор,
заведующий кафедрой «Информационные технологии, моделирование и управление» ФГБОУ ВПО «ВГУИТ»

Абрамов Геннадий Владимирович

Ведущая организация

федеральное государственное автономное образовательное учреждение высшего профессионального образования «Южный федеральный университет» г. Ростов-на-Дону

Защита состоится 23 мая 2012 г. в 15 часов 30 минут на заседании диссертационного совета Д 212.260.07 при ФГБОУ ВПО «ТГТУ» по адресу: г. Тамбов, ул. Ленинградская, д. 1, ауд. 160.

Отзывы на автореферат в двух экземплярах, заверенные гербовой печатью организации, просим направлять по адресу: 392000, г. Тамбов, ул. Советская, д. 106, ФГБОУ ВПО «ТГТУ», ученому секретарю диссертационного совета Д 212.260.07.

С диссертацией и авторефератом можно ознакомиться в библиотеке ФГБОУ ВПО «ТГТУ».

Автореферат разослан 22 апреля 2012 г.

Ученый секретарь диссертационного совета
доктор технических наук, доцент



С.Я. Егоров

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность исследования. Рассмотрим общую постановку задач, характерную для данной работы: необходимо решить уравнения исходной математической модели при помощи численного метода за заданный или меньший интервал времени. Ход решения задачи назовём процессом критичным по времени. Такие задачи часто возникают при организации оптимального управления либо когда время проведения численного эксперимента превышает время проведения физического эксперимента. Для решения таких задач за заданное время могут быть использованы параллельные вычисления. Данный подход оправдан с экономической и с технологической сторон. Параллельные вычисления обычно ассоциируются с суперкомпьютерами и кластерными системами. С распространением многоядерных процессоров и видеокарт с унифицированной шейдерной архитектурой появилась возможность реализовывать параллельные вычисления с помощью персональных компьютеров. Их стоимость на несколько порядков меньше, чем стоимость суперкомпьютеров. Существует ряд задач, которые можно решить на персональных компьютерах в режиме параллельных вычислений, например: распознавание изображений; расчет электрических и магнитных полей в различных средах; анализ аналоговых сигналов в реальном времени и др. Основные исследования проводились такими учеными как Mark Harris (основоположник направления GPGPU), Sha'Kia Boggan, Daniel M. Pressel, Ryoji Tsuchiyama, Takashi Nakamura, Takuro Iizuka, Akihiro Asahara, Satoshi Miki, B.B Воеводин, Вл.В. Воеводин, Б.Н.Четверушкин. Однако исследований в данной области проводилось немного, количество литературных источников сильно ограничено, существует ряд разрозненных статей, большинство из которых носит обзорный характер. Кроме того, нет единой стратегии для применения аппаратных средств с целью эффективного использования их вычислительных ресурсов.

Поэтому создание единой среды для проектирования, реализации и исследования эффективных численных методов для математического моделирования критичных по времени процессов, а также оптимизации использования аппаратных ресурсов с точки зрения вычислительной нагрузки, является **актуальной** научной и практической проблемой. Работа выполнена в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России 2009 – 2013 годы», государственный контракт № 02.740.11.0624.

Объект и предмет исследования. Объектом исследования являются критичные по времени процессы.

Предметом исследования являются математические модели критичных по времени процессов и алгоритмы решения уравнений таких моделей с использованием параллельных вычислений.

Цель и задачи исследования – уменьшение времени численного решения на ЭВМ уравнений математических моделей критичных по времени процессов при помощи параллельных вычислений.

Для достижения заданной цели в работе решались следующие **задачи**:

- разработка оптимальной, с точки зрения использования вычислительных ресурсов, модели загрузки имеющихся аппаратных средств при использовании параллельных вычислений для решения поставленной задачи за заданное время;

- создание численного метода решения задачи поиска оптимального вычислителя;
- разработка комплекса программ для создания, отладки и тестирования параллельных алгоритмов, на основе задачи определения оптимального вычислителя;
- исследование работоспособности полученного комплекса программ в производственных условиях.

Методы исследования. В диссертационной работе использовались методы математического моделирования, параллельного и объектно-ориентированного программирования.

Научная новизна:

1. Впервые предложена математическая модель для поиска оптимального вычислителя с точки зрения загрузки аппаратных средств отдельно взятой ЭВМ.

2. Модифицированы математические модели: для оптимизации процесса нанесения гальванического покрытия (добавлены уравнения, описывающие поиск оптимального размещения деталей на подвеске, с целью получения наименьшей неравномерности), для анализа аналогового сигнала с синусно-косинусного трансформатора (использован частотный фильтр), для построения трехмерного представления детали по ее двумерному представлению (создано более точное преобразование в полигональное представление через рецепторное).

3. Разработаны эффективные численные методы и алгоритмы решения уравнений модифицированных моделей расчета процесса нанесения гальванического покрытия, анализа аналогового сигнала, построения трехмерного представления детали, адаптированных к применению параллельных вычислений.

4. Предложена структура и внутренняя организация комплекса программ ЭВМ, реализующая приведенные параллельные и последовательные алгоритмы численных методов, а также предложен подход к созданию параллельных алгоритмов.

На защиту выносятся:

1. Математическая модель поиска оптимального вычислителя с точки зрения загрузки аппаратных средств.

2. Модифицированные математические модели оптимизации процесса нанесения гальванического покрытия, анализа аналогового сигнала с синусно-косинусного трансформатора, построения трехмерного представления детали по ее двумерному представлению.

3. Алгоритмы и численные методы решения уравнений моделей оптимизации нанесения гальванического покрытия, анализа аналогового сигнала с синусно-косинусного трансформатора, построения трехмерного представления детали по ее двумерному представлению.

4. Структура комплекса программ для реализации параллельных вычислений.

Практическая ценность работы состоит в разработке программного комплекса, при помощи которого созданы параллельные реализации следующих алгоритмов:

- решения уравнений математической модели распределения электрического поля и модифицированной модели определения оптимального размещения деталей на подвеске в гальванической ванне;
- анализа угла поворота датчика синусно-косинусного трансформатора в реальном времени;
- построения трехмерного представления детали по ее двумерному представлению, а также задача преобразования рецепторного представления в полигональное.

Внедрение. Разработанный комплекс программ успешно прошел производственные испытания в ООО «Гранит-М» (г. Уварово Тамбовской области) и принят к использованию при проектировании перспективного гальвано-оборудования, которое планируется выпускать на этом предприятии. Данный комплекс также принят к использованию в ОАО «Тамбовский завод «Электроприбор» (г. Тамбов) для анализа ряда аналоговых сигналов, поступающих с готовых изделий предприятия, и принятия решения о годности выпускаемой продукции.

Соответствие диссертационной работы паспорту специальности:

1. Разработаны новые математические модели: поиска оптимального вычислителя, оптимизации процесса нанесения гальванического покрытия, анализа аналогового сигнала, построения трехмерного представления детали по ее двумерному представлению. Пункт 1 паспорта специальности.
2. Проведены комплексные исследования разработанных алгоритмов параллельных вычислений с применением современной технологии математического моделирования и вычислительного эксперимента. Пункт 4.
3. Реализованы эффективные численные методы и алгоритмы вычислений уравнений разработанных математических моделей в виде комплекса проблемно-ориентированных программ. Пункт 5.

Апробация работы. Результаты диссертации докладывались и обсуждались на XXI – XXIII международных научных конференциях «Математические методы в технике и технологиях» (Саратов, 2008; Псков, 2009; Саратов, 2010), 7 Международной конференции «Покрyтия и обработка поверхности» (Москва, 2010), на VII Всероссийской научной конференции «Защитные и специальные покрытия, обработка поверхности в машиностроении и приборостроении» (Пенза, 2010).

Публикации. Основные положения диссертации отражены в 12 публикациях, в том числе в 4 статьях в рецензируемых журналах, а также в 2 программах, зарегистрированных в ФИПС.

Личный вклад автора в статьи. В статьях 1, 5, 6, 8 – 10 опубликованы общие принципы выбора вычислителя, используемые в данной работе, в статьях 2, 3, 7 разработаны модифицированные: математическая модель, численный метод и алгоритм построения трехмерного представления детали по ее двумерному представлению, в статьях 4, 8 описан реализованный комплекс

программ, модель выбора вычислителя, математические модели расчета электрического поля, преобразование в трехмерное представление, анализа аналогового представления сигнала с синусно-косинусного трансформатора. В работах 11, 12 реализованы алгоритмы и численные методы математического моделирования описанных в работе процессов.

Структура и объем работы. Диссертация общим объемом 124 страницы состоит из введения, четырех глав основного текста, заключения, списка литературы из 93 наименований и приложений.

ОСНОВНОЕ СОДЕРЖАНИЕ ДИССЕРТАЦИОННОЙ РАБОТЫ

Во введении обоснована актуальность темы исследования, определены цели и задачи диссертационной работы.

В первой главе рассматривается современное состояние процесса создания и реализации алгоритмов параллельных вычислений. Проанализированы современные программные и аппаратные средства для реализации параллельных вычислений.

В настоящее время для расчета уравнений математических моделей критичных по времени процессов используются, в основном, суперкомпьютеры и вычислительные кластеры. При этом компьютером, на котором строится решение искомой задачи, является обычный персональный компьютер. У данного подхода есть ряд ограничений и недостатков, связанных с монопольным использованием суперкомпьютера и вычислительного кластера. К тому же при использовании относительно не дорогих вычислительных кластеров существует ряд ограничений, связанных с производительностью локальной сети, объединяющей ноды кластера.

У этого подхода есть еще один серьезный недостаток: задачи, поступающие на рассмотрение и решение, имеют, в основном, фундаментальный характер, т.е. задача решается однократно, а потом долгое время анализируются результаты, полученные в ходе моделирования. Для задач, которые требуют постоянного многократного решения, приходится выделять отдельный суперкомпьютер, что во многих случаях нецелесообразно с экономической точки зрения и с точки зрения использования его ресурсов.

До недавнего времени альтернативы суперкомпьютерам и кластерам не существовало. В настоящее время компьютеры уже комплектуются средствами, позволяющими реализовать параллельные вычисления с достаточно высокой производительностью. К подобным техническим новшествам в первую очередь относятся видеокарты с унифицированной шейдерной архитектурой. Однако и у них есть свои недостатки: отсутствует удобная среда разработки и проектирования параллельных алгоритмов. До появления стандарта OpenCL, к тому же, не было единого языка программирования и средств исполнения.

На сегодняшний день имеются технические возможности для реализации критичных по времени процессов на персональном компьютере. В то же время возможности написания программных модулей для решения конкретных задач ограничены. В связи с этим поставлена задача исследования: соз-

дать систему для разработки, отладки и тестирования параллельных алгоритмов решения уравнений математических моделей, позволяющую выпускать готовое программное решение для повседневного применения на различных технических средствах. Кроме того, с развитием стандарта OpenCL планируется применение данной системы на суперкомпьютерах и вычислительных кластерах.

Во второй главе сформулирована в математическом виде задача применения параллельных вычислений и рассмотрена методика распараллеливания решения исходной задачи с целью увеличения скорости ее работы. Для этого необходимо решить следующие задачи:

1. Выявить возможность распараллеливания исходного алгоритма, а также последовательные этапы, где присутствует такая возможность.
2. Определить оптимальную, с точки зрения скорости работы параллельного алгоритма, степень разбиения задачи на подзадачи на каждом из этапов для всех присутствующих в данной конфигурации аппаратных средств (вычислителей) и выбрать наиболее подходящий вычислитель.
3. Сформировать параллельный алгоритм на каждом из последовательных этапов, задать входные и выходные данные этапов.
4. Проверить адекватность параллельного алгоритма путем сравнения результатов, полученных при помощи последовательного и параллельного алгоритмов.
5. Оценить эффективность параллельной реализации алгоритма.
6. Разработать комплекс проблемно-ориентированных программ для реализации параллельных алгоритмов.

Рассмотрим подробнее алгоритм задачи 2 (назовём его алгоритмом подбора оптимального, с точки зрения загрузки аппаратных средств, вычислителя).

Необходимо определить набор вычислителей Ω из набора вычислителей, присутствующих в системе P_i , а также степени разбиения на каждом из них N_i с использованием контрольного примера, построенного для текущей задачи Q , при которых разность заданного времени работы $T_{\text{зад}}$ и времени выполнения заданной задачи $T_{\text{рас}}$, полученной на основе выполнения контрольного примера Q и аппроксимированное до размерности и сложности искомой задачи, будет положительной и минимальной:

$$\begin{cases} T_{\text{зад}} - T_{\text{рас}} \geq 0, \\ T_{\text{зад}} - T_{\text{рас}} \rightarrow \min. \end{cases}$$

При следующих ограничениях:

$$\begin{cases} T_{iN}^{\exists} = \text{func}(i, N), \quad N \in [1, N_m], \quad i = 1, 2, \dots, N_D, \\ T_i = Q_T P_i^Q + Q_C P_i^I + (Q_i^S + Q_D) P_i^T \quad Q_i^S = Q_T P_i^L, \\ T_i^T = \frac{T_i N_T}{P_i^C} K_C, \quad T_i^{\exists T} = \frac{T_i^{\exists*} N_T}{N_m} K_C, \end{cases}$$

где T_{iN}^{\exists} – время исполнения на i -м вычислителе с разбиением на N задач; $\text{func}(i, N)$ – вычисляемая функция, выполняющая расчет на вычислителе, с

заданной степенью разбиения и возвращающая время исполнения задачи на нем; i – номер вычислителя; N – количество разбиений; N_m – количество элементарных подзадач (размерность по всем координатам контрольного примера); N_D – количество вычислителей в системе; Q_T – оценочная сложность элементарного разбиения; P_i^Q – производительность одного элемента i -го вычислителя (функция частоты); Q_C – связанность элементарных задач (0 для несвязанных); P_i^t – межэлементная передача данных (отношение производительности вычислителя ко времени передачи между элементами, маленькое для моноблочных вычислителей, большое для различных сетей); Q_i^S – оценочный объем передаваемого элементарного разбиения; Q_D – объем данных элементарного разбиения; P_i^T – время, затрачиваемое на передачу, между основной системой и i -м вычислителем; P_i^L – размер одной команды вычислителя; T_i^T – оценочное время исполнения искомой полноразмерной задачи на i -м вычислителе; P_i^C – количество вычислительных элементов i -го устройства; N_T – размерность исходной задачи; K_C – коэффициент, характеризующий степень упрощения исходной задачи для получения контрольного примера (для задач, в которых получение контрольного примера идет только за счет снижения размерности, коэффициент равен 1); T_i^{3T} – оценочное время исполнения искомой полноразмерной задачи на i -м вычислителе на основе экспериментальных данных при наиболее выгодном разбиении; T_i^{3*} – минимальное время работы при оптимальном разбиении.

Использовать несколько вычислителей в связке целесообразно, когда их производительности примерно равны, а также издержки на передачу данных между вычислителями при высокой связанности искомой задачи ниже времени расчета на них. Для определения набора вычислителей Ω используются следующие дополнительные уравнения для каждого C_j :

$$T_j^C = \sum_{k=1}^{N_j^C} \frac{T_{C_j^k}^{3*} N_{kjT}}{N_m}, \quad N_{kjT} = N_T \left(1/T_{C_j^k}^{3*} \right) / \left(\sum_{l=1}^{N_j^C} 1/T_{C_j^l}^{3*} \right), \quad j = 1, 2, \dots, N_C,$$

где C_j – j -й набор индексов с примерно одинаковой производительностью; N_C – количество наборов; N_j^C – количество элементов в j -м наборе.

При этом в каждый из наборов входит не обязательно максимальное количество вычислителей, кроме того, один вычислитель может участвовать в нескольких наборах. Наборы составляются таким образом, чтобы время выполнения задачи на них оказалось ниже $T_{\text{зад}}$, после чего добавление в набор прекращается. Двух одинаковых наборов не должно существовать. Из всех наборов, у которых время выполнения меньше $T_{\text{зад}}$, выбирается набор с наименьшим количеством вычислителей и наименьшим временем исполнения

при этом. Если наборов, у которых время исполнения ниже $T_{\text{зад}}$ нет, выбирается набор с наименьшим временем исполнения.

Для определения оценочного времени исполнения на нескольких вычислителях, с учетом распределения задачи по вычислителям, используется следующее соотношение:

$$T_j^C = \sum_{k=1}^{N_j^C} \left(N_T / \left(N_m \sum_{l=1}^{N_j^C} 1 / T_{C_l}^{э*} \right) \right).$$

Для решения данной задачи необходимо провести расчет для выбранной задачи ($\text{func}(i, N)$) на всех возможных вычислителях (множество конечное), и найти минимальное время, удовлетворяющее требованиям данной математической модели.

Данная математическая модель, представленная вышеприведенными выражениями, применяется при решении уравнений математических моделей для каждого из представленных в работе примеров, а также для любых других задач, которые будут решаться по данной методике. В начале подбирается вычислитель или набор вычислителей и степень разбиения на подзадачи, при помощи данной методики, после чего происходит расчёт самой задачи.

В третьей главе описан разработанный комплекс программ, позволяющий создавать, отлаживать и тестировать разработанные параллельные алгоритмы для моделирования критичных по времени процессов.

Исходными данными для системы служит параллельный или последовательный алгоритм, работу которого необходимо ускорить, либо повысить точность алгоритма путем увеличения размерности задачи. Для последовательного алгоритма определяются его составные части, решения которых не зависят друг от друга, и, соответственно, их решение может быть произведено в параллельном режиме. В случае невозможности подобного разбиения, необходимо рассматривать альтернативные алгоритмы, для которых подобное разбиение возможно. Для полученного параллельного (или для исходного параллельного) алгоритма выявляется степень его разбиения и определяется связанность отдельных этапов алгоритма. Эти данные необходимы для определения количества разбиений и этапов работы алгоритмов для различных аппаратных средств.

Для реализации разработанных алгоритмов предложена система на основе интерфейса OpenCL как наиболее гибкого и универсального инструмента написания программ с параллельными вычислениями. OpenCL является достаточно новым инструментом, поэтому систем на его основе создано не было. Система позволяет абстрагироваться от платформы и аппаратных средств, на которых производятся вычисления, и сразу приступить к написанию алгоритма решения задачи.

Структурная схема программного комплекса показана на рис. 1.

Программный комплекс для реализации параллельных вычислений состоит из набора модулей для реализации функций инициализации приложения, создания графического интерфейса пользователя, осуществления чте-

ния/сохранения настроек приложения в конфигурационные ini-файлы, сохранения загрузок данных, а также архивации/разархивации файлов при открытии/сохранении на основе открытого алгоритма LZMA. Для работы с внешними аналоговыми источниками данных реализован модуль для работы с аналого-цифровыми преобразователями фирмы ЗАО «Л-Кард». Также создан комплект быстрой разработки библиотек для реализации конкретного алгоритма параллельных вычислений.

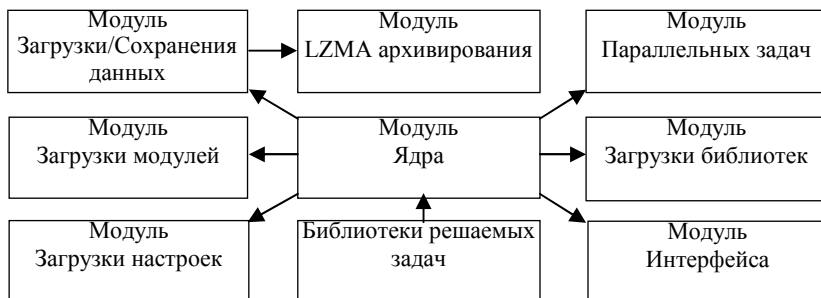


Рис. 1. Структурная схема программного комплекса

Основные классы модулей данной системы:

- модуль ядра реализует общую функциональность комплекса программ, осуществляет загрузку вспомогательных модулей;
- вспомогательные модули – расширяемый набор модулей, реализующих основную функциональность комплекса программ;
- библиотеки решаемых задач – расширяемый набор модулей для реализации конкретного параллельного алгоритма моделирования критичных по времени процессов. Модули можно переключать во время работы.

В четвертой главе рассмотрен ряд критичных по времени процессов, моделирование которых будет выполняться при помощи параллельных алгоритмов с использованием разработанного комплекса программ.

Для анализа увеличения скорости расчета для различных алгоритмов был использован следующий персональный компьютер AMD Athlon X2 3800+/4Гб/NVIDIA GeForce 8800 GTS (320 Mb)/Windows XP Pro SP2.

Пример I. Рассмотрим задачу размещения деталей на подвеске в гальванической ванне, при которой наносимое покрытие будет наиболее равномерным для всех деталей. На основе известного численного метода расчёта процесса нанесения гальванического покрытия используется модификация, состоящая в определении расположения деталей, при котором неравномерность покрытия будет минимальной. Эта задача ранее решалась экспериментальным путем на ограниченной вариации расположений деталей, что является дорогостоящей операцией. Ввиду неформализуемой зависимости неравномерности покрытия от расположения деталей, будем решать задачу полным перебором. Расхождение результатов расчёта неравномерностей покрытий для разных

размещений составляет более 60%. Результаты решения задачи сравнивались с экспериментальными данными, что подтвердило адекватность данного численного метода.

Необходимо найти координаты всех базовых точек деталей (x_i, y_i) , $t = 1, 2, \dots, n$, где n – количество деталей, при которых суммарная неравномерность гальванического покрытия:

$$R_{\Sigma} = \sum_{t=1}^n R_t \rightarrow \min,$$

где R_t – неравномерность покрытия t -й детали.

Неравномерность для каждой из деталей рассчитывается по формуле

$$R_t = \frac{1}{S_{kt}} \int_{S_{kt}} \frac{\delta_t(x, y, z) - \delta_t^{\min}}{\delta_t^{\min}} ds,$$

при следующих ограничениях:

$$\begin{cases} \frac{\partial^2 \varphi(x, y, z)}{\partial x^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial y^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial z^2} = 0, & \frac{\partial \varphi(x, y, z)}{\partial n} \Big|_{S_{\text{и}}} = 0, \\ (\varphi(x, y, z) + F_1(i(x, y, z))) \Big|_{S_a} = U, & (\varphi(x, y, z) + F_2(i(x, y, z))) \Big|_{S_{kt}} = 0, \\ i(x, y, z) = -\chi \text{grad}(\varphi(x, y, z)), & \delta_t(x, y, z) = \frac{\Theta}{\rho} i_t(x, y, z) \Delta \tau, \\ t = 1, 2, \dots, n, \end{cases}$$

где $\varphi(x, y, z)$ – потенциал в точке; $S_{\text{и}}$ – поверхность изолятора (стенки ванны из изолирующего вещества); $F_1(i)$ – функция анодной поляризации; $F_2(i)$ – функция катодной поляризации; U – напряжение между катодом и анодом; S_a – поверхность анода; S_{kt} – поверхность t -го катода; χ – электропроводность электролита; $\delta_t(x, y, z)$ – толщина гальванического покрытия в точке t -го катода; Θ – электрохимический эквивалент вещества; ρ – плотность металла покрытия; $\Delta \tau$ – время процесса.

При этом ограничения переведены в квазистатический вид и принято допущение о нулевом напряжении на катодах и положительном на аноде. Функции F_1, F_2 , в общем случае нелинейные, определяются обработкой экспериментальных данных.

Дополнительные ограничения:

- детали не могут пересекаться и выходить за границы ванны;
- технологическое ограничение силы тока $i_{\min} \leq i_{kt} \leq i_{\max}$;
- минимальная толщина покрытия больше заданной $\delta_t^{\min} \geq \delta_{\text{зад}}^{\min}$.

За базовые точки берутся центры деталей, для удовлетворения условию непересечения и невыхода за границы ванны, будем использовать прямоугольные параллелепипеды, описанные вокруг каждой из деталей, с центром в базовой точке детали, это позволит упростить проверку данных условий и, соответственно, ускорить работу алгоритма.

Было принято решение: ввиду конечности дискретного множества положений деталей в гальванической ванне решать оптимизационную задачу методом полного перебора.

При численном решении задачи в прямоугольных координатах разобьем пространство ванны сеткой. Для понижения порядка задачи используем принцип расщепления: будем рассматривать сечения объема ванны горизонтальными и вертикальными плоскостями XoY , YoZ (y – межэлектродное направление) и для каждого сечения решать двухмерную подзадачу. Получаем два набора не связанных подзадач для плоскостей XoY и YoZ :

$$\left\{ \begin{array}{l} \frac{\partial^2 \varphi(x, y, z)}{\partial x^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial y^2} = 0, \quad \frac{\partial \varphi(x, y, z)}{\partial n} \Big|_{S_n} = 0, \\ (\varphi(x, y, z) + F_1(i(x, y, z))) \Big|_{S_a} = U, \quad (\varphi(x, y, z) + F_2(i(x, y, z))) \Big|_{S_{kt}} = 0, \\ i(x, y, z) = -\chi \text{grad}(\varphi(x, y, z)), \quad \delta_t(x, y, z) = \frac{\partial}{\partial \tau} i_t(x, y, z) \Delta \tau, \quad t = 1, 2, \dots, n. \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial^2 \varphi(x, y, z)}{\partial y^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial z^2} = 0, \quad \frac{\partial \varphi(x, y, z)}{\partial n} \Big|_{S_n} = 0, \\ (\varphi(x, y, z) + F_1(i(x, y, z))) \Big|_{S_a} = U, \quad (\varphi(x, y, z) + F_2(i(x, y, z))) \Big|_{S_{kt}} = 0, \\ i(x, y, z) = -\chi \text{grad}(\varphi(x, y, z)), \quad \delta_t(x, y, z) = \frac{\partial}{\partial \tau} i_t(x, y, z) \Delta \tau, \quad t = 1, 2, \dots, n. \end{array} \right.$$

Обозначим через $\varphi_{i,j,k}^h$, $\varphi_{i,j,k}^V$ значения потенциала в узле сетки с координатами x_i , y_j , z_k , полученные при решении подзадач XoY и ZoY соответственно.

Тогда решение искомой задачи будет определяться как среднее арифметическое $\varphi_{i,j,k}^h$ и $\varphi_{i,j,k}^V$:

$$\varphi_{i,j,k} = \frac{\varphi_{i,j,k}^h + \varphi_{i,j,k}^V}{2}, \quad i = 0, 1, \dots, n_1, \quad j = 0, 1, \dots, n_2, \quad k = 0, 1, \dots, n_3.$$

При решении задачи на каждом из слоев для удовлетворения искомой функцией нелинейных краевых условий III рода применим итерационный метод, являющийся аналогом методов Ньютона и квазилинеаризации. Чтобы получить решение, будем использовать сочетание метода верхней релаксации с прогонкой по строке. Для перехода к дискретной форме будем теперь использовать вместо функции непрерывного аргумента $\varphi(x, y)$, сеточную функцию $\varphi(x_i, y_i)$, которую обозначим как $\varphi_{i,j}$.

Разностный оператор с использованием пятиточечного шаблона:

$$\left\{ \begin{array}{l} \frac{\varphi_{i-1,j} - 2\varphi_{i,j} + \varphi_{i+1,j}}{h_x^2} + \frac{\varphi_{i,j-1} - 2\varphi_{i,j} + \varphi_{i,j+1}}{h_y^2(j)} = 0, \quad \frac{\varphi_{2,j} - \varphi_{1,j}}{h_x} = 0, \\ \frac{\varphi_{n_1,j} - \varphi_{n_1-1,j}}{h_x} = 0, \quad \frac{\varphi_{i,1} + \varphi_{i,2}}{2} + F_1\left(\frac{\varphi_{i,2} - \varphi_{i,1}}{h_y(1)}\right) = U_a, \\ \frac{\varphi_{i,n_2} + \varphi_{i,n_2-1}}{2} + F_2\left(\frac{\varphi_{i,n_1} - \varphi_{i,n_1-1}}{h_y(n_2)}\right) = 0. \end{array} \right.$$

Краевые условия содержат нелинейные функциональные зависимости F_1 , F_2 , что не позволяет решать задачу как систему линейных уравнений.

Зададим начальное приближение функции распределения плотности тока на катоде, т.е. $i^0(i) = (\varphi_{i,n_1} - \varphi_{i,n_1-1})/h_y(n_2)$, $i = 1, 2, \dots, n_1$, и точность ε_1 выполнения краевого условия.

До начала работы метода прогонки необходимо задать начальное приближение функции во всех узлах сетки, т.е. $\varphi_{i,j}^{(k)}$, $k = 0$ и величину ε_2 точности решения (критерия окончания итерационного процесса метода прогонки).

На каждой внутренней строке $j = 2, 3, \dots, n_2 - 1$ методом прогонки решаем уравнение

$$\varphi_{i-1,j}^{(k+0,5)} - 2\varphi_{i,j}^{(k+0,5)} \left(1 + \frac{h_x^2}{h_y^2(j)} \right) + \varphi_{i+1,j}^{(k+0,5)} = \frac{h_x^2}{h_y^2(j)} (\varphi_{i,j+1}^{(k+0,5)} + \varphi_{i,j+1}^{(k)}),$$

где k – номер итерации, $(k + 0,5)$ -я итерация вводится для повышения точности.

Устойчивость метода проверялась с использованием сетки 80...120 шагов, полученные результаты позволяют говорить об устойчивости метода.

При решении задачи имеем следующие результаты вычислений по скорости:

- последовательный алгоритм – 7 сут 4 ч 15 мин;
- для разбиения на центральном процессоре (2 ядра) – 3 сут 22 ч 27 мин ускорение в 1,82 раза, эффективность использования вычислителей – 91%, теоретическое ускорение в 2 раза, исходя из закона Амдала;
- для разбиения на видеокarte 96 вычислителей (8 SIMD блоков 16 потоков данных в каждом) GeForce 8800GTS – 2 ч 6 мин 58,1 с, ускорение в 82,6 раз, эффективность – 85%, теоретическое ускорение в 96 раз.

Пример II. Рассмотрим задачу определения угла поворота датчика на основе синусно-косинусного трансформатора (СКТ) с фильтрацией помех входного сигнала для выделения полезного информационного. При этом используется модификация, основанная на совмещении частотной фильтрации сигнала с алгоритмом преобразования сигнала с СКТ в результирующее значение угла поворота. Фильтрация производится при помощи прямого и обратного преобразований Фурье и обнуления не существенных для нас частот. Этот метод позволяет без применения дополнительных электронных фильтров анализировать полезный сигнал, расхождение значений которого от нефильтрованного сигнала

составляет более 70%. Точность и адекватность метода подтверждена сравнением значений угла, полученных численным методом, и данных, полученных с помощью оптической делительной головки.

Необходимо найти в заданный момент времени t_n дискретное значение угла поворота датчика СКТ α_n на основании исходных данных синуса S_n и косинуса C_n угла при заданной частоте фильтрации сигнала Θ при следующих ограничениях:

$$\begin{cases} X_k = \sum_{n=0}^{N-1} S_n e^{-\frac{2\pi i k n}{N}}, Y_k = \sum_{n=0}^{N-1} C_n e^{-\frac{2\pi i k n}{N}}, S_n^* = \frac{1}{N} \sum_{k=0}^{N-1} X_k^* e^{\frac{2\pi i k n}{N}}, \\ C_n^* = \frac{1}{N} \sum_{k=0}^{N-1} Y_k^* e^{\frac{2\pi i k n}{N}}, \alpha_n = \arctg \frac{S_n^*}{C_n^*}, n = 0, \dots, N-1, k = 0, \dots, N-1, \end{cases}$$

где S_n, C_n – значения исходных сигналов с обмоток трансформатора; N – количество снятых за один цикл точек исходного сигнала (зависит от АЦП); X_k, Y_k – частотные характеристики входных сигналов; X_k^*, Y_k^* – частотные характеристики сигналов после фильтрации; S_n^*, C_n^* – дискретные значения результирующего сигнала синуса и косинуса соответственно с применением частотного фильтра; α_n – дискретные значения угла поворота СКТ.

Для фильтрации исходного сигнала воспользуемся дискретным преобразованием Фурье (ДПФ). Так как датчик работает на частоте 400 Гц, то можно фильтровать сигнал с частотой Θ более 800 Гц. После преобразования в обеих частотных характеристиках обнуляются частоты выше 800 Гц (больше полезного сигнала).

Рассмотрим параллельную и последовательную реализацию основной части алгоритма численного метода решения данного примера.

1. Съем исходных сигналов S_n и $C_n, n = 0, 1, \dots, N-1$.
2. Цикл вычисления прямого ДПФ $k = 0, 1, \dots, N-1$.

$$3. X_k = \sum_{n=0}^{N-1} \left(S_n \cos\left(\frac{\pi k n}{N}\right) \right). \quad 4. Y_k = \sum_{n=0}^{N-1} \left(C_n \cos\left(\frac{\pi k n}{N}\right) \right).$$

5. Увеличение k , если $k \leq N-1$, то переход к п. 3.
6. Обнуление частот характеристики больших Θ .
7. Цикл вычисления обратного ДПФ $n = 0, 1, \dots, N-1$.

$$8. S_n^* = \frac{1}{N} \sum_{k=0}^{N-1} \left(X_k^* \cos\left(\frac{\pi k n}{N}\right) \right) \quad 9. C_n^* = \frac{1}{N} \sum_{k=0}^{N-1} \left(Y_k^* \cos\left(\frac{\pi k n}{N}\right) \right).$$

10. Увеличение n , если $n \leq N-1$, то переход к п. 7.
11. Цикл вычисления угла поворота $n = 0, 1, \dots, N-1$.
12. $\alpha_n = \arctg(S_n^*/C_n^*)$.

13. Увеличение n , если $n \leq N-1$, то переход к п. 11.
14. Вывод результирующей дискретной зависимости угла $\alpha(t)$.

В данном алгоритме имеем три этапа: 2 – 5, 7 – 10 и 11 – 13, на первых двух этапах имеем по две подзадачи, на третьем – одну.

Для проверки устойчивости метода были использованы различные частоты дискретизации при снятии аналогового сигнала 90...110 кГц.

Для получения изменения угла поворота за период 1 с получаем:

- последовательный алгоритм – 3,1 с;
- для разбиения на центральном процессоре (2 ядра) – 1,9 с, ускорение в 1,63 раза, эффективность 82%, теоретическое ускорение в 1,98 раза;
- для разбиения на видеокарте 96 вычислителей (8 SIMD блоков 16 потоков в данных каждом) GeForce 8800GTS – 50 мс, ускорение в 62 раза, эффективность 65%, теоретическое ускорение в 66 раз.

Пример III. Задача построения трехмерного представления детали по ее двумерному представлению. Рассмотрена модификация численного метода преобразования двумерного представления детали в трёхмерное представление, предложенное в работе [2]. Модификация состоит в добавлении дополнительного этапа преобразования в рецепторное представление, а затем в итоговое полигональное представление. В результате добавления дополнительного этапа удастся сократить количество полигонов в итоговом представлении. Для примера при использовании сетки 100×100 и преобразовании представления куба из двумерного представления в трёхмерное получаем: для немодифицированного метода – 58 806 полигонов, для модифицированного – 12 полигонов. Адекватность метода подтверждена применением его для ряда представлений, с известным как двумерным так и трёхмерным видами.

Необходимо определить геометрическое место точек для задания поверхности результирующего трехмерного представления детали в полигональном виде:

$$P_i(x_1, y_1, z_1; x_2, y_2, z_2; x_3, y_3, z_3), \quad i = 1, 2, \dots, N,$$

где P_i – i -й полигон (в качестве полигонов используются треугольники), N – количество полигонов в итоговом представлении, при котором результирующее трехмерное представление будет соответствовать исходному двумерному с заданной точностью, определяемому шагом трехмерной сетки, используемому при нахождении точек поверхности детали.

Заданное исходное представление имеет вид

$$V_j(x_1, y_1, z_1; x_2, y_2, z_2), \quad j = 1, 2, \dots, M,$$

где V_j – отрезок исходного векторного двумерного представления детали; M – количество отрезков.

Входные данные (рис. 2) представляют собой набор отрезков, из которых состоит чертеж детали (дуги, эллипсы и окружности заменяются набором прямых), у каждого отрезка имеется параметр, определяющий тип линии (1 – основная линия, 2 – невидимая линия).

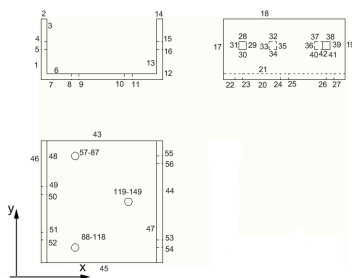


Рис. 2. Общий вид входных данных

Рассматриваемый алгоритм позволяет получить положение точек поверхности детали. В итоге получаем трехмерную матрицу для базового блока (рис. 3, а), после чего строим матрицы для всех блоков и объединяем их при помощи операции исключения лишнего или (рис. 3, б). Для построения полигональной модели по итоговой матрице будем использовать точки, хранящие номера отрезков. Внутренние точки, отмеченные в матрице единицами, не будут находиться на поверхности, следовательно, их можно исключить. Полигоны строятся с учетом непрерывности линий, их образующих, что позволяет получить наименьшее количество полигонов.

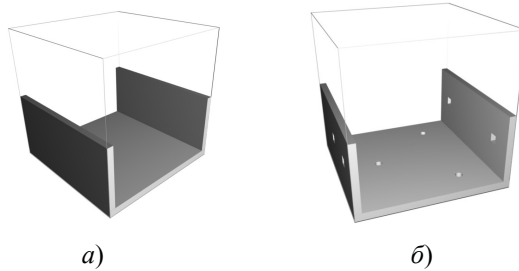


Рис. 3. Матрица базового блока (а). Итоговая матрица (б)

Рассмотрим реализацию основной части алгоритма численного метода.

1. Ввод исходных данных $V_j, j = 1, 2, \dots, M$.
2. Решение уравнений определения проекций $(V_j^f, M_f), (V_j^t, M_t), (V_j^s, M_s)$.
3. Выделение блоков $B_q^f, B_q^t, B_q^l, q = 1, 2, \dots, M_b$.
4. Построение куба и решение уравнений для нахождения и определения массивов точек $D_q(x, y, z)$.
5. Цикл для всех блоков $q = 1, 2, \dots, M_b$.
6. Цикл для всех нормалей $N_i^{ox} \ i = 1, 2, \dots, (n + 1)^2$
7. Решение уравнений для нахождения точек поверхности $D_q(x, y, z)$, для i -й нормали.
8. Увеличение i на 1, если $i \leq (n + 1)^2$, то переход к п. 6.
9. Цикл для всех нормалей $N_i^{oy} \ i = 1, 2, \dots, (n + 1)^2$.
10. Решение уравнений для нахождения точек поверхности $D_q(x, y, z)$, для i -й нормали, с учетом уже заполненных данных.
11. Увеличение i на 1, если $i \leq (n + 1)^2$, то переход к п. 9.
12. Цикл для всех нормалей $N_i^{oz} \ i = 1, 2, \dots, (n + 1)^2$.
13. Решение уравнений для нахождения точек поверхности $D_q(x, y, z)$, для i -й нормали, с учетом уже заполненных данных.
14. Увеличение i на 1, если $i \leq (n + 1)^2$, то переход к п. 12.
15. Проверка краевых точек на вхождение в $D_q(x, y, z)$.

16. Увеличение q на 1, если $q \leq M_b$, то переход к п. 5.

17. Объединение матриц $D_q(x, y, z)$.

18. Построение результирующего полигонального представления.

19. Вывод итогового трехмерного полигонального представления.

Цикл 5 – 16 и 6 – 8, 9 – 11, 12 – 14 можно вычислять параллельно.

Устойчивость метода проверялась с использованием сетки 80...120 шагов, результаты исследования позволяют говорить об устойчивости метода.

Варианты построения трехмерного представления для примера:

- последовательный алгоритм – 36,391 с;
- для разбиения на центральном процессоре (2 ядра) – 19,04 с, ускорение в 1,91 раза, эффективность 95%, теоретическое ускорение в 2 раза;
- для разбиения на видеокарте 96 вычислителей (8 SIMD блоков 16 потоков в данных каждом) GeForce 8800GTS – 420 мс, ускорение в 86,65 раза, эффективность 90%, теоретическое ускорение в 96 раз.

Рассмотрен лишь небольшой набор задач, решение которых невозможно за заданное время при использовании последовательной архитектуры. Адекватность всех математических моделей доказана сравнением с экспериментальными данными.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

1. Разработанные эффективные численные методы для решения уравнений модифицированных математических моделей позволяют ускорить вычисления для расчета: электрического поля в гальванической ванне, построения трехмерного представления детали по двумерному представлению, угла поворота датчика синусно-косинусного трансформатора в 83,87 и 62 раза соответственно.

2. Реализованная модель определения эффективного разбиения и подбора вычислителя позволяет выбрать оптимальную с точки зрения загрузки вычислительную платформу, что ускоряет и упрощает процесс разработки параллельных алгоритмов.

3. Разработан комплекс проблемно-ориентированных программ для реализации параллельных алгоритмов, позволяющий создавать, модифицировать и использовать готовые программные решения.

4. Практическая ценность результатов, полученных в диссертационном исследовании, подтверждена актами внедрения в работу на предприятиях ООО «Гранит-М» и ОАО «Тамбовский завод «Электроприбор».

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в рецензируемых журналах:

1. САПР гальванических процессов / А.С. Попов, Ю.В. Литовка, Г.А. Кириченко, М.А. Попова // Вестник Тамбовского государственного технического университета. – Т. 14. – № 4. – Тамбов, 2008. – С. 882 – 891.

2. Алгоритм формирования объемной геометрической модели детали из чертежа проекций / А.С. Попов, Ю.В. Литовка, В.В. Пэк, М.А. Попова //

Вестник АГТУ. Сер. «Управление, вычислительная техника и информатика». – Астрахань, 2009. – № 2 – С. 152 – 160.

3. Построение трехмерной сетки детали для расчета распределения гальванического покрытия по ее поверхности / А.С. Попов, Ю.В. Литовка, М.А. Попова // САПР и графика. – М., 2010. – № 1. – С. 68–69.

4. Система исполнения параллельных вычислительных процессов / А.С. Попов, Ю.В. Литовка // Радиотехника. – М., 2010. – № 5. – С. 60 – 67.

Прочие публикации:

5. Попов, А.С. Постановка задачи решения уравнения Лапласа при помощи параллельных вычислений / А.С. Попов, Ю.В. Литовка // Математические методы в технике и технологиях : сб. тр. XXI Междунар. науч. конф. – Т. 6. – Саратов, 2008. – С. 89–90.

6. Попов, А.С. О параллелизации вычислений в САПР гальванических процессов / А.С. Попов, Ю.В. Литовка // Математические методы в технике и технологиях : сб. трудов XXII Междунар. науч. конф. – Т. 10. – Псков, 2009. – С. 99 – 101.

7. Попов, А.С. Ввод графической информации в системе управления гальваническими процессами / А.С. Попов, Ю.В. Литовка, М.А. Попова // Математические методы в технике и технологиях : сб. тр. XXIII Междунар. науч. конф. – Саратов, 2010. – С. 47–48.

8. Попов, А.С. Система разработки и отладки алгоритмов параллельных вычислений / А.С. Попов // Математические методы в технике и технологиях : сб. тр. XXIII Междунар. науч. конф. – Саратов, 2010. – С. 157 – 159.

9. Система автоматизированного проектирования и управления гальваническими процессами / А.С. Попов, Ю.В. Литовка, Г.А. Кириченко, М.А. Попова // Тез. докл. 7 Междунар. конф. «Покрытия и обработка поверхности». – М., 2010. – С. 57–58.

10. Проблемы разработки комплексов программ моделирования и оптимизации гальванических процессов / Ю.В. Литовка, Г.А. Кириченко, М.А. Попова, А.С. Попов // Защитные и специальные покрытия, обработка поверхности в машиностроении и приборостроении : тез. докл. VII Всерос. науч. конф. – Пенза, 2010. – С. 46 – 48.

11. Свидетельство об официальной регистрации программы для ЭВМ № 2010614479. Система для разработки отладки и тестирования параллельных алгоритмов / А.С. Попов. – 2010.

12. Свидетельство об официальной регистрации программы для ЭВМ № 2010614480. Программа для построения трехмерной модели по двумерному чертежу / А.С. Попов, М.А. Попова. – 2010.

Подписано в печать 17.04.2012
Формат 60 × 84/16. 0,93 усл.-печ. л. Тираж 100 экз. Заказ № 192

Издательско-полиграфический центр ФГБОУ ВПО «ТГТУ»
392000, Тамбов, ул. Советская, д. 106, к. 14