

Глава 4. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ: ОТ МАШИННОГО ОБУЧЕНИЯ К ОПТИМАЛЬНЫМ РЕШЕНИЯМ

Эволюция человечества, говоря философским языком, непременно связана с получением новых знаний, с приобретением эффективных навыков жизнедеятельности и с принятием оптимизирующих решений. Еще на заре появления цифровых машин создатель прообраза ЭВМ Чарльз Баббидж вместе со своей сподвижницей, первой в мире программистской Адой Левлейс, отмечали, что числовое устройство может быть использовано для аналитических операций [1]. Уже во второй половине XX в., когда с помощью ламповых и полупроводниковых ЭВМ решались стратегические проблемы ядерной и ракетной безопасности, разрабатывались пионерские проекты по автоматизации формульных преобразований, построения алгоритмов и доказательным операциям, заложившим основы будущего искусственного интеллекта (ИИ). В те же времена после знаменитых работ Норберта Винера появилась наука кибернетика, акцентирующаяся отнюдь не на арифметике, а на управлении сложными системами, к которым относится в том числе и человеческое сообщество.

Острейшие дискуссии разгорались вокруг вопроса “может ли машина мыслить?”. И здесь поучительным примером является история с компьютерными шахматами. Чемпион мира в 1948–1963 гг. (с двумя годичными перерывами) Михаил Ботвинник придумал алгоритм игры в шахматы (по крайней мере, он так думал), основанный не на тривиальном переборе ходов, а на качественной оценке позиции. Он нашел сподвижников-энтузиастов среди математиков-программистов в Москве и новосибирском Академгородке, которые этой проблемой профессионально занимались и добились определенных успехов. В 1978 г. в руководимом Г.И. Марчуком ВЦ СО АН СССР шахматистом-перворазрядником Владимиром Бутенко была защищена по этой теме кандидатская диссертация. Кстати, в 1974 г. в Стокгольме во время проведения там очередного конгресса Международной федерации по информационным процессам (ИФИП, International

Federation on Information Processing) с многотысячной аудиторией состоялся Первый чемпионат мира по компьютерным шахматам, на котором победила советская программа Каисса. В том же году кем-то предлагалось баснословное пари о том, сможет ли шахматная программа играть в силу международного мастера? Как известно, история эту дилемму убедительно решила, и теперь машина обыгрывает даже чемпиона мира. Что поразительно, это делается в основном за счет сверхбыстрого перебора ходов, без каких-либо супералгоритмов. А это означает, что серьезнейшая интеллектуальная проблема решена главным образом за счет наращивания “мускулов” ЭВМ.

Чтобы покончить с шахматной темой, приведем очень важный с методологической точки зрения пример с “активными шахматами”. Как известно, современный гроссмейстер готовится к соревнованиям без компьютера не может, но во время игры его использование категорически запрещено. Тринадцатый чемпион мира Гарри Каспаров предложил альтернативный вариант: человек с компьютером играет против человека с компьютером. Такие соревнования были проведены, и оказалось, что побеждает не та пара, где сильнейший супергроссмейстер или суперкомпьютер, а та, где лучше всего взаимодействие человека с компьютером! Очень показательно, что этот спортивный результат был предсказан математически задолго до этого. В 1965 г. в Нью-Йорке на конгрессе ИФИП Г.И. Марчук сделал совместный доклад с А.П. Ершовым [2] о важности взаимодействия человека с ЭВМ при решении сложных задач математической физики.

Обращаясь к XXI в., необходимо отметить принципиальный переход количества в новое качество, происшедший в эволюции ЭВМ после воцарения пост-петафлопсной эры с суперкомпьютерами, имеющими 10^5 – 10^6 и более вычислительных ядер с пропорционально растущими огромными объемами памяти. В результате математическое моделирование стало главным орудием получения наукоемких знаний о сложнейших процессах макро- и микромира, связывая между собой теоретические и экспериментальные исследования, остающиеся главным критерием истины. Что касается ближайших перспектив, то нам сейчас надо начинать осваивать супервычисления на появившихся реально эксафлопсниках, хотя в литературе уже обсуждаются вызовы, обусловленные предстоящим появлением зеттафлопсных ЭВМ [3].

Одновременно происходит и лавинообразный рост разработок по искусственному интеллекту, включая распознавание образов, анализаторы текстов, переводчики и самые разнообразные приложения с машинным обучением, нейросетями, цифровыми двойниками, роботами и т.д. Весь

этот инновационный поток еще требует систематизации, методологического и даже философского осмысления. И никаким образом он не должен противопоставляться классическим наукам, а наоборот, ассимилироваться и служить катализатором прогресса во всевозможных технологиях развития, в первую очередь – в математическом моделировании, ставшим непременным атрибутом производственных отраслей.

Проблемы национального суверенитета, требующие опережающего научно-технологического прогресса в приоритетных направлениях государственного развития, в уже наступившую эпоху эксафлопсных суперкомпьютеров неизбежно должны опираться на массовую цифровизацию всех сфер жизнедеятельности человеческого общества с активным использованием современных достижений искусственного интеллекта, работы с огромными объемами данных, нейросетями и методологиями машинного обучения. Особое место в этих тенденциях принадлежит средствам получения новых фундаментальных и прикладных знаний, базирующимся на наукоемком математическом моделировании сложнейших процессов и явлений. Эти исследования, как правило, включают решение междисциплинарных прямых и обратных задач и обуславливают вовлечение экстремальных объемов вычислительных ресурсов с масштабируемым распараллеливанием алгоритмов на многопроцессорных вычислительных системах (МВС). Принципиальным моментом здесь является то, что в условиях непрерывно развивающихся математических и отраслевых наук проблемы моделирования неизбежно связаны с динамичными разработками, сопровождением и внедрениями больших размеров математического и программного обеспечения (МПО), высокая стоимость которого требует экономического отношения к профессиональной эксплуатации. С другой стороны, пользователи МПО, воспринимающие его в конкретных предметных областях, должны владеть суперкомпьютерными технологиями получения “активных” знаний, обуславливающих принятия решений на основе их анализа.

Фактически такие подходы связаны с построением цифровых моделей, или виртуальных реальностей, изучение и оптимизация характеристик которых позволяет качественно повышать эффективность труда с дополнительным выходом внутреннего валового продукта (ВВП). Не менее значимы проникновения интеллектуальных вычислительно-информационных новаций в социальные и гуманитарные сферы, не говоря уже о государственной безопасности. Образно говоря, компьютерные науки и инструментариум играют роль кровеносной или нервной системы, обеспечивающей жизнедеятельность человеческого сообщества как единого организ-

ма. И такая ситуация требует взглянуть по-современному на взаимосвязи высокопроизводительных вычислений с искусственным интеллектом и преобразованиями больших данных, составляющих “три кита” 4-й индустриальной революции.

Во всех этих трех направлениях за последние десятилетия достигнуты выдающиеся успехи, требующие глубокого анализа в плане гармоничного развития совокупности наук о получении знаний (см. [4–10]). Например, появившиеся возможности получения положительных практических результатов на основе когнитивного анализа больших данных с помощью, возможно, их простой статистической обработки иногда приводят к соображениям о появлении некоей “постнауки”, вроде Data Science, не требующей традиционных глубоких исследований. Другой иллюстрацией может служить следующее заманчивое рассуждение: “Если у меня появилась большая проблема, которую я сам не хочу решать, то можно взять готовую нейросеть, настроить ее должным образом, и она выдаст необходимый результат”.

Чтобы не ввести в заблуждение возможных доверчивых читателей такими призывами, мы в данной главе проведем системный анализ рассматриваемой проблематики. В разделе 4.1 описываются основные современные характеристики и тенденции самого математического моделирования как научной дисциплины и как технологии решения практических задач. Раздел 4.2 посвящен анализу функционального и системного наполнения программного обеспечения, реализующего стоящие перед ним технические требования. В разделе 4.3 излагаются методологические принципы основанного на вычислительном эксперименте машинного обучения, составляющего современную инструментальную базу для принятия решений и оптимального управления сложными системами в интеллектуальных новациях.

4.1. СУПЕРКОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ КАК ИНТЕЛЛЕКТУАЛЬНАЯ ОБРАБОТКА БОЛЬШИХ ДАННЫХ

Может показаться удивительным, но исторически такие на первый взгляд разные научные направления, как математическое моделирование и искусственный интеллект, развивались параллельно, бок о бок. В качестве примера можно привести первые этапы развития Вычислительного центра СО АН СССР, официально созданного в январе 1964 г. Первая защищенная в его стенах кандидатская диссертация (В.Л. Катков, науч-

ный руководитель Г.И. Марчук, 1966 г.) была посвящена системе КИНО (координаты инфетизимального оператора), реализующей идеи сподвижника М.А. Лаврентьева – Л.В. Овсянникова – по теории групп, которая только спустя много лет обрела актуальную прикладную значимость. После этого Владислав Катков возглавил лабораторию, которой лично Гурий Иванович придумал оригинальное название – лаборатория автоматизации построения алгоритмов (ЛАПА). Этот коллектив входил в состав руководимого А.П. Ершовым отдела программирования, создавшего язык, транслятор и развитую систему АЛЬФА и ставшего альма-матер получившей мировую известность Сибирской школы информатики, феномен которой в США называли “силиконовая тайга”. Молодые энтузиасты не только делали пионерские разработки, но и закладывали философско-методологические основы математического моделирования и технологий прикладного программирования для решения актуальных больших задач.

В 1973 г. была сформирована группа А.С. Нариньяни, в 1977 г. реорганизованная в лабораторию искусственного интеллекта, в которой велись исследования по общению с компьютером на языке естественного типа, математической лингвистике и вычислениям на неопределенных моделях [11]. Также была создана система программирования УНИКАЛЬК, в математическом плане реализующая оригинальный способ решения обратных задач идентификации параметров модели. Такой подход перекликается с попытками формализации интеллектуальной деятельности, активно развиваемой в течение многих лет Г.С. Альтшуллером и его последователями в системе ТРИЗ (теория решения изобретательных задач [12]). Удивительно, но факт: лаборатория ИИ успешно функционирует и в наше время под руководством Ю.А. Загорюлько в составе Института систем информатики (ИСИ СО РАН им. А.П. Ершова, выделившийся из Вычислительного центра СО АН в 1985 г.).

Философское изречение “бытие определяет сознание” имеет прямое отношение к нашей теме в том плане, что достижимый уровень искусственного интеллекта определяется мощностью ресурсов ЭВМ, включая быстродействие и объем памяти. Мы будем рассматривать только традиционные современные архитектуры кластерного типа, включающие вычислительные узлы с распределенной памятью, многоядерными процессорами (CPU с несколькими десятками ядер над общей памятью и, возможно, графическими ускорителями GPGPU). Хотя в нашей стране и за рубежом активно ведутся разработки новых поколений компьютеров, в том числе квантовых и реконфигурируемых, в ближайшие 5–10 лет они, по видимому, еще не станут массовыми и конкурентоспособными на мировом

рынке. Таковыми на указанный нами период времени можно рассматривать конфигурации с суммарным быстродействием около петафлопса и оперативной памятью в несколько десятков или сотен терабайт. В этом случае “большими задачами” (в терминологии Н.Н. Яненко [13, 14], решаемыми достаточно долго: часы или десятки часов, сутки или несколько суток и т.д.) приемлемо считать математические проблемы, постановки которых включают междисциплинарные многомерные прямые и обратные задачи с числом неизвестных функций около десятка, требующие для своего численного решения применение неструктурированных адаптивных сеток с числом узлов порядка 1000^3 .

Согласно сложившейся методологии математического моделирования (см. [13–18]), ее технологическая цепочка, при всем многообразии, включает относительно небольшое количество стадий, качественное содержание которых, однако, существенно меняется с развитием суперкомпьютерных поколений, вычислительных алгоритмов и инструментов. Мы будем рассматривать некую универсальную ЭВМ, или сеть машин, с программным обеспечением, интегрированным в том смысле, что оно позволяет решать самый широкий и непрерывно пополняемый круг задач, а также гибко расширять применяемые методы и технологии, автоматически адаптируясь к эволюции компьютерных платформ и к интеллектуальным требованиям растущего количественно и качественно контингента пользователей.

Изучение любого объекта начинается с формирования его модели, которая может представляться совокупностью дифференциальных и/или интегральных, а также дискретных уравнений и соотношений, с дополнительными ограничениями и условиями на оптимизацию, с большими объемами реально измеряемых данных, которые могут быть приближенными и даже противоречивыми, никак не подпадающими под строгие математические понятия существования, единственности и корректности решения. Такие сложные системы приходится анализировать гидрометеорологическим службам, например, при прогнозе погоды, изменении климата или для предупреждения природных катастроф, при использовании всемирной сети наземных и космических наблюдений. Другая иллюстрация – многофазные процессы в нефтегазовой отрасли с применением химических, электрофизических и других современных технологий добычи. Одна из экстремальных проблем – это последствия термоядерного взрыва, где перекрещиваются процессы и гидрогазодинамики, и физики твердого тела, и плазмы, и т.д. Главный тезис здесь заключается в постоянном росте математической сложности решаемых актуальных постановок.

Естественно, что стратегии и тактики моделирования должны обеспечивать максимум эффекта при минимизации вычислительных ресурсов, отнюдь не дешевых. Все науки быстро развиваются, а с ними растут, количественно и качественно, всевозможные модели, создавая свои иерархии, от простых и экономичных до более сложных и точных, с которыми надо высокопрофессионально работать. Речь здесь идет о взаимодействии человека с ЭВМ, требующего комфортные интерфейсы со входными языками, определяющие уровень интеллектуальности общения с пользователями самых различных специальностей. Предназначение данной стадии, которую назовем MODEL, заключается в задании для решения задач на ЭВМ той модели (или их последовательностей из возможного набора), которая требуется из контекста. Очевидно, что такая функциональность как соответствующей программной подсистемы, так и ее пользователя может быть достигнута только в результате машинного обучения путем накопления опыта эксплуатации и анализа получаемых результатов. Отметим, что “умные” реализации данной стадии могут включать качественный анализ математических свойств моделей, что, несомненно, повышает ценность такого продукта.

После конкретизации математической постановки на непрерывном уровне требуется ее дискретизация, т.е. построение сетки. Данный этап весьма важен для эффективности всего моделирования и представляет собой трудоемкую алгоритмическую проблему в многомерных задачах с реальными данными, включающими сложные геометрические конфигурации расчетных областей с кусочно-гладкими неодносвязными разномасштабными границами (в том числе движущимися) и контрастными материальными свойствами сред. В этих случаях зачастую приходится учитывать сингулярности решений и строить адаптивные неструктурированные сетки, для оптимизации которых вопросы пока остаются открытыми. Особую сложность представляют динамические сетки с перестраивающейся структурой в нестационарных задачах.

Необходимо еще отметить, что наиболее эффективные численные методы решения больших задач связаны с декомпозицией областей, являющейся основным орудием распараллеливания алгоритмов, а также применением многосеточных подходов, обеспечивающих асимптотическую оптимальность по порядку решений (при характерных шагах сетки $h \rightarrow 0$ общее число арифметических действий пропорционально числу неизвестных (см. [19])). Это ставит перед “сеточными генераторами” новые интересные задачи, связанные с операциями на графах. Данную стадию и соответствующую программную подсистему назовем DELAUNAY [20].

Для построения сеточных аппроксимаций различного порядка точности существует достаточно много подходов: методы конечных разностей, конечных объемов, конечных элементов, разрывных алгоритмов Галёркина и т.д. Создание многофункционального программного обеспечения с этой целью для различных типов сеток и видов операторов – актуальная и перспективно востребованная проблема (проект соответствующей подсистемы CHEBYSHEV описан в [21]). Отметим, что хотя проблема автоматизации построения алгоритмов для сеточных аппроксимаций достаточно старая [22], широкого распространения этот подход пока не получил.

Большинство практических задач является нелинейными и нестационарными, но после применения к ним квазилинеаризации и неявных по времени аппроксимаций неизбежно приходится решать системы линейных алгебраических уравнений (СЛАУ), как правило, с большими разреженными матрицами, симметричными и несимметричными. Основной подход здесь – это итерационные предобусловленные методы в пространствах Крылова (см. [23], соответствующая подсистема называется библиотека KRYLOV). Важно отметить, что с ростом практических требований к точности расчетов относительная роль вычислительной алгебры значительно преувеличивает в силу нелинейного закона увеличения ресурсоемкости.

Ключевую роль в машинном обучении играют оптимизационные методы решения обратных задач, поскольку именно на их основе отыскиваются и исследуются наилучшие сценарии процессов и явлений [24]. Типичная методология заключается в планировании серии вычислительных экспериментов, в которых на основе предыдущего опыта описывается минимизируемый целевой функционал, а после анализа полученных результатов формируется следующий сеанс машинного обучения. Такое взаимодействие человека и ЭВМ представляется безальтернативным во многих многокритериальных практических поисковых проблемах, когда детерминированный алгоритм их решения формализовать в принципе невозможно. Отметим такой важный момент, что варьируемые при этом параметры могут относиться не только к изучаемой модели, но и к самому вычислительному процессу, поскольку в патовых ситуациях он может или совсем не сходиться, или длиться неприемлемо долго. Таким образом, в получаемый итерационный цикл вовлекаются все рассмотренные стадии моделирования. К ним еще следует добавить методы и технологии постобработки, визуализации и анализа расчетных результатов, на основе которых функционируют средства принятия решений, на основе создаваемых онтологий предметной области и когнитивных подходов [25].

В качестве иллюстрации успешного применения такой методологии можно привести разработанную в ЛАПА систему автоматизации проектирования (ППП “Эфир”, ставшей основой кандидатской диссертации В.А. Катешова в 1984 г.) электронно-оптического преобразователя (ЭОП для формирования изображений в приборах ночного видения). Здесь требовалось минимизировать сложный многопараметрический целевой функционал с различными технологическими ограничениями, характеризующими качество прибора по строгому техническому заданию. При этом были реализованы сверхточные алгоритмы интегральных уравнений теории потенциала. Каждый расчетный сеанс с локальной оптимизацией занимал несколько часов на машине М-220 с производительностью 20 000 флопс (!), после чего вмешивался естественный интеллект математика и на основе оперативного анализа текущих результатов формировался следующий сеанс. Таким образом за несколько месяцев проблема была решена, по смоделированным результатам создали ЭОП, “боевые” характеристики которого совпали с требуемыми, и важное правительственное задание (проект носил романтическое название “Цветоножка”) было выполнено! А результаты научных исследований были опубликованы в монографии [26], которая по просьбе зарубежных коллег была переведена на английский язык. Очень важно понимать, что в этом рекордном для своего времени вычислительном эксперименте сыграла огромную роль незаметная на первый взгляд, но наиболее трудозатратная разработка большого объема наукоемкого программного комплекса.

4.2. МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КАК ЭКОСИСТЕМА

Прикладное программное обеспечение, как и системное, уже более полувека бурно развивается одновременно с вычислительной техникой, для которой всем на удивление до сих пор почти выполняется закон Мура (увеличение быстродействия в 1000 раз за 11 лет). Хотя надо признать, что за последние десятилетия темпы роста производительности программиста существенно отстают от компьютерных, и в определенном смысле можно говорить о кризисе программирования. Для исправления такого перекоса уже начинает активно привлекаться, прежде всего, искусственный интеллект.

К настоящему времени в мире накоплено огромное количество общедоступного (Open Source) и коммерческого “софта” в виде библиотек, специальных инструментов и проблемно-ориентированных пакетов прикладных

программ (ППП, см. [15]), представляющих собой высокую интеллектуальную ценность. Здесь можно назвать такие высокопрофессиональные разработки, как библиотеки PETSc, HYPRE, PARDISO, пакеты программ TRILINOS, FENIX, DEAL II, специализированные системы PARVIEW, MAPLE и многие другие. Отдельный мировой рынок составляют системы автоматизации проектирования (САПР, см. [27]), в котором за последние годы наметилась конвергенция с классическими ППП. Однако за последние годы в мире наметился переход от разношерстного “зоопарка” к интегрированным программным окружениям. Примерами таких больших проектов являются DUNE, OPEN FOAM, INMOST, а также базовая система моделирования (BCM, см. [17, 28–30]), являющиеся в первую очередь методо-ориентированными.

Функциональное наполнение BCM представляется набором автономных подсистем, каждая из которых отвечает за соответствующую технологическую стадию моделирования и связана с другими посредством согласованных структур данных. В результате работы подсистемы MODEL формируются функциональная и геометрическая структуры данных (ФСД и ГСД), являющиеся исходными для этапа генерации сетки. На основе получаемой сеточной структуры данных (ССД) в результате выполнения этапа аппроксимации строятся алгебраические информационные массивы в общепринятых мировых форматах (АСД), обеспечивающих высокопроизводительное решение широкого класса систем линейных алгебраических уравнений.

Такая архитектура позволяет осуществлять реализацию и развитие различных подсистем независимыми группами профессиональных разработчиков. При этом легко достигается гибкая расширяемость состава модулей и алгоритмов для каждого вычислительного этапа, в том числе с эффективным переиспользованием внешних программных продуктов и с адаптацией на новые компьютерные платформы. Итоговое интегрированное вычислительное окружение (ИВО) представляет собой самоподдерживающуюся экосистему с длительным жизненным циклом, ориентированную на успешную эксплуатацию широким кругом пользователей. Предусмотренный переизбыточный набор моделей и алгоритмов призван поддерживать методологии машинного обучения как самой экосистемы, так и ее пользователей.

Обеспечение богатой функциональности и эффективного использования ИВО требует создания разнообразного системного наполнения. С математической точки зрения это означает в первую очередь средства автоматизации построения алгоритмов, включая инструменты построения

многоверсионных конфигураций вычислительных модулей. Что касается языков программирования, то мы здесь не будем углубляться в эту специальную область, отметив только популярность сочетания стилей объектно-ориентированных подходов C++ и богатых изобразительных возможностей языка интерпретирующего типа Python, включающего и такие важные интеллектуальные компоненты, как компьютеризацию сложных аналитических выкладок. Это открывает путь к активному использованию аппроксимационных методов повышенной точности, которые в теоретическом плане имеют значительные перспективы в силу большого сокращения требуемой памяти и энергозатратных коммуникационных расходов, но имеют пока недостаточно широкое распространение из-за трудоемкого программирования с неизбежными человеческими факторами. В мире также ведутся разработки специализированных естественных языков для математиков-вычислителей (существует даже лозунг “программирование без программирования”). Хотя в публикациях в Интернете имеются такие заманчивые проекты по “фабрикам языков”, как SIDL (Scientific Interface Definition Language) и DSL (Domain Specific Languages), см. книгу [31], где насущная проблема трактуется как переход от “палеоинформатики к неоинформатике”, здесь ближайших прорывных перспектив не наблюдается, и технологии формирования алгоритмических библиотек остаются традиционными уже несколько десятилетий. В целом же языковой контент является одним из ключевых для уровня машинного обучения, а текущую ситуацию можно оценить как “мягкий выход из кризиса программирования”.

Несомненно, одним из главных качеств прикладной программы является ее производительность, определяемая в первую очередь качеством распараллеливания алгоритмов. В данном случае огромные возможности нам предоставляют ПЛИС (программируемые логические интегральные схемы), которые в принципе позволяют для заданного алгоритма спроектировать и собрать специализированный компьютер максимального быстродействия (здесь в принципе уже достигим феномен: робот воссоздают другие роботы). Однако у них имеется один главный недостаток – коммерческая неконкурентность в сравнении с типовыми суперкомпьютерами кластерного типа. Имеющиеся у них вычислительные узлы соединены шинами и обмениваются между собой данными с помощью очень простой библиотеки MPI, а расчеты на каждом из них осуществляются многоядерными процессорами с иерархической общей памятью (ее различные уровни имеют разные объемы и скорости обменов) под управлением программной системы типа OpenMP. Еще имеются быстрые графические

ускорители (GPGPU), но они имеют медленную связь с общей памятью, что существенно снижает эффективность их использования. Ситуацию с распараллеливанием вычислений можно считать парадоксальной ввиду отсутствия соответствующей массовой, или штатной, системы автоматизации программирования, за исключением нескольких малораспространенных языков или подсистем (например, SHAPEL и DVM – Distributed Virtual Machine [32]). Здесь уместно упомянуть драматическую историю с крупнейшим международным проектом HPF (High Performance Fortran), который в XX в. вовлек огромные средства и человеческие ресурсы на построение системы автоматизации распараллеливания языка Фортран, но был закрыт по причине идейного тупика у авторов.

Фактически сейчас распараллеливание алгоритмов на МВС – это глубоко “ручная” работа с экспериментально подбираемыми критериями роста производительности вычислений, которые измеряются двумя простыми параметрами: $S_p = T_1/T_p$, $E_p = S_p/p$ – коэффициентами ускорения и эффективности, где T_p есть время решения задачи на p процессорах. В качестве перспективных можно назвать исследования [33] по оптимизации распараллеливания на основе концепции D-детерминанта. Отметим, что эти перспективные работы были начаты еще в прошлом веке в Вычислительном центре СО АН СССР, где по этой теме в 1986 г. В.Н. Алеевой была защищена кандидатская диссертация, причем реальное распараллеливание алгоритмов осуществлялось на отечественной 16-процессорной машине ПС-2000, сделанной в Донецке. Исследования по распараллеливанию алгоритмов в ВЦ СО АН начались еще раньше, и обзорный доклад Г.И. Марчука, В.П. Ильина [34] по этим вопросам был сделан на Конгрессе ИФИП в Токио в 1980 г. Другим примером работ по автоматизации распараллеливания вычислительных процессов является разработанная В.А. Перепёлкиным под руководством В.Э. Малышкина в ИВМиМГ СО РАН система LUNA [35], интегрирующая средства “штатных” инструментариев MPI и OpenMP с общей и распределенной памятью.

С точки зрения машинного обучения можно сформулировать проблему о поиске самим суперкомпьютером наилучшего способа распараллеливания на основе проведения и анализа серии расчетов для конкретного класса задач. В разные годы делались попытки имитационного моделирования оптимальной архитектуры многопроцессорного суперкомпьютера, но оказалось, что для таких работ нужен еще более мощный суперкомпьютер. В целом же следует сказать, что распараллеливание – это достаточно интеллектуальная проблема, которая на примере задач вычислительной ал-

гебры исследуется в [36]. И в то же время оно обеспечивает высокую производительность, без которой не может успешно функционировать ИИ.

Очень важное значение для повышения производительности труда программиста имеет скорость прохождения таких неизбежных рутинных процедур, как отладка, тестирование, верификация и валидация реализуемого кода. Возникающие здесь проблемы неизбежно усугубляются при создании больших программных комплексов, которые как раз и подразумеваются в концепции ИВО, а связано это в первую очередь с драматическим усложнением информационных связей, а также внутренних междоульных и пользовательских интерфейсов. В больших профессиональных командах разработчиков операционных систем и компиляторов такие вопросы по необходимости давно решены с помощью известных компонентных технологий COM/DCOM и CCA (Common Component Architecture, см. [37]), но в современном прикладном программировании они пока находятся в ожидании перехода на индустриальное мышление.

Одним из главных компонентов ИВО является репозиторий – хранилище всего проекта, обеспечивающий целостность разработки и ее связи с внешним миром, поддерживающий свойства многоязыковости и кроссплатформенности, а также взаимодействия с разработчиками и пользователями. Широкое распространение в компьютерном сообществе получила система такого типа GITHUB с большим разнообразием сервисов.

В целом требования развития экосистемы МПО ставят ряд принципиальных проблем, без решения которых не смогут быть решены задачи устойчивого технологического прогресса [38]. Главная из них связана с масштабом глобализации процессов моделирования: от макро- до микромира, от производств до природоведения, от социальных отношений до гуманитарных, от здравоохранения до образования и т.д. Все это обуславливает неограниченный рост количества и качества новых моделей и методов их исследования, составляющих результаты бурного развития всех наук, реализуемых в интегрированном математическом и программном обеспечении ЭВМ, возможности которых с ожидаемым приходом квантовых и фотонных суперкомпьютеров мы даже представить себе не можем. Но готовиться к новой эпохе необходимо и технологически, и психологически, и философски, поскольку именно эти аспекты развития цивилизации следует считать наименее продвинутыми.

Создание МПО ставит нас перед мировой архитектурной дилеммой: воздвигнуть “Вавилонскую башню” или множество “софтвер-хаузов”, – решение которой включает множество экономических и политических вопросов.

В качестве наглядной иллюстрации можно представить ситуацию с отечественными нефтегазовыми корпорациями, которые из конкурентных соображений строятся по принципу “интегрированных нефтяных компаний”: “все свое” – от геологоразведки до бензоколонки. И этот предмет гордости включает, в частности, доморощенное МПО для прогнозирования месторождений и технологий добычи, составляющих большой фирменный секрет. Здесь альтернативой служит транснациональная компания Шлюмберже, разработавшая высокопрофессиональный программный продукт Eclipse, являющийся практически законодателем мод на мировом рынке нефтегазового МПО.

В первые десятилетия XXI в. появились новые веяния и даже “манифест” в создании современных архитектур и инфраструктуры для сверхбольших программных проектов. В зарубежной литературе появились такие термины, как Agile, Scrum, Extreme Programming (XP) и другие, связанные с “легкими” стилями управления взаимосвязанными коллективами. На самом деле возраст этих проблем – более 50 лет, здесь можно рекомендовать книги [39, 40].

В связи с неизбежным драматическим ростом новых массовых профессий по информационным технологиям (ИТ) все возрастающая роль принадлежит разработке учебных программ и курсов для самого широкого контингента участников: школьников, студентов и аспирантов, разработчиков и пользователей – специалистов в разных профессиях, сопричастных к внедрению технологий машинного обучения. От качества дидактических и других педагогических подходов в таком МПО в значительной степени зависит успех его применения.

Интенсивность развития функционального и системного наполнения экосистемы ИВО требует качественного повышения производительности труда при создании наукоемкого МПО для различных стадий математического моделирования, при участии различных групп разработчиков, в том числе в удаленном режиме. В значительной степени здесь успех зависит от эффективности применения современных стилей модульного, или сборочного, программирования на основе создания наборов типов объектов и структур данных. По аналогии с уже существующими “3D-принтерами” можно предвидеть создание Software-Printers, призванных кардинально ускорить разработку готовых программных продуктов. Здесь естественно включаются законы экономики производств, где виртуальные ИТ-объекты имеют огромные преимущества при массовом тиражировании, в сравнении с материальными изделиями, в силу практически бесплатном создании электронных копий.

Характеризуя в заключение комплекс рассматриваемых проблем создания МПО нового поколения, важно отметить, что они составляют новую отрасль человеческой деятельности, которая не сможет прогрессировать без придания ей необходимой инфраструктуры. В советской России существовал Государственный фонд алгоритмов и программ, который осуществлял большую систематизирующую и организационную деятельность, находившуюся в сфере компетенций ГКНТ (Государственный комитет по науке и технике СССР, руководимый в 1980–1986 гг. академиком Г.И. Марчуком). Создание аналогичной структуры типа госкорпорации ФАП представляется уже давно назревшим шагом к управлению прогрессом развития и инноваций российских информационных технологий.

4.3. О НЕКОТОРЫХ ПРИНЦИПАХ МАШИННОГО ОБУЧЕНИЯ И ПРИНЯТИЯ РЕШЕНИЙ

Всякое обучение заключается в приобретении знаний и навыков для какой-то сферы деятельности, в том числе в принятии решений на основе анализа содержательной информации. В отношении математического моделирования мы можем выделить три категории участников: неодушевленный объект – компьютер, состоящий из технического оборудования и программного обеспечения (hardware и software), сопровождаемый эксплуатационными службами и ответственными за внедрение дистрибьюторами, а также два типа субъектов – разработчики инструментов моделирования и конечные пользователи – специалисты в конкретных предметных областях, осуществляющие суперкомпьютерные инновации. Понятно, что любой человеческий персонал может состоять из людей разной квалификации, и между ними возникают отношения учитель-ученик.

В соответствии с терминологией [41], знания являются активными, если они приводят к каким-то действиям и результатам. Их накопление и систематизация для какой-то предметной области заключается в разработке соответствующей базы активных знаний (БАЗ), содержащей всевозможную информацию о своих объектах, их спецификациях, взаимосвязях и о возможных действиях над ними. С расширением и углублением добываемой информации появление БАЗ, как результат развития баз данных, следует считать неизбежным, поскольку объемы данных становятся непосильными для усвоения Homo sapiens (прошли времена энциклопедистов вроде Л.Д. Ландау). Структура БАЗ еще не является сложившейся, а пример проекта такой разработки для задач вычислительной алгебры приведен в [42]. Некоторым прототипом такой разработки можно считать

систему ALGOWIKI [43], созданную под руководством Дж. Дангарры и В.В. Воеводина. Многообразные вопросы искусственного интеллекта в математическом моделировании исследовались в [9].

В принципе, базы знаний должны содержать всю необходимую информацию по соответствующей теме. Если мы говорим о математическом моделировании для определенного класса проблем, то сюда входят описания моделей, вычислительных методов и технологий, примеры задач и результаты их решения (для чего должны быть специальные архивы), рекомендации по всевозможным применениям, а также литературные источники и доступные программные материалы с документацией. Другими словами, должна быть разработана своя онтология, позволяющая проводить разбор текстов, статистический анализ данных и другие возможные интеллектуальные действия. Типичной для базы знаний следует считать систему сбора и усвоения огромных объемов информации, например, космической, метеорологической, климатической, службы предупреждения землетрясений, цунами или других катастроф и т.д., которые необходимо интегрировать с оперативными расчетами.

Понятно, что получение новых знаний не является самоцелью, а служит средством для управления какой-то деятельностью. Показательно изучение модели объекта или процесса со свободными параметрами, для которых необходимо найти значения, обеспечивающие какие-то характерные свойства. Это приводит к решению обратных задач для систем дифференциальных и/или интегральных уравнений. Примером может быть проектирование небоскреба, не разрушающегося при каком-то предельном уровне землетрясения, или же прогнозирование наличия полезных ископаемых в исследуемом районе по данным георазведки.

Зачастую обратные задачи являются некорректными или условно корректными, когда небольшие возмущения исходных данных приводят к сильному изменению решений. Во избежание таких неустойчивостей развиты методы регуляризации, теория которых заложена А.Н. Тихоновым в ИПМ АН СССР им. М.В. Келдыша, а затем продолжена в научных школах академиков В.К. Иванова (Екатеринбург) и М.М. Лаврентьева (Новосибирск). Другой классический подход – оптимальное управление – основывается на принципе максимума Понтрягина. Универсальный инструмент к анализу сложных систем представляет собой аппарат сопряженных уравнений, развитый Г.И. Марчуком и его учениками (см. [4, 44]). Интересно заметить, что здесь приоритет открытия принадлежит физикам-ядерщикам. Они обнаружили, что решение задачи, формально сопряженное к кинетическому уравнению переноса Больцмана, имеет реальный фи-

зический смысл. Оно определяет ценность нейтрона, т.е. его способность внести вклад в развитие цепной реакции, и используется при расчете критической массы атомной бомбы.

Наиболее содержательными и универсальными постановками для машинного обучения являются проблемы, связанные с оптимизационными методами решения обратных задач, позволяющие достичь наибольшего практического эффекта, например, при идентификации параметров модели по результатам натурных измерений (иллюстрация – нефтегазовое месторождение), или оптимизации эксплуатационных режимов технической установки (самолет, корабль, заводской цех и т.д.). При этом постановка проблемы формулируется как поиск минимума некоторого целевого функционала

$$\Phi_0(\vec{u}(\vec{x}, t, \vec{p}_{opt})) = \min_{\vec{p}} \Phi_0(\vec{u}(\vec{x}, t, \vec{p})),$$

определенного на решении какой-то прямой начально-краевой задачи и подчиняющегося дополнительным линейным и/или нелинейным ограничениями:

$$p_k^{\min} \leq p_k \leq p_k^{\max}, \quad k = 1, \dots, m_1,$$

$$\Phi_l(\vec{u}(\vec{x}, t, \vec{p}))q \leq \delta_l, \quad l = 1, \dots, m_2,$$

$$\vec{p} = \{p_k\} \in \mathcal{R}^m, \quad m = m_1 + m_2.$$

Здесь $\vec{p}$ – m -мерный вектор оптимизируемых параметров, $t, \vec{x}$ – суть временная и пространственная координаты. Исходную прямую задачу, или уравнение состояния оптимизируемой сложной системы, формально можно представить в следующем виде:

$$L\vec{u} = \vec{f}(\vec{x}, t), \quad \vec{x} \in \bar{\Omega} = \Omega \bigcup \Gamma, \quad 0 < t \leq T < \infty,$$

$$l\vec{u} = \vec{g}(\vec{x}, t), \quad \vec{x} \in \Gamma, \quad \vec{u}(\vec{x}, 0) = \vec{u}^0(\vec{x}),$$

$$\bar{\Omega} = \bigcup \bar{\Omega}_j, \quad \Gamma = \Gamma^e \bigcup \Gamma^i,$$

$$\Gamma^i = \bigcup \Gamma_{j,k}^i = \bigcup (\bar{\Omega}_j \cap \bar{\Omega}_k),$$

где L есть в общем случае матричного типа оператор (в дифференциальной и/или интегральной форме), l – оператор граничных условий, $\bar{\Omega} \times [0, T]$ – расчетная область, которая зачастую состоит из подобластей $\bar{\Omega}_k$ с различными контрастными материальными свойствами, а также с внутренними и внешними границами Γ^e, Γ^i , в том числе имеющими разномасштабные детали и кусочно-гладкие многосвязные поверхностные сегменты. В общем случае речь идет о междисциплинарных неклассических

формулировках, когда даже вопросы существования, единственности и корректности остаются открытыми. Что касается проблемы условной минимизации функционала, то она требует многократного применения безусловной минимизации и сама при этом может быть локального типа или глобального. В последнем случае требуется найти все имеющиеся минимумы, что является наиболее ресурсоемкой задачей.

Постановки и алгоритмы решения оптимизационных задач отличаются большим разнообразием и представляют собой развитую область вычислительной математики, поддерживаемую высокопроизводительным программным обеспечением. Свои глубокие результаты имеются по задачам и линейного, и квадратичного, и нелинейного программирования. За последние десятилетия активно развиваются способы их эффективного решения: методы внутренних точек, последовательного квадратичного программирования, доверительных интервалов и т.д.

Можно привести следующего вида характерный пример машинного обучения. Пусть нам требуется в течение длительного времени (год или более) проводить оперативное оптимальное управление некоторой сложной системой, зависящей от 10 параметров, путем решения обратных и прямых задач, причем каждая из последних для своей реализации требует длительного времени расчета (часы или десятки часов). Отметим, что если в каждом параметре перебирать 10 его возможных значений, то общее количество вариантов составляет 10^{10} ("проклятие размерности")! В этом случае машинное "обучение компьютера" можно провести следующим образом. Сначала в течение нескольких, например, недель осуществляются решения сотен задач с помощью классических методов оптимизации, результаты которых запоминаются и как-то статистически обрабатываются (например, с помощью популярного вида обобщенной линейной регрессии – кригинга), формируя соответствующие аппроксимации в сеточном пространстве параметров. Затем на основе накопленных данных начинается оперативная работа машинно-программного комплекса, который быстро находит приближение к искомому состоянию системы, а затем с участием человека последовательно производится необходимое уточнение параметров.

В рассматриваемых подходах вошел в обиход термин "суррогатная оптимизация". Подразумевается, что поиск наилучшего решения производится не для реального объекта или процесса, а для его модели, возможно, даже слишком грубой. В целом решение сложных обратных задач с многоовражным поведением минимизируемых функционалов требует применения иерархии моделей, выбор которых требует высокого искусства. Фак-

тически в таких случаях работают мета-алгоритмы, включающие взаимодействие компьютера с пользователем, имеющим большой опыт в решении конкретного класса задач.

Развитие машинного обучения в вычислительной математике в первую очередь связано с выбором оптимального или “хорошего” алгоритма для каждой стадии технологической цепочки моделирования: генерация сетки, аппроксимация и дискретизация исходной задачи, решение получаемых систем линейных и нелинейных уравнений и т.д. В результате поиск наилучшего общего вычислительного процесса сводится к разноуровневым повторяющимся циклам с проведением многократных численных экспериментов, анализа промежуточных результатов и выработки итоговой стратегии практических расчетов. Зачастую здесь необходимо идти на компромисс типа “лучшее – враг хорошего”, поскольку сама оптимизация может оказаться дороже, чем применение уже известного достаточно эффективного подхода. При этом сами постановки вопроса о формулировке показателя “цена/качество” ждут своих специфических исследований. Здесь в качестве примера можно указать на работу [46], где с помощью “нейронных операторов” проводится оптимизация итерационных алгоритмов для решения СЛАУ.

Что касается появляющегося в последние годы огромного объема литературы по нейросетям и нейронным операторам (NN и NO, см. работу [47] по их применению к решению уравнений в частных производных, а также обзоры в [48, 49]), то к нему надо подходить с внимательной осторожностью, хотя Нобелевский комитет по физике присудил премию в 2024 г. за исследования по машинному обучению(!). Основой новых объектов NN и NO являются достаточно грубые параметризированные модели, которые быстро “обучаются”, т.е. позволяют найти необходимые приближения для решений с возможностью их дальнейшего уточнения. При этом фактически надо или использовать старые принципы аппроксимации, или искать новые, а что касается приставки “нейро”, то это, по-видимому, просто дань моде.

Аналогичная методология обучения применима и при использовании расчетов для других предметных наук (геофизика, химия, биология и т.д.) или производств: модель, компьютерный эксперимент, анализ результатов, новые знания. Однако это не является самоцелью, и далее должны следовать какие-то решения по оптимизации человеческой деятельности и инновациям – повышение производительности труда, количества и качества выпускаемого труда, достижение социального и/или гуманитарного эффекта и т.д. Фактически эти тенденции ведут к кардинальному изме-

нению и к появлению новых массовых профессий. Несомненно, машинное обучение должно значительно изменить педагогические подходы и саму систему образования – от дошкольного до высшего, и эти актуальные вопросы требуют своих исследований.

Машинное обучение является неотъемлемым атрибутом технологий получения новых знаний. Дальнейшая проблема искусственного интеллекта заключается в принятии человеком решений на основе анализа получаемых данных. Оптимизация этого вида деятельности – актуальнейшая, но отнюдь не новая проблема. В качестве иллюстрации можно привести уже упоминавшиеся нами работы Г.С. Альтшуллера (см. [12]) по созданию ТРИЗ – теории решения изобретательских задач. Современные подходы в этих направлениях базируются на построении онтологий рассматриваемых предметных областей, которые позволяют создавать когнитивные инструментари для средств принятия решений, принципы которых описаны, например, в [51, 52]. В целом же это направление захватывает очень разнообразные области знаний – и математическую логику, и семантические инструментари, и создание интеллектуальных интерфейсов и системных окружений и т.д. [53–56]. Надо сказать, что в данную область сейчас вкладывают большие средства серьезные частные компании типа OpenAI, что несомненно должно принести новые прорывные решения.

Универсальный характер машинного обучения и принятия на его основе необходимых решений по оптимальному управлению сложными системами обуславливает процесс его глобализации и экспансии, что в совокупности с роботизацией новых технологий неизбежно побуждает к философскому осмыслению процессов “цифровой” цивилизации общества, и не случайно появляются публикации о моральных аспектах искусственного интеллекта [57]. Очевидно, что эти инновации непосредственно влияют и на производственные вопросы государственного масштаба, и на проблемы национальной безопасности, и на всевозможные аспекты устойчивого развития цивилизации.

СПИСОК ЛИТЕРАТУРЫ

1. Ильин В.П. Вычислительная информатика – открытие науки. Новосибирск: Наука. Сиб. отд-ние, 1991. 197 с.
2. Ershov A.P., Marchuk G.I. Man-machine interaction in solving a certain class of differential equations // Proceed. IFIP Congress. N. Y., 1965. P. 550–551.
3. Liao X., Lu K., Yang C. et al. Moving from exascale to zettascale computing: challenges and techniques // Front. Inform. Technol. Electron. Eng. 2018. V. 19, N 1. P. 1236–1244.A.

4. *LeCun Y., Bengio Y., Hinton G.* Deep learning // *Nature*. 2015. V. 521. P. 436–444.
5. *Weinan E.* Machine learning and computational mathematics // *Commun. Comput. Phys.* 2020. V. 28. P. 1639–1670.
6. *Dongarra L., Grigori L., Higham, N.J.* Numerical algorithms for high performance computational science // *Philos. Transact. Ser.* 2020. V. 378. Iss. 2166.
7. *Xu Y., Zeng T.* Sparse deep neural network for nonlinear partial differential equations // *Numer. Math.: Theor., Meth. Appl.* 2023. Vol. 16. Iss. 1. P. 58–78.
8. *Ильин В.П.* Математическое моделирование: философия науки // *Математика, механика и информатика: сб. науч.-поп. статей*. 2017. С. 8–16.
9. *И'ин V.P.* Artificial intelligence problems in mathematical modeling // *Voevodin V., Sobolev S. (eds.). RuSCDays 2019. CCIS*. 2019. V. 1129. P. 505–516. Springer, Cham.
10. *Forrester A., Sobester A., Keane A.* Engineering design via surrogate modeling: A practical guide. N. Y.: Wiley, 2008.
11. *Загорюлько Ю.А., Загорюлько Г.Б.* Недоопределенные модели Нариньяни: становление, применение, проблемы и перспективы // *Развитие вычислительной техники в России, странах бывшего СССР и СЭВ: история и перспективы: труды 5-й Междунар. конф.* М., 2020. С. 126–132.
12. *Альтшуллер Г.С.* Найти идею. Введение в теорию решения изобретательских задач. Новосибирск: Наука, 1986. 209 с.
13. *Яненко Н.Н., Коновалов А.Н.* Некоторые вопросы теории модульного анализа и параллельного программирования для задач математической физики и механики сплошной среды // *Современные проблемы математической физики и вычислительной математики*. М.: Наука, 1982. С. 200–217.
14. *Яненко Н.Н., Рычков А.Д.* Актуальные проблемы прикладной математики и математического моделирования. Новосибирск: Наука, 1982.
15. *Ершов А.П., Ильин В.П.* Пакеты программ – технология решения прикладных задач. Новосибирск: ВЦ СО АН СССР, 1978. Препринт № 121.
16. *Самарский А.А., Михайлов А.П.* Математическое моделирование. М.: Физматгиз, 2002.
17. *Ильин В.П.* Как реорганизовать вычислительные науки и технологии? // *Вестн. РАН*. 2019. Т. 89, № 3. С. 232–242.
18. *Ильин В.П.* Математическое моделирование. Новосибирск: Изд-во СО РАН, 2017. Ч. 1. Непрерывные и дискретные модели.
19. *И'ин V.P.* Iterative preconditioned methods in Krylov spaces: Trends of the XXI century // *Comput. Math. and Math. Phys.* 2021. V. 61, N 11. P. 1750–1775. DOI: 10.1134/S0965542521110099.
20. *И'ин V.P.* Integrated computational environment for grid generation parallel technologies // *Springer Nature Switzerland AG 2020 / eds. L. Sokolinsky, M. Zymbler*. 2020. CCIS. V. 1263. P. 58–68.
21. *Бутюгин Д.С., Ильин В.П.* CHEBYSHEV: принципы автоматизации построения алгоритмов в интегрированной среде для сеточных аппроксимаций на-

чально-краевых задач // Труды Международной конференции ПАВТ'2014. Челябинск: Изд-во ЮУрГУ, 2014. С. 42–50.

22. *Валиуллин А.Н., Ганжа В.Г., Ильин В.П., Шапеев В. П., Яценко Н.Н.* Задача автоматического построения и исследования на ЭВМ разностных схем в аналитическом виде // Докл. АН СССР. 1984. Т. 275, № 3. С. 528–532.

23. *Бутюгин Д.С., Гурьева Я.Л., Ильин В.П., Перевозкин Д.В., Петухов А.В., Скопин И.Н.* Функциональность и технологии алгебраических решателей в библиотеке Krylov // Вестн. ЮУрГУ. Серия: Вычисл. математика и информатика. 2013. Т. 2, № 3. С. 92–105.

24. *Il'in V.* The integrated computational environment for optimization of complex systems // Proceedings of the 15th Int. Asian School-Seminar “Optimization Problems of Complex Systems” (OPCS 2019). P. 65–67. URL: <https://doi.org/10.1109/opcs.2019.888015>

25. *Il'in V.P., Skopin I.N.* About performance and intellectuality of supercomputer modeling // Progr. and Comp. Software. 2016. V. 42. Iss. 1. P. 5–16.

26. *Ильин В.П., Катешов В.А., Куликов Ю.В., Монастырский М.А.* Численные методы оптимизации эмиссионных электронно-оптических систем. Новосибирск: Наука, 1987. 191 с.

27. *Ушаков Д.М.* Введение в математические основы САПР. Новосибирск: Ледас, 2008.

28. *Bastian P., Blatt M., Dedner A. et al.* The dune framework: Basic concepts and recent developments // Comput. and Math. with Appl. 2020. URL: doi.org/10.1016/j.camwa.2020.06.007/

29. OpenFOAM. <https://www.openfoam.com/>.

30. INMOST: A toolkit for distributed mathematical modeling. URL: <https://www.inmost.org>

31. *Kleppe A.* Software language engineering: Creating domain-specific language using metamodels. N.Y.: Addison-Wesley, 2008.

32. DVM Systems. URL: <http://www.keldush.ru/dvm>

33. *Aleeva V.* Designing parallel programs on the base of the conception of determinant // Supercomputing. RuSCDays 2018 (Communications in Computer and Information Science (CCIS)) / V. Voevodin, S. Sobolev (eds.). Springer, Cham. 2019. V. 965. P. 565–577.

34. *Marchuk G.I., Ilin V.P.* Parallel computations in grid methods for solving mathematical physics problems // Proc. of IFIP Congr. Ser. 1980. V. 8. P. 671–676.

35. *Akhmed-Zaki D., Lebedev D., Malyshkin V.K., Perepelkin V.A.* Automated construction of high performance distributed programs in LuNA system // Parallel Computing Technologies. PaCT 2019 (Lecture Notes in Computer Science. V. 11657) / V. Malyshkin (ed.). Springer, Cham. P. 3–9. URL: https://doi.org/10.1007/978-3-030-25636-4_1

36. *Il'in V.P.* Parallel intelligent computing in algebraic problems / L. Sokolinsky, M. Zymbler (eds.) Parallel Computational Technologies. PCT 2021. Commun. in Comp. and Inform. Sci. 2021. V. 1437. Springer, Cham.

37. *Allan B., Armstrong R., Wolfe A. et al.* The CCA Core specification in a distributed Memory // SPMD Framework Concurrent Practice and Expedience. 2002. V. 14. P. 323–345.

38. *Ильин В.П.* Программирование ближайшего будущего: концепция и прагматика // Вестн. РАН. 2023. Т. 93, № 2. С. 150–161.

39. *Брукс Ф.* Мифический человек-месяц, или Как создаются программные системы. М.: Символ-Плюс, 2010.

40. *Скопин И.Н.* Основы менеджмента программных проектов: курс лекций, учебное пособие. М.: ИНТУИТ. РУ, 2004.

41. *Malyshkin V.E.* Active knowledge, LuNA and literacy for oncoming centuries // LNCS. 2015. V. 9465. P. 292–303.

42. *Ильин В.П.* Концепция и архитектура базы знаний систем линейных алгебраических уравнений // Материалы IX конф. Знания-Онтологии-Теории (ЗОНТ-2023), 2–6 октября 2023 г. Новосибирск, 2023. С. 143–15.

43. *Antonov A., Dongarra J., Voevodin V.* AlgoWiki project as an extension of the Top500 methodology // Supercomput. Frontiers and Innova. 2018. V. 5, N 1. P. 4–10.

44. *Марчук Г.И.* Сопряженные уравнения и анализ сложных систем. М.: Наука, 1992. 336 с.

45. *Марчук Г.И., Агошков В.И., Шутяев В.П.* Сопряженные уравнения и методы возмущений в нелинейных задачах математической физики. М.: Наука, 1993. 224 с.

46. *Rudikov A., Fanaskov V., Muravleva E., Laevsky Y.M.; Oseledets I.* Neural operators meet conjugate gradients: The FCG-NO method for efficient PDE solving // Proc. of the 41st Int. Conf. on Machine Learning, in Proc. of Machine Learning Res. 2024. V. 235 P. 42766–42782.

47. *Fanaskov V.S., Oseledets I. V.* Spectral Neural Operators / Dokl. Math. 2023. V. 108, suppl. 2. P. S226–S232.

48. *Бурнаев Е.В., Бернштейн А.В., Оселедец И.В. и др.* Фундаментальные исследования и разработки в области прикладного искусственного интеллекта // Докл. РАН. Математика, информатика, процессы управления. 2022. Т. 508. С. 19–27.

49. *Турдаков Д.Ю., Аветисян А.И., Оселедец И.В. и др.* Доверенный искусственный интеллект: вызовы и перспективные решения // Докл. РАН. Математика, информатика, процессы управления. 2022. Т. 508. С. 13–18.

50. *Марчук А.Г.* Модель ТРИЗ-системы, предназначенная для создания ассистирующей программы // Системная информатика. 2013. N 2. 1–10.

51. *Загорулько Ю.А., Загорулько Г.Б.* Онтологический подход к разработке системы поддержки принятия решений на нефтегазодобывающем предприятии // Вестн. НГУ. Сер.: Информационные технологии. 2012. Т. 10, вып. 1. С. 121–128.

52. *Zagorulko Y., Zagorulko G.* Architecture of extensible tools for development of intelligent decision support systems // New Trends in Software Methodologies, Tools and Techniques: Proc. of the 10th SoMeT_11. Hamido Fujita (eds.). Amsterdam: IOS Press, 2011. P. 457–466.