Параллельные процессы на этапах петафлопного моделирования

В.П. Ильин

Институт вычислительной математики и математической геофизики СО РАН,
Новосибирский государственный университет

1. Введение

Целью данной работы является рассмотрение современных вопросов масштабируемого распараллеливания алгоритмов при решении больших наукоемких задач математического моделирования. Эта тема тесно связана с проблемой высокопроизводительных вычислений на многопроцессорных вычислительных системах (МВС), и зачастую такая связь формулируется односторонне как отображение алгоритмов на архитектуру МВС. Однако неизбежно появление и встречного движения, которое можно сформулировать как конвергенцию алгоритмических структур и вычислительных архитектур, обсуждавшуюся в [1] и цитируемых там работах.

Появившиеся в конце "нулевых" годов петафлопные вычислители (которые были точно предсказаны еще за двадцать лет) и грядущий приход "экзафлопников" к концу первого десятилетия 21-го века переводят рассматриваемую проблематику на качественно новый уровень. В соответствии с прогнозами "дорожной карты" в [2], мировое сообщество ожидают в недалеком будущем компьютеры с десятками миллионов процессоров и миллиардами вычислительных ядер. С одной стороны, этот факт необозримо усиливает возможности вычислительных экспериментов для сложнейших явлений и процессов за реальное время, что делает математическое моделирование не просто третьим, а основным путем познания (наряду с теоретическими и натурными исследованиями). Но в то же время кардинально меняющиеся технические условия требуют обновления и модели вычислений, и технологий программирования, и концепции взаимодействия с динамически развивающимися суперкомпьютерами (хотя зачастую новое — это хорошо забытое старое).

Согласно общепринятым представлениям, архитектура МВС в ближайшее десятилетие не претерпит революционных изменений и будет эволюционировать в направлении гетерогенных многопроцессорно-многоядерных систем, использующих в вычислительных узлах ускорители типа графических процессорных элементов (GPU, которые уже успешно применяются и для арифметических задач) или/и программируемых логических интегральных схем (реконфигурируемых ПЛИСС, или FPGA, см. [3]).

Что касается конкретной аппаратной инфраструктуры, то здесь на крайних позициях находятся многоядерные "персональные суперкомпьютеры" и голиафы – "датацентры" (datacenter, или ЦОД – центр обработки данных) с петафлопной и более высокой производительностью. Промежуточное положение занимают мини- или миди-датацентры с быстродействием около десятков терафлопс, решающие частные корпоративные задачи.

Техническое оборудование (hardware, или "железо", сложнейший комплекс которого получил название IaaS – Infrastructure as a Service), естественно, неотделимо от системного программного обеспечения, включающего и операционные системы с компиляторами, и средства управления заданиями с диспетчеризацией и распараллеливанием, которые в совокупности составляют компьютерную платформу (PaaS – Platform as a Service) для пользователей – разработчиков прикладного программного обеспечения (приложений) в различных предметных областях. Относительно новой концепцией организации датацентров является формирование интеллектуального вычислительного сервиса (SaaS – Software as a Service) для конечных пользователей, заинтересованных только в результатах по решению конкретных задач из энергетики, биологии, геофизики и т.д. В целом описываемое программное окружение ЦОДа составляет информационные технологии (ИТ или IT) высокого уровня, предоставляющие пользователям универсальный доступ через Интернет и получившего название облачные вычисления (Cloud Computing, см. [4], [5]). В определенном смысле, предлагаемые здесь решения — это осовременные подходы к проблемам вычислительных центров коллективного пользования, обсуждавшиеся еще лет двадцать назад, например, в [6].

К настоящему времени характерные типы программных наукоемких средств, предоставляемых прикладным математикам, программистам и инженерам — это многочисленные пакеты программ (ППП) или библиотеки программ (популярные примеры – ANSYS и NETLIB, по которым, как и по многим другим, имеются обширные материалы в Интернет), существующие в виде свободного доступа или коммерческих продуктов. При этом пользователь или фактически разрабатывает свое приложение, решая сопутствующие трудоемкие вопросы распараллеливания и интерфейсов, или проводит расчеты в жестких рамках существующих ППП, где, возможно, он вставляет свои программные фрагменты, согласно предоставляемым ему правилам (UDF – User Defined Functions).

В настоящей работе развивается проблематика распараллеливания алгоритмов в рамках базовой системы моделирования (БСМ), ранее рассмотренная в [7], [8], которая может явиться конкретизацией технологий облачных вычислений. В п. 2 описывается на содержательном уровне достаточно общая постановка задач математического моделирования, включая междисциплинарные и обратные, а также основные технологические этапы крупномасштабного вычислительного эксперимента. В соответствии с таким представлением в п. 3 обсуждаются принципы построения главных компонент, архитектура, вопросы распараллеливания и пользовательских интерфейсов БСМ в рамках функционирования датацентров. Подчеркнем, что базовая система моделирования рассматривается как совокупность достаточно автономных вычислительных проблемно-независимых инструментариев, согласованных по смежным структурам данных и ориентированных на гибкую сборку широкого класса приложений по принципу детского конструктора.

2. Постановки задач и этапы наукоемкого моделирования

Среди всевозможных применений компьютера мы останавливаемся только на задачах математического моделирования, классические постановки которых описываются дифференциальными и/или интегральными уравнениями или соответствующими вариационными формулировками для обобщенных решений. Традиционно такая область называется математической физикой, хотя в последние десятилетия сюда добавляются вычислительная химия, вычислительная биология и т.д.

Междисциплинарность проблемы означает необходимость одновременного расчета процессов различной природы. Например, проектирование корабля требует моделировать его гидродинамические качества, прочность корпуса, тепловые режимы, электротехническое оборудование и т.п. С математической точки зрения это означает решение не одного, а системы уравнений, как правило, нелинейной и нестационарной, в которой начально-краевые условия и коэффициенты отражают материальные свойства сред и внешние воздействия. Реальные конфигурации расчетных областей с криволинейными кусочно-гладкими границами и острыми углами привносят свои сложности при формировании и обосновании математической постановки. Важно также разделить понятия прямых и обратных задач. К первым относятся те, у которых надо найти решение при полностью заданных начально-краевых условиях. К обратным мы относим оптимизационные постановки с параметризованными данными, которые надо определить по условию минимума некоторого целевого функционала. Такие задачи идентификации параметров модели требуют многократного решения прямых задач и применения алгоритмов нелинейного программирования.

Геометрическое функциональное моделирование. Задание пользователем исходных данных задачи и указаний по методам ее решения — это первая стадия вычислительного эксперимента, которую назовем этапом геометрического и функционального моделирования. Его содержанием является описание и возможное редактирование геометрических объектов расчетной области (вершины, ребра, грани, фигуры), вместе с указанием топологических связей между ними, а также функциональных объектов, характеризующих типы решаемых уравнений, краевые и начальные условия, а также представляемых в них коэффициентах (с привязкой к соответствующим подобластям и граничным поверхностным сегментам). Для обратных задач дополнительно указываются параметризованные исходные данные, совместно с описанием ограничений на возможные вариации параметров и формулировкой целевого функционала. Для обеспечения дружественного интерфейса пользователю должны на данном этапе предоставляться интеллектуальные графические и текстовые средства формирования математической модели и вычислительного задания. А результатом работы этой стадии является геометрическая и функциональная структуры данных (ГСД и ФСД), однозначно определяющие исходную информацию, необходимую для выполнения последующих этапов моделирования. В целях придания гибкости БСМ предусматривается возможность множественность представления форматов, с их взаимной конвертизацией. В частности, это необходимо для поддержки взаимодействия с распространенными внешними приложениями, например, графическими или САПРовскими продуктами, базирующимися на общепринятых геометрических форматах, см. подробнее [9], [10].

Дискретизация математической модели. Построение сетки представляет особо сложную проблему в многомерных случаях с кусочно-гладкими криволинейными и, возможно, движущимися границами (для последних специальный класс составляют т.н. свободные границы, положение которых заранее неизвестно). Сама сетка определяется совокупностью своих объектов различной размерности: узлы, ребра, грани, конечные объемы,– а также топологическими связями между ними. Между геометрической структурой расчетной области с подобластями и сеточной структурой существует большая аналогия, и они содержат по сути однотипные наборы объектов макро- и микро-уровня.

Существует большое количество критериев качества сетки, которые в значительной степени определяют точность и экономичность численного решения. Одно из главных требований сетке – адаптивность, означающая в том или ином смысле учет особенностей конфигурации границы и свойств искомого решения, которые определяются или теоретически априори или апостериори экспериментально, т.е. на основе предварительных расчетов.

В первую очередь требуется, чтобы вершины, ребра и граничные поверхности расчетной области составлялись из соответствующих сеточных объектов, или в крайнем случае аппроксимировались ими с возможно малой погрешностью. Второй аспект связан с возникающей сингулярностью, т.е. сильным ростом производных, в окрестности углов и ребер границы, что требует специального сгущения узлов в таких подобластях. А при наличии сильно меняющихся пространственно-временных особенностях решения сетки должны быть динамическими, т.е. регулярно или периодически перестраиваемыми.

Распространенные сеточные технологии включают триангуляции Делоне, ячейки Дирихле-Вороного и различные приемы контроля вырожденных случаев. Важным вычислительным средством являются многосеточные методы, основанные на построении последовательности вложенных сеток или на их локальном сгущении.

Наиболее просто конструируемыми и одновременно обеспечивающими больший порядок точности являются равномерные сетки, которые могут иметь различные типы конечных элементов: кубы или параллелепипеды, призмы, тетраэдры и т.д. Однако в реальных задачах сетки приходится строить неравномерные и нерегулярные, или неструктурированные, у которых для каждого узла номера его ближайших соседей можно задать только перечислением. Существуют компромиссные квазиструктурированные сетки, когда расчетная сеточная область состоит из сеточных подобластей, в каждой из которых сетка строится по своим правилам и может быть структурированной. Такие сетки могут быть несогласованными или согласованными - в последних случаях узлы на смежных границах раздела соседних сеточных подобластей являются совпадающими.

Методы построения сеток отличаются большим разнообразием и имеют обширную профессиональную литературу. Здесь используются и квазиконформные отображения, и вариационные принципы, и специальные метрические пространства, и многочисленные эмпирические приемы. Соответствующее программное обеспечение, или генераторы сеток, существует в широком ассортименте, или в свободном доступе через Интернет, или в качестве коммерческих продуктов. Зачастую процедуры построения сеток являются неотъемлемой частью проблемно-ориентированных пакетов прикладных программ (ППП), но они также существуют и в автономной форме, позволяющей их встраивать в различные приложения.

Согласно идеологии БСМ, стадия дискретизации задачи реализуется независимо от остальных, используя на входе только один из форматов ГСД, ГФСД и формируя на выходе сеточную структуру данных (ССД), также допускающую множественные форматы данных с конверторами. Подчеркнем, что методология квазиструктурированных сеток допускает использование внешних существующих сеточных генераторов, осуществляя таким образом актуальные возможности интеграции различных приложений и переиспользования программного кода.

Алгоритмы аппроксимации. Конечной целью дискретизации является переход от исходных функциональных соотношений к конечно-мерным уравнениям, неравенствам или рекурсиям. При этом фактически задача алгебраизируется, что достигается путем аппроксимации функций, производных и интегралов. Основные подходы к построению сеточных соотношений – это методы конечных разностей, конечных объемов, конечных элементов (МКР, МКО, МКЭ, см. [11], [12]), коллокаций и спектральные методы, связанные с разложениями в ряды Фурье. Не пытаясь делать обзора имеющегося огромного материала по данным вопросам, мы коротко остановимся главным образом на технологических аспектах наиболее универсальных МКО и МКЭ, позволяющих конструировать сеточные аппроксимации высоких порядков точности на различных типах конечных объемов для широкого класса задач математического моделирования.

Важным моментом этих двух подходов является поэлементная технология вычисления локальных матриц и векторов правых частей с последующей сборкой (ассемблированием) глобальных матриц и систем алгебраических уравнений (линейных или нелинейных – СЛАУ или СНАУ). Этот прием значительно упрощает программную реализацию и естественным образом обеспечивает эффективное распараллеливание данного вычислительного этапа, поскольку такие процедуры аппроксимации носят локальный характер, реализация которых использует свойства только топологически ближайших сеточных объектов, но никак не использует их дальние связи..

В нестационарных проблемах пространственная аппроксимация осуществляется на каждом временном шаге, а при наличии нелинейностей такая процедура повторяется итерационно для всех шагов. Наиболее трудоемкими оказываются задачи с динамическими сетками, если их приходится перестраивать на каждой итерации для каждого шага по времени.

Совокупность аппроксимационных алгоритмов для типовых задач математической физики может быть систематизирована и классифицирована по следующим признакам:

- по типу аппроксимируемого члена дифференциального уравнения (градиент, дивергенция и т.д.) или его механического смысла; примером могут быть матрицы жесткости и матрицы масс в МКЭ;

- по виду конечных элементов или объемов: тетраэдры, параллелепипеды, призмы и т.д.; при этом могут выделяться частные случаи, которые наиболее экономично реализуются (например, правильные фигуры);

- по характеру базисных функций, которые могут отличаться своими носителями, порядками и структурными свойствами (лагранжевые и эрмитовые, скалярные и векторные, и т.п.);

- по конфигурации сеточного шаблона, т.е. совокупности узлов, участвующих в одном уравнении, получаемом в итоге формирования алгебраических систем; наиболее экономичными оказываются компактные схемы, в которых участвуют только наиболее близкие геометрически соседние узлы; как правило, повышение порядка аппроксимации ведет к растягиванию сеточного шаблона и расширению ленточной структуры итоговых матриц, так что с точки зрения общей эффективности алгоритмов приходится искать золотую середину.

Результатом выполнения этапа аппроксимации является алгебраическая структура данных (АСД), аккумулирующая на дискретном уровне всю исходную информацию о решаемой проблеме, и для которой имеются общепринятые в мире сжатые матричные форматы, вместе с соответствующими переходниками – конверторами.

Задачи вычислительной линейной алгебры. Если решаемая проблема в целом является нестационарной и нелинейной, все равно после этапов временной аппроксимации и квазилинеаризации мы приходим к задачам линейной алгебры, из которых наиболее типичны – это решение СЛАУ или проблемы собственных значений.

Характерные матрицы, возникающие в сеточных методах решения дифференциальных задач, являются разреженными, ленточными и большими. Это означает, во-первых, что порядки N достигают десятков и сотен миллионов, а ненулевые элементы сосредоточены в некоторой полосе ширины m около главной диагонали, причем величина m и число ненулевых элементов в каждой строке не зависят от N. Дискретизированные алгебраические системы, возникающие из аппроксимации интегральных уравнений (определенных на границе или в объеме расчетной области) являются, наоборот, плотными, но зачастую имеют специальные структурные свойства (например, являются теплицевыми, квази- или блочнотеплицевыми).

Вычислительная линейная алгебра – хорошо продвинутая математическая дисциплина и содержит большое количество алгоритмов для решения задач с самыми разными типами матриц: вещественными и комплексными, квадратными и прямоугольными, эрмитовыми и неэрмитовыми, положительно определенными и знаконеопределенными. Имеется также соответствующее обширное программное обеспечение в виде библиотек или прикладных пакетов, как свободно распространяемых в Интернете, так и коммерческих.

Алгебраические задачи – “узкое горлышко” математического моделирования, поскольку потребляемые вычислительные ресурсы нелинейно растут с увеличением порядка N. Поэтому здесь особенно актуально распараллеливание алгоритмов на МВС с общей, разделенной и гибридной памятью.

Существенным моментом программного обеспечения для разреженных матриц является оптимизация кода, поскольку применение сжатых форматов данных позволяет кардинально сокращать объем хранимой информации, но существенно усложняет реализацию доступа к матричным элементам, что особенно критично при многоуровневой неоднородности кэша и оперативной памяти.

Методы оптимизации и решения нелинейных уравнений. Алгоритмы оптимизации решения обратных задач, сводящихся к проблеме условной минимизации функционала, являются бурно развивающейся в последние десятилетия областью вычислительной математики. Здесь развиты модификации методов множителей Лагранжа и внутренних точек, являющихся развитием подходов со штрафными функциями, использование доверительных интервалов для регулировки последовательности шагов, а также варианты алгоритмов Ньютона и последовательного квадратичного программирования для решения возникающих на промежуточных этапах СНАУ.

В данной тематике имеется много актуальных и далеко не решенных вопросов, связанных с поиском глобального минимума и оптимального управления, применения методов

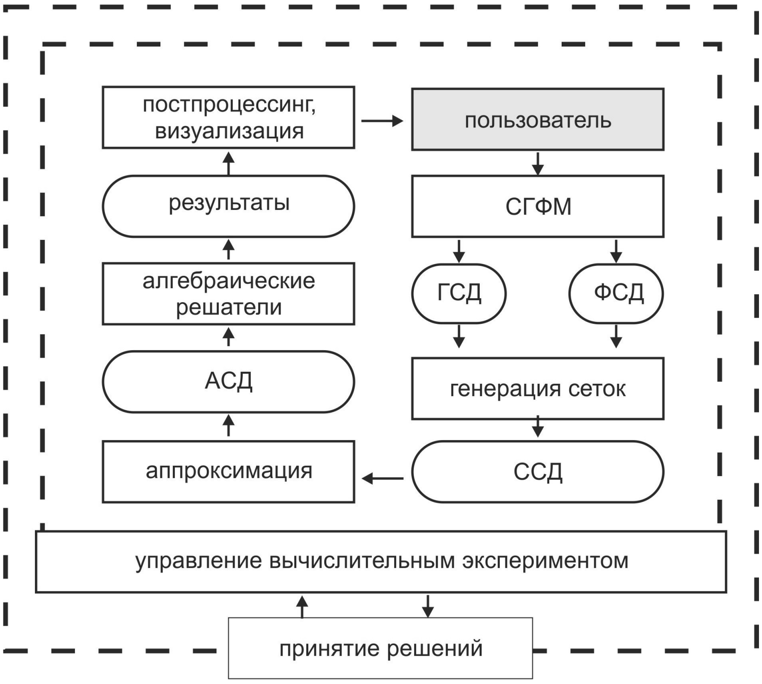

Рис. 1. Структура функциональных и информационных компонент БСМ

теории возмущений и сопряженных уравнений, нахождения градиентов функционалов или функций чувствительности к вариациям исходных данных. Зачастую постановки требуют штучного исследования устойчивости и корректности задачи для поиска соответствующего регуляризационного подхода. К этой же области можно отнести изучение нелинейных динамических систем, связанных с эффектами бифуркаций, самоорганизации и хаоса, странных аттракторов и т.д., где зачастую еще остаются открытыми методологические принципы математического моделирования.

Постобработка и визуализация результатов. Непосредственная реализация наукоемких алгоритмов в многомерных задачах может составлять отнюдь не главную долю общего вычислительного процесса, поскольку анализ получаемых результатов требует их презентабельной визуализации с предварительной обработкой сеточных функций, что требует выполнения ресурсоемких технологических операций по формированию сечений, изолиний и изоповерхностей, различных графиков с возможной анимацией многоцветных изображений. Такие технологические проблемы особенно актуальны в системах принятия решений по результатам математического моделирования.

3. Структура и принципы распараллеливания БСМ

Рассмотренные технологические этапы решения больших задач естественным образом отображаются на архитектуру и состав основных функциональных и информационных компонент БСМ, изображенных на рис. 1. Приведенная схема фактически представляет последовательность действий, осуществляемых при проведении полноформатного вычислительного эксперимента для изучения какого-то технического процесса или природного явления.

Для конечного пользователя (“модельера”) операционное окружение представляется, с одной стороны, входным интерфейсом, т.е. системой геометрического и функционального моделирования (СГФМ), включающей средства управления вычислительным процессом и принятия решений (последние аспекты представляют самостоятельную тему, выходящую за рамки данной работы). Выходной же интерфейс обеспечивается обработкой и визуализацией результатов расчета (постпроцессинг), на основе анализа которых могут меняться исходные данные, математическая модель или алгоритмы и формироваться новые вычислительные сеансы. Получаемые в итоге выполнения этого этапа информационные массивы ГСД и ФСД содержат, как правило, до нескольких сот объектов, в силу чего при его параллельной реализации на МВС формируемые данные целесообразно копировать в памяти всех процессоров, во избежание излишних коммуникационных потерь.

Библиотека сеточных генераторов (условное название – DELAUNAY), говоря формальным языком, является преобразователем данных ГСД+ФСД→ССД и, как уже отмечалось, служит интегратором программных процедур от различных разработчиков, на основе гибкой системы внутренних интерфейсов. Актуальную роль играет библиотека конечно-объемных и конечно-элементных "аппроксиматоров", которая на основе ССД, ГСД и ФСД формирует алгебраическую структуру данных.

На основе АСД функционирует библиотека алгебраических решателей KRYLOV, концепция и общая структура которой описана в [13].

Решение сложной математической проблемы организуется в общем случае по многократно вложенным циклам: численное интегрирование нестационарных задач по временным шагам, проведение на каждом шаге итераций по нелинейности, если таковая присутствует, расщепление по физическим процессам или/и по пространственным переменным, варьирование исходных данных в многовариантных расчетах или в обратных задачах. На всех этих уровнях могут строиться свои тактики и стратегии распараллеливания, но в любом случае одним из наиболее критичных моментов является параллельная реализация СЛАУ. Здесь главный подход заключается в алгебраической декомпозиции, которая на дискретном представлении отражает геометрическое разделение сточной расчетной области со сбалансированным разделением подобластей по соответствующим процессорам. Данная проблема представляет собой специальную задачу на графах с высокой вычислительной сложностью, и здравый компромисс заключается в поиске ее приближенного решения.

Следует сказать, что структура АСД с обычными сжатыми матричными форматами ориентирована на экономию памяти при решении больших разреженных СЛАУ и слабо "помнит" топологические свойства исходной расчетной области, которые могут иметь значительное влияние на эффективность метода декомпозиции. Поэтому разделение областей целесообразно начинать еще на этапе построения сетки, что должно отражаться в формировании и сеточных, и алгебраических структур данных.

При решении стационарной задачи, т.е. СЛАУ – на дискретном уровне,- алгоритм декомпозиции формально представляет собой крупно-блочный итерационный метод, скорость сходимости которого тем выше, чем меньше минимальное расстояние (в топологическом смысле) между подобластями. Поэтому при возможном разделении трехмерной расчетной области в одномерные ($1D$), двумерные ($2D$) или трехмерные ($3D$) информационные структуры предпочтение, естественно, следует отдавать последним. Однако эффективность $3D$ – декомпозиции при ее реализации на конкретной МВС сильно зависит от реальных физических связей между процессорами (идеальный случай, когда они составляют трехмерную вычислительную сеть).

Оптимизация массивного распараллеливания в конкретных случаях требует анализа коэффициентов ускорения и использования вычислительной системы

$$S_p = T_1/T_p, \quad E_p = S_p/p,$$

где T_p – время решения задачи (или реализации алгоритма) на p процессорах. К сожалению, сделать это непросто в силу отсутствия адекватных моделей машинных вычислений на МВС с иерархической памятью. При оценке эффективности производительности многопользовательского вычислительного центра ситуация значительно усложняется и требует учета распределения глобальных ресурсов между потоками задач.

Поскольку главное снижение производительности многопроцессорных компьютеров происходит из-за коммуникационных потерь, с точки зрения эффективности решения рассматриваемых задач перспективными являются следующие направления развития архитектур МВС:

- вычислительные сети (ВС) различной размерности (одно-, дву- и трехмерные);
- динамическая реконфигурация ВС с реализацией как структурированных, так и неструктурированных сеток;
- быстрые ближние связи и некоторые специальные коммуникации (типа гиперкуба);
- синхронизация обменов и вычислений, совмещение во времени двусторонних обменов;
- специализированные вычислительные устройства и гетерогенные МВС (например, аппаратная поддержка постобработки и визуализации, алгоритмов линейной алгебры).

В силу принципиальных появившихся технических возможностей, благодаря прогрессу "кремниевых технологий", становится реальным переход от лозунга "отображение алгоритмов на архитектуру МВС" к двустороннему движению (конвергенции) между данными категориями.

Список литературы

1. Ильин В.П. Об экзапроблемах математического моделирования. CAD/CAM/CAE Observer, N 2(54), 2010, 85-92.

2. Dongarra J., Beckman P., et. al. - IESP: International Exascale Software Project. Road Map, 18 Nov., 2009, www.exascale.org.

3. Каляев И.А., Левин И.И. Семейство реконфигурируемых вычислительных систем с высокой производительностью.//Вычислительные методы и программирование, т. 10, N 1, 2009, 207-214.

4. Armbrust M. et al. About the Clouds: A Berkeley View of Cloud Computing.–Technical Report No. UCB/EECS – 2009-28 http://www.eecs.berkeley.edu/Pubs.

5. Колесов А. IT-область. Сгущается облачность. Суперкомпьютеры. N 3, 2010, 8-13.

6. Алексеев А.С., Гололобов В.И., Ильин В.П., Карначук В.И. Комплексный центр математического моделирования: концепция программного обеспечения.–Новосибирск, ВЦ СО АН СССР, препринт N 821, 1988.

7. Ильин В.П. Параллельные алгоритмы для больших прикладных задач: проблемы и технологии.//Автометрия, N 2, 2007, 3-21.

8. Ильин В.П. Экзапроблемы математического моделирования.//Вестник ЮУрГУ, сер. "Математическое моделирование и программирование", вып. 6, N 35(211), 2010, 28-39.

9. Ильин В.П. Геометрическое и функциональное моделирование в задачах математической физики.–Новосибирск, Вычислительные технологии, т. 6, ч. 2, 2001, 315-321.

10. Ушаков Д.М. Введение в математические основы САПР.–Новосибирск, изд. комп. Ледас, 2006.

11. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений.–Новосибирск, изд. ИВМиМГ СО РАН, 2001.

12. Ильин В.П. Методы и технологии конечных элементов.–Новосибирск, изд. ИВМиМГ СО РАН, 2007, 370с.

13. Бутюгин Д.С., Ильин В.П., Ицкович Е.А., Петухов А.В., Кныш Д.В. Krylov: библиотека высокопроизводительных алгоритмов для решения разреженных СЛАУ.–Труды 13-й Всероссийской конференции “Современные проблемы математического моделирования”. – Ростов, изд. ЮФУ, 2009, 110-128.