CHEBYSHEV: концепция вычислительной интегрированной среды для сеточных аппроксимаций начально-краевых задач *

Д.С. Бутюгин [1], В.П. Ильин [1,2]

Институт вычислительной математики и математической геофизики СО РАН[1], Новосибирский государственный университет[2]

Рассматривается методология автоматизации построения алгоритмов для аппроксимации широкого класса сложных многомерных задач для уравнений математической физики на неструктурированных сетках различных типов с построением алгебраических структур данных на основе сеточной, геометрической и функциональной структур. Предлагаемая система CHEBYSHEV включает программные средскоетва аналитических преобразований для реализации методов конечных объемов и конечных элементов высоких порядков, разрывные алгоритмы Галеркина и спектральные подходы для аппроксимации основных видов дифференциальных и интегральных операторов на типовых конфигурациях сеточных ячеек и базисных функциях с разными степенями свободы, и поддерживает автоматическое сгущение адаптивных сеток, а также масштабированное распараллеливание в методологии декомпозиции областей.

1. Введение

В “дорожной карте” Международного экзамасштабного программного проекта IEPS [1] формулируется проблема о необходимости создания качественно нового программного обеспечения, ориентированного на постпетафлопсные суперкомпьютеры с многомиллионным количеством вычислительных ядер и/или процессоров. Для наукоемких задач математического моделирования это означает переход от “закрытых” пакетов прикладных программ (ППП), типа широко распространенных продуктов ANSYS [2] и NASTRAN [3], к открытым интегрированным вычислительным окружениям, примерами которых могут служить системы OpenFOAM [4] и DUNE [5].

В работах [6], [7] была представлена концепция базовой системы моделирования (БСМ), ориентированной на решение широкого класса прямых и обратных междисциплинарных задач, описываемых системами дифференциально-интегральных уравнений или соответствующими вариационными постановками. БСМ предусматривает поддержку всех основных технологических этапов крупномасштабного вычислительного эксперимента, включая геометрическое и функциональное моделирование, построение адаптивных квазиструктурированных сеток, сеточные аппроксимации исходных задач, решение возникающих алгебраических проблем, и т.д. В работах [8], [9], [10] описываются общие структуры и характерные принципы реализации некоторых из этих компонент.

Целью данной статьи является изложение методологических подходов к созданию библиотеки аппроксиматоров CHEBYSYEV как части БСМ, ответственной за автоматизацию построения аппроксимаций для широкого спектра задач математической физики. Формально данная стадия заключается в получении алгебраической структуры данных на основе сеточной, геометрической и функциональной структур данных, полностью отображающих постановку исходной проблемы, стационарной или нестационарной, линейной или нелинейной, которая описывается системой дифференциальных и/или интегральных уравнений в расчетных областях со сложной конфигурацией кусочно-гладких неодносвязных

*Работа поддержана грантом РФФИ № 11-01-00205, а также грантами Президиума РАН № 15.9-4 и ОМН РАН № 1.3.3-4.

границ и контрастными материальными свойствами в различных подобластях.

Система CHEBYSHEV включает программные средства аналитических преобразований для реализации методов конечных объемов и конечных элементов (МКО и МКЭ) высоких порядков, разрывные методы Галеркина (РМГ) и спектральные подходы для аппроксимации основных видов дифференциальных и интегральных операторов на типовых конфигурациях сеточных ячеек и базисных функциях с разными степенями свободы, обладающих рядом общих технологических свойств, см. [11], [12], [13] и цитируемую там литературу. Разрабатываемые технологии поддерживают современные подходы вычисления локальных матриц со сборкой глобальной матрицы задачи, автоматическое сгущение адаптивных сеток в окрестностях сингулярностей решений на основе апостериорного анализа промежуточных результатов, а также масштабированное распараллеливание в методологии декомпозиции областей.

Настоящая работа построена следующим образом. В п. 2 мы рассматриваем основные технологические принципы реализации различных типов аппроксимаций путем вычисления локальных поэлементных матриц и сборки глобальных матриц, а в п. 3 приводится описание методики автоматизации построения алгоритмов на примере конечно-элементных уравнений высоких порядков для трехмерной векторной задачи расчета электромагнитных полей в частотной области.

2. Методологические принципы построения сеточных аппроксимаций

Как отмечалось в [6], [7], проблема построения сеточных аппроксимаций начально-краевых задач для дифференциальных уравнений или соответствующих вариационных постановок формально может рассматриваться как преобразование информации, полученной после генерации сетки (сюда включаются геометрическая, функциональная и сеточная структуры данных — ГСД, ФСД и ССД), в алгебраическую структуру данных (АСД), которая на дискретном уровне полностью определяет решаемую задачу и сводит её к системе линейных или нелинейных алгебраических уравнений (СЛАУ или СНАУ).

Особенностью нашего подхода является ориентация на широкий класс задач: междисциплинарных, т.е. с несколькими неизвестными функциями (скалярными или векторными), прямых или обратных, многомерных со сложными конфигурациями расчетных областей и контрастными материальными свойствами сред, стационарных или нестационарных, линейных или нелинейных.

Из этих условий возникает требование применения универсальных алгоритмов, в первую очередь, связанных с построением адаптивной неструктурированной сетки, предусматривающей наличие сеточных подобластей с различными типами сеточных элементов (тетраэдры, призмы и т.д.). С другой стороны, необходимость сохранения эффективности, экономичности и “робастности” (надежности и безотказности) численных методов предъявляет к сеточным технологиям дополнительные условия о поддержке локальных сгущений или многосеточных подходов. Кроме того, обязательность масштабируемого распараллеливания актуальных сверхбольших СЛАУ с разреженными матрицами, представляемых в сжатых форматах и решаемых на многопроцессорных — многоядерных современных компьютерных архитектурах, делает вынужденной реализацию алгоритмов на основе декомпозиции областей с распределением данных по MPI-процессам.

Ориентация на широкие классы решаемых задач требует введения общего способа задания решаемых задач в различных подобластях и краевых условий на их границах. В общем случае можно ввести проблемно-ориентированный язык (Domain Specific Language, DSL), который будет решать эту задачу. Подобные подходы применяются во многих пакетах, ориентированных на решение мульти-дисциплинарных задач, например OpenFOAM, FEniCS, и др. Вводимый язык должен удовлетворять следующим требованиям:

- позволять задать метод построения аппроксимации задачи в целом, либо в некоторой подобласти, например метод конечных разностей (МКР), МКО или МКЭ;

- позволять указывать требуемый порядок аппроксимации для методов, которые поддерживают его изменение, а также задавать варианты адаптивного выбора порядка;

- содержать поддержку представительного набора дифференциальных и интегральных операторов, одно-, двух- и трехмерных, а также поверхностных операторов, позволяющих задавать и решать широкий класс уравнений матфизики с краевыми условиями первого, второго и третьего рода;

- для нелинейных задач — включать возможность задания параметризованных нелинейных якобианов;

- устанавливать ссылки на имеющиеся в области макро-объекты: подобласти, границы подобластей, контуры, и, соответственно, стандартизованно задавать функциональную связь между объектами языка и макро-объектами области;

- позволять задавать эстиматоры для оценки ошибки дискретных решений непрерывных задач;

- позволять указывать функционалы для пост-обработки получаемых решений.

Подробнее остановимся на особенностях реализации МКЭ в аппроксиматорах. Для ряда конечных элементов, например, H^1, H^{rot} и H^{div}-конформных, можно задать общую процедуру построения базисных функций произвольных порядков на симплектических элементах (треугольниках, тетраэдрах и т.д.). В таком случае можно описать исходное пространство полиномов, а также свойства ортогональности, которым должны удовлетворять соответствующие базисные функции. Такой подход позволяет построить конечно элементные базисы произвольных порядков.

Вторым достоинством такого подхода является возможность обеспечить поддержку всех базовых дифференциальных операторов, и соответствующих преобразований базисных функций, что в теории позволяет решать произвольные задачи математической физики, при условии, что они могут быть записаны в подходящей вариационной формулировке с использованием гильбертовых пространств, соответствующих поддерживаемым конечным элементам. Важно также то, что дифференцирование и интегрирование базисных функций конечных элементов может быть сделано аналитически аппроксиматором в режиме офф-лайн, благодаря чему существенно сокращаются вычислительные расходы на этапе аппроксимации, и повышается точность аппроксимаций, поскольку остутствует численное дифференцирование и интегрирование. Это оказывается очень полезным как для построения аппроксимаций, так и для пост-обработки решений при вычислении функционалов от решений.

Исходная информация для библиотеки аппроксиматоров включает полное описание всех сеточных объектов (узлы, ребра, грани, конечные объемы, или элементы), а также их взаимосвязи, мета-информация о макро-объектах области, осуществляющая связь сеточных объектов с ГСД и ФСД, передаваемых на вход генератору сетки, а также описание решаемых уравнений и методов аппроксимации на упоминавшемся выше DSL и дополнительные атрибуты, позволяющие определить типы решаемых уравнений в подобластях, их коэффициенты и виды краевых условий на граничных сеточных гранях.

На выходе имеем АСД для решения алгебраическими решателями, линейными или нелинейными. В зависимости от типа решаемой задачи, используемой компьютерной архитектуры и применяемых решателей, АСД может быть представлена либо в виде одной или нескольких (при использовании многосеточных методов) разреженных СЛАУ в стандартном формате CSR (Compressed Sparse Row), содержащем только ненулевые элементы матрицы и необходимые ссылки для их поиска в распределенной по процессорам памяти, либо в виде процедуры вычисления действия оператора задачи на дискретизованную функцию, представленную в виде вектора чисел с плавающей запятой, для использования в безматричных методах решения СЛАУ или в решателях СНАУ.

Реализация сеточных аппроксимаций экономично реализуется с помощью хорошо распараллеливаемых поэлементных технологий, основанных на синхронно вычисляемых локальных матрицах и сборке глобальной матрицы алгебраической системы.

Для МКО данный подход мы коротко проиллюстрируем на примере скалярного уравнения диффузии (см. подробнее [11])

$$\nabla p \nabla u = f(x, y). \tag{1}$$

Пусть D_k — ячейка Дирихле–Вороного около k-го узла сетки, имеющего i_k соседних узлов, а S_i — грань этой ячейки, перпендикулярная (i, k)-у ребру сетки. Тогда из уравнения (1) следуют интегральные соотношения баланса

$$-\sum_{i=1}^{i_k} \int_{S_i} \frac{\partial u}{\partial \vec{n}} p \mathrm{d}s = \int_{D_k} f(x, y) \mathrm{d}v, \tag{2}$$

которые можно переписать в следующем виде:

$$\sum_{i=1}^{i_k} J_i = \sum_{i=1}^{i_k} \int_{V_{k,i}} f(x, y) \mathrm{d}v = \sum_{i=1}^{i_k} f_{k,i}, \quad J = -p \frac{\partial u}{\partial n}. \tag{3}$$

Здесь объемные интегралы по ячейкам Дирихле–Вороного поделены на части, соответствующие разным полигонам, имеющим в основании грани S_i. Далее, участвующие в (3) интегралы можно еще детальнее разбить на фрагменты, принадлежащие разным конечным элементам, примером которых в двумерном случае является треугольник, имеющий вершинами узлы сетки.

Таким образом, после аппроксимации нормальных производных и применения квадратурных формул в (3) для каждого j-го m-угольного конечного элемента определяется локальный вектор $J^{(j)}$ дискретных потоков, выражаемых через вектор узловых значений $u^{(j)}$ искомой функции. Например, из соотношений вида (3) для ячеек Дирихле, пересекаемых с j-м треугольным элементом, определяются величины

$$u^{(j)} = \begin{bmatrix} u_1^{(j)} \\ u_2^{(j)} \\ u_3^{(j)} \end{bmatrix}, \quad J^{(j)} = \begin{bmatrix} J_1^{(j)} \\ J_2^{(j)} \\ J_3^{(j)} \end{bmatrix}, \tag{4}$$

$$J^{(j)} = A^{(j)} u^{(j)}, \quad A^{(j)} = \begin{bmatrix} a_{1,1}^{(j)} & a_{1,2}^{(j)} & a_{1,3}^{(j)} \\ a_{2,1}^{(j)} & a_{2,2}^{(j)} & a_{2,3}^{(j)} \\ a_{3,1}^{(j)} & a_{3,2}^{(j)} & a_{3,3}^{(j)} \end{bmatrix}, \tag{5}$$

Получаемая матрица $A^{(j)}$, элементы которой выражаются через характеристики только одного треугольника, называется локальной матрицей баланса. С помощью операторов, или матриц $P_j \in \mathcal{R}^{m,N}$, расширяющих m-мерные векторы $u^{(j)}$ до полного вектора u^h размерности N, равной числу узлов сетки, определяются "большие" локальные матрицы $\bar{A}^{(j)}$, из которых собирается глобальная матрица СЛАУ, аппроксимирующая исходную задачу (1):

$$P_j u^{(j)} = u^h, \quad \bar{A}^{(j)} = P_j^T A^{(j)} P_j, \tag{6}$$

$$A = \sum_j \bar{A}^{(j)} \in \mathcal{R}^{N,N}, \;\; Au^h = f \in \mathcal{R}^N, \;\; A = \{a_{i,k}\}. \tag{7}$$

Не останавливаясь на деталях (хотя и математически нетривиальных), отметим, что учет краевых условий разных типов в околограничных элементах может быть выполнен на локальном уровне без нарушения сборочной технологии вычислений, естественным образом распараллеливаемой на многоядерных процессорах с общей памятью.

Другим возможным вариантом решения однородной задачи Дирихле для уравнения диффузии (1) является МКЭ. Решение данного уравнения с помощью смешанных конечных элементов является, в определенном смысле, обобщением МКО и кратко может быть описаны следующим образом.

Прежде всего, краевая задача в ограниченной области Ω с границей Γ сводится к смешанной постановке, записываемой в виде системы дифференциальных уравнений первого порядка:

$$\vec{g} = p\nabla u, \;\; -\nabla\vec{g} = f, \;\; u|_\Gamma = 0. \tag{8}$$

Умножая первое и второе уравнения из (8) на некоторые пробные функции $\vec{v} \in V$ и $w \in S$ из подходящих функциональных пространств, и интегрируя их формально по некоторому множеству $G \in \Omega$ с границей ∂G, получаем слабую, или вариационную, формулировку смешанного МКЭ:

$$\begin{aligned} &\int\limits_G \vec{g}\cdot\vec{v}\,dV - \int\limits_G p\nabla u\cdot\vec{v}\,dV = 0, \\ &\int\limits_G \vec{g}\cdot\nabla w\,dV - \int\limits_{\partial G} \vec{n}\cdot\vec{g}w\,ds = \int\limits_G fw\,dV, \end{aligned} \tag{9}$$

где $\vec{n}$ есть внешняя нормаль к ∂G. Исходя из имеющихся уравнений и краевых условий задачи, можно выбрать функциональные пространства следующим образом: $\vec{g},\vec{v} \in H^{\mathrm{div}}$, а $u, w \in H_0^1$. Тогда последнее уравнение системы (9) может быть переписано в виде

$$\int\limits_G \vec{g}\cdot\nabla w\,dV = \int\limits_G fw\,dV. \tag{10}$$

Искомые функции для дискретных вариантов задач представляются в виде разложения по базисным функциям соответствующих конечномерных подпространств $V_h = \mathrm{span}\left\{\vec{\psi}_i\right\} \subset V$ и $S_h = \mathrm{span}\{\varphi_i\} \subset S$: $\vec{g} = \sum_i g_i\vec{\psi}_i$, $u = \sum u_i\varphi_i$. Подставляя эти выражения в уравнения (9), можем получить блочную систему

$$\begin{bmatrix} M & B \\ C & 0 \end{bmatrix} \begin{bmatrix} g \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix}, \tag{11}$$

где

$$\begin{aligned} M_{i,j} &= \int\limits_G \vec{\psi}_j\cdot\vec{\psi}_i dV, \quad & B_{i,j} &= -\int\limits_G p\nabla\varphi_j\cdot\vec{\psi}_i dV, \\ C_{i,j} &= \int\limits_G \vec{\psi}_j\cdot\nabla\varphi_i dV, & F_i &= \int\limits_G f\varphi_i dV. \end{aligned} \tag{12}$$

Для вычисления элементов матриц M, B и C, а также вектора F можно воспользоваться хорошо распараллеливающейся поэлементной технологией, описанной выше. При этом для того, чтобы вычислить элементы B и F в каждом конечном элементе необходимо либо использовать квадратурные формулы интегрирования, либо разложить функции p и f по подходящему конечно-элементному базису.

МКЭ, по-видимому, сейчас является наиболее универсальным и продвинутым подходом (как в теоретическом плане, так и в технологическом) для решения сложных начально-краевых задач. Однако в последние десятилетия стало развиваться достаточно самостоятельное, но примыкающее к МКО и МКЭ направление, которое получило название "разрывные методы Галеркина", основанное на вариационной смешанной постановке типа (9), но использующее на дискретном уровне более общего вида разрывные пробные функции.

А именно, в применение к рассмотренному выше примеру, мы введем следующие пространства пробных фукций:

$$\begin{aligned} V_h &= \{v \in L^2(G):\ v|_T \in \mathcal{P}_q(T)\ \forall T \in G_h\}, \\ W_h &= \{w \in [L^2(G)]^2:\ w|_T \in W(T)\ \forall T \in G_h\}, \end{aligned} \tag{13}$$

где G_h – некоторая дискретизация (триангуляция) множества G, T – какого-то типа конечный элемент (объем) из G_h, $\mathcal{P}_q(T)$ – пространство определенных на элементе T полиномиальных функций степени $q \geq 1$, а $W(T) = [\mathcal{P}_q(T)]^2$ – соответствующее пространство векторных функций. В этих обозначениях РМГ определяется следующим образом: найти $u_h \in V_h$ $\vec{g}_h \in W_h$ такие, что для всех $T \in T_h$ имеют место вариационные равенства

$$\begin{aligned} &\int\limits_T \vec{g}_h \cdot \vec{w}_h dV = -\int\limits_T p u_h \nabla \cdot \vec{w}_h dV + \int\limits_{\partial T} \hat{u}_T \vec{n}_T \cdot \vec{w}_h ds, \quad \forall \vec{w}_h \in W(T), \\ &\int\limits_T \vec{g}_h \cdot \nabla v_h dV - \int\limits_{\partial T} \hat{g}_T \cdot \vec{n}_T v_h\, ds = \int\limits_T f\, v_h\, dV, \qquad\qquad \forall v_h \in \mathcal{P}_q(T). \end{aligned} \tag{14}$$

Здесь через $\hat{u}_T$ и $\hat{g}_T$ обозначены следы соответствующих функций на границах конечных, выбор способа аппроксимаций которых, совместно с заданием порядка q, определяет один из конкретных алгоритмов РМГ.

3. Автоматизация построения конечно-элементных аппроксимационных алгоритмов

Мы рассмотрим методику автоматизации построения аппроксимаций дифференциальных уравнений на примере трехмерной векторной задачи расчета электромагнитных полей в частотной области. Данная задача может быть сведена к комплексному векторному уравнению Гельмгольца вида [14]

$$\nabla \times \nabla \times \vec{E} - \kappa \vec{E} = \vec{F}, \tag{15}$$

где $\vec{E}$ — комплексная амплитуда электрического поля, комплексный коэффициент κ связан с физическими параметрами среды, а $\vec{F}$ — вектор правой части. Данное уравнение может быть переписано в вариационной формулировке, для аппроксимации которой часто используют метод векторных конечных элементов [12].

В случае краевых условий Дирихле данная задача переписывается в вариационной форме как

$$\int\limits_G \left(\nabla \times \vec{E}\right) \cdot \left(\nabla \times \vec{\psi}\right) dV - \int\limits_G \kappa \vec{E} \cdot \vec{\psi} dV = \int\limits_G \vec{F} \cdot \vec{\psi} dV, \quad \vec{E}, \vec{\psi} \in H^{\mathrm{rot}}. \tag{16}$$

Автоматизация процесса конструирования конечно-элементных аппроксимационных алгоритмов заключается в следующем. Для построения аппроксимаций вариационных формулировок задач необходимо найти явные выражения для вычисления локальных матриц и векторов, а также для построения аппроксимаций функций, находящихся в правой части дифференциального уравнения. Помимо этого, для постобработки часто требуется генерация выражений для вычисления различных функций и функционалов зависящих от решения, предпочтительно, без использования численного дифференцирования и интегрирования. Данная задача является довольно трудоемкой, особенно в случае использования базисных функций конечных элементов высоких порядков.

В общем случае при автоматизации процесса построения аппроксимаций можно столкнуться с серьезными трудностями. В частности, при использовании МКЭ, вычислительная сложность построения аппроксимации зависит от типа используемых конечных элементов и составляющих элементов сетки. Однако при использовании тетраэдральных сеточных элементов (или, в общем случае, элементов на симплексах) задача существенно упрощается. Существующие обобщенные подходы (см. например [15], [16]) к построению конечно-элементных аппроксимаций позволяют строить дискретизации задач для широкого спектра конечных элементов и произвольных порядков базисных функций.

В частности, базисные функции конечных элементов Неделека, конформных пространству H^{rot}, в произвольном тетраэдре можно построить на основе барицентрических координат — линейных функций ξ_i, $i = 1, \ldots, 4$, принимающих значение 1 в одной из вершин тетраэдра и 0 — в остальных. Исходя из общего вида базисных функций [15] можно показать, что выражения для вычисления локальных матриц жесткости и масс, соответствующих первому и второму интегралам в (16), в тетраэдре будут иметь вид

$$\sum_j \sum_k \int\limits_T c_j c_k \xi_1^{p_{j,k,1}} \xi_2^{p_{j,k,2}} \xi_3^{p_{j,k,3}} \xi_4^{p_{j,k,4}} \left(\vec{F}_j(\xi_1, \xi_2, \xi_3, \xi_4) \cdot \vec{F}_k(\xi_1, \xi_2, \xi_3, \xi_4) \right) dV, \tag{17}$$

где T — тетраэдр, c_j — некоторые численные константы, а $\vec{F}_j(\xi_1, \xi_2, \xi_3, \xi_4)$ — константная вектор-функция. Аналитическое вычисление интегралов здесь может быть проведено с использованием формулы (см. [17])

$$\int\limits_T \xi_1^{p_1} \xi_2^{p_2} \xi_3^{p_3} \xi_4^{p_4} dV = \frac{p_1! p_2! p_3! p_4!}{(p_1 + p_2 + p_3 + p_4 + 3)!} \mathrm{mes} T. \tag{18}$$

Для автоматизации процесса построения аппроксимаций предлагается использовать подсистему специализированных аналитических вычислений. На вход данной системе подается описание выражений для вычислений, заданное в соответствии с некоторой формальной грамматикой, а также некоторое формализованное описание базисных функций для используемых конечных элементов.

Далее система строит внутреннее представление требуемых базисных функций, а также генерирует дерево вычислений локальных матриц и векторов. Стоит отметить, что во многих случаях для конечных элементов на симплексах удается провести большинство таких вычислений, как нахождение производных функций, интегралов линейных и билинейных форм по конечным элементам, аналитически в соответствии с заложенными в систему правилами.

Таким образом, одной из специфических компонент системы автоматизации построения аппроксимаций является модуль символьных вычислений.

Далее генерируются выражения для вычисления элементов локальных матриц и векторов в соответствии с полученным представлением и проводятся символьные оптимизации. Проводимые оптимизации состоят в приведении подобных слагаемых, поиске общих подвыражений, и организация потока вычисления для минимизации требуемого времени вычисления элементов матриц и векторов. В целом задача оптимизации является вычислительно сложной, и при ее решении используются различные эвристики.

Помимо генерации выражений для вычисления элементов локальных матриц и векторов также генерируется код для построения аппроксимации правой части. Для этого используются квадратурные формулы интегрирования, а вычисления на этапе конструирования аппроксиматора заключаются в предподсчете точек, в которых необходимо вычислить значения функций в соответствии с порядком выбранных конечных элементов для получения аппроксимации необходимой точности.

После проведения необходимых символьных вычислений и оптимизаций система генерирует исходный код на языке C++ для вычисления соответствующих выражений.

Можно отметить, что объем сгенерированного кода может быть довольно велик: например, для векторного уравнения Гельмгольца при аппроксимации элементами Неделка четвертого порядка общий объем сгенерированного кода превышает 9 мегабайт. Естественно, что такая система аналитических вычислений существенно упрощает поддержку построения аппроксимаций высоких порядков.

Помимо подсистемы аналитических вычислений и генератора кода, в аппроксиматор также входит библиотека поддержки времени исполнения. Туда включаются функции, общие для большинства поддерживаемых аппроксимаций — построение распределенных индексов степеней свобод, сопоставляющих последние с переменными в подобластях, алгоритмы на графах, включая покраску графов, применяемые в параллельной сборке матриц, процедуры параллельной сборки СЛАУ с использованием сгенерированных функций вычисления локальных матриц и векторов, матричные конверторы, и т.д.

Список литературы

1. IESP: URL: www.exascale.org/iesp (дата обращения: 01.12.2013).

2. ANSYS — Simulation Driven Product Development: URL: www.ansys.com (дата обращения: 01.12.2013).

3. MSC Nastran — Industry Leading Multidisciplinary FEA: URL: www.mscsoftware.com/product/msc-nastran (дата обращения: 01.12.2013).

4. OpenFOAM® — The Open Source Computational Fluid Dynamics (CFD) Toolbox: URL: www.openfoam.com (дата обращения: 01.12.2013).

5. DUNE Numerics: URL: www.dune-project.org (дата обращения: 01.12.2013).

6. Ильин В.П. Параллельные процессы на этапах петафлопного моделирования // Вычислительные методы и программирование, 2011. Т. 12, N. 1. С. 93–99.

7. Ильин В.П., Скопин И.Н. Технологии вычислительного программирования // Программирование, 2011. N. 4, С. 53–72.

8. Голубева Л.А., Ильин В.П., Козырев А.Н. О программных технологиях в геометрических аспектах математического моделирования // Вестник НГУ, серия Информационные технологии, 2012. Т. 10, N. 2. С. 25–33.

9. Ильин В.П. DELAUNAY: технологическая среда генерации сеток // СибЖИМ, 2013. Т. 16, N. 2(54). С. 83–97.

10. 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 // Вестник ЮУрГУ. Серия “Вычислительная математика и информатика”, 2013. Т. 2, N. 3. С. 92–105.

11. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений. Новосибирск: изд. ИВМиМГ СО РАН, 2001. 318 с.

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

13. Arnold D.N., Brezzi F., Cockburn B., Marini L.D. Unified analysis of discontinuous Galerkin methods for elliptic problems // SIAM J. Numer. Anal., 2002. Vol. 39, N. 5. P. 1749–1779.

14. Бутюгин С.Д., В.П.Ильин В.П. Параллельные методы и технологии моделирования электродинамических полей // Микроэлектроника СВЧ, Сборник трудов конференции, Санкт-Петербург, СПБГЭУ "ЛЭТИ", 2012. С. 223–227.

15. Ingelstrom P. A new set of H(curl)-conforming hierarchical basis functions for tetrahedral meshes // IEEE Transactions on Microwave Theory and Techniques. 2006. Vol. 54, N. 1. P. 160–114.

16. Kirby R.C, Logg A. A Compiler for Variational Forms // ACM Transactions on Mathematical Software. 2006. Vol. 32, N. 3. P. 417–444.

17. Соловейчик Ю.Г., Рояк М.Э., Персова М.Г. Метод конечных элементов для решения скалярных и векторных задач. Новосибирск: Изд-во НГТУ, 2007.