

ОРДЕНА ЛЕНИНА ИНСТИТУТ ПРИКЛАДНОЙ
МАТЕМАТИКИ
им. М. В. Келдыша
РОССИЙСКОЙ АКАДЕМИИ НАУК

В.Т. Жуков, О.Б. Феодоритова, Н.Д. Новикова

ИТЕРАЦИОННЫЙ МЕТОД
ДЛЯ КОНЕЧНО-ЭЛЕМЕНТНЫХ СХЕМ
ВЫСОКОГО ПОРЯДКА.

ЧАСТЬ II

Москва, 2003

Работа выполнена при частичной финансовой поддержке Российского Фонда Фундаментальных Исследований (проект 02-07-90027)

УДК.519.6

В.Т. Жуков, Н.Д. Новикова, О.Б. Феодоритова

Email: zhukov@kiam.ru, nn@kiam.ru, feodor@kiam.ru

Итерационный метод для конечно-элементных схем высокого порядка.

Часть II.

АННОТАЦИЯ

Изучены возможности итерационного алгоритма для hp -схем, получаемых при аппроксимации на неструктурных сетках дифференциальных уравнений с помощью стабилизированных вариантов метода конечных элементов (МКЭ). Приведены результаты численных экспериментов для задач, описываемых системами уравнений Эйлера и Навье-Стокса.

Стр. 32, табл. 4, рис. 17, библи. 8 назв.

O.B. Feodoritova, N.D. Novikova, V.T. Zhukov.

An iteration method for high order finite element schemes. Part II

ABSTRACT

An iterative algorithm for the solution of hp -discretization on unstructured grids is studied. It is applied for solution of Euler and Navier-Stokes equations discretized by stabilized finite element method (FEM). The results of numerical experiments are presented to show capability of the algorithm.

Содержание

1 Введение

Данная работа служит продолжением публикации [?], в которой приведен итерационный алгоритм для решения разностных схем высокого порядка, получаемых при дискретизации дифференциальных уравнений с помощью метода конечных элементов (МКЭ) или его стабилизированных вариантов. В той же работе обсуждаются результаты экспериментов по решению конечно-элементных схем для уравнений Пуассона, анизотропной диффузии, конвекции-диффузии; в [?] приведены примеры некоторых способов предобуславливания конечно-элементных схем. Здесь приводятся результаты, относящиеся к решению систем уравнений Эйлера и Навье-Стокса. Конечно-элементные hp -аппроксимации таких уравнений приводят к системам линейных алгебраических уравнений очень большой размерности, трудным для численного решения. Поэтому актуальной является разработка алгоритмов редукции систем МКЭ высокого порядка к задачам более простым и/или меньшей размерности. Теоретические аспекты разработки таких алгоритмов приведены, например, в [?].

Цель данной работы состоит в практическом изучении итерационного метода для hp -схем на неструктурных треугольных сетках. При этом предполагается, что на заданной сетке схема с линейными элементами ($p = 1$) может быть решена эффективно. Это естественное предположение, так как при использовании элементов высокого порядка сетка может быть значительно более грубой, чем в случае расчета с помощью схемы первого порядка. Решение системы линейных уравнений, соответствующей схеме порядка $p = 1$ на такой грубой сетке, может быть получено либо прямым методом, либо известными итерационными методами, например, многосеточным.

2 Дискретизация

2.1 Постановка задачи

Будем рассматривать системы линейных алгебраических уравнений, возникающие в процессе решения краевых задач для стационарных уравнений Эйлера и Навье-Стокса. Решение таких задач ищется установлением по времени, и системы линейных уравнений возникают в процессе линеаризации неявной нелинейной схемы на каждом временном слое. В расчетах использовались неструктурные треугольные сетки, полученные с помощью генератора сеток [?]. Дискретизация уравнений Эйлера и Навье-Стокса, алгоритм решения систем нелинейных алгебраических уравнений и программа `supgns` C^{++} разработаны Д.С. Каменецким, оказавшим также помощь в обеспече-

нии интерфейса между его программой и модулем **ASA F90** решения систем линейных уравнений, реализующим предложенный в [?] и изучаемый здесь итерационный алгоритм.

Полную формулировку конечномерных задач для уравнений Эйлера и Навье-Стокса можно найти, например, в [?], здесь для простоты приведем формулировку дискретной задачи в скалярном случае. Пусть в ограниченной области $\Omega \subset R^2$ с границей $\partial\Omega$ ищется решение $u(x, y)$ первой краевой задачи для линейного эллиптического уравнения:

$$Lu = f, \quad (x, y) \in \Omega, \quad (2.1)$$

$$u(x, y) = 0, \quad (x, y) \in \partial\Omega, \quad (2.2)$$

здесь $f(x, y)$ – заданная функция. Дискретизация задачи (2.1) – (2.2) проводится с помощью метода конечных элементов. В качестве проекционных рассматриваются семейства конечно-элементных подпространств V_h пространства Соболева $H_0^1 = \{v \in H^1(\Omega), v = 0 \text{ на } \partial\Omega\}$; подпространство V_h состоит из кусочно-полиномиальных функций степени $k \geq 1$ (здесь и далее степень полинома обозначается через k) на триангуляции $T_h \in T$, обращающихся в нуль на границе $\partial\Omega$.

Стандартная конечно-элементная формулировка задачи имеет вид: найти функцию $u_h \in V_h$ такую, что

$$B(u_h, v_h) = (f, v_h), \quad \forall v_h \in V_h, \quad (2.3)$$

здесь (\cdot, \cdot) – скалярное произведение в $L_2(\Omega)$, $B(\cdot, \cdot)$ – билинейная форма, соответствующая уравнению (2.1). В данной работе мы будем использовать стабилизированный вариант МКЭ, который приводится ниже в п.2.3.

2.2 Сетка, сеточные функции

В области Ω строится неструктурная треугольная сетка, полученная регулярной триангуляцией области. Такая сетка состоит из l треугольников T^1, \dots, T^l , которые будем называть ячейками, причем $Int T^i \cap Int T^j = \emptyset$ при $i \neq j$, где $Int T^j$ – внутренность треугольника, а их объединение совпадает со всей областью. Треугольники могут иметь общую сторону, или общую вершину или не иметь общих точек совсем. Стороны треугольников могут быть криволинейными для более точного описания границ. Введем сеточные аппроксимации функций u, f , сохраняя за ними прежние обозначения. Каждая ячейка сетки рассматривается как конечный элемент заданного порядка $k = 1, 2, 3, \dots$. Порядок элемента может меняться от ячейки к ячейке, но для простоты ограничимся постоянным для всех ячеек порядком k , который будем называть

порядком схемы. Для схемы порядка k значения сеточных функций задаются в вершинах треугольника, а также в некотором числе дополнительных узлов на сторонах (ребрах) треугольника и внутри него в зависимости от k . Полное число узлов сетки в треугольнике T^t есть $M = (k + 1)(k + 2)/2$, из них $(k - 1)$ узел лежит на каждой стороне и $(k - 1)(k - 2)/2$ узла лежит внутри треугольника. Сетку, состоящую из вершин треугольников и точек, лежащих на ребрах, будем называть основной и обозначать N_h (рис. 1 а). Полную сетку, включающую узлы сетки N_h и узлы внутри треугольников обозначим M_h . При $k = 1, 2$ сетки N_h и M_h совпадают. На сетке N_h будет записываться дискретная схема, сетка M_h используется для кусочно-гладкого восполнения сеточных функций. Пространство функций H_0^1 будем аппроксимировать пространством V_h непрерывных кусочно-полиномиальных функций, как в стандартном МКЭ. Функция $w \in V_h$, аппроксимирующая любую функцию $u \in H_0^1$, в каждой ячейке является интерполяционным полиномом степени k по узлам сетки M_h .

2.3 Стабилизированные схемы Галеркина

Для эллиптических задач с доминирующим конвективным переносом, а также для систем уравнений Эйлера и Навье-Стокса, обычно используют методы конечных элементов Петрова-Галеркина, или, как их еще называют, стабилизированные схемы Галеркина [?] ищется $u \in V_h$ такая, что для $\forall v \in V_h$

$$B(u, v) + \sum_t (\tau(Lu - f), \bar{L}v)_T = (f, v), \quad (2.4)$$

где τ – параметр схемы, суммирование проводится по всем ячейкам сетки; индекс T означает, что интеграл вычисляется по ячейке $T = T^t$. Стабилизирующий оператор \bar{L} определяет тип метода. Ограничимся двумя близкими случаями, когда в качестве \bar{L} выбирается исходный оператор L (Galerkin/least squares, GLS) или его конвективная часть (streamline upwind/Petrov-Galerkin, SUPG). Для систем дифференциальных уравнений параметр τ – матрица, элементы которой в каждой ячейке определяются ее размерами и коэффициентами уравнений в ней. Назначение этого параметра – регулировать величину искусственной вязкости, добавляемой в уравнения для подавления больших нефизических осцилляций.

2.4 Формирование системы линейных уравнений

Рассмотрим систему линейных уравнений, возникающую в результате численного решения систем нелинейных уравнений Эйлера или Навье-Стокса:

$$Au = b. \quad (2.5)$$

Здесь A – матрица жесткости размерности $N \times N$, u – вектор неизвестных u_h относительно конечно-элементного базиса в пространстве V_h . Конечно-элементный базис может быть выбран разными способами. В данной работе будут рассмотрены два базиса – лагранжев и иерархический, как правило используемые в МКЭ. В лагранжевом базисе локальные базисные функции в ячейках являются полиномами одной и той же степени k ; в иерархическом базисе они упорядочены по степеням: линейные функции, ассоциированные с вершинами ячеек, последовательно дополняются полиномами все более высокой степени, связанными со сторонами ячеек, а для $k > 2$ еще и с внутренними узлами в ячейках. Степени свободы внутри ячеек исключаются с помощью процедуры "static condensation" т.е. система уравнений (2.5) связывает неизвестные, ассоциированные только с узлами триангуляции и точками на ребрах.

Пусть матрица A и вектор b линейной системы (2.5) возникают в результате конечно-элементной дискретизации в каком-либо базисе Φ_1 . Тогда переход в другой базис Φ_2 описывается линейным преобразованием $\bar{A} = C^T A C$ и $\bar{b} = C^T b$, где C – матрица перехода от базиса Φ_1 к базису Φ_2 , T – операция транспонирования. Выбор базиса влияет на обусловленность линейной системы и, как следствие, на трудность ее численного решения. Если один из рассматриваемых базисов – лагранжев и иерархический – будет иметь заметное преимущество, то можно либо сразу записывать схему в таком базисе, либо делать преобразование матрицы жесткости на этапе решения системы линейных уравнений, как это делалось в настоящей работе.

3 Описание итерационного алгоритма

3.1 Описание сеточных структур: подробный и грубый уровни

В п. 2.1 для схемы порядка k было введено понятие основной сетки N_h как множества вершин треугольников и точек, лежащих на ребрах триангуляции. Сетку N_h будем называть сеткой первого уровня, или подробной. Сетка второго уровня, или грубая сетка в терминологии многосеточного метода, содержит только узлы триангуляции. Пример фрагментов сеток на каждом уровне показан на рис. 1. На сетке первого уровня мы имеем две группы неизвестных. Группа u_c связана с узлами триангуляции, назовем их c – узлами. Группа u_f связана с точками на ребрах ячеек, назовем их f – узлами. Представлением вектора неизвестных u в виде (u_c, u_f) порождает разбиение уравнений (2.5) на две группы. Уравнения первой группы соответствуют c – узлам, а уравнения второй – f – узлам, но шаблоны каждого уравнения включают узлы обоих типов.

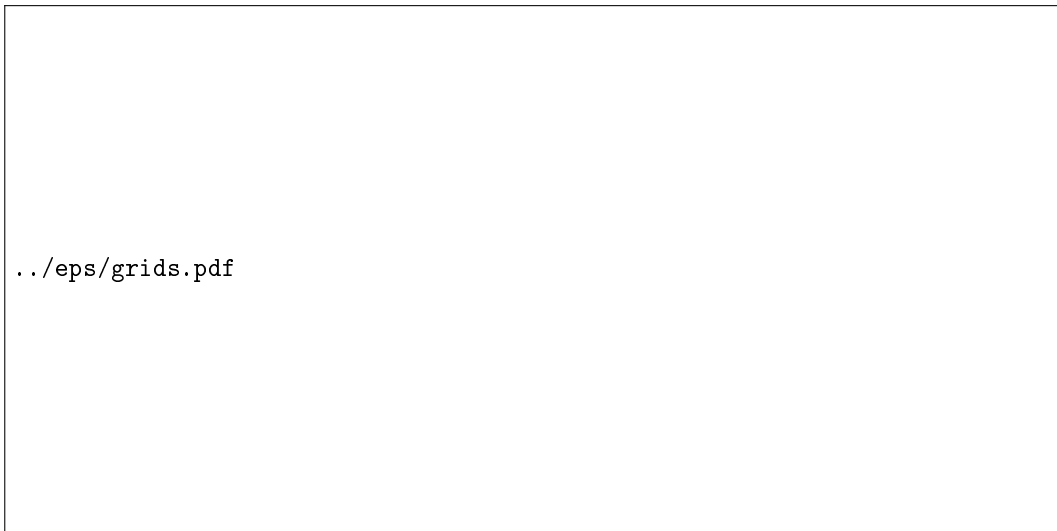


Рис. 1: **a**, **b** – фрагменты сеток первого и второго уровней

3.2 Операторы на каждой сетке.

Для схем порядка $k \leq 2$ дискретный оператор на подробной сетке определяется исходной матрицей жесткости A_h , связанной с соответствующей билинейной формой, а при $k > 2$ – матрицей системы, полученной после точного исключения неизвестных, ассоциированных с внутренними точками ячеек. В последнем случае правая часть системы уравнений (2.5) считается преобразованной соответствующим образом.

Конструкция оператора на грубой сетке зависит от выбранного конечно-элементного базиса. Напомним, что мы рассматриваем узловой (лагранжев) и иерархический базисы. В соответствии с представлением u в виде (u_c, u_f) матрицу системы (2.5) запишем в блочной форме:

$$A_h = \begin{pmatrix} A_{cc} & A_{cf} \\ A_{fc} & A_{ff} \end{pmatrix}.$$

В случае иерархического базиса в качестве оператора на грубой сетке естественно выбрать блок A_{cc} , соответствующий аппроксимации кусочно-линейными полиномами в случае стандартного МКЭ (2.3).

Геометрической конструкцией при построении грубого оператора в случае лагранжевого базиса является макроэлемент (МЭ). Макроэлемент — это произвольное связное множество ячеек сетки с введенной спецификацией узлов основной сетки, попавших в МЭ. Используются два вида макроэлементов:

- простой МЭ: объединение всех ячеек, имеющих выделенный узел в качестве одной из вершин;
- расширенный МЭ: подобласть расчетной области, являющаяся объединением некоторого числа ячеек. В модельных задачах такие МЭ могут

строится вручную, но обычно мы использовали программу METIS [?] для декомпозиции всей области на непересекающиеся подобласти, а затем присоединяли к каждой подобласти ячейки, у которых одна или две вершины попали на границу подобласти.

Каждый узел, принадлежащий МЭ, получает одну из следующих спецификаций $c/r/e$, в зависимости от которой определяется способ обработки узла. Все узлы на границе МЭ имеют спецификацию c или e ; к c -узлам относим узлы триангуляции, к e -узлам – точки, лежащие на граничных ребрах МЭ. Все внутренние узлы МЭ (т.е. узлы, шаблон которых полностью погружен в МЭ) имеют c и r спецификацию; как правило, к c -узлам относим узлы триангуляции, к r -узлам – точки на внутренних ребрах. Хотя, конечно, часть узлов триангуляции может быть добавлена к подмножеству r -узлов. На рис. 2 показан простой МЭ с указанием спецификации всех узлов. Соответствующим образом специфицируются и значения сеточной функции в МЭ.



Рис. 2: Простой МЭ с типичной спецификацией узлов

Алгоритм построения грубого оператора основан на приближенном исключении f -компоненты решения. Рассмотрим отдельный МЭ. Пусть u_k, u_{k+1} – два последовательных приближения к решению исходной линейной системы, $\delta = u_{k+1} - u_k$ – искомая поправка. Для внутренних узлов типа c и r данного МЭ система уравнений относительно поправки может быть представлена в следующем блочном виде:

$$\begin{bmatrix} A_{cc} & A_{cr} & A_{ce} \\ A_{rc} & A_{rr} & A_{re} \end{bmatrix} \begin{bmatrix} \delta_c \\ \delta_r \\ \delta_e \end{bmatrix} = \begin{bmatrix} g_c \\ g_r \end{bmatrix}, \quad (3.1)$$

где $g = b - Au_k$ – невязка. Перепишем эту систему в компактной форме

$$A_h^{me} \delta^{me} = g^{me}, \quad (3.2)$$

где A_h^{me} – матрица МЭ. Легко видеть, что эта система является недоопределенной. Величины δ_r могут быть исключены точно, после чего остаются уравнения относительно неизвестных δ_c и δ_e :

$$(A_{cc} - A_{cr}A_{rr}^{-1}A_{rc})\delta_c + (A_{ce} - A_{cr}A_{rr}^{-1}A_{re})\delta_e = g_c - A_{cr}A_{rr}^{-1}g_r. \quad (3.3)$$

Для исключения неизвестных δ_e применим приближенную формулу:

$$\delta_e = W\delta_c,$$

где W – заданная интерполирующая матрица. Она может быть определена различными способами. Наиболее простой и используемый в настоящей работе способ – линейная интерполяция вдоль ребер сетки.

После исключения граничных переменных δ_e получается система уравнений для c -переменных, соответствующих внутренним узлам МЭ.

$$A_H^{me} \delta_c = \bar{g}_c, \quad (3.4)$$

где

$$A_H^{me} = A_{cc} - A_{cr}A_{rr}^{-1}A_{rc} + (A_{ce} - A_{cr}A_{rr}^{-1}A_{re})W \quad (3.5)$$

– оператор на грубой сетке, а невязка на грубой сетке дается выражением

$$\bar{g}_c = g_c - A_{cr}A_{rr}^{-1}g_r.$$

3.3 Операторы интерполяции и сборки

Предыдущая процедура определяют оператор интерполяции P в МЭ, который ставит в соответствие сеточной функции g_c , определенной в c -узлах МЭ, сеточную функцию $g = [g_c, g_r, g_e]$, определенную во всех узлах МЭ, по формуле:

$$g = Pg_c, \quad \text{где} \quad P = [I; -A_{rr}^{-1}(A_{rc} + A_{re}W); W]^T, \quad (3.6)$$

Оператор сборки в МЭ имеет вид:

$$R = [I, -A_{cr}A_{rr}^{-1}]. \quad (3.7)$$

С помощью операторов интерполяции и сборки можно записать оператор на грубой сетке в виде

$$A_H^{me} = R A_h^{me} P, \quad (3.8)$$

т.е. уравнения на грубой сетке формируются с помощью процедуры, называемой в многосеточном методе галеркинской.

Приведенные выше операторы определены на шаблонах, ассоциированных с каждым МЭ. Уравнения на всей грубой сетке формируются суммированием по всем МЭ в области и имеют вид:

$$A_H \delta_c = g_c .$$

Результирующую матрицу A_H можно интерпретировать как некоторое приближение к дополнению Шура $S = A_{cc} - A_{cf} A_{ff}^{-1} A_{fc}$. "Расстояние" между точным и приближенным дополнением Шура зависит от размеров МЭ и выбора интерполирующей матрицы W . Точное дополнение Шура будет получено, если МЭ совпадет со всей расчетной областью. Ясно, что расчет S трудоемок и, по-видимому, не нужен, как это показано с помощью численных экспериментов в [?].

Оператор на грубой сетке в случае иерархического базиса. Для иерархического базиса оператор на грубой сетке, как и операторы межсеточного перехода, имеют более простой вид:

$$A_H = A_{cc}, \quad R = [I, 0], \quad P = R^* = [I, 0]^T.$$

Сглаживающая процедура. В стандартном многосеточном методе предполагается, что для перехода на более грубую сетку нужно на текущей подробной сетке обеспечить гладкость невязки, т.е. погасить с помощью достаточно простого итерационного метода высокочастотные компоненты невязки. Будем использовать в качестве сглаживающей процедуры несколько различных методов. Основным будет блочный метод Гаусса-Зейделя, основанный на декомпозиции области с помощью METIS[?], но будет использоваться и простейший вариант, когда каждый блок совпадает с простым МЭ. На рис. 3 показаны несколько блоков релаксации, в одном из них отмечены исключаемые неизвестные (для схемы порядка 2). Релаксацию, блоки которой устроены таким образом, будем называть "точечной". При расчете каждого блока соответствующие внутренние c - и r -переменные будут обновляться одновременно. Вариант, когда обновляются только внутренние r -переменные, также был использован в расчетах, но оказался неудачным.

Процедура сглаживания, т.е. переход от текущего итерационного приближения u_k к более "гладкому" \bar{u}_k можно записать с помощью оператора сглаживания S , который текущую ошибку $\varepsilon_0 = u_k - u$ за шагов сглаживания преобразует в ошибку $\varepsilon_1 = \bar{u}_k - u : \varepsilon_1 = S^\mu \varepsilon_0$. Конкретный вид оператора S не важен; можно, например, представлять его в виде $S = (I - B^{-1}A)$, где оператор релаксации B должен быть близок к A на негладкой части спектра.

Итерирующий оператор. Детальное описание алгоритма приведено ниже в 3.4. Полный итерирующий оператор M этого алгоритма имеет вид:

$$M = S^{\mu_1} (I - P A_H^{-1} R A_h) S^{\mu_2}, \quad (3.9)$$

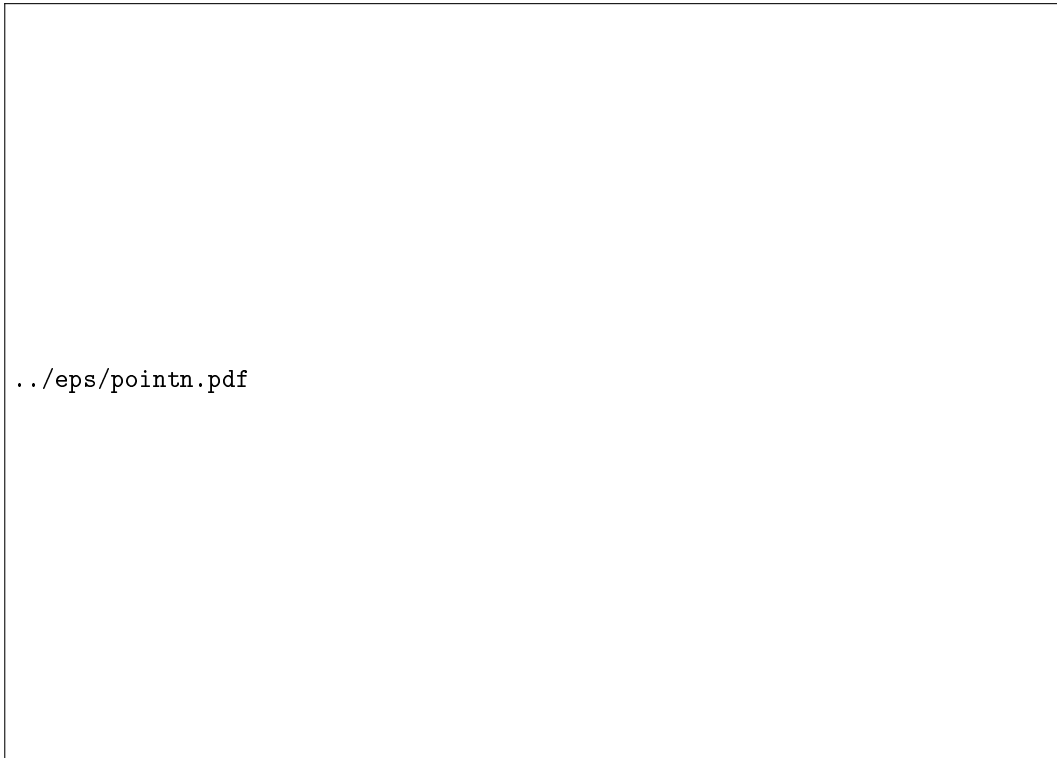


Рис. 3: Сетка вблизи тела. Примеры блоков "точечной"релаксации

где $(I - PA_H^{-1}RA_h)$ – оператор коррекции, μ_1, μ_2 – число шагов сглаживания до и после коррекции соответственно. Операторы интерполяции P и сборки R определены своими шаблонными представлениями (3.6) и (3.7).

Спектральный радиус $\rho(M)$ оператора M определяет скорость сходимости итераций; разумную теоретическую оценку $\rho(M)$ на неструктурных сетках дать практически невозможно, мы приведем некоторые экспериментальные данные о скорости сходимости алгоритма, полученные в конкретных расчетах.

3.4 Алгоритм

Пусть на основной сетке задано итерационное приближение u_k . Переход к новой итерации u_{m+1} происходит в три этапа:

1. Сглаживание невязки до коррекции:

По u_m вычисляем \bar{u}_m с помощью $\mu_1 (\geq 0)$ шагов сглаживающей процедуры;

2. Грубосеточная коррекция:

Вычисляем невязку $\bar{g} = b - A\bar{u}_m$;

Находим с помощью оператора сборки невязку \bar{g}_c на грубой сетке;

Решаем систему уравнений для поправки $\bar{\delta}_c$ на грубой сетке;

Находим $\bar{\delta}$ поправку на основной сетке интерполяцией $\bar{\delta}_c$ с грубой сетки;
 Проводим коррекцию: $\tilde{u}_m = \bar{u}_m + \bar{\delta}$;

3. Сглаживание невязки после коррекции:

По \tilde{u}_m вычисляем u_{m+1} с помощью $\mu_2(\geq 0)$ шагов сглаживающей процедуры.

Переход от начального приближения u_0 к u_m можно записать в виде

$$u_m = M^m u_0 + (I - M^m)A^{-1}b, \quad (3.10)$$

где итерирующий оператор M приведенного выше алгоритма дан в (3.9).

Оценка средней скорости сходимости итераций ρ рассчитывалась по формуле: $\rho = (r_m/r_0)^{1/m}$, где m – номер итерации, r_m – норма невязки на основной сетке, $r_m^2 = \sum_{n=1}^{n=N} (b - Au_m)_n^2$. Начальная невязка вычислялась при $u_0 \equiv 0$. Итерации проводились до достижения заданной относительной точности $\varepsilon : r_m/r_0 < \varepsilon$. Система для поправки решалась итерационным методом GMRES [?] с относительной точностью ε_c . Обычно, если не оговорено противное, задается $\varepsilon = 10^{-10}$ и $\varepsilon_c = 10^{-5}$. Точность ε_c – это параметр, который может влиять на эффективность метода (см. [?]).

Приведенный алгоритм будем называть AS–L и AS–H в случае лагранжевого и иерархического базисов соответственно.

Отметим, что данный алгоритм можно использовать как итерационный алгоритм решения исходной системы (тогда число итераций m определяется заданной точностью и может быть достаточно большим). Можно использовать его как предобуславливатель системы уравнений, задавая небольшое фиксированной число итераций, и не контролируя достигаемую точность. В этом случае предобуславливающий оператор C определяется соотношением $C^{-1} = (I - M^m)A^{-1}$, а оператор предобусловленной системы есть $C^{-1}A \equiv (I - M^m)$. Можно не требовать выполнения условия $\|I - M^m\| < 1$; для эффективности нужно, чтобы $C^{-1}A \simeq I$. Мы кратко затронем вопрос о возможностях алгоритма AS–L как предобуславливателя системы линейных уравнений в п. ??.

Переходя к описанию численных экспериментов заметим, что говоря об обращении матриц или операторов, мы обычно подразумеваем численное решение системы уравнений, а не нахождение обратной матрицы.