

K. Stüben A review of algebraic multigrid. 2001.

## **Аннотация**

С начала 1990-х годов резко возрос спрос на более эффективные методы решения больших разреженных неструктурированных линейных систем уравнений. Для практически актуальных размеров задач классические одноуровневые методы уже достигли своих пределов, и необходимо было разработать новые иерархические алгоритмы, чтобы обеспечить эффективное решение еще больших задач. В этой статье даётся обзор первого иерархического и чисто матричного подхода - алгебраический многосеточный метод (AMG). AMG может непосредственно применяться, например, для эффективного решения различных типов эллиптических уравнений в частных производных, дискретизированных по неструктурированным сеткам, как в 2D, так и в 3D. Поскольку AMG не использует какую-либо геометрическую информацию, он является решающим модулем, который можно применять даже к задачам без какого-либо геометрического фона при условии, что базовая матрица обладает определенными свойствами.

## **Введение**

Эффективное численное решение больших систем дискретных эллиптических уравнений в частных производных (PDE) требует иерархических алгоритмов, которые обеспечивают быстрое уменьшение как коротковолновых, так и длинноволновых компонент в разложении вектора ошибки. Прорыв, и, безусловно, одно из самых важных достижений за последние три десятилетия, был обусловлен многосеточным принципом. Любой соответствующий метод работает с иерархией сеток, заданной априори путем огрубления данной сетки дискретизации геометрически естественным образом («геометрический» многосеточный метод). Понятно,

что это просто для логически правильных сеток. Тем не менее, определение естественной иерархии может стать очень сложным для очень сложных, неструктурированных сеток, если возможно вообще.

Первая попытка автоматизировать процесс укрупнения была предпринята в начале 1980-х годов [10, 12, 13], в то время, когда так называемая интерполяция, зависящая от принципа Галеркина и оператора, была объединена в геометрический многосеточный метод для повышения его устойчивости (с целью решения задач диффузии со скачкообразными коэффициентами [1,20]). Эта попытка была мотивирована наблюдением, что разумная, зависящая от оператора, интерполяция и оператор Галеркина часто могут быть получены непосредственно из базовых матриц, без какой-либо ссылки на сетки. Результатом стал многосеточный подход, который не просто позволял выполнять процесс автоматического укрупнения, но мог быть непосредственно применен к (разреженным линиям) алгебраическим уравнениям определенных типов без какой-либо заранее определенной иерархии («алгебраический» многосеточный метод, AMG).

Первая довольно общая программа AMG была описана и исследована в [47,48,50]. Поскольку этот код стал общедоступным в середине 1980-х годов (AMG1R5), в течение многих лет не было никаких существенных дальнейших исследований и разработок в AMG. Однако с начала 1990-х годов, и даже больше, чем с середины 1990-х годов, резко возрос интерес к алгебраически ориентированным многоуровневым методам. Одной из причин этого, безусловно, была возрастающая геометрическая сложность приложений, которая технически ограничивала непосредственное использование геометрических многосеточных методов. Другой причиной был постоянно растущий спрос на эффективные «подключаемые» решатели. В частности, в коммерческих кодах это требование было обусловлено увеличением размеров проблем, которые прояснили пределы классических одноуровневых решателей, все еще используемых в большинстве пакетов.

Например, применение CFD в автомобильной промышленности затрагивает очень сложные области. Проточные системы обогрева и охлаждения, потоки снаружи автомобилей или потоки внутри пассажирских отделений рассчитываются на регулярной основе. Большие сложные сетки, обычно неструктурированные, используются для моделирования таких ситуаций. Требования к достижимой точности постоянно растут, что приводит к все более и более мелким сеткам. Локально улучшенные адаптации сетки введены для повышения точности с минимальным количеством дополнительных точек сетки. На рис. 1 приведен пример

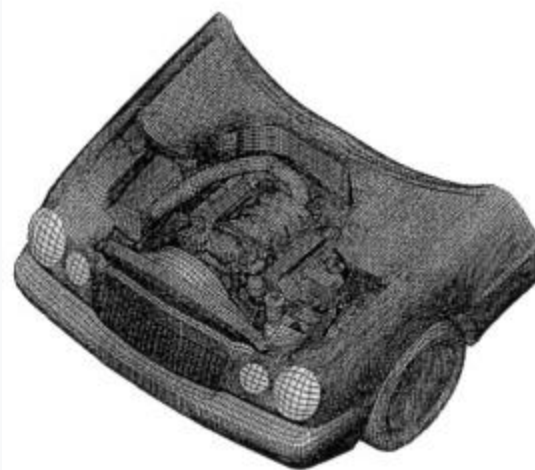


Рис. 1. Сетка для вычисления подкапотного пространства Mercedes-Benz E-Class.

В недавнем прошлом было исследовано несколько способов реализации конкретных алгоритмов AMG, и все еще продолжается быстрое развитие новых AMG и подобных AMG подходов и вариантов. Следовательно, пока нет единственного и лучшего подхода. Всякий раз, когда мы говорим о AMG в контексте конкретных числовых результатов мы фактически ссылаемся на код RAMG052 (подробно описанный в [51]), который является преемником исходного кода AMG1R5, упомянутого выше. Однако RAMG05 является совершенно новым и, в частности, включает в себя более эффективные и надежные стратегии интерполяции и укрупнения.

В этой статье дается обзор классической идеи AMG [48], некоторых улучшений и расширений и различных новых подходов. Основное внимание в разделах 2–6 уделяется фундаментальным идеям и аспектам, нацеленным на классы задач, для которых AMG разработан лучше всего, а именно на симметричные положительно определенные задачи (spd) того типа, которые возникают, например, из дискретизация скалярных эллиптических уравнений в частных производных второго порядка. Однако мы хотим отметить, что потенциальный диапазон применимости намного шире. В частности, AMG успешно применяется для различных несимметричных (например, конвекция-диффузии) и некоторых неопределенных задач. Кроме того, достигнут значительный прогресс в численной обработке систем ДУРЧП (главным образом, приложений Навье-Стокса и структурной механики). Тем не менее, серьезные исследования все еще продолжаются, и многое еще предстоит сделать, чтобы получить эффективность и надежность, сопоставимые со случаями скалярных приложений. В частности, в Разделе 7 устанавливаются ссылки на соответствующую литературу, где можно найти дополнительную информацию или более поздние подходы AMG. Хотя мы пытаемся охватить наиболее важные ссылки, этот список определенно не завершен в этой быстро развивающейся области исследований.

### **Алгебраический и геометрический многосеточные методы.**

В этой статье мы предполагаем, что у читателя есть базовые знания о геометрическом многосеточном методе. В частности, он должен быть знаком с основными принципами (сглаживание и коррекция на грубой сетке) и с рекурсивным определением многосеточных циклов. Потому, что для простоты мы ограничиваем наши основные соображения только двумя уровнями. Соответственно, всякий раз, когда мы говорим об эффективности,

конкретного подхода, мы неявно всегда предполагаем, что лежащий в основе двухуровневый метод рекурсивно расширен до реального многоуровневого метода, включающего лишь небольшое количество переменных (20 - 40, скажем) на самом грубом уровне. Что касается более подробной информации о геометрическом многосеточном методе, мы ссылаемся на [61] и обширный список ссылок, приведенных в нем.

Двухуровневый цикл AMG для решения (разреженной) s.p.d. системы уравнений

$A_h u^h = f^h \text{ или } \sum_{j \in \Omega^h} a_{ij}^h u_j^h = f_i^h \ (i \in \Omega^h)$	(1)
--	-----

формально определяется так же, как геометрический двухсеточный цикл на основе произведения Галеркина. Единственной разницей является то, что в контексте AMG  $\Omega^h$  это просто набор индексов, в то время как он соответствует сетке в геометрическом многосеточном методе. Соответственно, более грубый уровень,  $\Omega^H$ , просто соответствует (намного меньшему) набору индексов.

Если мы знаем, как отобразить H-векторы в h-векторы с помощью некоторого (полного ранга) оператора «интерполяции»  $I_H^h$ , оператор (s.p.d.) грубого уровня  $A_H$  определяется через

$$A_H := I_H^H A_h I_H^h \text{ где } I_H^H = (I_H^h)^T.$$

Затем выполняется один двухуровневый корректирующий шаг, то есть

$u_{new}^h = u_{old}^h + I_H^h e^H,$	(2)
--------------------------------------	-----

где поправка  $e^H$  является точным решением

$$A_H e^H = r^H \text{ или } \sum_{j \in \Omega^H} a_{ij}^H e_j^H = r_i^H \ (i \in \Omega^H)$$

с  $r^H = I_h^H(r_{old}^h)$  и  $r_{old}^h = f^h - A_h u_{old}^h$ . (Обратите внимание, что мы обычно используем букву  $u$  для численного решения и букву  $e$  для исправления или численной ошибки). Для соответствующих ошибок

$e^h = u_*^h - u^h$ , где  $u_*^h$  — обозначает точное решение (1). Также

$e_{new}^h = K_{h,H} e_{old}^h \text{ с } K_{h,H} := I_h - I_h^H A_H^{-1} I_h^H A_h,$	(3)
---	-----

будучи оператором коррекции грубой сетки ( $I_h$  обозначает тождественный оператор, единичный).

Напомним, что для любого сглаживающего оператора  $S_h$  для сглаживания — сходимость Галеркинских подходов легче всего исследовать с помощью w.r.t. Энергетическая норма,  $\|e^h\|_{A_h} = \sqrt{(A_h e^h, e^h)}$ . Предполагая  $\nu$  шагов релаксации, которые должны быть выполнены для (предварительного) сглаживания, следующее справедлив известный вариационный принцип (см., например, [51]),

$\ K_{h,H} S_h^\nu e^h\ _{A_h} = \underbrace{\min_{e^H}} \ S_h^\nu e^h - I_h^H e^H\ _{A_h}.$	(4)
--	-----

Как тривиальное следствие, сходимости двухуровневых циклов и, если она рекурсивно распространяется на любой число уровней, сходимость полных V-циклов всегда обеспечивается, как только метод релаксации сходится. Это верно для любой последовательности более грубых уровней и операторов интерполяции.

Что еще более важно (4) показывает, что скорость сходимости сильно зависит от эффективного взаимодействия между релаксацией и интерполяцией. Исходя из (4), мы хотим наметить основные концептуальные различия между геометрическим и алгебраическим многосеточным методом.

## Геометрический многосеточный метод.

В геометрическом многосеточном методе используются стратегии фиксированного укрупнения и обычно определяется интерполяция геометрически, как правило, с помощью линейной интерполяции. В зависимости от конкретной проблемы это обязательно предъявляет строгие требования к сглаживающим свойствам  $S_h$  (правая часть в (4) должна быть мала), а именно, что погрешность после релаксации изменяется геометрически гладко от точек сетки мелкого уровня до соседних точек грубого уровня. Другими словами, ошибка после релаксации должна быть геометрически гладкой относительно грубой сетки.

В качестве иллюстрации предположим, что более грубые уровни определяются стандартным геометрическим переходом  $h \rightarrow 2h$  огрубление в каждом пространственном направлении. Хорошо известно, что точечная релаксация геометрически сглаживает ошибку в каждом направлении, только если данная проблема по существу изотропна. В случае анизотропии проблемы, однако, сглаживание происходит только «в направлении сильных связей». В таких случаях всё больше требуются сложные сглаживатели, такие как чередующиеся линейные релаксации (ADI, Писмен Речфорд, алгоритм прогонки) или схемы типа ILU, для того чтобы все еще добиться хорошего взаимодействия между сглаживанием и интерполяцией и, таким образом, получить быструю многосеточную структурную сходимость.

В то время как построение «надежных сглаживателей» не является сложным в ситуациях 2D-модели, для 3D-приложений на сложных сетках их реализация становится довольно громоздкой. Например, надежным трехмерным аналогом релаксации по чередующейся линии является релаксация по чередующейся плоскости (например, 2D-сетка в каждой

плоскости), которую в сложных геометрических ситуациях становится очень сложно реализовать, если это вообще возможно. ILU сглаживатели, с другой стороны, теряют большую часть их сглаживающих способностей в общих 3D-ситуациях. Единственный способ ослабить требования к сглаживающим свойствам релаксации и при этом поддерживать эффективное взаимодействие релаксации и интерполяции - это использовать более сложные методы укрупнения. В геометрическом многосеточном методе шаги в этом направлении были, например, предприняты с использованием более чем одной более грубой сетки на каждом многосеточном уровне («множественное полуогрубление» [35,59,21,34]).

### **Алгебраический многосеточный метод**

В то время как, геометрический многосеточный метод по существу зависит от наличия надежных сглаживателей, AMG это противоположная точка зрения. Это предполагает простой процесс релаксации, который будет дан (как правило, простой релаксацией Гаусса – Зейделя), а затем пытается построить подходящую зависящую от оператора интерполяцию  $I_H^h$  (включая сам более грубый уровень). Согласно (4) эта конструкция должна быть такой, чтобы погрешность формы  $S_h^v e^h$  он достаточно хорошо представлен в диапазоне интерполяции,  $\mathcal{R}(I_H^h)$ . Чем лучше выполняется это условие, тем быстрее может быть сходимость. Обратите внимание, что здесь не важно, является ли релаксация сглаживающей ошибку в любом геометрическом смысле. Важно то, что ошибка после релаксации может быть охарактеризован алгебраически до такой степени, что она позволяет автоматически строить более грубые уровни и определяет интерполяции, которые локально адаптированы к свойствам данной релаксации. Эта



локальная адаптация является основной причиной гибкости AMG в адаптации его к проблеме и ее надежности в решении больших классов задач, несмотря на использование очень простых поточечных сглаживателей.

### Классическая amg аппроксимация.

В классическом AMG переменные грубого уровня мы рассматриваем как подмножество переменных первоначального уровня. То есть, мы предполагаем, что множество переменных 1-го уровня будет разбито на два непересекающихся подмножества,  $\Omega^h = C^h \cup F^h$ , с  $C^h$ , представляющим те переменные, которые должны содержаться на грубом уровне (C-переменные) и  $F^h$  - дополнительный набор (F-переменные). Учитывая такое C/F-расщепление, мы определяем  $\Omega^H = C^h$  и рассматриваем (полного ранга) интерполяции  $e^h = I_H^h e^H$  формы

$$e_i^h = (I_H^h e^H)_i = \begin{cases} e_i^H & \text{если } i \in C^h, \\ \sum_{k \in P_i^h} w_{ik}^h e_k^H & \text{если } i \in F^h, \end{cases} \quad (5)$$

где  $P_i^h \subset C^h$  называется множеством интерполяционных переменных. (По причинам разреженности  $A_H$ ,  $P_i^h$  должен быть достаточно малым подмножеством C-переменных «около»  $i$ .) Очевидно,  $\Re(I_H^h)$  сильно зависит как от конкретного выбора C-переменных, так и определения интерполяции. В данной ситуации можно легко представить себе «плохие» C/F-расщепления, которые просто не допускают никакой интерполяции, которая подходит в том смысле, который был изложен в предыдущем разделе. То есть конструкция конкретного C/F-расщепления и определение интерполяции являются тесно связанными процессами.

Конкретные алгоритмы, используемые на практике, в значительной степени эвристически мотивированы. В разделе 3.1 мы в основном обобщить основные идеи, как описано в [48] и некоторые модификации, введенные в [51]. В разделе 3.2, мы более подробно рассмотрим некоторые теоретические и практические аспекты в случае, если  $A_h$  содержит только неположительные вне-диагональные записи (М-матрица). Чтобы упростить запись, мы обычно опускаем индекс  $h$  в формулах, например, мы пишем  $S$ ;  $A$ ;  $K$  и  $e$  вместо  $S_h$ ,  $A_h$ ,  $K_{h,h}$  и  $e^h$ . Более того, вместо из (5), мы просто пишем

$e_i = \sum_{k \in P_i} w_{ik} e_k \quad (i \in F).$	(6)
--	-----

### 3.1 Основная идея.

Классический AMG использует простую релаксацию Гаусса – Зейделя для сглаживания. Из некоторых эвристических аргументов, Можно видеть, что ошибка  $e$ , полученная после нескольких шагов релаксации, характеризуется тем, что (масштабированный) остаток (невязка) в среднем для каждого  $i$  намного меньше самой ошибки  $|r_i| \ll a_{ii}|e_i|$ .

Это означает, что локально  $e_i$  может быть хорошо аппроксимировано

$e_i \approx - \left( \sum_{j \in N_i} a_{ij} e_j \right) / a_{ii} \quad (i \in \Omega),$	(7)
---	-----

где  $N_i = \{j \in \Omega: j \neq i; a_{ij} \neq 0\}$  обозначает окрестность любого  $i \in \Omega$ . Такая ошибка называется алгебраически гладкой. Согласно замечаниям в конце раздела 2, именно такая ошибка который должен быть хорошо представлен в  $\mathfrak{R}(I_H^h)$  То есть общая цель состоит в том, чтобы построить C/F-расщепления и определить множество интерполяционных переменных  $P_i \subset C$  ( $i \in F$ ) вместе с соответствующими весами  $w_{ik}$  такими что (6) дает разумное приближение

для каждого алгебраически гладкого вектора ошибки  $e$ . Очевидно, что очень «точная» интерполяция в этом смысле получается непосредственным использованием (7), что есть, выбрав  $P_i = N_i$  и  $w_{ik} = -a_{ik}/a_{ii}$ . Однако для этого потребуется выбрать C/F-расщепление, так что для каждого  $i \in F$  все его соседи содержатся в  $C$ . Хотя любой такой выбор можно предположить, но он дает прямой решатель, и этот подход не имеет реальной практической значимости, поскольку с точки зрения вычислительной работы и потребности в памяти, результирующий метод, как правило, будет чрезвычайно ресурсоёмок и приводит к потере разреженности оператора Галёркина даже если он будет рекурсивно расширен до иерархии уровней [51].

На практике мы хотим добиться быстрой сходимости с помощью небольших наборов интерполяционных переменных  $P_i$ , настолько малых насколько это возможно (для того, чтобы учесть быстрое огрубление и получить достаточно разреженный оператор Галеркина). Различные подходы были проверены на практике, которые не могут быть подробно описаны здесь. Далее мы просто дадим схему некоторых типичных подходов.

### 3.1.1. Прямая интерполяция.

Мы говорим о прямой интерполяции, если  $P_i \subseteq N_i$ . Такая интерполяция может быть немедленно получена из (7), если мы знаем, как аппроксимировать «неинтерполирующую часть» (то есть эту часть суммы в (7) который относится к  $j \in N_i \setminus P_i$ ) для алгебраически гладкой ошибки. Это приближение является наиболее критическим шагом в определении интерполяции. Например, для  $M$ -матриц  $A$  такое приближение можно получить, наблюдая, что алгебраически гладкая ошибка медленно меняется в направлении сильных (больших) связей. Особенно, чем более сильные связи

любой переменной  $i$  содержатся в  $P_i$ , тем лучше алгебраически гладкая ошибка удовлетворяет

$$\frac{1}{\sum_{k \in P_i} a_{ik}} \sum_{k \in P_i} a_{ik} e_k \approx \frac{1}{\sum_{j \in N_i} a_{ij}} \sum_{j \in N_i} a_{ij} e_j \quad (i \in \Omega)$$

Подставляя это в (7), получим интерполяцию (6) с положительными весами

$w_{ik} = -\alpha_i a_{ik}/a_{ii}$ где $\alpha_i = \frac{\sum_{j \in N_i} a_{ij}}{\sum_{\ell \in P_i} a_{i\ell}}$ .	(8)
---	-----

Практически это означает, что мы должны построить C/F-расщепления так, чтобы каждый  $i \in F$  имел достаточно большое количество сильно связанных C-соседей, которые затем принимаются в качестве набора интерполяционных переменные  $P_i$ . (См. Раздел 3.2 относительно некоторых важных дополнительных аспектов.)

В случае (скалярных) эллиптических ДУРЧП самые большие недиагональные элементы обычно отрицательны. Если там также являются положительные недиагональные записи, такой же процесс, как и раньше, может применяться при условии, что такие записи являются относительно небольшими: небольшие положительные связи можно просто игнорировать, просто рассматривая их как слабые. Однако ситуация становится менее очевидной, если  $A$  содержит большие положительные недиагональные элементы. Во многих таких случаях можно предположить, что алгебраически гладкая ошибка колеблется вдоль таких связей (например, в случае слабо диагонально доминирующих дополнительных матриц  $A$  [26,51]). Это можно использовать для обобщения приведенного выше подхода, например, путем подходящего разделения положительных и отрицательных связей, приводя к интерполяционным формулам, содержащим как положительные, так и

отрицательные веса. Соответствующий подход было предложено в [51] и может быть формально применен к произвольным s.p.d. матрицам. Однако результирующая интерполяция эвристически оправдана только до тех пор, пока для любого  $i$  эти компоненты ошибок  $e_k$  которые соответствуют большим положительным связям  $a_{ik} > 0$ , медленно меняются между собой (если  $a_{ik}$  очень мала, и в этом случае его влияние можно игнорировать).

На практике эти простые алгебраические подходы для построения интерполяции охватывают большой класс приложений. Однако пока нет лучшего способа автоматически построить интерполяцию, которая хороша для произвольной s.p.d. матрицы, по крайней мере, не просто рассматривая размер и знак коэффициентов.

Например, в случае конкретных одноэлементных дискретизаций высшего порядка или дискретизаций билинейными элементами на четырехсторонних сетках с большими пропорциями, получающиеся матрицы обычно содержат значимые положительные вне-диагональные элементы и далеки от того, чтобы быть слабо диагонально доминирующими.

В таких случаях показатели AMG могут быть только неоптимальными. Если это происходит, это часто помогает использовать некоторую информацию о происхождении этих положительных связей, а не полагаться только на информацию, непосредственно содержащуюся в матрице. Например, можно попытаться структурно упростить данную матрицу перед применением AMG (см., например, [41]). Альтернативные подходы, определяющие укрупнение и интерполяцию изложены в разделе 7.

***Более сложные интерполяции.***

Есть несколько способов улучшить прямую интерполяцию из предыдущего раздела. Чтобы обозначить некоторые возможности, предположим, что C/F-расщепление и для каждого  $i \in F$  задан набор (сильно связанных) интерполяционных переменных  $P_i \in N_i \cap C$ . Вместо немедленной аппроксимации неинтерполаторной части уравнения (7), как это сделано выше, можно сначала (приблизительно) устранить все сильно связанные пары  $e_j$  ( $j \notin P_i$ ) с помощью соответствующего  $j$ -го уравнения. Идеи, изложенные в предыдущем разделе, могут затем аналогичным образом применяться к полученному расширенному уравнению для  $e_i$ , что приводит к интерполяции с увеличенным набором интерполяционных переменных. Соответствующая интерполяция (называемая стандартной интерполяцией в [51]), как правило, значительно более устойчива на практике. В качестве альтернативы можно получить улучшенную интерполяцию, применив шаг F-релаксации (более подробно см. «Якоби-интерполяция» в разделе 5) для прямой или стандартной интерполяции.

В обоих подходах, по сравнению с прямой интерполяцией, «радиус» интерполяции увеличивается, что, в свою очередь, уменьшает разреженность результирующего оператора Галеркина. Однако интерполяционные веса, соответствующие переменным «далеко» от переменных, обычно намного меньше, чем самые большие. Поэтому перед вычислением оператора Галеркина следует усечь оператор интерполяции, игнорируя все маленькие записи (и масштабировать оставшиеся веса так, чтобы общая сумма оставалась неизменной). Обратите внимание, что из-за вариационного принципа усечение интерполяции является «безопасным процессом»; в худшем случае общая сходимость может замедлиться, но расходимость не может возникнуть. С другой стороны, усечение операторов Галеркина может быть опасным, так как это разрушает обоснованность вариационного принципа и, если его не применять с большой осторожностью, может даже привести к сильному расхождению на практике.

Помимо других незначительных различий, оригинальная интерполяция AMG, предложенная в [48] (и реализованная в коде AMG1R5), может рассматриваться как компромисс между прямой интерполяцией и стандартной интерполяцией, описанной ранее. Там была предпринята попытка заменить все сильно связанные пары ( $j \notin P_i$ ) в (7) средними, включающими только переменные в  $P_i$ . Однако, чтобы это было разумным, основываясь на определенных критериях, новые C-переменные должны были быть добавлены к заданному расщеплению апостериори «окончательный выбор C-точки» в [48]). Хотя этот подход работал довольно хорошо в тех случаях, которые рассматривались в [48], обычно слишком много дополнительных C-переменных требуется в геометрически сложных трехмерных ситуациях, вызывая недопустимо высокое заполнение в сторону более грубых уровней (см. Примеры в разделе 4.2). На практике стандартная интерполяция, описанная выше (в сочетании с разумным усечением), оказалась более устойчивой и часто значительно более эффективной.

Вышеуказанные улучшения интерполяции обычно приводят к более быстрой сходимости, но также в некоторой степени увеличивают вычислительную работу за цикл и требуемую память. Платит ли это в конечном итоге, зависит от конкретного приложения. Если память является проблемой (как это часто бывает в коммерческих средах), можно вместо этого захотеть упростить интерполяцию за счет уменьшения скорости сходимости. Один из способов добиться этого - разрешить интерполяцию из переменных, которые не находятся в непосредственной близости. Такая «дальняя» интерполяция [48], как правило, позволяет значительно ускорить грубое огрубление и резко увеличивает разреженность на более грубых уровнях. Подробнее о простом подходе, который был проверен на практике, см. [51] («агрессивное огрубление» и «многопроходная интерполяция»).

### 3.2. Случай M матриц.

На практике оказывается, что сходимость V-цикла AMG в значительной степени не зависит от размера проблемы. К сожалению, это невозможно доказать, основываясь только на алгебраических аргументах. Тем не менее, некоторые важные аспекты могут быть исследованы теоретически, в частности, не зависящие от матрицы двухуровневая сходимость может быть доказана для различных классов матриц, если интерполяция определена правильно. Здесь мы рассмотрим класс M-матриц. Обобщения на другие классы и соответствующие доказательства можно найти в [51].

### 3.2.1. Двухуровневые отображения.

Следующая теорема показывает, что прямая интерполяция на основе (8) обеспечивает независимую от матрицы двухуровневую сходимость, если для каждого  $i \in F$  связность, представленная в  $P_i$ , является фиксированным подмножеством полной связи.

**Теорема 1.** Пусть  $A$  симметричная слабо диагонально-доминантная M-матрица. С фиксированным  $0 < \tau \leq 1$  выберите C/F-расщепление так, чтобы; для каждого  $i \in F$ ; существует множество  $P_i \subseteq C \cap N_i$ , удовлетворяющее

$\sum_{k \in P_i}  a_{ik}  \geq \tau \sum_{j \in N_i}  a_{ij} $	(9)
---	-----

и определить интерполяцию согласно (8). Тогда двухуровневый метод; используя один шаг релаксации Гаусса – Зейделя для (пост-) сглаживания; сходится со скоростью, которая зависит только от  $\tau$ ,

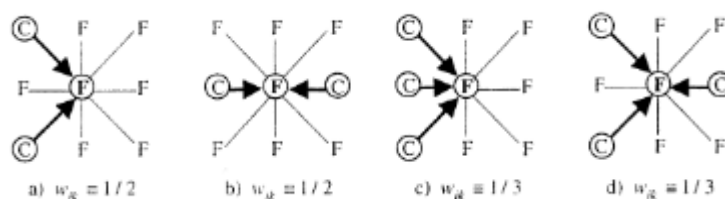




Рис. 2. Различные C / F-размещения и соответствующие интерполяционные формулы.

*но не от данной матрицы*

$$\|SK\|_A \leq \sqrt{1 - \tau/4}.$$

Приведенная выше теорема подтверждает, что для интерполяции важны сильные связи, в то время как использование слабых связей увеличит вычислительную работу, но вряд ли повлияет на сходимость. Чем более сильные связи используются для интерполяции, тем лучше может быть сходимость. Обратите внимание, что это косвенно означает, что огрубление будет происходить «в направлении плавности», что является основной причиной того факта, что сходимость AMG не зависит от анизотропии. Более того, интерполяцию AMG можно рассматривать как алгебраическое обобщение операторно-зависимой интерполяции, введенной в [1,20], что объясняет, почему производительность AMG не чувствительна к большим разрывным изменениям коэффициентов данной системы уравнений. Для иллюстрации, см. Раздел 4.1

С практической точки зрения вышеупомянутая оценка сходимости является оценкой наихудшего случая, поскольку данная проблема имеет некоторый геометрический фон (что обычно и происходит). Причина в том, что алгебраическое требование (9) не учитывает расположение интерполяционных C-точек относительно F-точек. Для иллюстрации рассмотрим 9-точечную дискретизацию оператора Пуассона.

$$\frac{1}{3h^2} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}_h.$$

Из геометрического многосеточного метода известно, что линейная интерполяция дает быструю сходимость. Алгебраическая интерполяция, однако, не может различить геометрически «хорошее» и «плохое» C / F-расщепление. Например, на рис. 2а и b мы используем один и тот же общий

вес для интерполяции, но второе расположение явно приведет к более лучшей сходимости. Точно так же расположение на рисунке 2d, хотя оно не дает точно второй порядок, будет намного лучше, чем на рисунке 2.c

Это показывает, что конкретное расположение C/F-расщепления будет иметь существенное влияние на качество интерполяции и, следовательно, на конечную сходимость. Чтобы строго обеспечить оптимальную интерполяцию, нам пришлось бы использовать геометрическое расположение (сильно связанных) точек друг относительно друга. На практике, однако, оказывается достаточным основать конструкцию C / F-расщепления на следующей дополнительной цели. Как правило, следует организовать C / F-разбиение так, чтобы набор C-переменных строил (приблизительно) максимальное множество с тем свойством, что C-переменные не сильно связаны друг с другом («максимально независимое множество») и что F- переменные «окружены» своими интерполяционными C-переменными. Это может быть обеспечено в достаточной степени, просто используя информацию о связности, содержащуюся в матрице (алгоритм см. В [48,51]). Обратите внимание, что сильная связь не обязательно должна быть через прямое соединение. На практике достижение этой цели существенно улучшит сходимость, даже если используются только небольшие наборы интерполяционных переменных.

### **Обобщение до многоуровневых циклов**

К сожалению, предположения об интерполяции в теореме 1 достаточны для равномерной двухуровневой, но не для равномерной сходимости V-цикла. Хотя, выбрав  $\tau \geq \frac{1}{2}$ , можно гарантировать, что все рекурсивно определенные операторы Галеркина остаются слабо диагонально-доминантными M-матрицами и, следовательно, формальное продолжение до полных V-циклов

является прямым, A-независимая сходимость V-циклов не может быть доказана. Причина - ограниченная точность чисто алгебраически определенной интерполяции, рассмотренная в предыдущем разделе. Мы вернемся к этой проблеме в Разделе 6, где мы рассмотрим сценарий наихудшего случая в контексте подходов AMG «агрегационного типа».

На практике, однако, можно наблюдать сходимость V-цикла, которая в значительной степени не зависит от размера задачи, если принять во внимание дополнительную цель предыдущего раздела. Кроме того, оказывается, что это не важно заставить матрицы грубого уровня точно переопределить M-матрицы. Напротив, такое требование накладывает слишком много ограничений на процесс укрупнения, в частности, на более низкие уровни, где размер операторов Галеркина может существенно возрасти.

В этом контексте мы хотим еще раз подчеркнуть, что для эффективной общей производительности скорость конвергенции является лишь одним аспектом. Не менее важным аспектом является сложность (разреженность) более грубых матриц уровней, создаваемых AMG (что напрямую влияет как на время выполнения, так и на общее требование к памяти). Надо рассматривать совместно в системе скорость сходимости и сложность оператора.

$$c_A = \sum_{\ell} |m_{\ell}|/|m_1|,$$

( $m_{\ell}$  обозначает число ненулевых элементов, содержащихся в матрице на уровне  $\ell$ ), ограничено независимо от размера A, имеем ли мы асимптотически оптимальную производительность. Типичное исполнение AMG в случае некоторых сложных проблем приведено в разделе 4.2.

## **AMG как предобуславливатель**

Чтобы повысить надежность геометрических многосеточных подходов, в последние годы стало очень популярно использовать многосеточные методы не в качестве отдельного решателя, а, скорее, комбинировать ее с методами ускорения, такими как сопряженный градиент, BI-CGSTAB или GMRES. Это развитие было обусловлено наблюдением, что зачастую не только проще, но и более эффективно использовать ускоренные многосеточные подходы, а не пытаться оптимизировать взаимодействие между различными многосеточными компонентами, чтобы улучшить сходимость отдельных многосеточных циклов.

Это оказалось аналогичным для AMG, который первоначально был разработан для использования в автономном режиме. Практический опыт ясно показал, что AMG также является очень хорошим предобуславливателем, намного лучше, чем стандартный (одноуровневый) метод ILU предобуславливания. Эвристически, основная причина заключается в том, что AMG, в отличие от любого одноуровневого предобуславливателя, стремится к эффективному сокращению компонентов ошибок, как на коротких, так и на больших расстояниях. Однако, хотя AMG пытается уловить все соответствующие влияния путем надлежащего огрубления и интерполяции, его интерполяция вряд ли когда-нибудь будет оптимальной. Вполне может случиться, что уменьшение ошибок значительно менее эффективно для некоторых очень специфических компонентов ошибок. Это может привести к тому, что несколько собственных значений итерационной матрицы AMG будут значительно меньше чем все остальные. Если это происходит, коэффициент сходимости AMG ограничивается медленной скоростью сходимости только нескольких исключительных компонентов ошибок, в то время как большинство компонентов ошибок сокращаются очень быстро. Ускорение, например, благодаря сопряженному градиенту, как правило, очень эффективно устраняет эти конкретные частоты. Альтернатива, а именно попытка предотвратить такие ситуации,

направляя больше усилий на построение интерполяции, обычно будет намного дороже. Даже тогда нет окончательной гарантии, что таких ситуаций можно избежать. (Отметим, что это происходит с помощью «надежных» геометрических многосеточных методов.)

### **Приложения и производительность.**

Разумеется, гибкость AMG и простота его использования имеют свою цену: подготовительная фаза AMG, в которой анализируется данная проблема, строятся грубые уровни и собираются все операторы, должна быть завершена до начала фактической фазы решения. Эти дополнительные издержки являются одной из причин того, что AMG обычно менее эффективен, чем геометрические многосеточные подходы (если применяются проблемы, для которых геометрический многосеточный может быть применен эффективно). Другая причина заключается в том, что, как правило, нельзя ожидать, что компоненты AMG будут «оптимальными». Они всегда строятся на основе компромиссов между численной работой и общей эффективностью. Тем не менее, если применять стандартные тестовые задачи эллиптического типа, вычислительные затраты фазы решения AMG (без учета стоимости установки), как правило, сопоставимы со стоимостью решения надежного геометрического многосеточного решателя [47].

Однако AMG не следует рассматривать как конкурента геометрическому многосеточному методу. Сильными сторонами AMG являются его надежность, его применимость в сложных геометрических ситуациях и его способность решать даже те проблемы, которые недоступны геометрическому многосеточному методу, в частности, проблемы с отсутствием геометрического или непрерывного фона. В таких случаях AMG следует рассматривать как альтернативу стандартным численным методам, таким как метод сопряженных градиентов, ускоренный типичными (одноуровневыми) предобуславливателями. В статье показываются

некоторые конкретные сравнения производительности в Разделе 4.2. Прежде, однако, демонстрируется гибкость AMG в настройке локального процесса его укрупнения в соответствии со сглаживающими свойствами релаксации с помощью простого, но характерного модельного уравнения.

#### 4.1 Модельная проблема для иллюстрации.

Рассмотрим модельное уравнение

$-(au_x)_x - (bu_y)_y + cu_{xy} = f(x, y),$	(12)
---	------

определенное на единичном квадрате с граничными условиями Дирихле. Мы предполагаем, что  $a = b = 1$  везде за исключением верхней левой четверти единичной квадратной области (где  $b = 10^3$ ) и нижней правой четверти (где  $a = 10^3$ ). Коэффициент  $c$  равен нулю, за исключением верхней правой четверти, где мы устанавливаем  $c = 2$ .

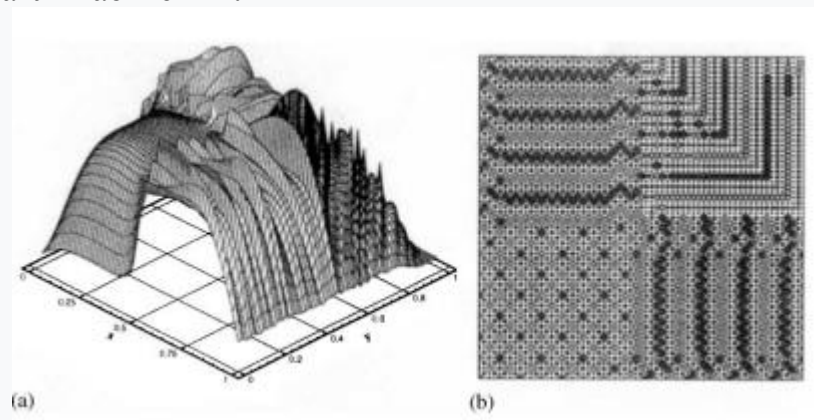


Рис. 3. (а) «Плавная» ошибка в случае задачи (12). (б) 1-й и три последовательных уровня, созданные стандартным алгоритмом огрубления AMG.

Полученная дискретная система изотропна в нижней левой четверти единичного квадрата, но сильно анизотропный в остальных кварталах (всего четыре квартала). На рис. 3, а показано, как выглядит «гладкая» ошибка на

самом лучшем уровне сетки после применения нескольких шагов релаксации Гаусса – Зейделя к однородной задаче, начиная со случайной функции. Различные анизотропии, а также разрывы по всей интерфейсные линии четко отражены на рисунке.

Эвристически ясно, что такую ошибку можно эффективно уменьшить только с помощью более грубой сетки, если эта сетка получена путем существенного огрубления в направлениях, в которых ошибка действительно изменяется гладко в геометрическом смысле, и если интерполяция правильно обрабатывает разрывы. Действительно, можно увидеть в разделе 3.2, это именно то, что делает AMG. Во-первых, операторная интерполяция обеспечивает правильное лечение разрывов. Во-вторых, сокращение AMG в направлении сильной связи, то есть в направлении плавности. На рис. 3, б изображены одна и три последовательные сетки, созданные с использованием стандартного укрупнения AMG и интерполяции. Самые маленькие точки отмечают точки сетки, которые содержатся только в 1-й сетке, квадраты отмечают те точки, которые также содержатся на более грубых уровнях (чем больше квадрат, тем дольше соответствующая точка сетки остается в процессе укрупнения). На рисунке видно, что укрупнение является равномерным в нижней левой четверти, где проблема является изотропной. В других кварталах AMG приспосабливается к различным анизотропиям, локально огибая их в правильном направлении. Например, в нижней правой четверти укрупнение происходит только в направлении X. Так как AMG принимает только сильные связи при укрупнении и поскольку все соединения в направлении y слабы, отдельные линии укрупнены независимо друг от друга. Следовательно, огрубление соседние x-линии не «синхронизированы»; это на самом деле вопрос «совпадения», когда грубое



начинается в каждой строке. Это должно наблюдаться при интерпретации картины укрупнения в верхнем правая четверть: в пределах каждой диагональной линии огрубление происходит по существу в направлении этой линии.

#### 4.2. Промышленные приложения.

Для демонстрации эффективности AMG мы рассмотрим некоторые сложные проблемы такого типа, которые обычно решаются в двух коммерческих кодах, предназначенных для моделирования нефтяных пластов и для расчета гидродинамики соответственно. В обоих кодах числовое ядро требует быстрого решения скалярного эллиптического уравнения. Хотя при моделировании нефтяного пласта геометрии, как правило, довольно просты, но лежащие в основе проблемы имеют сильно анизотропные и разрывные коэффициенты (скачки на несколько порядков почти случайным образом), в вычислительной гидродинамике эти задачи подобны пуассоновским, но определены на очень сложных, неструктурированных сетках. Подробнее об этих кодах см. [51].



Рис. 4. Охлаждающая рубашка четырехцилиндрового двигателя.

Рассматриваются следующие тестовые случаи: 3

(1) Водохранилище. Базовый резервуар соответствует простой геометрии расчётной области, дискретизированному сеткой, состоящей из 1,16



миллионов ячеек. Изменение абсолютной проницаемости приводит к разрыву изменение коэффициентов на четыре порядка.

(2) Охлаждающая рубашка. Расчет течения через рубашку охлаждения четырехцилиндрового двигателя. Основная сетка состоит из 100 000 тетраэдрических ячеек (см. Рис. 4).

(3) Угольная печь. Расчет течения внутри модели угольной печи. Основная сетка состоит из 330 000 шестигранников и нескольких тысяч пентаэдров, в том числе многих локально адаптированных сеточных ячеек.

(4) underhood. Вычисление недоделанного обтекания модели Mercedes-Benz E-класса. Сетка является очень сложным и состоит из 910 000 ячеек (см. рис. 1).

(5) E-класс. Вычисление внешнего течения над моделью E-класса Mercedes-Benz (см. Рис. 5). Исходная сетка состоит из 10 миллионов клеток. Из-за ограничений памяти наш тест выполняется всего на двух сетках уменьшенных размеров, состоящие из 2,23 и 2,82 миллиона ячеек соответственно. (Обратите внимание, что базовая сетка также включает в себя все детали моделирования предыдущего случая с underhood.)

Требование к памяти - главная проблема для любого коммерческого поставщика программного обеспечения. Промышленные пользователи коммерческих кодов всегда сводят моделирование к пределам возможностей своих компьютеров, нехватка памяти серьёзна. По этим причинам в коммерческой среде AMG подходы с низким объемом потребления оперативной памяти представляют первостепенный интерес, даже если сокращение потребности в памяти происходит за счет (ограниченного) увеличения общего вычислительного времени. По сравнению со стандартными одноуровневыми решателями, накладные расходы памяти в

несколько десятков процентов, безусловно, приемлемы. В любом случае, однако, оператор сложности  $s_A$  (11) не должна быть больше 2,0 скажем. Поэтому в следующих тестовых прогонах мы используем стратегия агрессивного огрубления (см. раздел 3.1) и, чтобы восполнить результирующее сокращение скорости сходимости, используйте AMG в качестве предварительного кондиционера, а не в качестве отдельного решателя.

На рис. 6 показаны истории сходимости V-цикла AMG для каждого из указанных выше случаев, основанные на коде RAMG05 который упомянут во введении. Результаты отражают общее впечатление, что конвергенция AMG зависит в ограниченной степени от типа используемых элементов, а также от типа проблемы, но вряд ли от размера проблемы. В частности, три сетки Mercedes сопоставимы по своему типу, но их размер варьируется более чем в три раза. Конвергенция, очевидно, изменяется лишь незначительно.

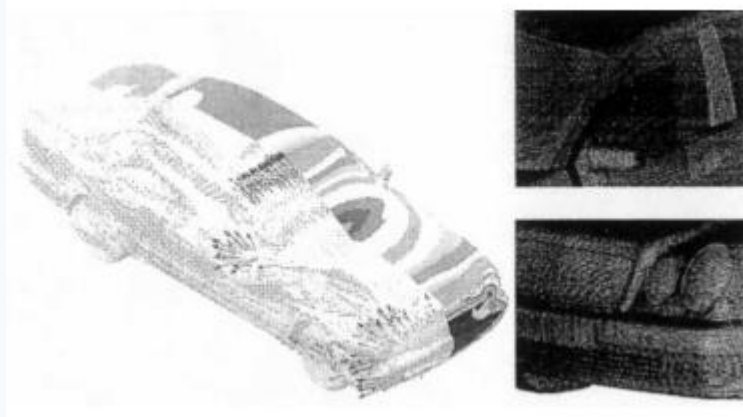


Рисунок 5. Модель Mercedes Benz E-класса.

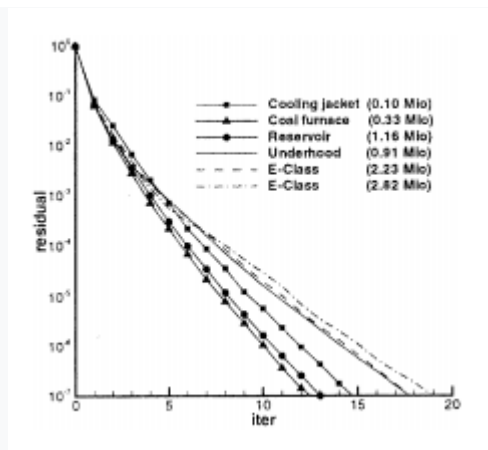


Рис. 6. RAMG05 истории сходимости для различных задач.

problem	RAMG05/cg			ILU(0)/cg		AMG1R5
	time	iter	$c_A$	time	iter	$c_A$
jacket	12.3	21	1.44	218.2	926	5.35
furnace	45.7	18	1.47	292.8	286	7.06
reservoir	165.0	18	1.41	2707.0	720	7.66
underhood	172.9	25	1.43	1364.0	461	5.64
eclass (2.23)	438.8	25	1.46	8282.0	1151	6.24

Рис. 7. Производительность RAMG05 по сравнению с  $ILU(0) = cg$ .

AMG уменьшает количество итераций до 46 раз. С точки зрения времени выполнения, AMG до 19 раз быстрее. На рисунке также показано, что промышленные требования с точки зрения памяти, упомянутые ранее, полностью выполнены. На самом деле, A-сложность (11) очень удовлетворительная для всех случаев, а именно  $c_A \approx 1,45$ . Для сравнения, последний столбец на рисунке показывает недопустимо высокие значения сложности предшественника RAMG05, AMG1R5. Как уже упоминалось в

разделе 3.1, AMG1R5 обычно выполняется довольно хорошо в случае двумерных задач. Однако в сложных трехмерных случаях, как рассматриваются здесь, результаты четко продемонстрировали одно из преимуществ различных методов укрупнения и интерполяции которые используется в RAMG05 (рис. 7). (Для получения дополнительной информации о различиях в двух кодах мы отсылаем в [51]).

## 5. Алгебраический многосеточный метод базирующийся на F-релаксации.

В этом разделе мы рассмотрим совершенно другой подход [28, 51], который можно использовать для правой части (4), чтобы сделать её маленькой. Для описания мы предполагаем, что векторы и матрицы упорядочены так, что, при данном C / F-разбиении, система уравнений (1) может быть записана в блочном виде

$A_h u = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} u_F \\ u_C \end{pmatrix} = \begin{pmatrix} f_F \\ f_C \end{pmatrix} = f.$	13)
---	-----

Соответственно, оператор интерполяции переписывается как  $I_H^h = (I_{FC}, I_{CC})^T$ , с  $I_{CC}$  являющимся единичным оператором. Вместо  $e^h = I_H^h e^H$  мы просто пишем  $e_F = I_{FC} e_C$ .

### 5.1. Базовая идея.

Упомянутый выше подход основан на том факте, что субматрица  $A_{FF}$  очень хорошо обусловлена, если мы просто выбираем C / F-расщепление соответственно. Например, для всех проблем, которые мы здесь имеем в виду, мы можем легко заставить  $A_{FF}$  наилучшим образом доминировать

$a_{ii} - \sum_{j \in F, j \neq i}  a_{ij}  \geq \delta a_{ii} \quad (i \in F)$	(14)
---	------

по диагонали с некоторым фиксированным  $\delta > 0$ . Предполагая, что это имеет место в следующем, мы можем эффективно аппроксимировать решение F-уравнений (с замороженным  $e_C$ ),

$A_{FF}e_F + A_{FC}e_C = 0$	(15)
-----------------------------	------

например, путем релаксации (в дальнейшем называемой F-релаксацией). Используя это в качестве основы для определения как сглаживания, так и интерполяции, мы можем заставить правую часть (4) стать малой, как нам хочется. Чтобы быть более точным, при любом  $e_C$  интерполяция определяется с помощью шагов  $\mu$ F-релаксации, чтобы приблизительно решить (15). Чтобы сохранить результирующий оператор как можно более «локальным», мы рассматриваем только релаксацию Якоби (ниже мы будем называть это интерполяцией Якоби). То есть мы итеративно определяем последовательность операторов,

$I_{FC}^{(\mu)} = P_{FF}I_{FC}^{(\mu-1)} - D_{FF}^{-1}A_{FC} \text{ где } P_{FF} = I_{FF} - D_{FF}^{-1}A_{FF},$	(16)
---	------

начиная с некоторого разумного оператора интерполяции первого предположения,  $I_{FC}^{(0)}$ . Из-за (14) мы имеем быструю сходимость  $(I_H^h)^{(\mu)}e_C \rightarrow \hat{e}(\mu \rightarrow \infty)$  со скоростью, которая зависит только от  $\delta$ . Здесь  $\hat{e} := (\hat{e}F, e_C)^T$  где  $\hat{e}F := -A_{FF}^{-1}A_{FC}e_C$  обозначает решение уравнения (15).

Аналогичным образом мы также используем F-релаксацию для сглаживания (далее называемую F-сглаживанием ниже). То есть мы определяем один шаг сглаживания с помощью  $u \rightarrow \hat{u}$ , где

$Q_{FF}\bar{u}_F + (A_{FF} - Q_{FF})u_F + A_{FC}u_C = f_F, \quad \bar{u}_C = u_C.$	(17)
--	------

В отличие от интерполяции, здесь мы обычно используем релаксацию Гаусса – Зейделя, то есть  $Q_{FF}$  - нижняя треугольная часть  $A_{FF}$  (включая диагональ). Соответствующий оператор сглаживания легко увидеть

$$S_h^v e = \begin{pmatrix} S_{FF}^v(e_F - \hat{e}_F) + \hat{e}_F \\ e_C \end{pmatrix} \quad \text{where } S_{FF} = I_{FF} - Q_{FF}^{-1} A_{FF}. \quad (18)$$

Как и в случае интерполяции, для любого заданного  $e = (e_F, e_C)^T$  мы имеем быструю сходимость  $S_h^v e \rightarrow \hat{e} (v \rightarrow \infty)$ .

## 5.2. Двухуровневая сходимость.

Для различных классов матриц можно показать, что F-сглаживание и якоби-интерполяция могут использоваться для получения независимой от матрицы сходимости на двух уровнях, если интерполяция первого предположения,  $I_{FC}^{(0)}$ , выбрана правильно. Более того, двухуровневая сходимость становится сколь угодно быстрой, если  $v, \mu$  выбираются достаточно большими. В качестве примера снова рассмотрим класс M-матриц (см. Теорему 1).

**Теорема 2.** Пусть  $A$  - симметричная слабо диагональная M-матрица. Определите интерполяцию путем аппликации и шагов  $\mu \geq 0$  Якоби F-релаксации, используя интерполяцию, как определено в теореме 1, как первое предположение (с фиксированным значением  $0 < \tau \leq 1$ ). Затем, если для (пред-) сглаживания используются шаги  $v \geq 1$  Gauss – Seidel F-релаксации, выполняется следующая двухуровневая оценка сходимости,

$$\|KS^v\|_A \leq \|S_{FF}\|_{A_{FF}}^v + \hat{\tau} \|P_{FF}\|_{A_{FF}}^\mu,$$

где  $\|S_{FF}\|_{A_{FF}} < 1$  и  $\|P_{FF}\|_{A_{FF}} < 1$ , и обе нормы, а также  $\hat{\tau}$  зависят только  $\tau$ , но не зависят от матрицы  $A$ .

В этой теореме мы использовали интерполяцию из теоремы 1 в качестве первого предположения. В частности, мы предполагаем, что C/F-расщепление удовлетворяет (9), что легко увидеть, чтобы обеспечить сильное диагональное доминирование (14)  $\delta = \tau$ . Хотя можно подумать о различных других способах определения интерполяции первого предположения, мы хотим отметить, что правильный выбор интерполяции первого

предположения важен для получения двухуровневой сходимости, независимой от матрицы (например, этого недостаточно просто выбрать  $I_{FC}^{(0)} = 0$ ). В общем случае интерполяция первого предположения должна быть такой, чтобы соответствующий ей оператор Галеркина,  $A_H^{(0)}$ , спектрально эквивалентен дополнению Шура,  $A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}$ , для всех матриц в рассматриваемом классе. Более подробную информацию и обобщения приведенной выше теоремы, а также доказательства см. [51].

Обратите внимание, что подход AMG, обсуждаемый здесь, на самом деле не в духе стандартных многосеточных схем, в обычном смысле ничего не используется. На самом деле, роль F-сглаживания заключается в том, чтобы просто сгладить ошибку всей системы. Это, вместе с Якоби-интерполяцией, является подходом «грубой силы» для создания  $\|S^v e - I_H^h e_C\|_A$  малой для любых  $e = (e_F, e_C)^T$ .

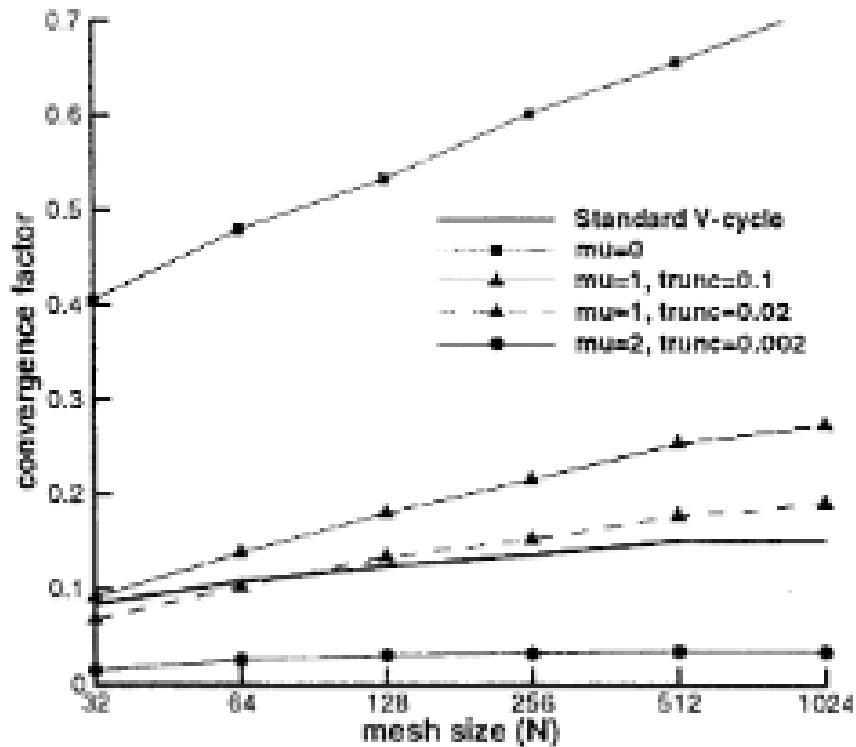


Рисунок 8. История сходимости многосеточного метода базирующегося на F-релаксации.

### 5.3. Практические замечания.

Тот факт, что AMG можно заставить сходиться так быстро, как мы хотим, на практике имеет мало значения. Каждый шаг F-релаксации, применяемый к интерполяции, увеличивает ее «радиус» на единицу дополнительный слой связей, вызывающий увеличение заполнения у оператора Галеркина. Полученный выигрыш в скорости сходимости, как правило, более чем съедается соответствующим увеличением матрицы сложности в сторону более грубых уровней. Следовательно, основной проблемой является компромисс между сходимостью и числовой работой (который напрямую связан с требованиями к памяти). Обратите внимание, что это в некотором смысле, прямо противоположно геометрическому многосеточному методу, где числовая работа за цикл известна, и управляема, но сходимость не может быть удовлетворительной.

Для практической реализации интерполяции Якоби важно соблюдать несколько вещей. Первый, большинство новых записей, вводимых каждым дополнительным шагом релаксации, будут относительно небольшими и могут быть обрезаны (до вычисления оператора Галеркина) без серьезного ущерба для сходимости (см. также раздел 3.1). Во-вторых, обычно нет необходимости выполнять F-релаксацию с полной матрицей  $A_{FF}$ . Вместо этого можно вполне игнорировать все те записи  $A_{FF}$ , которые относительно малы (их нужно добавить к диагонали, а также сохранить суммы строк интерполяции). Наконец, мы хотим заметить, что, хотя теорема 2 утверждает быструю сходимость, только если на практике достаточно велика,  $\mu < 2$  вряд ли когда-либо потребуется (по крайней мере, если  $\delta$  не слишком мал).



$-\left((1 + \sin(x + y))u_x\right)_x - (e^{x+y}u_y)_y = f(x, y),$	(19)
--	------

На рис. 8 показаны некоторые коэффициенты сходимости V-цикла как функция размера ячейки для модельного уравнения, дискретизированного на единичном квадрате с однородным размером ячейки  $h = 1/N$ . Сначала мы наблюдаем быструю независимую от  $h$  сходимость «стандартного» V-цикла AMG (что соответствует подходу, описанному в разделе 3, с использованием одного полного шага релаксации Гаусса – Зейделя как для предварительного, так и для последующего сглаживания). Конвергенция резко падает и становится сильно  $h$ -зависимым, если мы просто заменим каждый полный шаг сглаживания с помощью двух шагов F-сглаживания и оставьте интерполяцию без изменений (выбор  $\mu = 0$ ). Этого следует ожидать так как определение интерполяции в классическом AMG основано на предположении, что ошибка после релаксации алгебраически гладкая (см. раздел 3.1). Это, очевидно, не верно, если только частичное релаксация, такая как F-релаксация, выполняется. Однако, если мы используем только один шаг F-релаксации Якоби для улучшения интерполяции ( $\mu = 1$ ), сходимость становится сравнимой со сходимостью стандартного цикла. Результаты показаны с использованием двух различных параметров усечения, 0,1 и 0,02, соответственно. Наконец, выбор  $\mu = 2$  (и четыре шага частичной релаксации для сглаживания, а не два) дает сходимость примерно в два раза быстрее, чем у стандартного цикла.

Отметим, что если в этом примере учтены вычислительные затраты времени и памяти, стандартный V-цикл более эффективен, чем другие. В частности, цикл, использующий  $\mu = 2$  существенно уступает, в основном из-за значительно более высокой стоимости установки. Это типично для приложений, для которых алгебраически гладкие ошибки, в смысле раздела

3.1, могут быть охарактеризованы достаточно хорошо. Эвристическая причина в том, что тогда, использование полных шагов сглаживания, относительно простой интерполяции типа, описанного в разделе 3.1, обычно достаточны для того, чтобы алгебраической аппроксимацией сгладить ошибки и получить быструю сходимость. Это больше не верно, если используется простое F-сглаживание и, как правило, необходимо приложить дополнительные усилия для «улучшения» интерполяции с помощью F-релаксации для того, чтобы справиться со всеми теми компонентами ошибок, на которые не влияет простое F-сглаживание. (В частности, отметим, что все компоненты ошибок вида  $\hat{e}$  не уменьшаются совсем за счет F-сглаживания.) В целом, однако, когда характеристика алгебраически гладких ошибок менее прямолинейна, использование F-релаксации дает возможность улучшить сходимость. Дальнейшие численные эксперименты, использующие F-сглаживание и Якоби-интерполяцию, можно найти в [28,51]. F-релаксация это частный случай «совместимой релаксации», который в более общем контексте рассматривается в [11].

## **6. Алгебраический многосеточный метод агрегативного типа.**

В предыдущих разделах мы рассматривали все более сложные интерполяционные подходы. В этом разделе мы вернемся и рассмотрим самый простой случай, когда каждая F-переменная интерполируется только из одной единственной C-переменной. Мы уже указывали в разделе 3.2, что использование таких «односторонних» интерполяций не рекомендуется. Фактически, одна из важных целей дополнительной цели, представленной в разделе 3.2, заключалась в том, чтобы просто избежать экстремальных

интерполяций. С другой стороны, полученный метод настолько прост в реализации, что, тем не менее, привлек к себе некоторое внимание. Мы опишем основные проблемы с этим подходом в Разделе 6.2 и суммируем три возможности улучшения в Разделах 6.3–6.5. Поскольку мы просто хотим выделить основные идеи, мы ограничиваем наши мотивации очень простыми, но характерными (Пуассоновскими) задачами.

### 6.1. Базовая аппроксимация.

Рассмотрим  $C / F$ -расщепления и интерполяции (6) где для каждого  $i \in F$ ;  $w_{ik} = 1$  только для одного конкретного  $k \in C$  и нуля в противном случае. Следовательно, общее количество переменных можно подразделить на «агрегаты»  $I_k$  ( $k \in C$ ), где  $I_k$  содержит (кроме самого себя) все индексы, соответствующие  $F$ -переменным, которые интерполируются из переменной  $k$  (см. Рис. 9). С этим обозначением вычисление оператора Галеркина становится очень просто. Это легко

$$I_h^H A_h I_H^h = (a_{kl}^H) \text{ где } a_{kl}^H = \sum_{i \in I_k} \sum_{j \in I_l} a_{ij}^h \quad (k, l \in C), \quad (20)$$

то есть коэффициент  $a_{kl}^H$  представляет собой просто сумму всех перекрестных связей между  $I_k$  и  $I_l$ . Очевидно, что в отношении коэффициентов  $a_{kl}^H$  особая роль переменных  $k$  и  $l$  (как  $C$ -переменных) не отличается от других переменных; оператор Галеркина просто зависит от определения агрегатов. Следовательно, мы могли бы также связать каждый агрегат  $I_k$  с некоторой «новой» переменной грубого уровня, которая не имеет прямого отношения к  $C$ -переменной  $k$ . Приведенная выше интерполяция

представляет собой не что иное, как кусочно-постоянную интерполяцию от этих новых переменных грубого уровня к ассоциированным агрегатам.

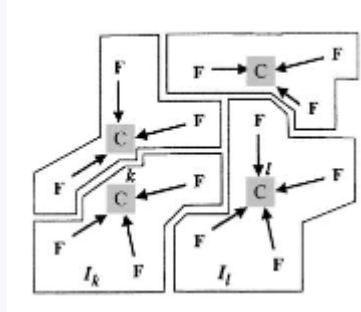


Рис. 9. Подразделение переменных мелкого уровня на агрегаты. Стрелки показывают, из какой С-переменной F-переменная интерполируется.

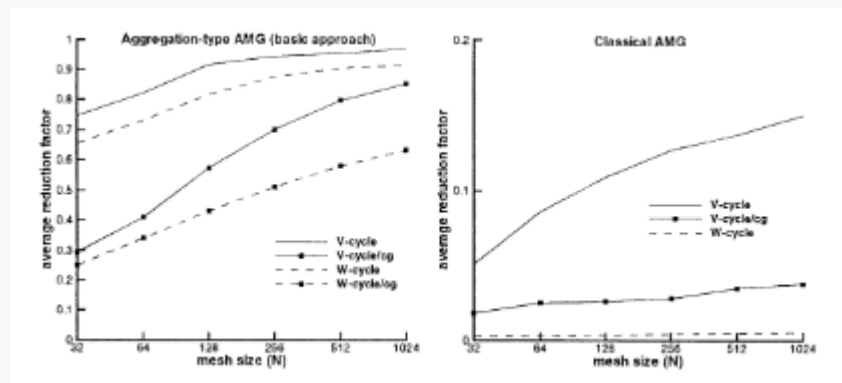


Рис. 10. Сходимость (а) AMG агрегационного типа, (б) классического AMG

Первоначально такие подходы как агрегативный алгебраический многосеточный метод [52,53,9] были разработаны другим путем: огрубление определяется путем построения агрегатов (а не построения С / F-расщепление), переменная нового грубого уровня ассоциируется с каждым агрегатом, а интерполяция  $I_H^h$  определена как кусочно-постоянная. (То есть набор переменных грубого уровня, как правило, не рассматривается как подмножество переменных подробного уровня.) Ясно, что для того, чтобы данное подразделение на агрегаты было разумным, все переменные в одном агрегате должны сильно зависеть друг от друга. В противном случае кусочно-постоянная интерполяция не имеет реального смысла.

Как и ожидалось, немедленная реализация этого простого подхода к укрупнению и интерполяции приводит к очень неэффективным решающим алгоритмам, даже если они используются только в качестве предобуславливателей. На рис. 10а показана типичная сходимость как V-, так и W-циклов, используемых, как автономные и как предобуславливатели при решении модельного уравнения (19). Сходимость действительно очень медленная и демонстрирует сильную  $h$ -зависимость. Для сравнения, гораздо лучшая производительность классического AMG изображена на рис. 10б.

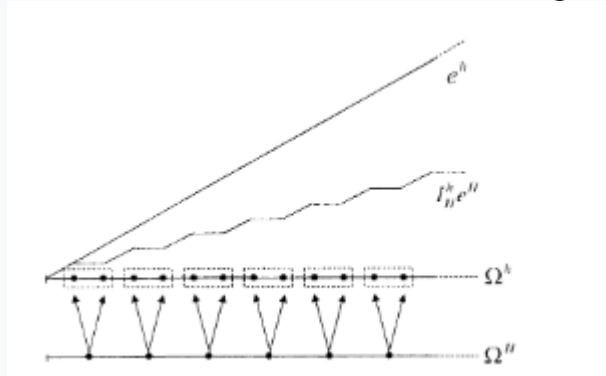


Рис. 11. Оптимальное приближение  $I_h^h e^H$  к  $e^h$  в энергетической норме.

## 6.2. Причина медленной сходимости.

Основная причина этой неудовлетворительной сходимости состоит в том, что кусочно-постоянная интерполяция не имеет возможности аппроксимировать значения гладкой ошибки, если аппроксимация основана на энергетической норме (см. (4)). На самом деле порядок приближения становится нулевым.

Чтобы проиллюстрировать это, рассмотрим самый простой случай, когда  $A_h$  получается из дискретизации  $-\frac{d^2 u}{dx^2}$  на единичном одномерном интервале с размером ячейки  $h$ , т.е. строки  $A_h$  соответствуют разностному шаблону

$$\frac{1}{h^2}[-1 \quad 2 \quad -1]_h$$

с граничными условиями Дирихле. Пусть  $e^h$  - ошибка, удовлетворяющая однородным граничным условиям. Согласно вариационному принципу, соответствующая двухуровневая коррекция  $I_H^h e^H$ , оптимально в том смысле, что это минимизирует  $\|e^h - I_H^h e^H\|_{A_h}$  w.r.t. все возможные исправления в  $\mathfrak{R}(I_H^h)$ . (В этот момент, конкретный выбор  $I_H^h$  не имеет отношения.) Это означает, что  $I_H^h e^H$  минимизирует

$$\|v^h\|_{A_h}^2 = (A_h v^h \cdot v^h) = \frac{1}{2h^2} \sum_{i,j} (v_i^h - v_j^h)^2 + \sum_i s_i (v_i^h)^2, \quad (21)$$

где  $v^h = e^h - I_H^h e^H$  и  $s_i = \sum_j a_{ij}^h$ . (Штрих означает, что суммирование происходит только по соседним переменным  $i$  и  $j$ .) Это, в свою очередь, означает, что вдали от границы (где мы имеем  $s_i \equiv 0$ ), Евклидова норма наклона  $v^h$  минимальна. На самой границе  $s_i \neq 0$ , а  $v^h$  равно нулю.

Предполагая теперь, что агрегаты будут построены путем объединения пар соседних переменных, результат этой минимизации проиллюстрирован на рис. 11 (см. также [9,10]). Здесь мы рассмотрим гладкую ошибку  $e^h$  в окрестности левой границы единичного интервала. На каждом агрегате интерполяция постоянная и наклон  $I_H^h e^H$  обязательно исчезает. На оставшихся интервалах евклидова норма наклона  $v^h$  становится минимальным, если наклон  $I_H^h e^H$  равно  $e^h$ . Следовательно,  $I_H^h e^H$  имеет, в среднем только половина склона  $e^h$  (независимо от  $h$ ). То есть результирующее приближение примерно в 0.5 раза меньше по сравнению с наилучшим приближением в евклидовом смысле. (Обратите внимание, что последующее сглаживание «покачивания», но не улучшает качество исправление.) Соответственно, оператор Галеркина, который легко

вычисляется, оказывается слишком большим в два раза больше по сравнению с «естественной»  $2h$ -дискретизацией  $-\frac{d^2u}{dx^2}$ .

Если та же самая стратегия теперь используется рекурсивно для введения более грубых и более грубых уровней, выше аргументы переносятся на каждый из промежуточных уровней и, в частности, на каждую более грубую сетку. Оператор Галеркина на новом сеточном уровне больше в 2 раза, по сравнению с предыдущим оператором на предыдущем сеточном уровне. Простой рекурсивный аргумент показывает, что ошибки накапливаются от сетки к сетке, а коэффициент сходимости асимптотического V-цикла не может ожидать быть лучше, чем  $1 - 2^{-m}$ , где  $m$  обозначает количество более грубых уровней. Это Сходимость V-цикла сильно зависит от  $h$ .

### *6.3. Улучшение путем повторного масштабирования операторов Галеркина.*

Тот факт, что кусочно-постоянная интерполяция производит плохо масштабируемые компоненты AMG, было основой для улучшения, введенного в [9] метода. В этой статье показано, что сходимость можно существенно улучшить, просто умножив поправки  $I_H^h e^H$  на некоторый подходящий коэффициент  $\alpha > 1$  («Сверх-коррекции»). Это эквивалентно повторному масштабированию оператора Галеркина на  $1/\alpha$

$$I_h^H A_h I_H^h \rightarrow \frac{1}{\alpha} I_h^H A_h I_H^h$$

и оставляя все остальное без изменений.

В случае простого модельного уравнения  $-\frac{d^2u}{dx^2}$ , рассмотренного в предыдущем разделе,  $\alpha = 2$  будет оптимальным выбором. Тем не менее, основные аргументы переносятся на уравнение Пуассона в 2D и 3D,

предполагая, что равномерная сетка и агрегаты должны быть построены из  $2 \times 2$  и  $2 \times 2 \times 2$  блоков соседних переменные соответственно. В случае более общих проблем и  $\alpha=2$  или различных сеток, оптимальный вес больше не  $\alpha=2$ . Тем не менее, в [9] было продемонстрировано, что слегка уменьшенное значение  $\alpha = 1.8$  (чтобы уменьшить риск «перерегулирования») дает существенно улучшенную конвергенцию V-цикла для различных типов проблем, если многосеточный цикл используется в качестве предобуславливателя и если число грубых уровней сохраняется фиксированным (в [9] всегда используются четыре уровня). Сглаживание выполняется симметричными релаксациями по Гауссу-Зейделю.

Сравнение рис. 10а и 12а показывает улучшение сходимости при повторном масштабировании:  $\alpha=1.8$  применяется к модельному уравнению (19). (В отличие от [9], мы здесь не ограничили количество более грубых уровней.) Рис. 12а показывает, что действительно существует риск «перерегулирования»: для более крупных ячеек V-цикл начинает расходиться. (Обратите внимание, что вышеупомянутое повторное масштабирование разрушает действительность вариационного принципа, и процесс итерации может значительно отличаться.) Использование V-цикла в качестве предобуславливателя устраняет проблему.

Мы хотим отметить, что приведенное выше сравнение показывает только тенденцию улучшений, обусловленных до масштабирования, конкретное усиление зависит от того, как точно выбраны агрегаты (что не является оптимальным здесь и, безусловно, может быть улучшено в некоторой степени). В любом случае выигрыш в сходимости, надежность и эффективность этого (очень простого и легко программируемого) подхода несколько ограниченный, одна причина состоит в том, что хорошее значение



параметра  $\alpha$  зависит от различных аспектов, таких как конкретная проблема, тип сетки и, в частности, тип и размер агрегатов. Например, если агрегаты состоят из трех соседних переменных (а не двух) в каждом пространственном направлении, те же аргументы, что и в предыдущем разделе, показывают, что в случае Уравнения Пуассона. Если размер агрегатов сильно варьируется по домену, это становится трудным определить хорошее значение для  $\alpha$ .

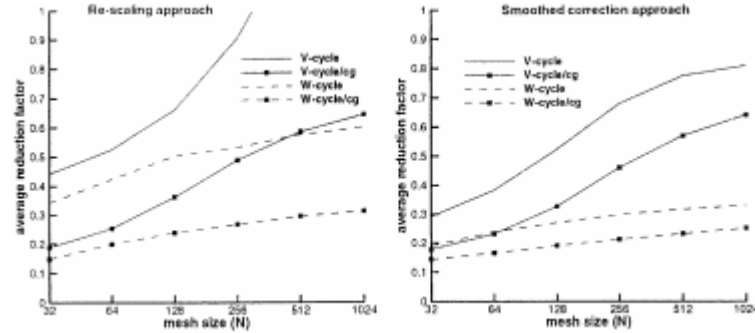


Рис. 12. (а) Подход с изменением масштаба ( $\alpha = 1.8$ ), (б) Сглаженный подход к коррекции.

#### 6.4. Улучшение путем сглаживания коррекций

Вместо того, чтобы явно указывать коэффициент масштабирования  $\alpha$ , как раньше, разумное масштабирование также может быть выполняется автоматически. Идея состоит в том, чтобы изменить шаг грубой коррекции (2), заменив истинная (кусочно-постоянная) коррекция  $e^h = I_H^h e^H$  в некотором приближении,  $e_0^h$ , а затем вычислить  $u_{new}^h$  по формуле

$$u_{new}^h = u_{old}^h + \alpha e_0^h \text{ с } \alpha = \frac{(f^h - A_h u_{old}^h, e_0^h)}{(A_h e_0^h, e_0^h)}, \quad (22)$$

вместо (2). Обратите внимание, что  $\alpha$  определяется так, что энергетическая норма погрешности  $u_{new}^h$  становится минимальным.

Ясно, что для того, чтобы эта минимизация была осмысленной, выбор  $e_0^h$  имеет решающее значение. Самое главное,  $e_0^h$  должно быть достаточно гладким приближением к  $e^h$ . (Выбор  $e_0^h = e^h$  не даст усиление: вариационный принцип просто подразумевает  $\alpha = 1$ .) Один из возможных вариантов выбора

$$e_0^h = S_h^\nu e^h, \quad (23)$$

что требует применения сглаживающих шагов к однородным уравнениям (начиная с  $e^h$ ).

Обратите внимание, что, грубо говоря, этот процесс оставит «гладкую часть»  $e^h$  практически без изменений; только его «высокочастотная часть» будет уменьшена. Следовательно, применяются правильные шаги сглаживания  $u_{new}^h$  после коррекции грубой сетки, будет эффективно исправить это.

Эффект этого процесса сглаживания поправок, продемонстрирован на рис. 12б (с использованием  $\nu = 2$ ). Помимо того, что по сравнению с подходом повторного масштабирования (см. Рис. 12а), сходимость немного лучше, здесь нет риска «перерегулирования», как раньше, так как процесс «контролирует себя». С другой стороны, дополнительные этапы сглаживания увеличивают общее время вычислений.

Хотя сглаживание поправок является простым средством автоматического исправления неправильных масштабов в некоторой степени его возможности ограничены. В любом случае итоговая общая производительность обычно хуже, чем у классического AMG.

#### 6.5. Улучшение за счет сглаживания интерполяции.

Более сложный (но и более дорогой) способ ускорения AMG базового типа агрегации подход разработан и проанализирован в [52–54]. Здесь кусочно-постоянная интерполяция рассматривается только как интерполяция первого угадывания, которая улучшается за счет некоторого процесса сглаживания («сглаженная агрегация») перед вычислением оператора Галеркина.

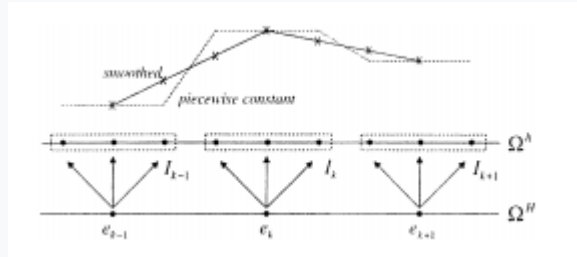


Рис. 13. Кусочно-постоянная и сглаженная интерполяция.

В [52,53] это сглаживание предлагается быть сделано, применяя один шаг  $\omega$ -Якоби-релаксации.

Чтобы быть более конкретным, обозначим оператор, соответствующий кусочно-постоянной интерполяции, через  $\tilde{I}_H^h$ .

Тогда фактически используемая интерполяция определяется как

$$I_H^h = (I_h - \omega D_h^{-1} A_h^f) \tilde{I}_H^h,$$

где  $D_h = \text{diag}(A_h^f)$  и  $A_h^f$  получается из исходной матрицы  $A_h$  путем сложения всех слабых связей по диагонали («фильтрованная матрица»). То есть, учитывая некоторый вектор грубого уровня  $e^H$ ;  $e^h = I_H^h e^H$  определяется применяя один шаг релаксации  $\omega$ -Якоби к однородным уравнениям  $A_h^f v^h = 0$ , начиная с кусочно-постоянный вектор  $\tilde{I}_H^h e^H$ . (Обратите внимание, что этот процесс увеличит «радиус» интерполяции и, следовательно, разрушить простоту основного подхода. Кроме того, интерполяция, как правило, не будет особой формы (5) уже нет. Отметим также, что здесь релаксация Якоби служит совершенно другой цели, чем Якоби-F-релаксация, как

рассматривается в разделе 5. В частности, Якоби-релаксация здесь используется как более гладкое, примененное к полной системе уравнений, которая требует использования нижнего параметра релаксации,  $\omega < 1$ .)

Чтобы проиллюстрировать этот процесс, мы снова рассмотрим одномерный случай  $-\frac{d^2u}{dx^2}$  и предположим, что агрегаты состоят из трех соседних переменных (соответствующих типичному размеру агрегатов, используемых в [52,53] в каждом пространственном направлении). Во-первых, обратите внимание, что, поскольку все связи сильны, мы имеем  $A_h^f = A_h$ .

На рис. 13 изображены как кусочно-постоянная интерполяция (пунктирная линия), так и сглаженная интерполяция, полученная после применения одного Якоби-шага с  $\omega = 2/3$  (сплошная линия). Очевидно, сглаженный Интерполяция просто соответствует линейной интерполяции, если переменные грубого уровня рассматриваются как точные аналоги тех переменных, которые находятся в центре совокупностей. Линейная интерполяция, однако не имеет проблемы масштабирования, как описано в разделе 6.2 для кусочно-постоянной интерполяции.

Конечно, в более общих ситуациях ослабление кусочно-постоянной интерполяции не даст точную линейную интерполяцию и хороший выбор  $\omega$  зависит от ситуации. Тем не менее, даже если  $\omega = 2/3$  остается фиксированным, сглаженная интерполяция обычно будет намного лучше, чем кусочно-постоянная интерполяция (На самом деле реальное преимущество сглаживания по сравнению с кусочно-постоянной интерполяцией в том, что ошибки, полученные после интерполяции с более грубого уровня, имеют гораздо меньшую «энергию»; см. также раздел 7.) Это продемонстрировано в [53] с помощью различных примеров с использованием смеси итерационных методов Гаусса – Зейделя и SOR для

сглаживания ошибок. Тенденция состоит в составлении совокупностей из трех соседних переменных в каждом пространственном направлении. Обратите внимание, что хорошее значение для  $\omega$  зависит не только от проблемы и основной сетки, а также от размера агрегатов. Если вместо этого только два соседа будут агрегированы в каждом пространственном направлении, один легко это подтвердит  $\omega \approx 0.5$  должно быть выбран.

В целом, классические AMG и AMG, основанные на сглаженной агрегации, работают сравнительно, если их применять к относительно гладким (пуассоновским) задачам. Определенное преимущество агрегации типа AMG является то, что он часто требует меньше памяти, чем классический AMG (из-за его очень быстрого огрубления, которое вызывает особенно низкую сложность оператора  $s_A$ ). С другой стороны, это оплачивается сниженной устойчивостью: код типа агрегации, кажется, требует ускорения с помощью сопряженного градиента чтобы поддерживать приемлемую эффективность и надежность в более сложных ситуациях. С другой стороны классический AMG прилагает больше усилий к построению интерполяции и выполняет более медленное укрупнение, его производительность обычно в меньшей степени зависит от таких аспектов, как сильные разрывы коэффициентов матрицы СЛАУ (для некоторых примеров см. в [51]).

## **7. Дальнейшие разработки и выводы**

Алгебраический многосеточный метод обеспечивает общий подход к разработке надежных и эффективных методов решения больших классов матричных уравнений, таких как те, которые обычно возникают в численном решении эллиптических ДУРЧП, как на структурированных, так и на

неструктурированных сетках, в 2D и 3D. Строительство подходящих операторов интерполяции (включая процесс укрупнения) имеют решающее значение для получения быстрой и (почти)  $h$ -независимой сходимости. Как правило, чем больше усилий вкладывается в эту конструкцию, тем быстрее скорость сходимости может быть достигнута, но, к сожалению, требуемая числовая работа может возрасти еще быстрее. Также основной проблемой при разработке эффективных алгоритмов AMG является компромисс между конвергенцией и численной работой, и сохранение баланса между ними является конечной целью любого практического алгоритма решения СЛАУ. Мы суммировали и обсудили различные возможности для реализации конкретных алгоритмов AMG. Большинство приложений типа методов, обсуждаемых в этой статье, основанных на классическом подходе AMG по-прежнему принадлежат к наиболее эффективным. Обширный список экспериментов, основанный на оригинале – коде AMG1R5, можно найти в [18]. Надежность и эффективность могут быть существенно улучшены, особенно в случае сложных трехмерных сеток, с использованием модификаций, упомянутых в разделах 3.1 и 5 (подробнее см. [51]). Методы AMG на основе сглаженной агрегации (см. Раздел 6.5) является эффективной альтернативой классическому AMG, по крайней мере, если он используется в качестве предобуславливателя к алгоритму Крыловского типа, а не в одиночку. Дальнейшие разработки и приложения, которые близки к классическим идеям AMG содержатся, например, в [16,23,24,26,32,33,40,62,63]. Связанные подходы, но с акцентом на различные стратегии укрупнения и интерполяции, например, найдены в [22,27]. Применения подхода агрегационного типа в вычислительной гидродинамике можно найти в [30,39,46,60].

Тем не менее, есть еще много приложений, для которых алгебраически определённая интерполяция, и, следовательно, результирующие характеристики AMG пока не являются удовлетворительными. Например, одна из основных текущих исследовательских деятельности в AMG направлена на его обобщение для эффективного решения таких систем ДУРЧП, как проблемы линейной упругости. Хотя AMG успешно применяется в различных случаях (см., Например, [9,14,31,48,53]), его развитие еще не достигло состояния, когда конкретный подход хорошо улажен. Однако даже для скалярных приложений все еще остаются вопросы о лучших способах определения, например, огрубление и интерполяция, если заданная матрица  $s.p.d.$ , содержит относительно большие положительные недиагональные записи, и матрица далека от того, чтобы быть слабо доминирующей по диагонали. В таких случаях показатели классического AMG могут быть только неоптимальными. Часто можно избежать таких ситуаций, упрощая данную матрицу перед применением AMG [41]. Можно также представить ситуации, когда было бы выгодно (и легко) обеспечить AMG с некоторой дополнительной информацией о проблеме под рукой. Например, информация о геометрии (с точки зрения расположения точек) или более конкретных описаний того, что является «алгебраически гладкой» ошибкой выглядит (например, в виде некоторого предоставленного пользователем «тестового вектора (ов)»). Эта дополнительная информация может использоваться для первой интерполяции AMG, чтобы приблизить определенные типы компонентов ошибок особенно хорошо. Прямые возможности уже были указаны в [48].

Далее мы кратко изложим несколько более поздних подходов для определения интерполяции, которые стремятся к повышению надежности в таких случаях, как упомянутые выше.

Новый способ построения интерполяции (AMGe, [14]) начинается с того факта, что алгебраически гладкая ошибка - не что иное, как ошибка, которая медленно сходится к  $w.r.t.$  в процессе релаксации. Следовательно, алгебраически гладкая ошибка, как правило, соответствует собственным векторам  $A$ , принадлежащим до наименьших собственных значений. Вместо определения интерполяции путем прямого приближения (7), цель в [14] состоит в том, чтобы определить интерполяцию, так чтобы чем меньше соответствующее собственное значение, тем лучше собственные векторы интерполированы. Чтобы удовлетворить это путем явного вычисления собственных векторов, конечно, слишком дорого, однако в случае методов конечных элементов - при условии известности матрицы жесткости элемента, которые нужно знать - можно вывести меры (связанные с мерами, используемыми в классической многосеточной теории), минимизация которой позволяет определять локальные представления алгебраически гладких компоненты ошибки в вышеприведенном смысле. Дополнительная надежность была продемонстрирована в [14] с помощью определенных модельных приложений. Тем не менее, подход все еще находится в зачаточном состоянии. В частности, значительный необходимо приложить усилия для развития, чтобы связать процессы укрупнения и интерполяции определение для того, чтобы получить оптимальный алгоритм. В любом случае, это интересный новый подход который может привести к более общеприменимым подходам AMG. Другие алгебраические подходы, предназначенные для решения уравнений, полученных из конечно-



элементных дискретизации, были рассмотрены в [31,58]. Оба подхода основаны на агрегации и базисные функции грубого пространства определены так, что их энергия в некотором смысле минимизирована. (В конечно-элементный контекст, естественно определить неявно интерполяцию путем построения грубого пространства базисные функции.) Это не требует известности матриц жесткости элементов, но приводит к глобальной (минимизирующая) проблема минимизации, точное решение которой было бы очень дорого. Тем не менее, итерационные процессы решения предлагаются в обеих статьях для получения приближенных решений, указывает на то, что дополнительная работа (вложенная на этапе установки) является приемлемой. В то время как Wan et al. [58] сосредоточиться на скалярных приложениях, расширение для систем ДУРЧП (от линейной эластичности) является одним из основных аспектов в [31]. Особое внимание уделяется правильной обработке мод с нулевой энергией (например, моды твердого тела в случае линейной упругости): такие моды должны содержаться в диапазоне базисные функции грубого пространства, по крайней мере, вдали от границ Дирихле. (Обратите внимание, что для типичных скалярных задач, это соответствует требованию, чтобы константы были точно интерполированы вдали от границ Дирихле, ср. (8).) Интересно, что подход в [31] можно рассматривать как продолжение более ранней работы [53] по сглаженной агрегации: если только одна итерация шаг выполняется для приблизительного решения проблемы минимизации энергии, полученный метод совпадает с подходом сглаженной агрегации. В отличие от последнего, однако, дальнейшие итерации не будут увеличивать поддержку базисных функций (то есть радиус интерполяции). Несколько тестовые примеры в [31] показывают преимущества этой новой

интерполяции с точки зрения сходимости скорость. К сожалению, однако, это преимущество существенно компенсируется за счет минимизации количества шагов.

Существуют различные другие документы, посвященные разработке многосеточных методов для решения конечно-элементные задачи на неструктурированных сетках. Хотя некоторые из них также основаны на алгоритмическом компоненты, которые более или менее определены алгебраически, большинство из них не должны быть алгебраическими многосеточными решателями в том смысле, который рассматривается в этой статье.

Поэтому мы не хотим обсуждать такие подходы здесь далее, но скорее ссылаются, например, на [15] и ссылки, приведенные в нем.

В подходе [56]  $A$  не предполагается асимметричным, а интерполяция и ограничение построены отдельно. Например, интерполяция построена так, что гладкая ошибка  $S_h e^h$  равна особенно хорошо интерполировал евклидову норму,  $\|\cdot\|_2$ . Точнее, попытка сделать так, чтобы евклидова норма  $\|S_h e^h - I_H^h e^H\|_2$ , была как можно меньше (часть (4)), т.е. чтобы  $e^H$  обозначает прямой впрыск  $S_h e^h$  до грубого уровня.

В [56] это приводит к определенным локальным минимизациям, которые используются для 1-й, для каждой переменной, пары соседние переменные, которые позволили бы хорошую интерполяцию в вышеуказанном смысле, и, в то же время, вычислить соответствующие веса (как интерполяции, так и ограничения). На основе этой информации строится C/F-расщепление, что позволяет каждой F-переменной интерполировать из одного из пар, найденных ранее. Эвристический алгоритм используется для минимизации общего числа C-переменных.

В этом контексте мы хотим отметить, что, хотя классический AMG был разработан в вариационной структуре он успешно применяется для большого числа несимметричных задач без каких-либо изменений. Это можно объяснить эвристически, но нет теоретического обоснования доступно в это время. В контексте сглаженной агрегации AMG теоретический анализ может можно найти в [25].

Важным аспектом, который не был рассмотрен в этой статье, является распараллеливание AMG.

Эффективное распараллеливание классического AMG довольно сложно и требует определенного алгоритмического модификации с целью ограничения стоимости связи без существенного ущерба для конвергенции. Большинство подходов к распараллеливанию, исследованных до настоящего времени, либо относятся к простым агрегированным вариантам (например, [46]) или используют прямые методы декомпозиции области (такие как чередующийся метод Шварца) для распараллеливания. Стратегия распараллеливания, которая остается очень близкой к классическому подходу AMG представлен в [29]. Результаты для сложных трехмерных задач демонстрируют, что этот подход достаточно хорошо масштабируется на компьютерах с распределенной памятью, если число неизвестных на процессор не так уж мало. Метод, рассмотренный в [56], также доступен параллельно. Там, например, в Боннском университете и Национальные лаборатории LLNL [17] и LANL, но к настоящему времени результаты не опубликованы.

Это выходит за рамки этой статьи, чтобы также обсудить разнообразие иерархических алгебраических подходов которые на самом деле не связаны с многосеточной идеей в том смысле, что эти подходы не основаны на фундаментальных принципах многосеточного метода, сглаживания и грубой

коррекции. На самом деле есть быстрое и очень интересное постоянное развитие таких подходов. Для полноты, однако, мы включили некоторые избранные ссылки. Различные подходы, основанные на приближении блочного исключения Гаусса (Методы «Шур-декомпозиции») можно найти в [2–5,19,36 –38,42,57]. Многоуровневые структуры также были введены в предобуславливатели типа ILU, например, в [49]. Совсем недавно какой-то гибрид были разработаны методы, использующие идеи ILU и многосеточного метода [6–8,43–45]. Для дальнейшего обсуждения см. также [55].

Подводя итог, разработке иерархически работающих алгебраических методов для эффективного решения. Решение больших разреженных, неструктурированных систем уравнений, в настоящее время принадлежит к одной из самых активных направлений исследований в численном анализе. Многие различные методы были исследованы, но на данный момент ни один из них не способен эффективно справиться со всеми практически актуальными проблемами. Все методы, кажется, имеют диапазон применимости, но все они могут быть неэффективными в определенных других приложениях. Следовательно, развитие в этой интересной области исследований должно продолжиться в течение следующих лет.

## **Ю. Саад**

**Мотивация.** Описанные до настоящего времени (геометрические) многосеточные методы с иерархией (геометрических) сеток, от самой грубой ( $l = 0$ ) до самой подробной, на всех уровнях, кроме самого грубого, будет применена итерационная сглаживающая процедура, например, метод Гаусса

Зейделя и система самого грубого сеточного уровня, обычно решается точно. Однако возникает следующий вопрос:

- Что следует сделать, если имеющаяся грубая сетка обладает уже таким количеством степеней свободы, что использование прямого решателя занимает слишком много времени?

Такая ситуация будет часто происходить, если проблема дается в сложной области в  $\mathbb{R}^d$ ,  $d \in \{2, 3\}$ , см. на рисунке 7.1 (академический) пример. Сложные расчетные области очень вероятно, будут встречаться в приложениях. Затем программа генератора сетки часто приводит к (самым грубым) сеткам, которые настолько подробны, что уточнение приведет к такому большому количеству степеней свободы, что эффективное моделирование проблемы не представляется возможным.

Есть всего только одна сетка.

Чтобы справиться с ситуацией самой грубой сетки со многими степенями свободы, существует как минимум две возможности.

- Одноуровневая итерационная схема. В случае, если есть геометрическая сетка иерархии, но грубая сетка уже велика, можно использовать простой итерационный метод, например, сглаживающий метод Зейделя, чтобы решить систему на самой грубой сетке приблизительно. Тогда режимы сглаживания ошибок на этой сетке не демпфируются. Однако опыт показывает, что этот подход работает на практике иногда довольно хорошо.

Если доступна только одна сетка, метод подпространства Крылова можно использовать для решения возникающих линейных систем уравнений.

- Итерационная схема с многоуровневыми идеями. Построить более сложный итерационный метод, который использует своего рода многосеточную идею для решения системы уравнений на грубой геометрической сетке. Реализация этой многосеточной идеи должна основываться только на информации, которая может быть получена из матрицы на самой грубой сетке.

Этот тип решателя называется алгебраическим многосеточным методом (AMG).

### **Заключение non-Galerkin amg**

Мы ввели метод без потерь, чтобы уменьшить объем работы, требуемой при работе параллельной версии алгебраического многосеточного метода, путем удаления слабых или неважных записей из операторов грубой сетки после формирования многосеточной иерархии. Эта альтернатива оригинальному методу не Галеркинских грубых сеток также способна уменьшить затраты на связь на грубых уровнях, что приводит к общему сокращению времени решения. Кроме того, этот метод сохраняет исходную иерархию Галеркина, что позволяет многим ограничения не-Галеркины должны быть ослаблены. В результате удаленные записи легко сосредоточены непосредственно на диагонали, что значительно снижает затраты на настройку, а также уменьшает сложность коммуникаций на этапе решения. Кроме того, по мере добавления записей по диагонали записи, удаленные из матрицы, сохраняются и адаптивно повторно вводятся в иерархию, если это необходимо для сходимости. Следовательно, компромисс между сходимостью и расходами на связь контролируются во время решения с небольшим количеством дополнительной работы.

**An efficient algebraic multigrid preconditioned conjugate gradient solver.** Chihiro Iwamura , Franco S. Costa , Igor Sbarski , Alan Easton , Nian Li

В этой статье мы представляем надежный и эффективный алгебраический многосеточный предобусловленный сопряженный градиент решатель для системы линейных уравнений, возникающие при дискретизации методом конечных элементов скалярного эллиптического уравнения в частных производных второго порядка на неструктурированных

сетках. Алгебраический многосеточный метод (AMG) является одним из наиболее перспективных методов решения больших систем линейных уравнений, возникающих из неструктурированных сеток. Обычный метод AMG обычно требует дорогостоящего времени установки, особенно для трехмерных задач, так что обычно оно не используется для небольших систем и систем среднего размера или приближения низкой точности. Наш решатель имеет фазу быстрой настройки для метода AMG и быстрый цикл итерации. Это позволяет нам применять этот решатель не только для больших систем, но и для малых и средних системы линейных уравнений, а также для систем, требующих приближения малой точности.

## **Введение**

При численном моделировании потока полимерного литья под давлением, систем линейных уравнений с решены симметричные положительно определенные (SPD) матрицы размером от малого до очень большого. Конечный элементная дискретизация уравнения давления гидродинамики на неструктурированных тетраэдрических сетках часто производит плохо обусловленную матрицу. Поэтому трудно получить решения таких систем линейного уравнения эффективно с использованием многих распространенных итерационных методов. В этой статье мы используем малый, средний и большой системы размеров, которые классифицируются по количеству неизвестных: менее 3000, 3000–20 000 и более 20000 соответственно.

Для плохо обусловленных матричных задач метод предобусловленного сопряженного градиента (PCG) может быть эффективным и надежным. Для задач сравнительно небольшого размера, распространенные (одноуровневые) решатели PCG, такие как симметричные последовательное переобусловленное предобусловленное сопряженный градиент (SSORPCG)

достаточно эффективен. Тем не менее, эффективность часто ухудшается при проблемах среднего и очень большого размера. Для этих типов проблем, Многосеточный (многоуровневый) метод PCG может быть методом быстрого решения. В этом методе многосеточный цикл используется в качестве предварительного условия для метода сопряженных градиентов (CG).

Наши системы линейных уравнений являются результатом конечно-элементной дискретизации скалярной эллиптической дифференциальное уравнение в частных производных (ДУРЧП) второго порядка на неструктурированных тетраэдрических сетках при инъекции полимера

литье моделирования. Геометрия наших моделей может быть очень сложной, а коэффициенты дифференциального оператора могут сильно варьироваться.

Для такого рода задач геометрический многосеточный метод не подходит, потому что обычно требуется равномерный шаблон укрупнения и дифференциальные операторы с непрерывным

медленно меняющиеся или постоянные коэффициенты. Кроме того, чтобы обеспечить простоту внедрения в существующие коммерческие коды, нас интересует только «черный ящик», который требует только информации содержится в данной системе матрицы. Поэтому мы не преследовали тех многосеточных методов, которые основаны на

при знании формулировки элемента или узловых координат. По этим причинам мы полагаем, что метод CG, предобусловленный алгебраической многосеткой (AMG), является наиболее подходящим для нашей цели.

Идеальные матрицы для метода AMG, описанные Руге и Штубеном [1,2], слабо диагональны доминантные симметричные М-матрицы. Матрица - это симметричная М-матрица, если она SPD и имеет неположительные недиагональные значения. Общедоступный решатель AMG, созданный Ruge,



Stuben и Hempel (AMG1R5), может использовать в качестве автономного решателя AMG (AMG1R5 / SA) или цикл AMG можно использовать в качестве предварительного условия для CG решатель (AMG1R5 / CG). AMG1R5 очень эффективен для решения систем с симметричной М-матрицей. Однако эффективность быстро ухудшается, если матрица отклоняется от симметричной М-матрицы.

Мы обнаружили, что простая модификация алгоритма AMG AMG1R5 может привести к более

надежный и эффективный решатель проблем, когда модифицированный цикл AMG используется в качестве предварительного условия для CG.

Надежность существенно улучшена путем создания соответствующего оператора интерполяции AMG и

строгое выполнение цикла AMG симметричным образом для получения предварительного кондиционера SPD AMG. Мы обнаружили, что наши системы линейных уравнений часто содержат до 20–35% ненулевых элементов, являющихся недиагонально положительными, и что они сильно отклоняются от слабо диагонально доминирующей симметричной М-матрицы. Тем не менее, наш подход надежно решает эти системы линейных уравнений, когда этот предобуславливатель AMG используется для CG.

Эффективность особенно повышается за счет уменьшения числа ненулевых входов в грубой системе матрицы с использованием очень небольших наборов интерполяционных точек. Во-первых, мы создали очень маленькие наборы интерполяции точки только от прямых соседей. Это дает простой, быстрый процесс укрупнения и более простые грубые системы (меньшее количество ненулевых элементов для матриц более грубой системы). Используя этот подход, небольшой итог число ненулевых элементов матриц

грубой системы в основном достигается за счет существенного уменьшения количество ненулевых элементов первой матрицы грубой системы, а также некоторое уменьшение во второй и третьей матрицы грубой системы. Если используется стратегия укрупнения AMG1R5, количество ненулевых записей Матрица первой грубой системы часто примерно одинакова или даже больше, чем число ненулевых записи оригинальной (лучшей) матрицы. Это большое количество ненулевых записей первой грубой системы вносит существенный вклад в общее объединенное число ненулевых записей всех систем. Мы понимаем, что с нашим методом будет увеличение коэффициента сходимости, однако, преимущество более простого грубого Системы значительно перевешивают этот недостаток. Кроме того, дальнейшее уменьшение общего числа ненулевых записи возможны без значительного увеличения коэффициента сходимости путем аппроксимации матрицы исходная система с более простой симметричной М-матрицей. Цикл AMG применяется для решения так называемой априорная система предварительной обработки, которая использует эту симметричную М-матрицу. Эта симметричная М-матрица создана из исходной матрицы системы, просто добавив любые положительные недиагональные записи к диагональному элементу та же строка и замена положительных недиагональных записей на ноль. Способ априорной предварительной обработки.

Созданная матрица подразумевает меньшее количество ненулевых записей. Наш грубый подход может быть применен к этой симметричной М-матричной системе для создания значительно меньшего общего числа ненулевых элементов. Грубые системы для AMG, созданный с помощью этих подходов, целесообразно использовать при предубеждении, и они дают

три важных Преимущества: сокращение времени настройки, сокращение времени цикла AMG и уменьшение потребности в памяти. Ради точности необходимо использовать двойные (64-битные) переменные точности в CG метод для большинства наших проблем. Однако нет необходимости использовать переменные двойной точности, когда оценка решения априорной системы предварительного кондиционирования. Поэтому дорогой (с точки зрения как память и вычислительная работа) многосеточный цикл может быть выполнен с одинарной (32-битной) точностью переменные. Следовательно, эффективность существенно улучшается, в то время как требования к памяти остаются маленькими.

Были проведены тесты, чтобы продемонстрировать, что наш метод CG, подготовленный AMG, обеспечивает эффективную стратегию решения при решении любого размера плохо обусловленной системы линейных уравнений. В тестах система линейных уравнений для давления для задач трехмерного (3D) моделирования потока была решена для фаза наполнения литьем полимеров. Решатели, используемые для решения этих систем линейных уравнений:

- SSORPCG: симметричная последовательная CG с предварительным условием избыточного расслабления, общий (одноуровневый) решатель PCG;
- AMG1R5 / SA: автономный решатель AMG от Ruge, Stuben и Hempel;
- AMG1R5 / CG: решатель AMG Ruge, Stuben и Hempel, где цикл AMG используется в качестве предобуславливателя для CG;
- SAMG: быстрый коммерческий решатель AMG (SAMG, выпуск 20b1), который является преемником RAMG05 [2] где RAMG05 является

преемником AMG1R5, в этих тестах цикл AMG используется в качестве предварительного кондиционера для метода сопряженных градиентов;

- AMGPCG0: наш CG-решатель с предварительной обработкой AMG, который использует исходную матрицу для априорной системы предварительной обработки и использует более простые грубые системы для AMG,
- AMGPCG1: наш предопределенный CG-решатель AMG, который всегда использует симметричную M-матрицу для априорная система предварительной обработки и использует более простые грубые системы для AMG.

На протяжении всей статьи AMGPCG используется для обозначения группы AMGPCG0 и AMGPCG1, которая имеют общий алгоритмом огрубления.

## 2. AMG в качестве предобуславливателя.

### 3. Робастый оператор многосеточного V-цикла.

В этом разделе мы ничего не говорим об эффективности для многосеточного оператора V-цикла, однако мы показываем, что  $\|M\|_B < 1$ , так что надежность гарантирована, а многосеточный предобусловливатель является законным. Операторы сглаживания Гаусса – Зейделя  $S$  (как  $S^{(pre)}$ , так и  $S^{(post)}$ ) будут иметь следующее соотношение для любой матрицы SPD при разумных предположениях (см. [2, с. 30]):

$$\|S_l e_l\|_{B_l}^2 \leq \|e_l\|_{B_l}^2 - \sigma \|e_l\|_{B_l D_l^{-1} B_l}^2, \quad (16)$$

где  $\|*\|_{B_l}$  - энергетическая норма относительно SPD-матрицы  $B_l$ ,  $\|*\|_{B_l D_l^{-1} B_l}$  является нормой в

отношении SPD матрица  $B_l D_l^{-1} B_l$  (то есть,  $\|e_l\|_{B_l D_l^{-1} B_l}^2 = (B_l D_l^{-1} B_l e_l, e_l)$ ), здесь  $D_l^{-1}$  - обратная диагональная матрица  $B_l$ ,  $e_l$  - вектор ошибок на уровне  $l$  и  $\sigma > 0$ . Имеем

$$\|e_l\|_{B_l D_l^{-1} B_l}^2 \geq \xi \|e_l\|_{B_l}^2, \quad (17)$$

#### 4. Априорные сведения о предобуславливании матрицы алгоритмом AMGPCG1.

Очевидным преимуществом замены исходной матрицы  $A$  на более простое приближение  $B$  является уменьшение потребности в памяти и вычислительной работы, необходимой как для установки, так и для цикла AMG.

Однако недостатком является то, что отношение двух констант  $\alpha_B$  и  $\beta_B$  неравенства (12)  $\beta_B / \alpha_B$  становится больше, так что отношение  $\alpha_Q$  и  $\beta_Q$  неравенства (15)  $\beta_Q / \alpha_Q$  может стать больше. Этот недостаток минимизируется путем аппроксимации исходной матрицы  $A$  симметричной  $M$ -матрицей, которая близка к исходной матрице  $A$ . Это связано с тем, что наш оператор интерполяции  $I_{l+1}^l$  может быть определен более точно, если матрица является симметричной  $M$ -матрицей. Это подразумевает более точный оператор грубой коррекции сетки  $T_{l+1}$ , так что мы можем иметь гораздо меньшее значение  $\|M\|_*$  для этого вида матрицы. Детали определения  $I_{l+1}^l$  обсуждаются в разделе 5.2, а оценки  $\|M\|_*$  приведены в разделе 6 для проблем, возникающих при моделировании потока литья под давлением полимера. Аппроксимация матрицы  $A$  создается

$b_{ii} = a_{ii} + \sum_{j \in N_i^+} a_{ij}, \quad N_i^+ = \{j: j \neq i, a_{ij} > 0\},$ $b_{ij} = 0 \text{ если } a_{ij} \geq 0, i \neq j,$ $b_{ij} = a_{ij} \text{ если } a_{ij} < 0, i \neq j,$	(28)
---	------

где  $a_{ij}$  и  $b_{ij}$  - коэффициенты матриц  $A$  и  $B$  соответственно. В этом приближении матрица  $B$  становится симметричной  $M$ -матрицей [7], которая приблизительно слабо диагонально доминирует, потому что большинство строк матрицы  $A$  имеют сумму элементов в каждой строке, равное приблизительно нулю для наших задач. Это удовлетворяет первой части предположения (i), что априорная матрица предобуславливания является SPD. Кроме того, все собственные значения  $B^{-1}A$  находятся в диапазоне  $(0; 1]$  [7]. Следовательно,  $\kappa(B^{-1}A)$  ограничен

$\kappa(B^{-1}A) \leq 1/\lambda_{\min}(B^{-1}A),$	(29)
---	------

где  $\lambda_{\min}(B^{-1}A)$  является наименьшим собственным значением  $B^{-1}A$ . Неравенство (29) гарантирует существование двух положительных значений  $\alpha_B$  и  $\beta_B$ , что удовлетворяет неравенству (12). Наши эксперименты также подтвердили, что собственные значения  $B^{-1}A$  находятся в диапазоне  $(0; 1]$  и  $\lambda_{\min}(B^{-1}A)$  для наших задач обычно между 0.1 и 0.5, а максимальное собственное значение  $B^{-1}A$  равно 1.0. Следовательно, вторая часть предположения (i) также удовлетворяется этим подходом.

## 5. Детали процесса огрубления в AMGPCG.

В методе AMG алгебраически гладкие ошибки, которые не могут быть быстро устранены сглаживателем (стандартный итерационный метод, такой как Гаусс-Зейдель), эффективно устраняются путем использования соответствующим образом отрегулированных грубых системы. Однако метод решения AMG имеет ряд недостатков. К ним относятся дорогие стоимость

установки этих грубых систем и большой объем памяти по сравнению с обычным (одноуровневым) итерационным решателем, таким как SSORPCG. Для преодоления этих недостатков мы используем соответствующие грубые системы, которые имеют меньшее число ненулевых элементов в своих матрицах. В AMGPCG эти соответствующие грубые системы достигаются за счет значительного уменьшения числа ненулевых элементов первого грубого матрица системы, а также несколько уменьшая матрицы второй и третьей грубой системы, как указано в следующие разделы.

Чтобы описать процесс укрупнения AMGPCG, от подробного уровня  $\ell$  до грубого уровня  $\ell+1$ , мы переписываем уравнение.

(4) как

$$B_{\ell}x_{\ell} = r_{\ell} \text{ или } \sum_{j \in \Omega_{\ell}} b_{ij,\ell} x_{j,\ell} = r_{i,\ell}, \quad i \in \Omega_{\ell}$$

где  $\Omega_{\ell}$  - набор индексов  $\{1; 2; 3; \dots; n_{\ell}\}$ . Чтобы вывести более грубую систему уровней, нам нужно разделить  $\Omega_{\ell}$  на непересекающиеся подмножества

$$\Omega_{\ell} = C_{\ell} \cup F_{\ell},$$

где  $C_{\ell}$  содержит точки, также представленные на более грубом уровне (С-точки, то есть  $\Omega_{\ell+1} = C_{\ell}$ ), а  $F_{\ell}$  представляет собой дополнительный набор (F-точек). После деления  $\Omega_{\ell}$  на  $C_{\ell}$  и  $F_{\ell}$  система AMG грубого уровня

$$B_{\ell+1}x_{\ell+1} = r_{\ell+1} \text{ или } \sum_{j \in \Omega_{\ell+1}} b_{ij,\ell+1} x_{j,\ell+1} = r_{i,\ell+1}, \quad i \in \Omega_{\ell+1}$$

построена с использованием формулы Галеркина. Оператор Галеркина,  $B_{\ell+1}$ , определен в формуле. (8).  $r_{\ell+1}$  и  $x_{\ell+1}$  фактически соответствуют невязке и коррекции (ошибке) уравнения (30) соответственно. Следовательно, уравнение (32) может быть выражается как

$$I_{\ell}^{\ell+1} B_{\ell} I_{\ell+1}^{\ell} e_{\ell+1} = I_{\ell}^{\ell+1} (r_{\ell} - B_{\ell} v_{\ell}),$$

где  $v_{\ell}$  является расчетное решение уравнения (30) после применения оператора предварительного сглаживания. Так как у нас есть  $I_{\ell}^{\ell+1} = (I_{\ell+1}^{\ell})^T$ ,

определяя оператор интерполяции  $I_{\ell+1}^\ell$  является одной из основных задач процесса укрупнения.

### Наборы связей между точками сетки

Прежде чем определить оператор интерполяции, нам нужно ввести множества, которые характеризуют связи между точками сетки. Первый набор является прямой окрестностью точки, которую я определил как

$$N_i = \{j \in \Omega: j \neq i, b_{ij} \neq 0\},$$

где  $\Omega$  - набор индексов. Далее мы говорим, что я сильно отрицательно связан с другой точкой  $j$ , если  $-b_{ij} \geq \varepsilon_{STR} \max |b_{ik}^-|$  при  $0 < \varepsilon_{STR} < 1$ , где  $b_{ik}^- = b_{ik}$  если  $b_{ik} < 0$  и  $b_{ik}^- = 0$ , если  $b_{ik} \geq 0$ , а  $\varepsilon_{STR}$  является константой которая определяет сильные отрицательные связи. Тогда мы обозначаем множество всех сильных отрицательных связей точки  $i$  по  $S_i$

$$S_i = \{j \in N_i: -b_{ij} \geq \varepsilon_{STR} \max |b_{ik}^-| \text{ с фиксированным } 0 < \varepsilon_{STR} < 1\}$$

Также через  $S_i^T$  обозначим множество точек, сильно отрицательно связанных с  $i$

$$S_i^T = \{j \in \Omega: i \in S_j\}$$

В AMG1R5  $\varepsilon_{STR}$  имеет значение по умолчанию установлено на 0,25. В нашем методе AMGPCG для систем, возникающих из 3D неструктурированные сетки, мы устанавливаем  $\varepsilon_{STR}$  на очень высокое значение (то есть,  $\varepsilon_{STR} = 0,98$ ) для самого лучшего и первого грубого уровня,  $\varepsilon_{STR} = 0,75$  для второго грубого уровня и  $\varepsilon_{STR} = 0,25$  для остальных грубых уровней ( $l = 4; \dots; l_{\max}-1$ ). Причина будет обсуждаться ниже. Наконец, для любого множества  $P$   $|P|$  обозначает количество элементов в множестве  $P$

### Определение весов интерполяции



Предположим, что  $\Omega$  разбито на непересекающиеся подмножества  $C$  и  $F$ . Затем мы определяем вес интерполяции так, что операторы интерполяции и ограничения могут быть определены.  $i$ -й компонент ошибки  $e$  на уровне штрафа  $\ell$  определяется как

$e_{i,\ell} = (I_{\ell+1}^\ell e_{\ell+1})_i = \begin{cases} e_{i,\ell+1} & \text{if } i \in C_\ell \\ \sum_{k \in C_{i,\ell}} w_{ik} e_{k,\ell+1} & \text{if } i \in F_\ell \end{cases}$	(37)
---	------

где  $C_i$  называется набором интерполяционных точек и определяется ниже. Наша цель состоит в том, чтобы определить вес интерполяции согласно уравнению (37).

Ошибка  $e$  считается алгебраически гладкой, если  $\|Se\|_B = \|e\|_B$ , где  $S$  - оператор сглаживания и  $\|e\|_B = \sqrt{(Be, e)}$ . Когда ошибка является алгебраически гладкой, имеет место следующее соотношение [1,2,4]:

$b_{ii}e_i + \sum_{j \in N_i} b_{ij}e_j \approx 0.$	(38)
---	------

Алгебраически гладкая ошибка медленно меняется в направлении сильных отрицательных связей [1,2,4]. Поэтому для каждого  $i \in F$  точки в  $N_i$  обычно делятся на три категории для всех уровней, кроме самых грубых:

- множество точек  $C$ , с которыми точка  $i$  имеет сильные отрицательные связи, определяемые как  $C_i = S_i \cap C$ ,
- набор  $F$ -точек, с которыми у точки  $i$  имеются сильные отрицательные связи, определяемые как  $D_i^S = S_i - C_i$ , и
- множество точек, с которыми у точки  $i$  есть слабые связи, включая положительные, которые могут содержать как  $C$ -, так и  $F$ -точки, определенные как  $D_i^W = N_i - S_i$ .

Набор  $D_i^W$  содержит как отрицательные, так и положительные связи. Алгебраически гладкая ошибка обычно не меняется медленно в направлении положительных связей [2]. Следовательно, мы разбиваем  $D_i^W$  на следующие два непересекающихся множества

$D_i^{W-} = \{j \in D_i^W : b_{ij} < 0\},$ $D_i^{W+} = \{j \in D_i^W : b_{ij} \geq 0\}.$	(39)
--	------

чтобы по-разному относиться к положительным связям. Поэтому перепишем (38) как

$b_{ii}e_i + \sum_{j \in C_i} b_{ij}e_j + \sum_{j \in D_i^{W-}} b_{ij}e_j + \sum_{j \in D_i^{W+}} b_{ij}e_j + \sum_{j \in D_i^S} b_{ij}e_j \approx 0$	(40)
---	------

Чтобы установить вес интерполяции  $w_{ik}$  уравнения (37), необходимо оценить  $e_j$  ( $j \neq C_i$ ) из (40), используя или  $e_i$  или  $e_k$  ( $k \in C_i$ ). Наша цель состоит в том, чтобы создать этот  $w_{ik}$  как можно более эффективно, сохраняя при этом точность  $w_{ik}$ , так как эффективная конструкция  $w_{ik}$  позволяет сократить время установки и получить точную  $w_{ik}$  меньший коэффициент сходимости. Во-первых, мы приближаем  $e_j$  ( $j \in D_i^{W-}$ ) с  $e_i$ . Потому что любая оценка ошибки, вызванная этим присвоением относительно незначительным, если  $|b_{ij}|$  мало, а  $e_j$  и  $e_i$  близки друг к другу, если  $|b_{ij}|$  велико. Эта оценка  $e_j$  ( $j \in D_i^{W-}$ ) очень быстрый и достаточно точный.

Для положительной связи обычно также достаточно оценить  $e_j$  ( $j \in D_i^{W+}$ ) по  $e_i$ , так как положительные связи, как правило, небольшие. Тем не менее, точность оценки для этого  $e_i$  может быть улучшена с помощью  $e_k$  ( $k \in C_i$ ), если существуют сильные отрицательные пути из точки  $i$  в  $j$  и из  $j$  в  $i$ , где сильные отрицательные пути означает, что путь между точками  $i$  и  $j$  через точки  $k \in C_i$  с относительно большими отрицательными связями существует. При наличии сильных отрицательных путей из точек  $i$  в  $j$  алгебраически

гладкая ошибка все еще может медленно меняться между этими точками. Далее, если есть сильные отрицательные пути в обоих направлениях между точки  $i$  и  $j$ , мы можем быть более уверенными в предположении, что алгебраически гладкая ошибка изменяется медленно между этими точками.

Поэтому мы приближаем  $e_j$

$$e_j \approx \frac{\sum_{k \in C_i} b_{jk}^{S-} e_k}{\sum_{k \in C_i} b_{jk}^{S-}},$$

где  $b_{jk}^{S-} = b_{jk}$  если  $k \in C_i^{j+} \subset C_i$  или  $b_{jk}^{S-} = 0$ . Множество  $C_i^{j+}$  определено

### Эффекты изменения $\varepsilon_{\text{str}}$

Значение  $\varepsilon_{\text{str}}$  настраивается таким образом, чтобы время установки, время цикла AMG и требования к памяти могли быть уменьшаются без значительного ущерба для поведения конвергенции. Эти цели могут быть достигнуты, если мы можем создать

более простые грубые системы, в то же время используя достаточно точный оператор интерполяции. Используете ли вы большое значение  $\varepsilon_{\text{str}}$  в алгоритме для «Предварительного выбора C-точки» и «Final C-Point выбор и определение весов интерполяции» будут увеличивать или уменьшать  $|C|$ , неясно. Однако наш опыт показывает тенденцию уменьшения  $|C|$  для матриц, возникающих из неструктурированных трехмерных сеток, и матрицы соответствующих первых нескольких более грубых систем. Кроме того, увеличение  $\varepsilon_{\text{str}}$  обычно подразумевает

меньшее количество элементов в множестве интерполяционных точек  $C_i$ . Так как  $|C_i|$  маленький и  $C_i$  создается только из непосредственных соседей определение веса интерполяции становится очень простым и, следовательно, Алгоритм «Окончательного выбора точки C и определения весов интерполяции» может быть выполнен быстро.

Кроме того, меньшее значение  $|C_i|$  означает, что число ненулевых элементов более грубой матрицы уровня меньше. Это потому, что грубая матрица строится с использованием уравнения. (8) с оператором интерполяции  $I_{\ell+1}^\ell$ , который имеет

меньший  $|C_i|$ . Еще одним важным преимуществом использования большого  $\varepsilon_{\text{str}}$  является то, что существует тенденция иметь меньше положительные недиагональные элементы или меньшие значения положительных недиагональных элементов в более грубой матрице. Это позволяет оператор интерполяции  $I_{\ell+2}^{\ell+1}$  будет определено более точно с использованием нашего алгоритма. Записи грубее матрица может быть выражена как

$$b_{km,\ell+1} = \sum_{i,j} \bar{w}_{ik} b_{ij,\ell} \bar{w}_{jm} = b_{km,\ell} + \sum_{i \in F_\ell} w_{ik} \left( b_{im,\ell} + \frac{1}{2} \sum_{j \in F_\ell} w_{jm} b_{ij,\ell} \right) + \sum_{i \in F_\ell} w_{im} \left( b_{ik,\ell} + \frac{1}{2} \sum_{j \in F_\ell} w_{jk} b_{ij,\ell} \right), \quad (47)$$

где  $k, m \in C_\ell$ ,  $\bar{w}_{ik} = \delta_{ik}$ , если  $i \in C_\ell$ , и  $\bar{w}_{ik} = w_{ik}$ , если  $i \in F_\ell$ . Здесь  $\delta_{ik}$  обозначает символ Кронекера. Обычно  $w_{ik}$  и  $w_{im}$  положительны, а большое  $\varepsilon_{\text{str}}$  подразумевает, как правило, большие  $-b_{im,\ell}$  и  $-b_{ik,\ell}$ . Следовательно,  $b_{km,\ell+1}$  имеют тенденцию становиться отрицательный или меньший положительный для недиагональных записей. Следовательно, применяя очень большое значение  $\varepsilon_{\text{str}}$  для лучшие и первые несколько грубых уровней могут привести к созданию значительно более простых и подходящих грубых систем и, следовательно, сократить время настройки и время цикла AMG, не увеличивая при этом существенно коэффициент сходимости.

Время настройки и время цикла AMG практически напрямую связаны с общим количеством ненулевые записи матриц всех уровней. Требование к памяти тесно связано с общим количеством ненулевые записи, хотя также требуется некоторая дополнительная память. К сожалению, меньший  $|C_i|$  часто приводит к увеличению коэффициента сходимости. Тем не менее, когда  $C_i$  создается с очень большим значением  $\varepsilon_{\text{str}}$ , алгоритм AMG

укрупняется в направлении очень медленно меняющейся алгебраически гладкой ошибки и имеет тенденцию создавать более грубую матрицу с меньшим количеством положительных недиагональных элементов или меньшими значениями положительных недиагональных элементов. Это означает, что интерполяция может быть достаточно точной, даже если у нас есть маленький  $|C_i|$ .

увеличение коэффициента сходимости не является существенным, так что его легко преодолеть уменьшением необходимой вычислительная работа и, следовательно, общая польза больше. использование этой интерполяции достаточно точный, когда метод AMG используется в качестве предварительного условия для CG. Мы увидим этот аспект в следующем разделе.

## Результаты

Были проведены тесты для решения различных систем линейных уравнений для давления, чтобы сравнить эффективность SSORPCG, AMG1R5 / SA, AMG1R5 / CG, SAMG, AMGPCG0 и AMGPCG1 для 3D проблемы. В этих тестах все решатели AMG настроены на использование одной итерации Гаусса-Зейделя в качестве сглаживателя и V цикла для многосеточного цикла. Другие параметры оптимизированы для достижения конвергенции в кратчайшие сроки возможное время сглаживатель AMG1R5 / CG был модифицирован для выполнения сглаживания с порядком точек наоборот. Это потому, что оператор пост-сглаживания  $S_\ell^{post}$  должен быть присоединен к оператору предварительного сглаживания  $S_\ell^{pre}$  в  $A_\ell$  энергии внутреннего продукта, когда AMG используется в качестве предварительного кондиционера для CG. Три разных моделирования методом конечных

элементов моделировалось на этапе заполнения моделирования потока литья под давлением полимера и решается при заполнении примерно на 90% по объему. Также для одной из моделей мы решили системы линейного уравнения с самого раннего этапа моделирования заполнения, чтобы продемонстрировать эффективность решения малых до проблемы среднего размера с нашими решателями. Допуски сходимости были установлены на  $1,0 \cdot 10^{-5}$ , что является значение по умолчанию, используемое в промышленности, так что каждый решатель прекратил итерацию, и истекшее время ЦП было измерено, когда относительный остаток был уменьшен до значения, меньшего, чем этот допуск. Относительный остаток определяется как евклидова норма остаточного вектора, деленная на евклидову норму вектора правой части уравнения. (1) (т.е.  $\|f - Av\| = \|f\|$ , где  $v$  - вектор оцененного решения).

Две из тестовых моделей являются пластинами, и их единственное отличие заключается в используемой сетке. Проблема Пластины-А в основном имеет элементы с относительно небольшими пропорциями (самая длинная сторона / самая короткая сторона) и проблема Plate-B имеет много элементов с высоким соотношением сторон (см. рис. 1–3). Основным источником плохой подготовки к нашей проблеме являются элементы с высоким соотношением сторон. Поэтому проблема Plate-B является хорошим тестом для надежности решателей. Количество узлов Plate-A и Plate-B составляет 29 587 и 170 028 соответственно.

Третья тестовая модель - это проблема электрического разъема, которая имеет сложную геометрию (см. рис. 4). Она имеет 200 060 узлов.

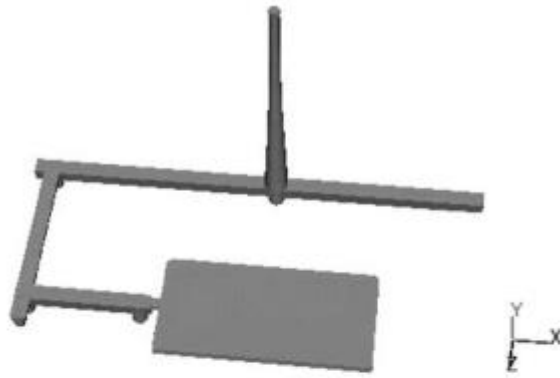


Fig. 1. Plate model.



Fig. 2. Mesh Plate-A (small aspect ratio).

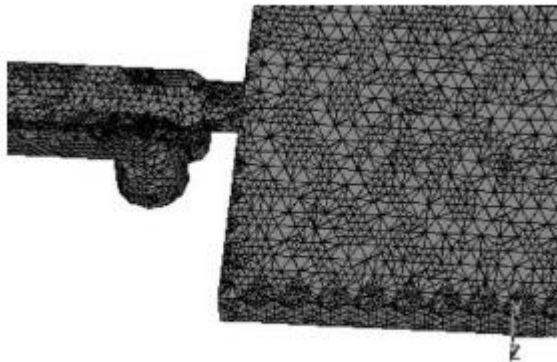


Fig. 3. Mesh Plate-B (large aspect ratios).

В таблице 1 приведены результаты для сложностей  $C_K$  и  $C_G$ , прошедшего времени и количества итераций.  $C_K$  и  $C_G$  - операторные и сеточные сложности, соответственно определенные как

$c_k = \sum_{\ell=1}^{\ell_{max}} \frac{ K_{\ell} }{ A_1 }, c_G = \sum_{\ell=1}^{\ell_{max}} \frac{ N_{\ell} }{ N_1 },$	(48)
---	------

где  $|K_{\ell}|$  - количество ненулевых элементов для матрицы  $K_{\ell}$ , а  $N_{\ell}$  - количество строк матрицы  $K_{\ell}$ . Для AMGPCG0, AMG1R5 и SAMG,  $K_1$  является исходной

матрицей  $A$  (то есть  $A_1$ ). Для AMGPCG1  $K_\epsilon$  является приближением  $A$  (см. Уравнение (28)).  $C_K$  почти пропорционален необходимой вычислительной работе и приблизительно пропорционален требуемой памяти. Время установки и время цикла AMG также почти напрямую связано со значением  $C_K$ .

На рис. 5  $\|M\|$  AMG-схемы AMGPCG оценивается с помощью векторов ошибок системы линейных измерений уравнения  $Bx = 0$ , где  $B$  - априорная матрица предобуславливания. Так как правая часть равна нулю, ошибка равна расчетному решению  $\|M\|$  можно выразить как

$$\|M\| = \sup_{e^{(m)} \neq 0} (\|Me^{(m)}\|/\|e^{(m)}\|) = \sup_{e^{(m)} \neq 0} (\|e^{(m+1)}\|/\|e^{(m)}\|), \quad (49)$$

где  $\|e^{(m)}\|$  и  $\|e^{(m+1)}\|$  Евклидовы нормы ошибки на  $m$ -ом и  $m+1$  первом шаге.

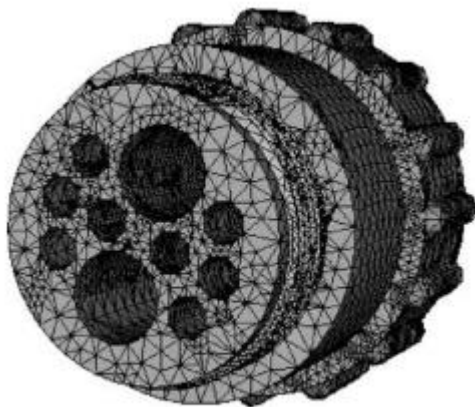


Fig. 4. Electrical connector.



Table 1  
Complexities and computation time

Method	Complexities		Elapsed time (s)/Pentium II 233 MHz			Number of iterations
	$C_K$	$C_G$	Setup time	Per cycle	Total	
<i>Plate-A</i>						
SSORPCG	1.00	1.00	0.0	0.17	61.0	352
AMG1R5/SA	3.32	2.05	4.4	0.54	12.0	14
AMG1R5/CG	3.32	2.05	4.4	0.66	9.0	7
SAMG	1.23	1.19	1.4	0.36	5.0	10
AMGPCG0	1.74	1.78	1.1	0.30	3.5	8
AMGPCG1	1.41	1.78	1.0	0.26	3.6	10
<i>Plate-B</i>						
SSORPCG	1.00	1.00	0.0	1.1	667.0	604
AMG1R5/SA	4.17	2.24	41.0	Did not converge		
AMG1R5/CG	4.17	2.24	41.0	5.1	143.0	20
SAMG	1.36	1.25	11.0	2.4	42.0	13
AMGPCG0	1.89	1.82	8.0	2.1	29.0	10
AMGPCG1	1.42	1.81	7.0	1.7	33.0	15
<i>Electrical connector</i>						
SSORPCG	1.00	1.00	0.0	1.3	160.0	119
AMG1R5/SA	5.33	2.27	77.0	6.1	426.0	57
AMG1R5/CG	5.33	2.27	77.0	7.3	187.0	15
SAMG	1.38	1.22	15.0	3.0	36.0	7
AMGPCG0	2.03	1.84	11.0	2.8	28.0	6
AMGPCG1	1.61	1.84	10.0	2.3	31.0	9

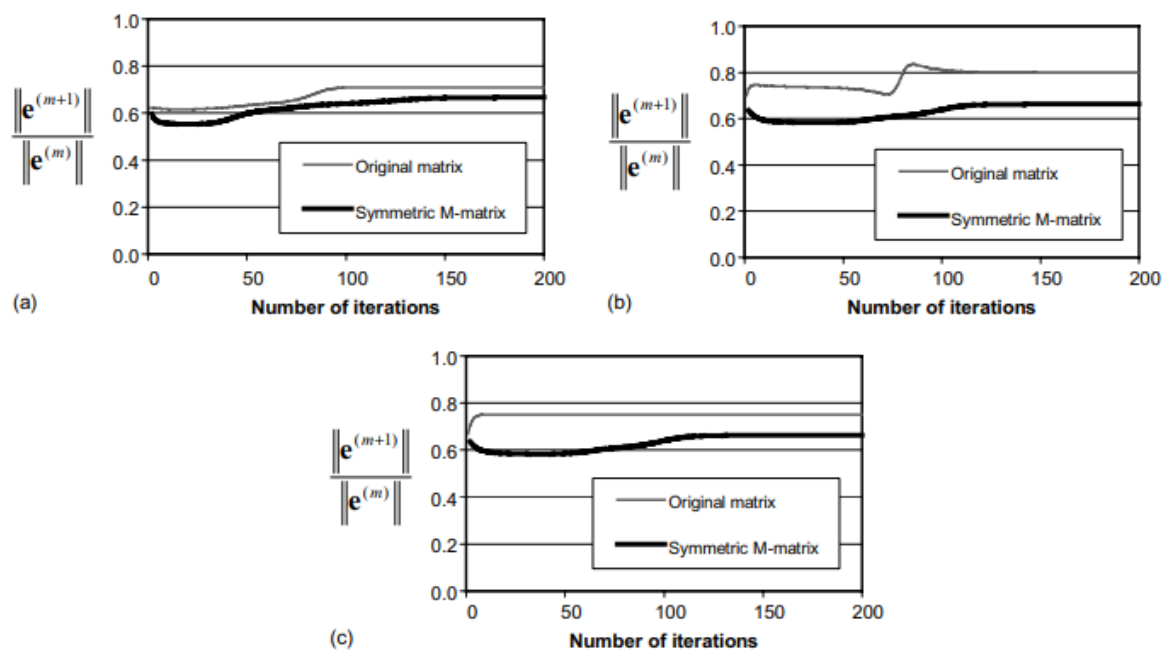


Рис. 5. (a) Оценка  $\|M\|$  для Plate-A, (b) оценка  $\|M\|$  для Plate-B и (c) оценка  $\|M\|$  для электрического соединителя

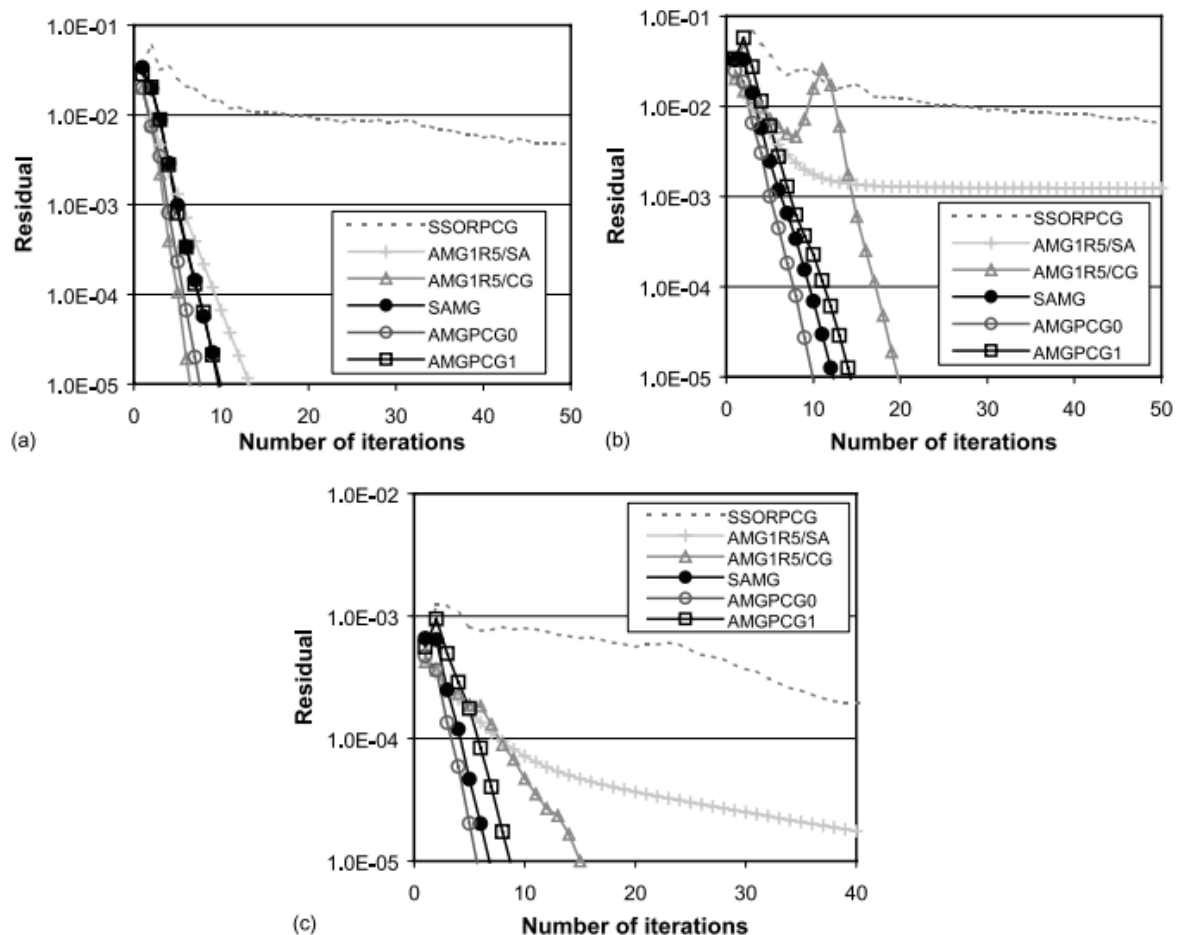


Рис. 6. (a) Графики изменения относительной невязки для Plate-A, (b) изменения относительной невязки для Plate-B и (c) изменения относительной невязки для электрический разъем.

На рис. 6 показаны истории изменения относительных невязок. На рис. 7 совокупное быстроедействие показано для фазы установки и фазы решения, чтобы показать эффективность решателей AMGPCG. Рис. 8 показывает улучшение быстрогодействия на решение, полученное с помощью AMGPCG для ранних этапов имитация заполнения Plate-A (то есть, решение систем малого и среднего размера).

## Дискуссия

AMGPCG0 строго удовлетворяет условиям (i) - (iv), и мы показали, что  $\|M\|_B < 1$  теоретически в Раздел 3. Кроме того, мы наблюдали  $\|e^{(m+1)}\|/\|e^{(m)}\| < 1$  с использованием нашей схемы AMG для возникающих проблем в литье под давлением полимера (см. также рис. 5). AMGPCG1 также строго удовлетворяет предположениям (i) - (iv).

Мы показали существование двух положительных постоянных, которые удовлетворяют неравенству спектральной эквивалентности отношение для матрицы B к A. Оцененные  $\|M\|$  схемы AMG намного меньше, когда оригинал матрица заменяется симметричной M-матрицей (см. рис. 5). Это связано с тем, что оператор интерполяции определен более точно, а с неравенства (19) имеет тенденцию уменьшаться при использовании нашего подхода с тенденцией к становятся меньше, потому что диагональная запись становится больше, когда есть положительные недиагональные записи в тот же ряд. Таким образом, наш метод AMG является законным для использования в качестве предварительного условия для CG.

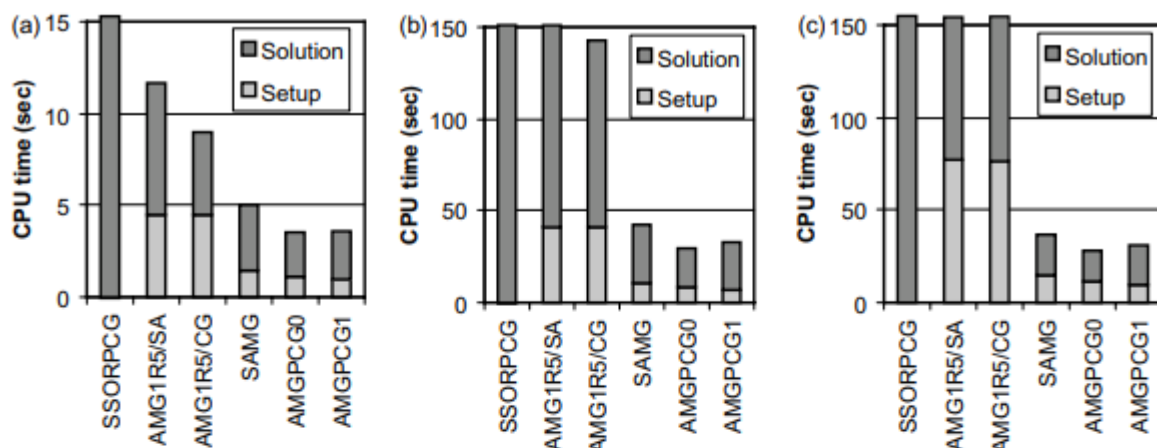


Fig. 7. Elapsed CPU time for (a) Plate-A, (b) Plate-B and (c) electrical connector.

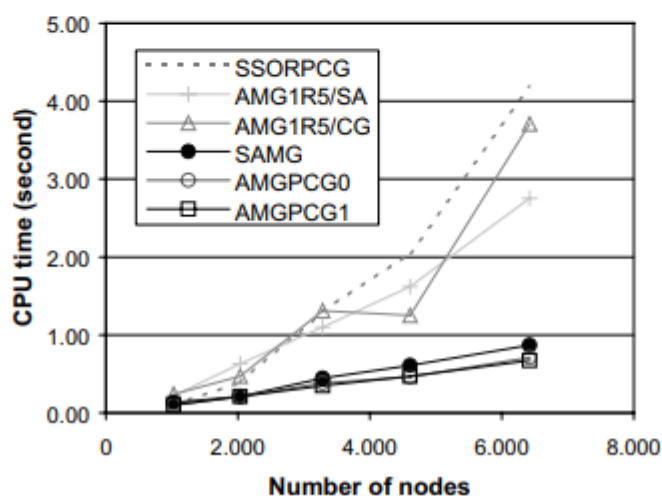


Fig. 8. Elapsed CPU time for small to medium systems for the Plate-A.

Мы протестировали три задачи с нерегулярными трехмерными тетраэдрическими сетками. Plate-A в основном имеет элементы с небольшими пропорциями, однако некоторые элементы все еще довольно искажены с большим соотношением сторон (см. рис. 2). Plate-B является наиболее плохо обусловленной проблемой из всех протестированных здесь, так как многие элементы имеют большой соотношение сторон (см. рис. 3). Электрический разъем имеет сложную геометрию, а некоторые элементы имеют высокую пропорции (см. рис. 4).

AMG1R5 / SA и AMG1R5 / CG работают быстрее, чем SSORPCG для решения проблемы Plate-A. Однако для проблема Plate-B, эффективность

решателя AMG1R5 / SA значительно ухудшается и не может достичь допуск сходимости в течение 200 итераций. Как видно из рис. 6 (б), после примерно 15 итерации остаток AMG1R5 / SA больше не уменьшается. Хотя AMG1R5 / CG может достичь сходимости достаточно быстро, мы видим отклонение остатка от AMG1R5 / CG для этой проблемы. Это связано с тем, что проблема Плате-В плохо обусловлена множеством положительных недиагональных входов и следовательно, вес интерполяции, созданный AMG1R5, может не подходить. В случае проблемы с электрическим разъемом AMG1R5 / SA может достичь допуска к сходимости с 57 итерациями, однако остаток AMG1R5 / SA имеет очень постепенное снижение. Эта модель имеет несколько элементов с высоким соотношением сторон (см. рис. 4). Поэтому это также не вполне обусловленная проблема. AMG1R5 / CG достигает конвергенции в 15 итераций, однако, общее требуемое время больше, чем SSORPCG. Особенно это связано с дорогостоящими настройками алгоритма AMG AMG1R5.

Когда SAMG оптимизирован по скорости, он очень быстро достигает конвергенции, используя стратегию «агрессивного огрубления» [2] для решения этих проблем. Эта грубая стратегия приводит к очень малым сложностям оператора  $C_K$  и сеточные сложности  $C_G$  (см. таблицу 1). Сложности операторов  $C_K$  SAMG для этих задач 4 – на 15% меньше, чем  $C_K$  AMGPCG1.  $C_G$  сложности SAMG более чем на 30% меньше, чем AMGPCG для этих проблем. SAMG значительно быстрее, чем SSORPCG и AMG1R5 для этих проблем. SAMG намного быстрее, чем AMG1R5, потому что сложности оператора значительно меньше, чем AMG1R5, в то время как необходимое количество итераций мало. Меньшая сложность оператора

приводит к меньшей вычислительной работе на этапе настройки и многосеточном цикле, чтобы ускорить как настройку, так и многосеточный цикл.

Сложности операторов обоих решателей AMGPCG также намного меньше, чем сложности операторов решателей AMG1R5. Сеточные сложности решателей AMGPCG лишь немного меньше, чем сеточные сложности решателей AMG1R5. Это означает, что меньшая сложность оператора получается путем создания более грубой матрицы с меньшим количеством ненулевых недиагональных элементов, чем существенное уменьшение  $|C|$ . AMG методы с агрессивным укрупнением, такие как SAMG, могут привести к гораздо меньшей сложности оператора, чем наш метод путем значительного снижения  $|C|$  (то есть небольшой CG). Хотя сложность нашего оператора больше чем сложность SAMG, у нашего подхода есть важные преимущества. Наш подход производит гораздо меньшие наборы интерполяционных точек, которые создаются только из лучших соседей на первых нескольких грубых уровнях. Наши решатели также имеют относительно большую сложность сетки по сравнению с SAMG. Во-первых, это означает, что определить вес интерполяции просто с помощью нашего алгоритма, так что настройка намного быстрее, несмотря на большую сложность. Во-вторых, на этапе решения большая сложность оператора означает что работа, необходимая для сглаживания, больше. Тем не менее, гораздо меньшие наборы интерполяционных точек в этих уровнях и относительно большая сложность сетки сокращают вычислительную работу, необходимую для интерполяции

(приведение решения грубой системы к штрафной системе) и ограничение (приведение остатка штрафа система в грубую систему). Следовательно, используя наш подход, вся вычислительная работа для многосеточной цикл может быть меньше, несмотря на большую сложность оператора. Кроме того, мы используем переменные одинарной точности для того чтобы оценить решение априорной системы предобуславливания (4). Это уменьшает требования к памяти для наша схема AMG. Это также уменьшает время настройки и время цикла AMG примерно на 10–20% по сравнению с использованием переменных двойной точности при тестировании на нашем 32-битном оборудовании. Поэтому настройка и мультисетка цикл AMGPCG быстрый (см. таблицу 1).

Оба решателя AMGPCG имеют быструю настройку и могут достигать допуска конвергенции в течение небольшого количества итераций (в основном менее 20 итераций) и имеют устойчивое остаточное уменьшение для любых задач мы проверили. AMGPCG1 немного увеличивает количество необходимых итераций по сравнению с AMGPCG0.

Однако, как видно из таблицы 1, время установки и время цикла являются быстрыми, поэтому общее требуемое время примерно одинаково или лишь незначительно увеличивается по сравнению с AMGPCG0 для этих проблем. Это потому, что AMGPCG1 имеет меньшую сложность оператора. Это примерно 75–82% AMGPCG0 для этих проблемы. В некоторых случаях, особенно когда модель в основном имеет элементы с небольшими пропорциями, AMGPCG1 сходится быстрее, чем AMGPCG0. Это потому, что положительные недиагональные записи оригинала системные матрицы имеют тенденцию уменьшаться и, следовательно, соотношение двух

постоянных  $\alpha_B$  и  $\beta_B$  неравенства (12)  $\beta_B / \alpha_B$  становится меньше. Это означает, что AMGPCG1 едва увеличивает количество необходимых итераций по сравнению с AMGPCG0, в то время как его сложность оператора меньше, чем сложность оператора AMGPCG0.

Решатели AMGPCG всегда самые быстрые среди решателей, которые мы здесь тестировали. Для плохо обусловленной задачи Plate-B оба решателя AMGPCG более чем в 20 раз быстрее, чем SSORPCG, более чем в четыре раза быстрее, чем AMG1R5 / CG, и немного быстрее, чем SAMG. Наша AMGPCG намного быстрее, чем AMG1R5 / CG, потому что время установки и каждого цикла очень быстрое, а вес интерполяции установлен соответствующим образом (то есть, нет больших колебаний остатка, как наблюдается в AMG1R5 / CG). Для проблемы Plate-B, многие веса интерполяции становятся отрицательными для AMG1R5, в то время как отрицательные веса интерполяции для AMGPCG не возникали. Также AMGPCG быстрее, чем SAMG для этих проблем, потому что установка и время цикла быстрее, а необходимое количество итераций аналогично SAMG. Даже не смотря на то что SAMG немного медленнее, чем AMGPCG для этих проблем, его требования к памяти меньше, чем AMGPCG. Кроме того, SAMG очень общий. Он может эффективно решать множество различных типов матриц.

Все решатели используют решение предыдущего временного шага в качестве начальной оценки для текущего шага. Остаток требуется только уменьшение на два порядка для проблемы с электрическим разъемом и приблизительно на четыре порядка для проблем с пластиной-A и пластиной-B, чтобы удовлетворить сходимость толерантность. Поэтому быстрая настройка очень важна для этих проблем. Оба решателя AMGPCG завершают расчет для



оценки решения еще до завершения этапа настройки AMG1R5 для этих три проблемы (см. таблицу 1 и рис. 7).

На рис. 8 показано общее время, необходимое для решения системы линейных уравнений для давления на ранних этапах моделирования потока впрыска полимера. Как видно из рисунка 8, решатели SAMG и AMGPCG очень быстро по сравнению с другими решателями. Когда система имеет 2000 узлов, SAMG и AMGPCG уже быстрее, чем SSORPCG и на 6000 узлов, эти решатели примерно в пять раз быстрее, чем SSORPCG и в три-четыре раза быстрее, чем оба решателя AMG1R5. Поэтому и SAMG, и решатели AMGPCG могут последовательно использоваться для любого размера задачи матрицы.

Часто в промышленных задачах, включая моделирование потока литьевого формования полимера, очень высокая точность решения не требуется. Поэтому значение допуска не устанавливается на очень маленькое значение. Кроме того, достаточно точная оценка часто доступна с предыдущего временного шага. Поэтому AMG решатель с быстрой установкой фазы идеально подходит для этого типа моделирования.

## **Заключение**

Системы линейных уравнений, возникающие при дискретизации методом конечных элементов уравнения давления на неструктурированные сетки в литье под давлением полимеров были решены SSORPCG, AMG1R5 / SA, AMG1R5 / CG, SAMG, AMGPCG0 и AMGPCG1, чтобы продемонстрировать, что наш подход использования AMG в качестве предварительного условия для CG может произвести очень эффективный и надежный решатель.

Во-первых, надежность существенно улучшается путем создания соответствующего оператора интерполяции для метода AMG и

использование этого метода AMG в качестве предварительного условия для CG.

Во-вторых, более простые грубые системы (меньшая сложность оператора) получаются путем сокращения первых грубые системы в основном за счет использования очень малого  $|C_i|$ , где  $C_i$  создается только из очень сильно отрицательно связаны прямые соседи. В случае AMGPCG1 дальнейшее снижение сложности оператора получается простым приближением исходной системной матрицы для априорной матрицы предварительного кондиционирования. Кроме того, наша стратегия укрупнения проста. Следовательно, мы можем выполнить этап установки очень быстро и использование наших простых грубых систем обеспечивает быстрые итерационные циклы, но не увеличивает коэффициент сходимости.

Следовательно, наши решатели очень надежны и эффективны для решения систем линейных уравнений любого размера возникающие в результате дискретизации скалярного эллиптического ДУРЧП на неструктурированных трехмерных сетках. Кроме того, наши решатели могут эффективно решать задачи, требующие только приближения с низкой точностью.

## Algebraic Multigrid Methods

### Определение

Алгебраические многосеточные (AMG) методы используются для нахождения решений (разреженных) линейных систем уравнений, используют многоуровневую стратегию релаксации и коррекцию на грубой сетке, используемую в геометрических многосеточных (GMG) методах. В то время как уравнения в частных производных (ДУРЧП) часто являются источником этих линейных систем, целью AMG является обобщить

многоуровневый процесс для решения задач, где не очевидно как решается грубосеточная задача - например, неструктурированные сетки, задачи на графах или структурированные проблемы, где равномерное уточнение не эффективно. В GMG многоуровневая иерархия определяется по структурированному огрублению проблемы с последующим определением релаксации и операторы интерполяции. Напротив, в способе AMG выбран способ релаксации, например Гаусса-Зейделя - и грубые задачи и интерполяция конструируются автоматически.

## **Обзор**

Ранние работы в многосеточных методах полагались на геометрическую структуру для построения грубосеточных операторов. Это было обобщено в [11] Маккормиком, где многосеточный метод анализируется в условиях свойств матрицы. Этот алгебраический подход к теории получил дальнейшее развитие Мандал в [10], и вместе с [3] Брандтом, эти работы составляют основу для многих работ раннего периода развития, которое привело к так называемой Руге-Штубен или классической алгебраической многосеточный (CAMG) метод в [13]. Одним из отличительных аспектов CAMG является то, что грубая проблема определяется на подмножестве степеней свободы исходной проблемы, что приводит как к грубой, так и к тонким моментам, приводящие к термину AMG на основе CF. Другой вариант алгебраического многосеточного метода появился в [14] как сглаженная агрегация на основе AMG (SA), где коллекции степени свободы (или агрегаты) определяют грубую степень свободы. Вместе алгоритмы-программы AMG на базе CF и SA привели к ряду разработок по расширению AMG для более широкого класса проблем и архитектур.

Существует ряд программных библиотек, которые реализуют различные формы AMG для разных целей. Оригинальный алгоритм и варианты SAMG доступны как `amglr5` и `amglr6` [13]. Библиотека HYPRE поддерживает параллельную реализацию на основе C/F AMG в пакете BoomerAMG [8]. Пакет Trilinos включает ML [7] в качестве параллельного решателя AMG на основе SA. Наконец, PyAMG [2] включает несколько вариантов AMG для тестирования, и Cusp [1] распространяет свободно стандартную реализацию SA для использования на графических процессорах (GPU).

### Терминология

Цель решателя AMG - найти решение системы линейных алгебраических уравнений

$Ax = b,$	(1)
-----------	-----

где  $A \in \mathbb{R}^{N \times N}$  разреженный, симметричный и положительно определенный. Прекрасная проблема (1) определяется на множестве точных индексов  $\Omega_0 = \{0, \dots, n-1\}$ . Метод AMG обычно определяется в два этапа: этап настройки и этап решения. Этап настройки ответственный за построение грубых операторов  $A_k$  для уровня  $k$  иерархии, вместе с оператором интерполяции  $P_k$ . Базовая иерархия, например, состоит из ряда операторов  $\{A_0, A_1, \dots, A_m\}$  и  $\{P_0, P_1, \dots, P_{m-1}\}$ .

При такой иерархии фаза решения затем выполняется так же, как и в геометрическом многосеточном методе, как в алгоритме 1 для двухуровневого метода; метод уровня  $m$  распространяется аналогично. Здесь оператор  $G(\cdot)$  обозначает метод релаксации, такой как взвешенный Якоби или Гаусс-Зайдель.

**Algorithm 1: AMG Solve Phase**

$\mathbf{x} \leftarrow \mathcal{G}(A_0, \mathbf{x}, \mathbf{b})$	{Pre-relaxation on $\Omega_0$ }
$\mathbf{r}_1 \leftarrow P_0^T \mathbf{r}$	{Restrict residual $\Omega_1$ }
$\mathbf{e}_1 \leftarrow A_1^{-1} \mathbf{r}_1$	{Coarse-grid solution on $\Omega_1$ }
$\hat{\mathbf{e}} \leftarrow P_0 \mathbf{e}_1$	{Interpolate coarse-grid error}
$\mathbf{x} \leftarrow \mathbf{x} + \hat{\mathbf{e}}$	{Correct fine-grid solution}
$\mathbf{x} \leftarrow \mathcal{G}(A_0, \mathbf{x}, \mathbf{b})$	{Post-relaxation on $\Omega_0$ }

**Теоретические наблюдения**

Двух сеточный процесс, определенный в алгоритме 1, можно рассматривать как распространение ошибки оператора интерполяции. Во-первых, пусть  $G$  представляет оператор ошибки для релаксации - например,  $G = I - \omega D^{-1}A$  для взвешенного метода Якоби. Кроме того, грубые операторы  $A_k$  обычно определяются через произведение Галеркина:

$A_{k+1} = P_k^T A_k P_k$ . Таким образом, для начального предположения  $\mathbf{x}$  и точного решения  $\mathbf{x}^*$  в (1) ошибка  $\mathbf{e} = \mathbf{x}^* - \mathbf{x}$  для двухсеточного метода с одним проходом предварительной релаксации определяется через

$$\mathbf{e} \leftarrow \left( I - \underset{\substack{\uparrow \\ \text{interpolate}}}{P_0} \left( \underset{\substack{\uparrow \\ \text{coarse solve}}}{P_0^T A_0 P_0} \right)^{-1} \underset{\substack{\uparrow \\ \text{restrict}}}{P_0^T} \underset{\substack{\uparrow \\ \text{residual}}}{A_0} \right) \underset{\substack{\uparrow \\ \text{relax}}}{G} \mathbf{e}$$

correct

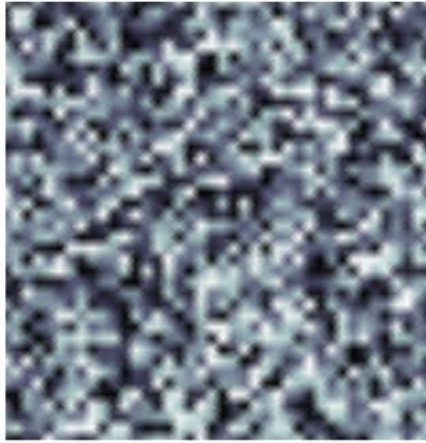
Ключевое наблюдение следует из (2) при определении методов AMG: если оставшаяся после релаксации ошибка содержится в диапазоне интерполяции, обозначаемом  $\mathfrak{R}(P)$ , тогда решатель точен. То есть, если  $G\mathbf{e} \in \mathfrak{R}(P)$ , то грубая сеточная поправка определена по формуле  $T = I - P(P^T A P)^{-1} P^T A$  уничтожает ошибку. Одним из важных свойств  $T$  является что это  $A$ -ортогональная проекция, которая подчеркивает тесную связь с другими методами подпространственной проекции.

## Методы

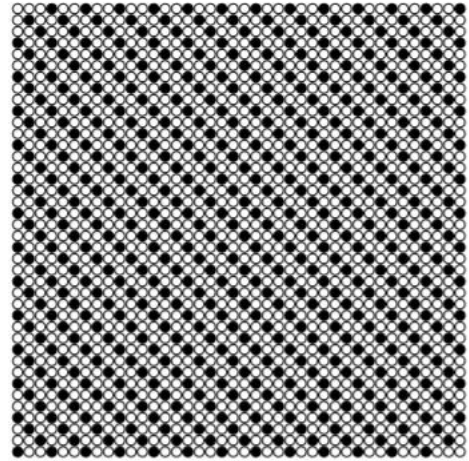
Фаза установки AMG определяет метод, однако есть несколько общих особенностей

1. Определение силы связи между степенями свободы;
2. Определение грубых степеней свободы;
3. Построение интерполяции,  $P$ ; а также
4. Формирование грубосеточного оператора через произведение Галеркина  $P^T A P$ .

Алгебраические методы определяют грубые сетки и результирующие операторы интерполяции, чтобы дополнить ограничения релаксации. То есть интерполяция должна захватывать компоненты ошибки которые не адекватно подавляются, например, взвешенным методом Якоби. Ошибка, не уменьшенная релаксацией, называется алгебраически гладкой ошибкой. Идентифицировать сглаженная ошибка, ребро на графике матрицы  $A$  считается сильным, если воспринимается ошибка медленно меняться по этому ребру. Это позволяет автоматически огрубить, чтобы соответствовать особенностям работы итерационного метода релаксации. В качестве примера рассмотрим случай анизотропного диффузионного оператора  $-u_{xx} - \epsilon u_{yy}$  повернутый на  $45^\circ$  вдоль оси координат и дискретизирован конечными элементами  $Q1$  на равномерной сетке. При увеличении анизотропного поведения ( $\epsilon \rightarrow 0$ ) происходит равномерное укрупнение и с геометрической многосеткой приводит к ухудшению производительности. В алгебраическом методе огрубление происходит в направлении гладкой ошибки, которая следует за линией анизотропии как показано на рисунке 1. Здесь, укрупнение выполняется (автоматически) только в направлении гладкой ошибки и приводит к высокой скорости сходимости.



(a) Error after relaxation for a random guess.



(b) Coarse points (•) and fine points (○).

Рисунок 1. Огрубление при работе C/F основанного AMG.

### AMG на основе C/F разбиения.

AMG на основе C/F начинается с  $A_k$ , матрицы k-уровня, и определяет сильные ребра в соответствии с

$$-A_{ij} \geq \theta \max_{k \neq i} (-A_{ik}),$$

(3)

где  $\theta$  - некоторый порог. Этот процесс дает матрицу прочности  $S$  (см. Алгоритм 2),

который идентифицирует ребра, где ошибка гладкая после релаксации. В свою очередь,  $S$  используется для разделения множества узлов данного уровня на C-точки или F-точки (см. рисунок 1b), требуя, чтобы F точки сильно связаны по крайней мере с одной C-точкой (для интерполяции). С C / F-точками

определены веса  $W$ , чтобы сформировать оператор интерполяции вида

$$P = \begin{bmatrix} W \\ I \end{bmatrix}$$

Наконец, грубый оператор строится через произведение Галеркина,  $P^T A P$ , которое имеет доминирующая стоимость для большинства методов AMG.



---

**Algorithm 2:** CF-based AMG

---

**Input:**  $A$ :  $n \times n$  fine level matrix

**Return:**  $A_0, \dots, A_m, P_0, \dots, P_{m-1}$

**for**  $k = 0, \dots, m - 1$  **do**

$S \leftarrow \text{strength}(A_k, \theta)$  {Compute strength-of-connection}

$C, F \leftarrow \text{split}(S)$  {Determine  $C$ -points and  $F$ -points}

$P_k \leftarrow \text{interp}(A_k, C, F)$  {Construct interpolation from  $C$  to  $F$ }

$A_{k+1} = P_k^T A_k P_k$  {Construct coarse operator}

---

### Алгебраический многосеточный метод сглаженной агрегации

Методы AMG на основе SA имеют важное отличие: они требуют априорного знания медленной сходимости или гладкой ошибки, обозначаемой  $B$ . Общий выбор для них векторы при отсутствии дополнительных знаний о проблеме  $B \equiv 1$ , постоянный вектор. Алгоритм SA (см. Алгоритм 3) сначала строит силу соединения матрица, аналогичная AMG на основе C/F, но с использованием симметричного порога

$$|A_{ij}| \geq \theta \sqrt{|A_{ii}A_{jj}|}$$

Исходя из этого, агрегаты или наборы узлов формируются (см. Рисунок 2) и представляют грубые степени свободы. Далее,  $B$  ограничен локально для каждого агрегата, чтобы сформировать предварительный оператор интерполяции  $T$ , так что  $B \in R(T)$ . Затем, чтобы повысить точность интерполяция,  $T$  сглаживается (например, с помощью взвешенного Якоби), чтобы получить интерполяцию оператор  $P$ . Это показано на рисунке 2b, где кусочно-постоянные функции образуют основа для базиса  $T$ , в то время как основа для базиса  $P$  напоминает кусочно-линейный функции. Наконец, грубый оператор вычисляется через произведение Галеркина.



---

**Algorithm 3:** SA-based AMG

---

**Input:**  $A$ :  $n \times n$  fine level matrix

$B$ :  $n \times c$  vectors representing  $c$  smooth error components

**Return:**  $A_0, \dots, A_m, P_0, \dots, P_{m-1}$

**for**  $k = 0, \dots, m - 1$  **do**

$S \leftarrow \text{strength}(A_k, \theta)$  {Compute strength-of-connection}

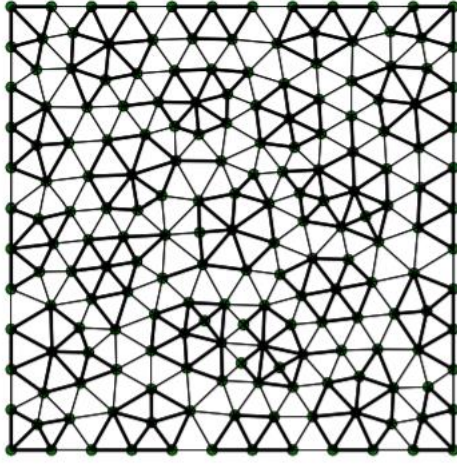
$Agg \leftarrow \text{aggregate}(S)$  {Aggregate nodes in the strength graph}

$T_k \leftarrow \text{tentative}(B, Agg)$  {Construct tentative interpolation operator}

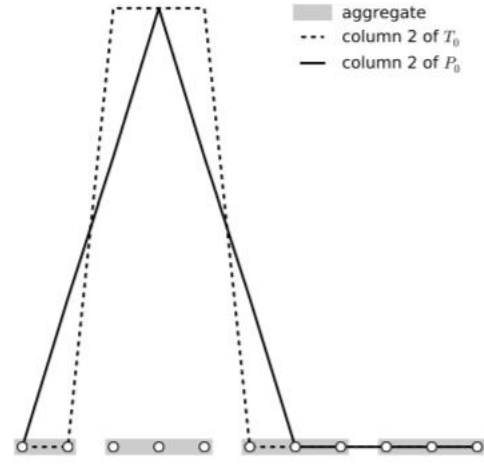
$P_k \leftarrow \text{smooth}(A_k, T_k)$  {Improve interpolation operator}

$A_{k+1} = P_k^T A_k P_k$  {Construct coarse operator}

---



(a) Aggregation of nodes on a mesh.



(b) Column of  $T$  and  $P$  on an aggregate.

Рисунок 2. Алгебраический многосеточный метод на основе сглаженной агрегации в 2D и в 1D.

### Практические соображения

Алгебраические многосеточные методы обычно используются в качестве предобуславливателей - например, для перезапущенный GMRES или метод сопряженных градиентов Крылова - приводящий к снижению количества итераций. Однако общая стоимость предварительно обработанной итерации требует оценки как коэффициента сходимости  $\rho$ , так и объёма работы в каждом многосеточном цикле.

Чтобы измерить объем работы в V-цикле используется так называемая операторная сложность иерархии

$$c_{op} = \frac{\sum_{k=\rho^m} nnz(A_k)}{nnz(A_0)},$$

При этом оценивается общая работа на цифру точности  $c_{op} / \log_{10} \rho$ . Это связывает стоимость цикла AMG со стоимостью стандартного разреженного матрично-векторное умножение. Это также показывает соотношение цены и точности

в AMG, однако, это может быть «скрыто», если стоимость этапа настройки не включена.

И в AMG на основе C/F, и в AMG на основе SA оператор интерполяции играет большую роль как в эффективности, так и в сложности алгоритма. В каждом случае, интерполяция может быть обогащена - например, путем расширения шаблона интерполяции или путем увеличения  $B$  в случае SA - приводит к более быстрой конвергенции, но более дорогостоящим итераций. Есть несколько способов, которыми AMG был расширен или переработан чтобы повысить устойчивость к более широкому кругу проблем или повысить его эффективность. Например, адаптивные методы [4; 5] попытка построить улучшенную иерархию путем изменения фазы установки на основе его производительности на  $Ax = 0$ . Другие работы сосредоточены на отдельных компонентах, таких как обобщение силы связи [12] или огрубление, такое как работа совместимой релаксации [9], где грубые сетки выбираются непосредственно посредством релаксации. И новые методы продолжают появляться как теория поддерживающая АМГ становится более развитой и обобщенной [6].

**Samg user manual**

Для эффективного численного решения больших систем дискретных эллиптических уравнений с частными производными (ДУРЧП) требуются иерархические алгоритмы, обеспечивающие быстрое уменьшение как коротких, так и длинноволновых ошибок. Прорыв, и, безусловно, один из самых важных достижений за последние три десятилетия, был связан с появлением многосеточного принципа. Любой соответствующий метод действует не только на данной сетке дискретизации, а скорее на иерархии сеток, определенных априори путем огрубления данной сетки геометрически естественным образом («геометрическая» многосетка). С начала девяностых наблюдается сильный рост интереса к алгебраически ориентированным иерархическим методам которые автоматически строят разумную иерархию на основе алгебраической информации (явно или неявно) содержится в матрице дискретизации), а не сетка информации. Одной из причин этого, безусловно, является возрастающая геометрическая сложность приложений, которая технически ограничивает непосредственное использование геометрических многосеточных методов. Другая причина - это постоянно растущий спрос на эффективные «подключаемые» решатели, которые могут быть интегрированы в существующие программные пакеты без необходимости полной реструктуризации кода. В частности в коммерческих кодах, это требование обусловлено увеличением размеров проблемы, которые явно демонстрируют пределы классические одноуровневые решатели до сих пор используются в большинстве пакетов. Миллионы степеней свободы в базовых численных моделях требуют иерархических подходов для эффективного решения.

**Алгебраический многосеточный метод.**

Первым иерархическим матричным подходом был алгебраический многосеточный (AMG), который расширяет классический идеи геометрической многосетки (основанные на сглаживании и грубой сетке) для некоторых классов линейных систем уравнений. AMG работает не по иерархии сеток, а по иерархии меньших линейных систем уравнений, которые построены полностью автоматически. В частности, строительство операторов (матрицы), используемые для передачи информации между различными уровнями (ограничение невязок и интерполяция поправок) основана на элементах матрицы, а матрицы на более грубых уровнях вычисляются на основе по так называемому принципу Галеркина (матрицы Галеркина). Этот автоматизм является основной причиной гибкости AMG в адаптации к конкретным требованиям решаемой проблемы и является основной причиной ее надежности в решении больших классов задач, несмотря на использование очень простых сглаживателей. Хотя происхождение AMG восходит к началу восьмидесятых годов [1-5], оно все еще является одним из наиболее привлекательных алгебраических подходов. Тем не менее, серьезные исследования в области AMG все еще продолжаются, в основном с целью расширения диапазона применимости. Основное исследование сосредоточено на приложениях, включающих связанные системы ДУРЧП для надежности и эффективности которые еще не были достигнуты в той же степени, что и для скалярных PDE. Тем не менее, был достигнут существенный прогресс, и в ближайшем будущем. Для некоторого обзора о состоянии техники, мы ссылаемся на [8]. Подробное введение в AMG представлено в В [7] результаты его распараллеливания найдены в [9]. Относительно методов и приложений для выбранной пары системы ДУРЧП, см. [14].

### **Samg пакет.**

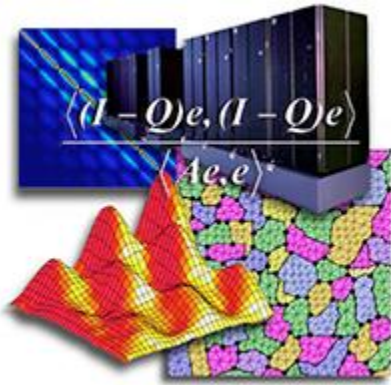
SAMG, написанный на FORTRAN90, основан на подходе AMG, который, по нашему опыту, оказался быть очень гибким, надежным и эффективным на практике. Основные компоненты SAMG были реализованы уже в коде RAMG [7], который является преемником исходного (общедоступного) кода AMG1R5, описан в [3]. Однако по сравнению с RAMG SAMG гораздо более общий. В частности, SAMG может применяться к как скалярные, так и связанные системы «эллиптических» ДУРЧП.

С точки зрения пользователя, SAMG является решателем «подключаемого модуля». По сути, только (разреженная) линейная система уравнения для решения,

$$Au = F,$$

должен быть передан в SAMG. В общем, нет информации относительно формы домена или структуры базовая сетка должна быть предоставлена. Таким образом, помимо его надежности и эффективности, основным практическим преимуществом SAMG заключается в том, что его можно напрямую применять для решения определенных классов ДУРЧП на неструктурированных сетки, как в 2D, так и в 3D. Более того, SAMG может даже применяться к задачам без какого-либо геометрического фона, при условии, что лежащие в основе матрицы имеют «подобный тип» как те, которые возникают из эллиптических ДУРЧП.

## SCALABLE-LINEAR-SOLVERS-COLLAGE-SM-LLNL.PNG



HYPRE библиотека линейных решателей из Ливермора делает возможным более крупные, более детальные моделирования при решении задач быстрее, чем традиционные методы в больших масштабах. Она предлагает полный набор масштабируемых решателей для крупномасштабного научного моделирования, показывая параллельные методы Многосеточный как структурированных, так и неструктурированных проблем сетки.

Библиотека HYPRE очень портативная и поддерживает несколько языков программирования.

Работа над HYPRE началась в конце 1990-х годов. С тех пор она была использована научно-исследовательскими институтами и частными компаниями для моделирования потока подземных вод, магнитные термоядерной энергии плазмы в токамаках и стеллараторах, поток крови через сердце, поток текучей среды в парогенераторах для атомных электростанций, а также насосную деятельность в нефтяных пластах, названию всего в нескольких областях. В 2007 году HYPRE выиграл R & D 100 Award от R & D Magazine как один из наиболее значимых технологических прорывов в год.

Команда HYPRE был одним из первых развивать алгебраические многосеточные алгоритмы и программное обеспечение для экстремального масштаба параллельных суперкомпьютеров. Команда ведет активную роль в многосеточной научного сообщества и признается за его ведущую роль в обоих алгоритмов и разработки программного обеспечения.

## 5 METHOD

Классический алгоритм AMG реализуется HSL MI20 подробно описана в разделе 7 книги [1]. Эта реализация из AMG игнорирует положительные внедиагональные записи во время огрубление, добавляет любой положительный вне диагональный коэффициент к диагонали, когда вычисления весов интерполяции, и использует прямую интерполяцию.

Метод AMG был первоначально разработан как линейный решатель, и несмотря на то что наша цель сделать из него эффективный переобуславливатель, логичнее при дальнейшем описании думать о AMG как о решателе при описании метода. Важно также отметить, что AMG может обеспечить хорошую предобуславливатель даже тогда, когда его использование в качестве чисто солвера (решателя) терпит неудачу (например, блок предобуславливанием внутри более крупной алгоритмической системы).

Рассмотрим линейную систему  $Au = f$ . В основе Многосеточный метод представляет собой серию когда-либо грубых представлениях полной матрицы  $A$ . Учитывая аппроксимацию  $f$  к решению  $u$ , рассмотрим решение  $Ae = g$ , чтобы найти ошибки  $e$ , где  $g$  невязка  $g = f - Af$ . Многосеточный применяет «сглаживающий итерационный процесс» (например, Гаусса-Зейделя или метод нижней релаксации Jacobi), чтобы удалить высокую частоту компоненты вектора ошибок. В этом случае задача может быть представлена меньшей (более грубой) системой  $A_c e_c = g_c$ , которую дешевле решить. Эта идея может быть применена рекурсивно, производя ряд грубых уровней и грубых исправления ошибок к решению. Грубый (наименьший) уровень решается с помощью прямого метода или простой итерационной схемы, такие как Гаусс-Зейделя. Решение грубый уровень должен затем быть продлен на каждом из уровней, более подробных сеток. При использовании в качестве линейного решателя (как, когда параметр `krylov solver` имеет значение 0), весь многосеточный процесс применяется итерационно до получения решения с желаемой точностью. Если это не получается есть способ который может быть использован для эффективного предобуславливания линейной системы. Как правило, предобуславливание является одной многосеточной итерацией. Использование многосеточного метода как предобуславливателя для ускорения сопряженных градиентов (cg

– congruate gradients), MINRES, GMRES или BiCGStab встроен в MI20 решить; смотрите соответствующий HSL документация для деталей этих методов.

Многосеточный метод требует некоторых средств производства коэффициент грубого уровня матрицы  $A_c$ , вместе со средствами передачи остаточных и ошибок векторов между уровнями; в AMG это придумали, как сделать чисто алгебраически не используя геометрическую информацию.

## 5.1 Setup

Грубый уровень матрицы  $A_c$  и интерполяция / ограничение матриц  $I_{cf}$  созданы на этапе установки, который включает следующие шаги.

### 5.1.1 Finding F and C points in HSL\_MI20

Для описания процедуры AMG огрубления, мы связываем строки матрицы  $A = \{a_{ij}\}$  с точками (так что строки  $i$  связаны с точкой  $i$ ) и рассмотрим связь между точками. Будем говорить, что точка  $i$  подключается к точке  $j$  если  $a_{ij} < 0$ .

Для создания следующего грубого уровня, точки разделены на точки C (те точки, которые будут существовать на следующем уровне), F точки (их значения должны интерполироваться из точек C), и не связанных между собой пунктов (C точки не связаны между собой – максимальное независимое множество). Это деление основано на понятии сильной (strong) связи. Если  $i$  подключен к  $j$  и  $|a_{ij}| \geq \theta \cdot \max\{|a_{ij}|: a_{ij} < 0\}$ , где  $0 < \theta < 1$  соответствует контролю управляющий параметр `control%st parameter i`, то мы говорим, что  $i$  имеет сильную связь с  $j$  и  $j$  имеет сильную транспонирование подключение к  $i$ .

**NOTE:** Параметр `pkgtype st parameter` используется при определении того, какие соединения являются сильными а какие слабыми (смотрите раздел 5.1.1 Детали). Значение по умолчанию равно **0.25**, но для некоторых приложений (*особенно в 3D*), это может быть выгодно использовать большее значение. Ограничение:  $0,0 \leq \text{st parameter} \leq 1.0$ .

После удаления несвязанных точек, каждый шаг процесса укрупнения протекает следующим образом. Каждая еще не рассмотренная точка имеет вес, который первоначально равен числу его сильных соединений



транспонирование. Еще не рассмотренная точка с максимальным весом выбирается так, чтобы стать новой точки С, а точки с сильной связи транспонированной к новой точке С стали F точки. Веса затем увеличиваются на число сильных связей с новыми F точками. Этот процесс повторяется до тех пор, пока все точки не будут назначены либо в виде точек F или С, или пока все остальные не определившихся точки имеют вес, равный нулю. Если параметр `control%one pass coarsen` равен `.FALSE.`, далее выполняется проверка, которая направлена на повышение качества укрупнение путем, возможно, делая дополнительные точки типа С. Если  $n_c$  и  $n_f$  являются число С и F точек, соответственно, укрупнение стагнирует, если  $n_c \geq n_f * \text{\_control\%reduction}$  и, в этом случае построение иерархии уровней прекращается.

**NOTE** : Огрубление `pkgtype.reduction` контролирует огрубление. Если два последовательных уровней имеют  $n_c$  и  $n_f$  точек, соответственно, то укрупнение продолжается, пока  $n_c \leq n_f * \text{reduction}$ . Снижение должно быть не менее 0,5 и не более одного 1. Значение по умолчанию значение равно 0,8.

В противном случае, укрупнение продолжается до тех пор либо запрошенную максимальное число уровней не было достигнуто (`control%max levels`) или число точек было уменьшено ниже выбранного порога (`control%max points`).

**NOTE:** `int max levels` имеет максимальное количество грубых уровней в многосеточном структуре, которая генерируется `mi20` установки. Значение по умолчанию равно 100. **Ограничение:** `max levels`  $\geq 1$ .

На протяжении огрубление, необходимо знать сильные связи (для F точек) и сильная транспонирование соединения (для точек С). Легко проверить, является ли подключение к точке сильны (для точки  $i$  данные, необходимые содержится в строке  $i$  матрицы, и это доступно, так как мы проводим матрицы при хранении сжатого строчного формата (CRS)). Тестирование для сильных соединений транспонировать не так просто. Для того, чтобы найти прочные связи транспонирования до точки  $j$ , мы должны знать, какие строки имеют ненулевую запись в колонке  $j$ . Для общей разреженной матрицы, состоявшейся в CRS формат, мы должны проходить через всю матрицу, проверяя каждую строку. Тем не менее, если матрица имеет симметричный разреженность шаблон, все проще, так как если строка  $i$  имеет запись в колонке  $j$ , строка  $j$  имеет запись в колонке  $i$ .

Для того, чтобы уменьшить количество ненулевых элементов в матрицах грубых уровней, весовые коэффициенты интерполяции могут быть усечены. Если управляющий `control%trunc parameter > 0.0`, вес интерполяции будут удалены из матрицы интерполяции, если их значение меньше или равно `control%trunc parameter` умноженный на наибольший вес интерполяции в их ряду интерполяция матрицы. После этого остальные веса масштабируются таким образом, чтобы суммы строк остается неизменной.

**NOTE:** `pkgtype` параметр `trunc parameter` контролирует усечение весов интерполяции. Значение по умолчанию 0.0 (вес интерполяции не усеченной (усечение не используется)). **Ограничение:**  $0,0 \leq \text{control.trunc параметр} < 1,0$ .

Для матриц с симметричным портретом разреженности, HSL MI20 предлагает два метода для нахождения сильного транспонирование соединения. Первый из них, выбирается путем установки `control%st method = 1`, , выполняет тестирование по мере необходимости; Этот метод часто быстрее. Альтернативный метод (`control%st method = 2`) находит сильные связи транспонирования перед укрупнением и затем сохраняет информацию; этот метод используется всегда, когда  $A$  имеет несимметричный портрет (рисунок) разреженности.

### 5.1.2 Generate interpolation matrix and $A_c$

Метод прямой интерполяции используется для расчета весов интерполяции, и это полностью описано в [1], при этом положительные внедиагональные коэффициенты эффективно удаляются путем их добавления к диагонали. Матрица грубый коэффициент уровня  $A_c$  генерируется из матрицы подробного уровня  $A_f$  и матрица интерполяции  $I_{cf}$  используя Галеркина соотношение  $A_c = I_{cf}^T * A_f * I_{cf}$ . Если `control%aggressive > 1`, , более чем на один шаг огрубление выполняется перед  $A_c$  вычисляется; это известно как **агрессивное** укрупнение.

## 5.2 Preconditioning phase

MI20 который используется в качестве предобуславливателя принимает предоставленный пользователем вектор  $z$  и возвращает  $x = Mz$ , где  $M$  AMG переобуславливатель. Переобуславливатель осуществляет `control%v`

iterations V-циклов. Обозначая матрицу интерполяции от уровня  $k$  уровню,  $k + 1$  как  $I_k^{k+1}$  и матрица на уровне  $k$  через  $A_k$ , то V-циклы выполняются в HSL MI20 следующим образом:

*User-supplied*  $z$ .

Initialise  $x=0$ ;  $A_1=A$ ; it  $s=0$ .

**do outer**

if (it  $s == \text{max\_it } s$ ) stop

$z_1 = z - Ax$

**do**  $k=1, ml-1$

Initialise  $e_k=0$

Pre-smooth  $e_k$  (using damped Jacobi or Gauss-Seidel)

Compute  $r_k = z_k - A_k e_k$

Restrict  $r_{k+1} = I_k^{k+1} r_k$

Set  $z_{k+1} = r_{k+1}$

**end do**

Solve the coarse grid error correction problem  $A_{k+1} e_{k+1} = z_{k+1}$

**do**  $k=ml-1, 1, -1$

Prolong and then update the error correction  $e_k \leftarrow e_k + (I_k^{k+1})^T e_{k+1}$

Post-smooth  $e_k$  (using damped Jacobi or Gauss-Seidel with the sweep direction reversed)

**end do**

Update  $x \leftarrow x + e_1$

It  $s = \text{it } s + 1$

**end outer**

Здесь `max_iters` и `ml` являются контрольные параметры управления `control%v iterations` и `control%max levels`, соответственно.

Более подробная информация о методе реализуется в рамках HSL MI20, вместе с результатами численных расчетов приведены в [2].

### 5.3 Solve Phase

Вызов `MI20_solve` решает линейную систему либо с помощью AMG в качестве итерационного метода, когда

`solve control%krylov solver = 0`, или с помощью метода подпространств Крылова с MI20 в качестве предобуславливателя.

Reference:

[1] K. Stüben. An introduction to algebraic multigrid. In U. Trottenberg, C. Oosterlee, A. Schüller, eds, 'Multigrid', Academic Press, 2001, pp 413-532.

[2] Boyle, Jonathan, Milan Mihajlovi, and Jennifer Scott. HSL MI20: an efficient AMG preconditioner for finite element problems in 3D. International journal for numerical methods in engineering 82.1 (2010): 64-98. APA

Система `HSL_MI20` несимметричный: алгебраические многосеточный переобуславливатель

Дано  $N \times N$  разреженная матрица `A` `HSL_MI20` имеет две функции.

Учитывая  $n$ -вектор `z`, `HSL_MI20` вычисляет вектор  $x = Mz$ , где  $M$  является алгебраическим многосеточный (AMG)  $V$  цикла предобуславливатель -  $A$ .

В качестве альтернативы, учитывая правая стороне  $n$ -вектор `b`, `HSL_MI20` решает линейную систему уравнений  $Ax = b$  с помощью метода AMG с или без ускорителя Крылова.

Классический метод AMG используется, как описано в [1] (см также раздел 5 ниже для краткого описания алгоритма). Матрица  $A$  должна иметь положительные диагональные элементы и (большую часть) вне-диагональные элементы должны быть отрицательными (диагональ должна

быть велика по сравнению с суммой вне- диагональных элементов). Во время процесса многосеточной огрубления, положительные вне- диагональные элементы игнорируются и, при расчете весов интерполяции, положительная вне- диагональные элементы добавляются к диагонали.

Точность: По крайней мере 8 байт арифметики рекомендуется (тип double).

Отказ огрубления:

`int c fail` , управляет критериям отказа огрубления.

Значение 1 указывает на то, что огрубление завершается, если **любая** строка матрица грубого уровня имеет по меньшей мере один строго положительный элемент, но нет отрицательных внедиагональных коэффициентов.

Значение 2 указывает на то, что укрупнение прекращает действовать, если **все** строки в матрице грубого уровня имеют по крайней мере один строго положительный запись и отсутствие отрицательных вне диагональных элементов или если недостаточное количество отрицательных внедиагональных коэффициентов вызывает огрубление потерпеть неудачу.

*по умолчанию 1. Ограничение:* С терпеть неудачу = 1 или 2.

## SUMMARY

Алгебраическое многосеточная (AMG) является одним из наиболее эффективных итерационных методов для решения больших, разреженных линейных системы, полученные из дискретизацией второго порядка скалярного эллиптического самосопряженного частными производными уравнения. Он также может быть использован в качестве предобуславливателя для методов подпространств Крылова. В этой связи, мы сообщаем о разработке и развитии надежной, эффективной и портативных Fortran 95 осуществления классический Рунге-Штубена AMG, который

доступен как пакет HSL MI20 в HSL математической ПО для создания библиотеки. Процедура может быть использована как прекондиционере "black-Box", но он также предлагает пользователю широкий выбор опций и параметров. Правильная настройка этих параметров для конкретного приложения может существенно повысить производительность на AMG-переобусловленного Крылова решателя. Это проиллюстрировано с помощью ряда примеры возникающих в неструктурированной конечных элементов дискретизацией диффузии, конвекции-диффузии, и уравнений Стокса, а также переходные проблемы тепловой конвекции, связанный с Буссинеском приближение Навье-Стокса в 3D.

## 1. INTRODUCTION

Математические модели важных физических систем и явлений часто выражаются в терминах дифференциальных уравнений (ДУРЧП). Численные алгоритмы решения ДУРЧП обычно исходят из некоторая форма дискретизацией непрерывной задачи, в результате чего в системе линейных или нелинейных уравнения. В большинстве случаев, матрицы коэффициентов являются большими, разреженными и плохообусловленными (смотри, пример [1, с. 57, 148]). Исследования в области разработки эффективных алгоритмов как прямой и итерационное решение таких систем привело к множеству надежных и эффективных методов (Введение обеспечивается стандартными текстами [2, 3]). Одним из желательных свойств линейного решателя оптимальное масштабирование времени выполнения и требуемые вычислительные ресурсы с дискретный размер проблемы, независимо от других параметров задачи. Современные разреженные прямые решатели являются не способными достичь оптимального масштабирования; в частности, памяти и числа операций для прямой Solver как правило, увеличивается гораздо быстрее, чем от размера системы. Это ограничивает пригодность прямые решатели для решения систем линейных уравнений, полученных из дискретизацией уравнения в частных производных (ДУРЧП) в трех пространственных измерениях.

Итерационные методы решения, такие как методы подпространств Крылова [3, глава 7], имеют оптимальные требования к хранению и вычислительные затраты на одну итерацию, таким образом предлагая потенциал, чтобы быть оптимальными решателями. К сожалению, общее число итераций для сходимости, как правило, не известны заранее и могут потенциально быть очень большим (это определяется спектральными свойствами матрица коэффициентов системы и ограничена размером линейной системы). В таких случаях, оптимальное свойство решателем теряется, хотя это может быть достигнуто (по крайней мере, приблизительно), если система надлежащим образом предобусловлена. Хороший переобуславливатель [1, раздел 2.2], [3, глава 10], [4, раздел 9.2] представляет собой матрицу (или линейный оператор), который является спектрально близок к коэффициенту системы матрица, но вычислительно дешевы, чтобы собрать и применить действие его обратного к вектору. Разработка оптимальных предобуславливателей неизбежно проблема конкретных; они были разработаны для широкого круга важных проблем в строительной механике [5, 6], механике жидкости [1, 7] и электромагнетизма [8, 9].

Важный класс оптимальных итерационных решателей для линейных систем, которые получаются из общие дискретизациях скалярных, эллиптические, самосопряжённых PDE основывается на многосеточных (MG) методов [10, 11]. В дальнейшем, мы предполагаем, что PDE дискретизируется подходящим способом, таким как метод конечных элементов (МКЭ) или метод конечных объемов (МКО), на заданном разбиении непрерывной области (Упоминаемый как сетки), что приводит к системе линейных уравнений. Размерность этой системы пропорционально числу точек (узлов) в сетке, с каждой дискретной неизвестность ассоциированной с точки сетки. MG использует наблюдение, что простые итерационные методы, такие как Гаусса-Зейделя (GS) или метод нижней релаксации Jacobi (Down relaxation Jacobi) [3, глава 4] (которые также упоминаются как методы релаксации

или сглаживателей) способны уменьшать только определенные компоненты погрешности решения, а именно: высокочастотные (колебательная) компоненты [10, глава 2]. Такое поведение является следствием их локальный характер, обновление каждого неизвестного влияет только очень немногие соседних неизвестные. С другой стороны, эффективное сокращение медленно меняющейся низкой частоты (или гладкая) компоненты погрешности решения требует глобальной коммуникации между всеми неизвестными в системе. Гладкие компоненты ошибки будут становиться все более колебательное, если представлены на сетки, которые грубее, чем сетка, на которой дискретизована исходная проблема. Таким образом, основная концепция геометрическая многосеточный (GMG) является снижение, за счет применения метода релаксации, решение компоненты ошибок, которые считаются высокочастотном на текущей сетке, а затем выбрасывающего остальные ошибки в более грубой сетки. На этой сетке, некоторые из компонентов, ошибка будет дополнительно снижается релаксацией, перед другим проекцией на еще более грубой сетки. Реализация GMG требует знания геометрической информации для вложенной иерархии прогрессивно грубее сетки, дискретное представление задачи на этих сетках, а также средства связи информация между сетками. Эти требования часто трудно удовлетворить. Например, при моделировании методом конечных элементов реалистичных задач в промышленных приложениях включает в себя сложную пространственную область геометрические формы, часто с внутренними границами. В таких случаях неструктурированных сетках или композитный адаптивно утонченные сетки, используются для которых не представляется возможным или удобным для генерации вложенной последовательность грубых сеток [10, с. 172]. В попытке преодолеть эти трудности при сохранении выгоды от multigrid концепции, алгебраические многосеточный (AMG) была введена [12]. В AMG, последовательность последовательно более грубых (более мелких) проблем порождается автоматическим укрупнением процедура. Эвристика этой процедуры



основаны на свойствах дискретного скалярного, эллиптические, самосопряженные операторы и не требуется геометрическая информация о сетке. Это значительно расширяет применение многосеточного подхода и даже позволяет использовать его на проблемы, которые непосредственно не связанных с сеткой. Есть целый ряд различных стратегий огрубления, используемых в данном контексте, в том числе и классический Рунге-Штубена огрубление и его варианты [13], сглаженная агрегации (SA) AMG [14], и агломерация AMG [15, 16]. Есть обобщения огрубление Рунге-Штубен Процедура направлена на решение систем уравнений в частных (см раздел 2.4 для получения более подробной информации). Классическая процедура огрубления Рунге-Штубен можно распараллелить, используя разложение домена [17].

Цель этой статьи имеет два аспекта: введение более широкой инженерного сообщества новый код HSL MI20 в математической библиотеки программного обеспечения HSL [18] и продемонстрировать как правильный выбор параметров в пределах HSL MI20 приводит к весьма эффективным и надежным AMG-выдержано Крылов решатели для скалярных эллиптических задач в области 3D и AMG на базе блока переобуславливатели для систем ДУРЧП, которые возникают в механике жидкости (задача Стокса и Моделирование переходного процесса тепловой конвекции проблем задачи Буссинеска). HSL MI20 вычисляет предобуславливатель AMG, используя классическую процедуру огрубления Рунге-Штубен.

Работа организована следующим образом. В разделе 2 мы вводим основы классической AMG метод. В разделе 3 описаны основные особенности кода HSL MI20. Раздел 4 сообщает численные Результаты для Крылова решателей AMG обусловлена (используя HSL MI20) при применении к дискретным диффузии и конвекции-диффузии в 3D и, в рамках блока предобуславливания, к задаче Стокса в 3D и переходных уравнений

Буссинеска в 3D. Проблема дискретизируется МКЭ с использованием неструктурированных тетраэдрических сеток. Для стандартной модельной задачи, мы представим краткое сравнение производительности между HSL MI20 и хорошо зарекомендовавших кодов AMG BoomerAMG [19] (часть библиотеки hypre [20]) и ML (часть библиотеки Trilinos [21]).

И, наконец, в разделе 5, мы суммируем наши выводы.

## **1. ALGEBRAIC MULTIGRID (AMG)**

В этом разделе мы представляем краткий обзор принципов и методов, развернутых в классической Рунге-Штубена метод AMG [22]; Этот подход реализован в нашем новом коде HSL MI20. Основным ингредиент любого алгоритма GMG является существование иерархии вложенных сеток [23]. Такая иерархия может быть получена, исходя из первоначальной грубой сетки, и путем добавления последовательно более подробных Уровни дискретизация (например, с помощью вложенного или невложенных глобального и локального измельчения сетки). Тем не менее, многие практические проблемы, определяются на доменах со сложной геометрией и внутренними границами, или есть решения, которые демонстрируют мелкомасштабные явления, такие как ударов, особенностей, и пограничные слои. В таких случаях, неструктурированная сетка или набор из композитного блока структурированных Сетки [10, с. 172] должны решать все особенности области и решения проблемы. Таким образом, создание вложенной последовательности прогрессивно более грубых сеток трудно (если, конечно, возможно вообще). Кроме того, точный учет геометрической информации, связанной с такой сеткой иерархия требует сложных структур данных и отнимает много памяти. Это может негативно влиять на производительность программного обеспечения, которое реализует их. Кроме того, некоторые практические проблемы приводят к большим разреженным линейным системам, которые не связаны с сетчатыми

дискретизациям. Применять принципы MG, подходящая 'сетка' метод укрупнение необходимо, что в явном виде не использовать какие-либо геометрическая информация. В основе AMG представляет собой последовательность прогрессивно меньше (грубее) представления линейной матрицы коэффициентов системы. Такой метод основан на эвристике, развился из чисто алгебраических соотношений между неизвестными в матрице системы. AMG Алгоритм требует определенных компонентов, а именно, учитывая дискретный оператор  $A = A_1$  на самом подробном уровне, последовательность операторов грубых уровней  $A_\ell$ ,  $\ell = 2, \dots, L$  ожидается ( $L-1$  является количество грубые уровни), вместе с методом релаксации, последовательность интерполяционных операторов передачи  $P_\ell^{\ell+1}$  и  $P_{\ell+1}^\ell$  и решатель для линейной задачи на уровне  $L$  грубосеточное строительство этих компонентов основывается на двух фундаментальных эвристики: алгебраически гладких ошибок и принцип сильной связи.

### *1.1. Two fundamental AMG principles*

Рассмотрим линейную систему

$$Ax = b \quad (1)$$

где  $A = \{a_{ij}\} \in \mathbb{R}^{n \times n}$  является матрица коэффициентов,  $x = \{x_i\} \in \mathbb{R}^n$  является вектор неизвестных, и  $b = \{b_i\} \in \mathbb{R}^n$  является данный вектор правой части. Если вычисленное решение обозначим через  $\tilde{x}$ , мы можем определение погрешности решения будет  $e = x - \tilde{x}$  и невязка быть  $r = b - A\tilde{x}$ . Ошибка решения и невязка связаны уравнением остаточного

$$Ae = r \quad (2)$$

Эффективность MG в снижении погрешности решения основано на двух взаимосвязанных числовых Процедурах: с фиксированной точкой итерации (также называемый сглаживающий или релаксации метод), и грубый Коррекция сетки. Если ошибка решение разлагается на отдельные компоненты Фурье [10, глава 2], можно показать, что с фиксированной

точкой итерации является эффективным в снижении его высокой частоты (осциллирующий) компоненты, в то время как коррекция грубой сетки снижает низкочастотные компоненты. В GMG, то Ошибка решения, полученный после применения нескольких шагов с фиксированной точкой итерации (сглаживающего) Алгоритм выглядит геометрически гладкая, и грубые сетки выбраны с целью представления эта ошибка с достаточной точностью. В AMG, нижележащей сетки явным образом не известна, и, таким образом, Фурье-анализ не может быть использован для определения гладкости погрешности решения. Вместо этого мы сначала выбрать процедура релаксации, а затем определить алгебраически гладкой, чтобы ошибка была гладкой, которая не может быть уменьшена эффективно релаксации. Обратите внимание, что алгебраически гладкая ошибка не может быть гладкой функция в геометрическом смысле (см [10, стр. 140]). Алгебраически гладкая ошибка характеризуется малыми остатками, то есть,

$Ae \approx 0$	(3)
----------------	-----

Другим важным свойством алгебраически гладкой ошибки является то, что его значение в любой момент может интерполировать точно по средневзвешенным своих соседей (соседних неизвестных данного неизвестного  $x_i$  определяются ненулевых элементов  $a_{ij}$   $i \neq j$  в  $I$ -й строки в матрице  $A$ ). Это свойство имеет значение при определении процедуры интерполяции. Для уменьшения гладких ошибок. Кроме того, они должны быть представлены меньшим уравнением (грубый уровень) дефект формы (2), что дешевле, чем решение исходной задачи (1). Коррекция грубого уровня является процесс передачи остаточного  $g$  от штрафа до грубого уровня, решая уравнение дефекта (точно или приблизительно) и проецирование (интерполяцией) коррекции  $e$  обратно в подробном уровне. Целый Процесс может быть применен рекурсивно путем введения иерархии грубых уровней и путем решения Уравнение дефекта примерно на каждом уровне (на грубом одной стороны, это может быть решена точно).

В линейной системе (1), каждое уравнение  $i$  представляет собой ограничение однозначно определять неизвестно  $x_i$ . Хотя все уравнения необходимы для вычисления  $x_i$  точно, то можно определить из  $i$ -го уравнения множество соседних неизвестных  $x_j$  ( $j \neq i$ ), что наибольшее влияние неизвестный  $x_i$ . Эвристически, они являются неизвестными  $x_j$ , для которых абсолютное значение соответствующие записи  $a_{ij}$  велики по сравнению с другими недиагональных элементов в  $i$ -го уравнения. Такие неизвестные  $x_j$  являются хорошими кандидатами для определения значения  $x_i$  путем интерполяции, то есть, они являются кандидатами на грубых точках уровня. Этот Эвристическое аргумент может быть формализован путем определения неизвестная  $x_i$  быть сильно связан со всеми неизвестными  $x_j$ , для которого

$$-a_{ij} \geq \theta \max \{-a_{ik}, k \neq i\} \quad (0 < \theta < 1) \quad (4)$$

т.е.  $x_i$  сильно связан со всеми неизвестными, чьи записи  $a_{ij}$  сравнимы по величине с самым большим внедиагональной коэффициентом в  $i$ -го уравнения. В противном случае,  $x_i$  считается слабо подключен к  $x_j$ . Обратите внимание, что в выражении (4) предполагается, что все недиагональные элементы в  $A$  являются отрицательными, т.е.  $A$  является  $M$ -матрицей [3, с. 28]. Если  $x_i$  сильно (соответственно, слабо) соединены до  $x_j$ , то  $x_j$  называется сильно (соответственно, слабо) влияние  $x_i$ . Параметр  $\theta$  в формуле (4) упоминается как сила пороговой зависимости, и (4) представляет собой формальное выражение силы принципа зависимости. Важным свойством гладкой ошибки является то, что она медленно меняется вдоль сильные связи.

## 1.2. *The interpolation operator*

Предположим, что неизвестные  $x_i$  были разбито на  $n_c$  грубой (C) и  $n_f$  подробной (штрафа) (F) Неизвестными уровня. С помощью этого разделения предполагается, что неизвестные C сформирует следующий грубый уровня, а F неизвестные будут иметь свои значения интерполируются от неизвестных

С. Определим окрестности каждого штрафа уровня неизвестной  $x_i \in F$ , как множество неизвестных  $N_i = \{x_j \mid j \neq i, a_{ij} \neq 0\}$ .

Каждая неизвестная принадлежность к  $N_i$  либо грубый уровень неизвестен, что сильно влияет на  $x_i$  ( $C_i$ ), неизвестный штраф уровень, который сильно влияет на  $x_i$  ( $D_i^s$ ), или либо  $C$  или  $F$  неизвестно, что слабо влияет на  $x_i$  ( $D_i^w$ ).

Очевидно

$N_i = C_i \cup D_i^s \cup D_i^w$	(5)
-----------------------------------	-----

Оператор интерполяции  $P \in \mathbb{R}^{n^F \times n^C}$  затем покомпонентно определяется следующим образом:

$(Pe)_i = \begin{cases} e_i & \text{if } x_i \in C \\ \sum_{j \in C_i} w_{ij} e_j & \text{if } x_i \in F \end{cases}$	(6)
---	-----

где  $W_{ij}$  являются неизвестными весами интерполяции. Эти веса получаются из уравнения (3), которая может быть выражена в покомпонентном виде

$a_{ii} e_i \approx \sum_{j \in N_i} a_{ij} e_j$	(7)
--	-----

Используя (5), и добавление вкладов от  $D_i^w$  до диагональных членов (см [10, стр. 143] больше подробности), выходы

$w_{ij} = - \frac{a_{ij} + \sum_{m \in D_i^s} (a_{im} a_{mj} / \sum_{k \in C_i} a_{mk})}{a_{ii} + \sum_{n \in D_i^w} a_{in}}$	(8)
---	-----

Выбор (8) интерполяции весов  $W_{ij}$  гарантирует, что постоянные функции интерполируются точно (Матрица  $P$  имеет свои строки суммы, равные 1). Приближения, участвующие в выводе Формула (8), если предположить, что  $x_i$  и  $x_j$  сильно связаны  $F$  неизвестных, должно быть, по крайней мере одну общую точку в своих соответствующих грубых множеств  $C_i$  и  $C_j$  (см [10, стр. 143-144] для более подробной информации). Оператор сужения  $R$  в AMG

обычно выбирают транспонированный оператор интерполяции, то есть  $R = P^T$ .

### 1.3. Coarse level operators

После того, как операторы интерполяции  $P_{\ell-1}^\ell$  между каждой парой уровней  $\ell-1$  и  $\ell$  были собраны, операторы грубого уровня (матрицы)  $A_\ell, \ell = 2, \dots, L$  ( $A_1 = A$ ), построены рекурсивно с использованием проекция Галеркина

$A_\ell = (P_{\ell-1}^\ell)^T A_{\ell-1} P_{\ell-1}^\ell$	$(9)$
---	-------

В качестве интерполяции матрицы  $P_{\ell-1}^\ell$  имеет полный ранг, любые свойства матрицы  $A_{\ell-1}$  (Например, симметрия, положительная определенность) будет унаследован его грубое представление  $A_\ell$  [3, с. 423]. Заметка, тем не менее, к которым относится данная функция только тогда, когда коэффициент матрицы  $A$  возникает в дискретизаций скалярные ДУРЧП и не распространяется на системы уравнений в частных производных.

### 1.4. The coarsening procedure

Процесс сборки операторов интерполяции предполагает разбиение  $\{x_i\}$  в грубые и подробные неизвестными уровня. Критерии этого разделения основаны на принципах AMG гладких ошибок и сила зависимости. В GMG, выбор грубой сетки влияет следующие (взаимоисключающие) требования:

- (I) гладкие ошибки должны быть аппроксимирована точно,
- (II) гладкие функции должны быть интерполированы от грубой до подробной сетки точно, и
- (III) размер каждой грубой сетки должна быть значительно меньше, чем у предыдущей подробной сетки чтобы обеспечить вычислительную эффективность.

Обозначим через  $S_i$  множество неизвестных, которые сильно влияют на  $x_i$ , и  $S_i^T$  множество неизвестных которые прочно связаны с  $x_i$ , на основе эвристики (4). Процедура AMG огрубление для определение множества  $C$  и  $F$  (введенный в § 2) основано на двух эвристических критериев:

**H1.** Для каждого  $x_i \in F_i$ , каждого неизвестного  $x_j \in S_i$  должно быть либо в  $C_i$  или быть тесно связано с по меньшей мере, одно неизвестное в  $C_i$ .

**H2.** Множество  $C$  должно быть максимальное подмножество всех неизвестных с тем свойством, что нет двух неизвестные от него сильно связаны друг с другом.

Для того, чтобы сделать приближения, приводящие к формуле интерполяции (8), каждая из неизвестного  $x_j \in D_i^S$  должны быть прочно соединены, по крайней мере, с одной неизвестной в  $C_i$ ; Эвристический **H1** обеспечивает это. В этом случае, неизвестные  $x_j \in D_i^S$  которые не выбраны, чтобы быть  $C$  неизвестными, представлены в интерполяции по формуле (8) через их значения, которые распределены в точках  $C_i$ . Эвристический **H2** обеспечивает баланс в размерах грубых уровней. Если число неизвестных  $C$  не намного меньше, чем число неизвестных на уровне подробной, интерполяция, вероятно, будет точной и AMG будет больше эффективным, но за счет увеличения стоимости вычислений. Эвристический **H2** контролирует размер самого грубые уровни, требуя, чтобы не существует сильная зависимость между неизвестными  $C$ , при этом сохраняя неизвестные величины  $C$  при достаточном расстоянии " друг от друга. С другой стороны, требование множество  $C$  должно быть максимальное подмножество обеспечивает качество интерполяции.

Требования **H1** и **H2**, как правило, взаимно исключают друг друга. На практике, **H1**, как правило, применяются строго говоря, в то время как **H2** выполнено приблизительно (обычная компромисс между точностью и эффективностью).

Такой подход, однако, приводит к грубым уровням, которые часто больше, чем это необходимо. Для 2D задач это, как правило, выгодно, но и для 3D



задач дискретизированных методами конечных элементов или конечных разностей, слабое исполнение **H2** может привести к значительному увеличению размера грубых уровней, и, следовательно, время выполнения. В недавнем исследовании [24] показывает, что грубые уровни с более низким сложностями (т.е. с меньшим числом неизвестных, чем полученная путем строгого соблюдения **H1**) могут по-прежнему дает хорошие скорости сходимости, когда AMG используется в качестве предобуславливателя к алгоритму сопряженных градиентов, а не как решатель. Мы нашли ту же тенденцию в наших численных исследований для скалярных эллиптических задач в 3D (см. Результаты, приведенные в разделе 4). Эта тенденция распространяется и на случай исследования систем уравнений в частных производных.

Стандартный алгоритм огрубление использует два прохода данных. При первом проходе, предварительный разбиение на C и F неизвестных делается с целью обеспечения соблюдения **H2**, сохраняя при этом качество интерполяции. Во-вторых, некоторые из первоначальных F неизвестных становятся C неизвестными, чтобы строго следить за соблюдением **H1** (и, таким образом, нарушают **H2**). Если второй проход не выполняется, укрупнение Стратегия упоминается как один проход огрубления (RS1); в противном случае, у нас есть два прохода укрупнение (RS2). Эффективность обеих стратегий оценивается в Разделе 4.

В последнее время, обобщения стандартных методов укрупнения Руге-Штубена были введены для применения AMG к системам уравнений в частных производных. Классическая процедура огрубления AMG базируется от аппроксимации переменных, при котором все неизвестные обращались одинаково. Если не очень слабая связь между неизвестными, это не работает эффективно для систем ДУРЧП и некоторые модификации и расширения необходимы для достижения хорошей сходимости и устойчивости (см [25]). Там существует несколько обобщений подхода на основе переменной. Неизвестный подход, основанный, где различные типы неизвестных

рассматриваются отдельно, это самый простой способ обобщающих AMG для систем уравнений в частных производных. Это эффективно работает, если диагональные блоки близки к М-матриц (например, для системы нескаларных ДУРЧП, такие как линейной упругости [26]). Дальнейшие обобщения (точка на основе Подход) были введены в коде SAMG [27]. Здесь все неизвестные одни и те же сетки иерархии, и процесс огрубление основан на вспомогательной матрице, называется первичным матрица. Эта матрица должна быть репрезентативной паттернов подключения всех неизвестных система. Хотя значительный прогресс был достигнут в разработке AMG для систем ДУРЧП, до сих пор ни один метод, который работает универсально хорошо для широкого класса задач. Некоторые конкретные реализации, однако, работают хорошо для некоторых важных применений, таких как механики жидкости, многофазный поток в пористых средах [28], строительной механики и полупроводниковых процессов и моделирования устройства [29, 30].

### *1.5. The algorithm complexity*

Многосеточные методы являются привлекательными в качестве переобусловливателей, потому что, при использовании в сочетании с Крыловым решатели, они стремятся решать большие разреженные линейные системы порядка  $n$  за  $O(n)$  операций, демонстрируя тем самым (Близкое к оптимальному) масштабирование. Это оптимальность достигается за счет (почти) постоянным числом итераций нужности для получения заданной точности, независимо от размера задачи или любой другой проблемные параметры, и оптимальной расчетной стоимости каждой итерации. Больше, чем необходимо операторов грубого уровня увеличивать вычислительные затраты итераций AMG и, в крайнем случае, может подорвать алгоритм оптимального свойства масштабирования.

Чтобы дать нам возможность контролировать как выбор стратегии укрупнения (RS1 или RS2) и сила порога зависимости  $\theta$  из (4) влияют на

качество огрубление и размер грубого операторы уровня, мы вводим стандартные количественные показатели: сложности сетки  $C_G$ , оператор сложность  $c_A$ , а средний размер трафарета  $c_s$ .

Сложность сетки  $C_G$  определяется как

$c_G = \frac{1}{n_1} \sum_{\ell=1}^L n_\ell$	(10)
--	------

где  $n_\ell$  это размер матрицы  $A_\ell$  ( $\ell = 1, \dots, L$ ), и  $A_1$  подробной матрицы сетки. Если в 3D дискретизация задачи на последовательности вложенных друг в друга, равномерно утонченных сеток, а также полный укрупнение является используется (обычный случай в GMG), то  $C_G = 8/7$  (См [10, с. 45-48]). Полная огрубление униформе Сетка предполагает сокращение количества точек сетки в каждом пространственном направлении с коэффициентом 2. Большое расхождение от этого значения указывает на сильное нарушение эвристического **H2**.

Сложность оператора  $c_A$  определяется как

$c_A = \frac{1}{nnz(A_1)} \sum_{\ell=1}^L nnz(A_\ell)$	(11)
--	------

где  $nnz(A_\ell)$  число ненулевых записей в  $A_\ell$  ( $\ell = 1, \dots, L$ ).  $c_A$  обеспечивает индикацию требования к хранению AMG (хранение матриц интерполяции не входит). В случае полной огрубление задачи, определенной на последовательности вложенных друг в друга, равномерно утонченных сеток, и прямое дискретизация ДУРЧП на каждом уровне (подход, как правило, принимается в GMG),  $c_A$  приближенно равно  $c_G$ . В AMG, разница в этих двух параметров может происходить (особенно для 3D-задач), что указывает на матрицы грубого уровня имеют, в среднем, гораздо больших размеров шаблона аппроксимации (количество ненулевых элементов для каждой строки), чем подробной матрицы сетки  $A_1$ . Это также означает, что матрицы

интерполяции достаточно плотные, что делает ограничение и операции интерполяции более дорогим применением. К тому же, применение GS сглаживанием становится все более дорогим. Значение  $C_A$  может быть уменьшено (за счет уменьшения эффективности переобусловливателя) путем повышения прочности зависимости порог  $\theta$  (Раздел 4).

Средний размер трафарета  $C_s$  определяется как

$c_s = \frac{1}{L} \sum_{\ell=1}^L \frac{nnz(A_\ell)}{n_\ell}$	(12)
--	------

Эта величина должна быть сравнима со средним размером трафарета исходной задачи  $C_s^1 = nnz(A_1) / n_1$ . Большие различия между  $C_s$  и  $C_s^1$  указывают, что матрицы грубого уровня имеют больше, чем трафареты исходной матрицы. Следует отметить, что из-за небольших вносов от грубейшие уровни,  $C_s$  могут представлять собой оптимистическую оценку (смотрите примеры в разделе 4).

### 1.6. AMG solver and preconditioner

Когда все компоненты алгоритма AMG собираются, фазы решения состоит из повторяя V-циклов, пока заданный уровень точности достигается. V-цикл, по существу, одинаков в обоих ГМГ и AMG и подробно описан в работе [10, с. 40-41]. Короче говоря, V-цикл состоит приближенного решения системы (1) небольшим числом (nit1) с фиксированной точкой итераций, и проекция остаточного до грубого уровня  $\ell$  используя оператор сужения. На каждом из них уровни  $\ell = 2, \dots, L$ , остаточное уравнение (2) решается (точечным итерации фиксированной для  $\ell = 2, \dots, L-1$  и точно или приближенно для  $\ell = L$ ). Процесс приближенного решения (1) и (2) по с фиксированной точкой итерации упоминается как пре-сглаживанием. Полученный в результате коррекции к решению должно быть спроецирована (с интерполяцией) для  $\ell = L, \dots, 1$  с помощью оператора интерполяции. Этот процесс

вводит в компоненты ошибки высокочастотные возмущения. Для их устранения на каждом уровне, нам нужно выполнить небольшое количество ( $\text{nit}_2$ ) сглаживающих поточечных итераций. Этот процесс упоминается как пост-сглаживание. V-цикл с шагом  $\text{nit}_1$  предварительно сглаживающих и  $\text{nit}_2$  пост-сглаживания шагов символически обозначается в V ( $\text{nit}_1, \text{nit}_2$ ).

Хотя наш новый пакет HSL MI20 может быть использован в качестве автономного AMG решатель для скаляра, эллиптические, самосопряженные задачи, нашей основной задачей было разработать эффективный "черный ящик" предобуславливатель для скалярных эллиптических задач, а также разрешить использование пакета в пределах блока предобуславливанием рамки для более сложных систем ДУРЧП, которые могут быть применены в сочетании с, например, Крылова решателей [3, глава 6], доступные в библиотеке HSL. Конструкция HSL MI20 обсуждается в следующем разделе.

При использовании стандартного V-цикла, то это обычно бывает, что AMG-выдержано Крылова солвер выполняет более надежно, чем автономный AMG решателя, особенно применительно к некоторым нетривиальные скалярные эллиптические задачи (например, анизотропной диффузии, диффузии с разрывными коэффициентами [23], проблемы конвекции с рециркуляционным полем конвективных и т.д.). Обратите внимание, что в последнее время K-цикл был введен [31], который направлен на повышение надежности AMG решатель с помощью Крылова ускорение (как правило, состоящие из небольшого числа итераций решателя Крылова) в каждой грубый уровень.

## 2. DESIGN OF HSL MI20

Учитывая разреженная матрица  $A \in \mathbb{R}^{n \times n}$  и вектором  $z \in \mathbb{R}^n$ , HSL MI20 вычисляет вектор  $y = \text{AMG}(A, z)$ , где  $\text{AMG}(A, z)$  обозначает приближенное решение линейной системы  $Ay = z$  по ряд AMG V-циклов. Матрица  $A$  (который может быть симметричным или несимметричным) должна в идеале быть "близка" к M-матрице, то есть, он должен иметь положительные диагональные элементы и в идеале (В большинстве случаев) недиагональ-

записей должен быть отрицательным. Кроме того, по диагонали не должна быть небольшой по сравнению с суммой модулей внедиагональных элементов [10, с. 138]. Сильное нарушение этих требования может привести к плохой сходимости характеристик AMG решателя / preconditioner. Для того, чтобы количественно оценить термин «близкий к М-матрицы», в разделе 4 мы сообщаем число положительных внедиагональных элементов в процентах от общего количества внедиагональными записей и величины крупнейшего внедиагонального элемента в  $A$ .

### *2.1. The user interface and derived data types*

HSL является Fortran библиотекой и HSL MI20 написана на Fortran 95. Одной из главных причин этого выбора было, потому что это позволяет намного более простой пользовательский интерфейс, чем если бы был использован Fortran 77. Обратный Интерфейс связи используется со следующими процедурами, доступными для пользователя:

MI20 setup принимает матрицу  $A$ , при необходимости проверяет его на наличие ошибок (дубликатов и вне диапазона записи не допускается), а затем генерирует все данные, необходимые для preconditioner AMG. То есть, он выбирает грубые уровни, строит операторы интерполяции, и строит грубую матрицы уровня. Матрица  $A$  должна быть предоставлена в сжатом формате разреженным ряд (CSR), то есть, ненулевые записей в должны храниться строка за строкой.

MI20 precondition выполняет операцию preconditioning  $y = AMG(A, Z)$ , где  $Z$  представляет собой поставляемого пользователем вектор (как правило, остаточный вектор, если AMG используется в качестве preconditioner Крылов решатель). Более один вызов MI20 precondition может быть сделано после того, как сделан один вызов MI20 setup.

MI20 finalize должны вызываться после всех других вызовов, для настройки MI20 setup и MI20 precondition завершения для данной задачи. Это освобождает все компоненты массива из производных типов данных, созданных HSL MI20.

Свежие ноябрь 2019. Просто тематика.

Мы представляем расширение хорошо известного классического алгебраического многосеточного алгоритма, где огрубление основано на выборе уравнения для неявно связанной системы давление-скорость, дискретизированной методом конечного объема (FV). Результирующая матрица коэффициентов линейной системы упорядочена по точкам, и для каждой ячейки FV имеется 4 уравнения: три компонента скорости и давление. Серьезная проблема ставится с точки зрения распараллеливания процесса последовательного укрупнения и расчет грубой матрицы уровня по проекции Галеркина. В этой статье мы представляем стратегию распараллеливания и выполнение блока - выбор AMG.

.....

Мы расширяем возможности и возможности уже существующего пространственно-параллельного метода, основанного на блоке сглаживания Якоби. Сначала сформулируем более подробный критерий пространственного укрупнения, который позволяет методу иметь дело с неструктурированными сетками и различными параметрами материала. Дальше мы исследовать приложение к вихретоковому уравнению, где нетривиальное ядро оператора curl вызывает серьезные проблемы. Это исправлено коррекция узлового вспомогательного пространства. Мы приступаем к определению скорости сходимости по локальный анализ Фурье и численные эксперименты. Наконец, мы представляем числовой эксперимент, который демонстрирует свои отличные свойства масштабирования.

Примерно в 2004 году тактовая частота процессора перестала расти из-за физических ограничений. Поэтому мы вместо этого испытали

экспоненциальный рост числа процессорных единиц. Это проявляется в современных процессорных кластерах с числом процессоров от 1000 до более десяти миллионов по состоянию на 2019 год. Между тем параллельные в пространстве приложения часто не могут эффективно использовать более нескольких тысяч процессоров (в зависимости от размера проблемы). Освоить эту вычислительную мощность требует массивно параллельных алгоритмов. Для моделирования задач зависящих от времени временное измерение предлагает потенциал для дальнейшего распараллеливания, однако классическая концепция с временным шагом является принципиально последовательной. В первую очередь для параболических эволюционных уравнений, параллельные по времени, методы были разработаны, чтобы снять это «проклятие последовательности». После первого применения для систем обыкновенных дифференциальных уравнений более 55 лет назад [24] расцвела целая семья параллельных по времени методов всех видов, см. [9] для исторического обзора несмотря на то, что метод Хакбуша [12] все еще был последовательным во времени, он заложил некоторые основы для параллельных по времени многосеточных схем и бросил вызов идее точного решения проблемы во времени. Любич и Остерманн использовали другой подход, представив многосеточный метод пространственно-временного типа для релаксации формы волны [19]. В глубине

Обсуждение методов релаксации формы волны можно найти в монографии [38]. Следующий вариант описан как пространственно-временная параллельная многосеточная релаксация формы волны с циклическое восстановление можно найти в [16]. Современный метод, похожий на наш, - многосеточное сокращение времени [7], которая была активно расширена



командой, сосредоточенной вокруг Лоуренса Ливерморского Национального Лаборатория [7, 6, 8, 18, 11, 27]. Дальнейшие результаты можно найти на веб-сайте проекта XBraid5. Другой очень активной группой является Jülich Forschungszentrum с разнообразными публикациями о параллельных по времени методах [33, 31, 34, 32, 36, 40, 26, 35, 20, 17, 25, 30, 3, 29, 4].

Метод, который может быть описан как предшественник следующего метода, является пространственно-временным мультисетка с точечной релаксацией [15]. Метод, использованный в этой статье [10], заменяет точечная релаксация с блочной релаксацией. Таким образом, это может быть описано как пространство-время многосетка с блочной релаксацией. Последние публикации показали, что этот метод может также применяться к стохастическим дифференциальным уравнениям [23] и в сочетании с методы изогеометрического анализа [14]. Это побудило нас расследовать заявление нашего метода в уравнение вихревых токов. Кроме того, мы обобщаем понятия чтобы расширить наши возможности для решения более сложных проблем.

Уравнения вихревых токов используются для изучения низкочастотных электромагнитных применений, таких как электродвигатели / генераторы, индукционные печи, неразрушающий контроль и многое другое. На практике гармонические по времени подходы часто предпочтительнее переходных модели из-за более низких вычислительных затрат. Однако гармонические по времени подходы борются с нелинейными материалами (эффектами насыщения) и движущейся геометрией. Как шаг для решения этих более сложных задач мы рассмотрим только линейный случай с постоянной геометрией в этой статье.

.....

Мультисетка является одним из наиболее эффективных методов решения эллиптических уравнений в частных производных. Он является алгоритмически оптимальным и устойчивым в сочетании с методами Крылова. Алгебраическая многосетка особенно привлекательна благодаря своей природе черного ящика. Однако это происходит за счет увеличения затрат на настройку, которые могут быть значительными в случае систем, в которых системная матрица часто изменяется, что затрудняет амортизацию затрат на установку. В этой работе мы исследуем несколько стратегий выполнения отложенных обновлений многосеточной иерархии, соответствующих изменениям в матрице системы. К ним относятся отложенные обновления, обновления значений без изменения структуры, обработка локальных изменений и полные обновления. Мы демонстрируем, что во многих случаях накладные расходы на построение иерархии AMG могут быть уменьшены для быстро меняющихся системных матриц.

.....

Плохое предобуславливание матрицы системы является общеизвестным усложнением методов погруженных конечных элементов и усеченного изометрического анализа. Элементы с небольшими пересечениями с физической областью дают проблемные собственные значения в матрице системы, что, как правило, ухудшает эффективность и надежность итерационных решателей. В этом вкладе мы исследуем спектральные свойства погруженных систем конечных элементов, обработанных методами типа Шварца, чтобы установить их пригодность в качестве сглаживателей в многосеточном методе. На основе этого исследования мы разработали геометрический многосеточный предобусловливатель для погруженных методов конечных элементов, который обеспечивает независимую от ячеек и не зависящую от разреза скорость сходимости. Этот метод предварительной

обработки применим к дискретизации высшего порядка и позволяет решать крупномасштабные погруженные системы с вычислительными затратами, которые линейно масштабируются с числом степеней свободы. Производительность предварительного кондиционера продемонстрирована для обычных базисных функций Лагранжа и для изогеометрических дискретизаций как с равномерными В-сплайнами, так и с локально уточненными аппроксимациями на основе усеченных иерархических В-сплайнов.

---

### **A robust and scalable multi-level domain decomposition preconditioner for multi-core architecture with large number of cores**

С развитием высокопроизводительных вычислений многоядерные и многоядерные системы стали общей чертой новых аппаратных архитектур. Необходимые усилия по программированию, вызванные введением этих архитектур, являются сложными из-за растущего числа ядер. Модели параллельного программирования, основанные на модели потока данных и парадигме программирования задач, предназначены для решения этой проблемы. Итеративные линейные решатели являются ключевой частью моделирования нефтяных пластов, поскольку они могут составлять до 80% общего времени вычислений. В этих алгоритмах стандартные методы предварительной обработки больших, разреженных и неструктурированных матриц, такие как неполная факторизация LU (ILU) или алгебраическая многосетка (AMG), не масштабируются на архитектурах с разделяемой памятью с большим количеством ядер. Представленные недавно предварительные условия многоуровневой декомпозиции доменов (DDML) кажутся как численно устойчивыми, так и масштабируемыми на новых архитектурах из-за их параллельного характера. В этой статье предлагается

параллельная реализация этих предварительных кондиционеров с использованием парадигмы программирования задач с моделью потока данных. Этот подход проверен на линейных системах, извлеченных из реалистичных симуляций нефтяных пластов. Это показывает, что при наличии соответствующего грубого оператора в таких предварительных кондиционерах метод имеет хорошие скорости сходимости, в то время как наша реализация обеспечивает интересную масштабируемость на многоядерных архитектурах.

.....

Мы предлагаем метод адаптивного алгебраического многосеточного (PC -  $\alpha$ AMG) покрытия траектории для решения линейных систем с лапласианами взвешенного графа, который также может быть применен к дискретизированным эллиптическим дифференциальным уравнениям в частных производных второго порядка. PC -  $\alpha$ AMG основан на несглаженной агрегации AMG (UA-AMG). Чтобы сохранить структуру сглаженной ошибки вплоть до грубых уровней, мы приближаем множества уровней сглаженной ошибки, сначала формируя непересекающееся с вершиной покрытие пути с путями, следующими за наборами уровней. Агрегации затем формируются путем сопоставления вдоль путей в покрытии пути. Таким образом, мы можем построить многоуровневую структуру с низкими вычислительными затратами. Предложенный PC -  $\alpha$ AMG предоставляет механизм для эффективной перестройки многоуровневой иерархии во время итераций и приводит к быстрому нелинейному многоуровневому алгоритму. Традиционно для UA-AMG требуются более сложные циклические методы, такие как AMLI-цикл или K-цикл, но, как показывают наши численные результаты, предложенный здесь PC -  $\alpha$  AMG приводит к почти оптимальному стандартному алгоритму V-цикла для решения линейные

системы с лапласианами взвешенного графа. Численные эксперименты для некоторых реальных графических задач также демонстрируют эффективность и надежность РС -  $\alpha$ AMG, особенно для плохо обусловленных графов.

**Введение.** Взвешенные графы часто используются как представления данных для описания широкого спектра областей применения, включая социальные, сенсорные, энергетические, и нейронные сети [3, 51, 21, 11]. Связанный граф Лапласиан естественно возникает в крупномасштабных вычислениях различных областей применения. Например, решение систем с лапласианами взвешенного графа является ключевым компонентом для решения ранжирования и проблемы с рекомендациями пользователей [20, 24, 14]. В [12, 13, 31] сходства белков рассчитываются путем решения графа лапласовых систем, связанных с сетями взаимодействия белков. Кроме того, эти сходства используются в кластеризации и маркировке белков. Кроме того, брак графа лапласиана и популярной информатики такие темы, как сверточные нейронные сети и тензорная декомпозиция доминирующая тенденция [18, 9, 26, 19, 10, 43, 30]. Продвинутое алгоритмы, которые адаптируют граф Свойства Лапласа для улучшения задач включают реконструкцию изображений, кластеризацию наборов данных изображений и классификацию [1, 37, 23, 28]. Для того чтобы эффективно решить граф Лапласа системы, алгебраическая многосетка (AMG) часто применяется. Стандартный метод AMG был предложен для решения дифференциальных уравнений в частных производных (ДУРЧП) и в основном включает в себя две части: сглаживание высокочастотных ошибок на подробных уровнях и устранение низкочастотные ошибки на грубых сетках [49, 47, 42, 5, 40, 41]. AMG доказано быть одним из самых успешных итерационных методов в

практических приложениях и многих методы AMG были разработаны для решения графовых лапласовых систем, таких как комбинаторная многосетка [27], худой AMG [32] и алгебраический многоуровневый предобуславливатель на основе сопоставлений / агрегаций [25, 38, 6, 15].

Традиционные методы AMG обычно строят многоуровневые структуры заранее и нет никакой взаимосвязи между оставшимися ошибками и грубыми схемами. Это приводит ко многим вычислительным недостаткам. Например, применяется велосипедная схема та же многоуровневая иерархия на этапе решения, даже когда скорость сходимости медленно, что неэффективно. Чтобы уменьшить вычислительную избыточность, адаптивность становится необходимым. Существенные усилия были предприняты для включения адаптивности в итерационные методы. Еще в 1984 году был предложен оригинальный адаптивный алгоритм AMG [5], который заложил основу для развития самообучения и начальной загрузки AMG.

Подход начальной загрузки AMG получил дальнейшее развитие в [35, 4] и начинается с нескольких тестовых векторов, в то время как адаптивные подходы обычно начинаются только с одного теста вектор. Режиссер теории сглаженной агрегации AMG (SA-AMG) разработал в [46, 45], Brezina et al. введен адаптивный SA-AMG, определяющий интерполяцию операторы, основанные на информации из самой системы, а не на явном знании околоядерного пространства [7]. В следующем году те же авторы далее предложил оператор-индуцированный подход интерполяции, который автоматически представляет гладкие компоненты [8]. В обеих этих работах основная идея состоит в том, чтобы в настоящее время построенный метод AMG (первоначально это просто простая релаксация) к тривиальное решение

$x = 0$  однородной задачи ( $Ax = 0$ ). Эти учения начните со случайного начального предположения. Таким образом, медленно сходящиеся ошибки выставляются после нескольких итераций процесс адаптивного укрупнения улучшается за счет точной интерполяции алгебраически гладких ошибок. МакЛахлан, Мантейфель и Маккормик дальнейшее развитие двухуровневой нелинейной адаптивной AMG на основе сокращения и достижение локальная сходимость и, возможно, глобальная сходимость в особых случаях [34]. Разработка метода несглаженной агрегации AMG (UA-AMG) было начато довольно недавно.

В [15] Д'Амбра и Василевский применили схему укрупнения, в которой используются совместимые взвешенное сопоставление, позволяющее избежать зависимости характеристик прочности соединения в определении как грубого пространства, так и операторов интерполяции. Там также есть были другие идеи о том, как обеспечить хорошее приближение к гладким ошибкам от диапазоны адаптивно построенных интерполяционных операторов. Однако в большинстве существующих работ используется допущение, что огрубление (агрегация или выбор грубых узлов) уже сделано и не зависит от правой части или ошибки распределение во время итераций. Это побуждает нас сосредоточиться на алгоритмах, которые используют гладкая ошибка не только для построения операторов интерполяции, но и для определения агрегаты. Далее мы следуем этой идее и в этой статье мы предлагаем адаптивный метод AMG на основе UA-AMG [2], где агрегаты формируются вдоль множества постоянства ошибки медленного схождения. Мы устанавливаем многоуровневую структуру, сначала находя покрытие пути [39], которое аппроксимирует наборы уровней гладкой ошибки, затем огрубляя пути так, чтобы структура сглаженной ошибки хорошо

представлена на грубых уровнях, и здание плавная ошибка в диапазоне оператора интерполяции для его эффективного устранения. Фаза настройки активируется, когда скорость сходимости становится медленной, что делает схему, адаптирующуюся к ошибкам с медленной сходимостью. Вместо использования AMLI- или Kcycles для стандартного UA-AMG [6, 22, 47] мы просто используем V-цикл и достигаем почти равномерная сходимости для модельных задач. Дизайн сочетания UA-AMG и V-цикл имеет преимущества в своей простоте и эффективности и поэтому является благоприятным для решение большой линейной системы с плохо обусловленными матрицами. В то время как существующие адаптивные методы должны сначала решить однородную проблему,  $Ax = 0$ , чтобы определить компоненты почти нулевого пространства и использовать эти знания для решения  $Ax = b$ , наш адаптивный метод решает  $Ax = b$  напрямую, для общей правой части  $b$ . Оставшаяся часть теста организована следующим образом. В разделе 2 мы обсуждаем путь алгоритм поиска обложки, который мы используем для огрубления. Наш основной адаптивный алгоритм AMG обсуждается в разделе 3. Числовые результаты приведены в разделе 4. Наконец, в разделе 5, Мы предоставляем заключительные замечания и перечисляем возможные направления в будущем.

1 декабря 2000 ddemidov

Аннотация.

Разреженные приближенные инверсии рассматриваются как сглаживатели для геометрических и алгебраических многосеточных методов. Они основаны на SPAI-алгоритме (Grote and Huckle, 1997), который строит разреженный приближенный обратный оператор  $M$  матрицы  $A$ , минимизируя



I-MA в норме Фробениуса. Это приводит к новой иерархии по сути параллельных сглаживателей: SPAI-0, SPAI-1 и SPAI ( $\epsilon$ ). Для геометрического многосеточного метода характеристики SPAI-1 обычно сопоставимы с характеристиками сглаживания Гаусса-Зейделя. В более сложных ситуациях, когда ни сглаживатели Гаусса-Зейделя, ни более простые сглаживатели SPAI-0 или SPAI-1 не являются адекватными, дальнейшее уменьшение  $\epsilon$  автоматически улучшает сглаживание SPAI ( $\epsilon$ ) там, где это необходимо. В сочетании со стратегией алгебраического укрупнения (Ruge и Stüben, 1987) полученный метод дает надежную, параллельную и алгебраическую многосеточную итерацию, легко настраиваемую даже не экспертом. Численные примеры демонстрируют полезность сглаживателей SPAI как в последовательной, так и в параллельной среде.

Существенными преимуществами сглаживателей SPAI являются: повышенная надежность, присущий параллелизм, независимость от упорядочения и возможная локальная адаптивность.

## ROBUSTNESS AND SCALABILITY OF ALGEBRAIC MULTIGRID [18]

### **Аннотация.**

Алгебраический многосеточный метод (AMG) в настоящее время возрождается в популярности, в связи с частичным резким ростом необходимости решать физические проблемы, поставленные на очень большие, неструктурированные сетки. Хотя AMG доказала свою полезность для различных типов проблем, она обычно не понимается насколько широка область применения метода. В этом исследовании мы демонстрируем, что диапазон применимости, в то же время описывая некоторые из последних

достижений в технологии AMG. Кроме того, в свете из императивов современных компьютерных сред мы также рассматриваем AMG с точки зрения масштабируемости. Наконец, мы показываем некоторые ситуации, в которых стандартный AMG не работает, и указываем текущие указания, предпринятые исследователями AMG, чтобы смягчить эти трудности.

**1. Вступление.** Алгебраический многосеточный метод (AMG) был впервые введен в начале 1980-х годов [11, 8, 10, 12] и сразу же вызвал значительный интерес [31, 27, 29, 28]. Исследования продолжались скромными темпами в конце 1980-х и начале 1990-х годов [18, 14, 20, 24, 19, 25, 21]. Однако в последнее время наблюдается значительный всплеск интереса к «классическому» AMG, как определено в [28], а также для множества других многоуровневых методов алгебраического типа [3, 16, 33, 6, 2, 4, 5, 15, 32, 17, 34, 35 36]. Во многом это возрождение исследований AMG связано с необходимостью решения все более крупных систем, с сотнями миллионов или миллиардами неизвестных на неструктурированных сетках. Размер этих проблем диктует использование крупномасштабной параллельной обработки, которая в свою очередь требует алгоритмов, которые хорошо масштабируются, так как размер проблемы увеличивается. Многосеточные методы хорошо известно, что они масштабируемы для эллиптических задач на регулярных сетках. Тем не менее, многие современные проблемы включают в себя чрезвычайно сложные геометрии, делая структурированные геометрические сетки чрезвычайно трудно реализуемыми, если не невозможно, использовать. Разработчики кода приложений превращаясь в возрастающем количестве в очень большие неструктурированные сетки, и AMG виден многие из них, как один из наиболее перспективных методов решения возникающий в этом контексте.

Это исследование состоит из четырех компонентов. Сначала мы рассмотрим производительность «классического» AMG по ряду проблем, имеющих регулярную структуру, с целью определения его надежности. Во-вторых, мы исследуем производительность AMG на том же наборе проблемы, но теперь с неструктурированными сетками и / или нерегулярными доменами. В-третьих, мы изучили алгоритмическую масштабируемость AMG, изучив его производительность на нескольких проблемы с использованием сеток увеличивающихся размеров. Наконец, мы вводим новый метод для вычисления весов интерполяции, и мы показываем, что в некоторых проблемных случаях это может значительно улучшить производительность AMG.

Наше исследование отличается от предыдущих отчетов о эффективности AMG (например, [28, 29]) в первую очередь благодаря нашему исследованию масштабируемости, нашему акценту на неструктурированных сетках и введение нового алгоритма для вычисления весов интерполяции. В разделе 2 дано описание некоторых деталей алгоритма AMG, чтобы обеспечить понимание результатов и последующего обсуждения. В разделе 3 мы представляем результаты AMG применительно к ряду симметричных скалярных задач с использованием конечных дискретизаций на структурированных и неструктурированных 2D и 3D сетках. AMG также тестируется на несимметричных задачах, как на структурированных, так и на неструктурированных сетках, и результаты представлены в Разделе 4. Испытана версия AMG, разработанная для систем уравнений, с упором на проблемы в упругости. Результаты обсуждаются в разделе 5. В разделе 6 мы представляем и сообщаем о тестах нового метода для вычисления интерполяции веса. Мы завершаем с некоторыми замечаниями в разделе 7.

**7. Выводы.** Потребность в быстрых решениях для многих типов проблем, особенно те, которые дискретизированы на неструктурированных сетках, является четким признаком того, что существует рынок для программного обеспечения с возможностями, которые предлагает AMG. Наше исследование здесь демонстрирует надежность AMG в качестве решателя по широкому кругу проблем. Наши тесты показывают, что он может быть расширен и что можно разработать надежные эффективные коды для проблем, которые очень трудно решить другими методами. AMG также показано имеет хорошую масштабируемость на модельных задачах. Эта масштабируемость имеет тенденцию ухудшаться отчасти с увеличением сложности проблемы, но факторы конвергенции остаются поддающимися решению даже в худшей из этих ситуаций.

Для Вики.

С начала 1990-х годов наблюдается растущий спрос на более эффективные методы решения крупных разреженных, неструктурированных линейных систем уравнений. Для практически актуальных размеров проблем классические одноуровневые методы уже достигли своих пределов, и необходимо было разработать новые иерархические алгоритмы, чтобы обеспечить эффективное решение еще больших проблем. В этой статье дается обзор первого иерархического и чисто матричного подхода, алгебраического многосеточного метода (AMG). Например, AMG может применяться, например, для эффективного решения различных типов эллиптических уравнений с частными производными, дискретизированных на неструктурированных сетках, как в 2D, так и в 3D. Поскольку AMG не

использует никакой геометрической информации, это «плагин», который может быть применен к проблемам без какого-либо геометрического фона при условии, что базовая матрица обладает определенными свойствами.

На этапе разработки оптимизационного резервуара 3DSL, оптимизированного по линии тока, amglr5 был эффективно использован для решения (7). Недавно созданная компания Streamsim (Сан-Франциско) хотела коммерциализировать этот симулятор и продолжать использовать amglr5 в качестве решателя. Поскольку amglr5 был свободно доступным исследовательским кодом, не было бы никакой проблемы, чтобы дать разрешение. Однако, согласно законодательству США, каждое соглашение требует «рассмотрения», что означает, что компания Streamsim должна предложить что-то ценное, пусть и маленькое, в обмен на предмет соглашения, здесь amglr5. Поскольку amglr5 был учебным кодом без какого-либо обслуживания или поддержки, был установлен официальный лицензионный договор и согласован лицензионный сбор в размере одного доллара (см. Рис. 3)



Рис. 3. Лицензирование бесплатно для amglr5: согласно законодательству США, каждое соглашение требует «рассмотрения», что означает что компания Streamsim должна предложить что-то ценное, сколь бы маленьким оно ни было, в обмен на предмет соглашения (здесь amglr5). Обратите

внимание: хотя чек выдается в день дурака, это не шутка. Опубликовано с любезного разрешения Fraunhofer SCAI 2016, Все права защищены.

Хотя этот сбор был действительно маргинальным, связанные усилия были более чем погашены. Никто не ожидал, что эта «лицензионная история» проложит путь для работы SCAI по моделированию оптимальной формы резервуара много лет спустя. Фактически, как только развитие SAMG началось несколько лет спустя, Streamsim стал первым «настоящим» клиентом.

## **Parallel Algebraic Multigrid**

**Hans-Joachim Plum, Arnold Krechel, Sebastian Gries, Bram Metsch, Fabian Nick, Marc Alexander Schweitzer, and Klaus Stüben**

### **Параллельная версия алгебраического многосеточного метода**

#### **1. Введение**

Типичные размеры проблем в промышленных приложениях сегодня варьируются от нескольких миллионов до миллиардов степеней свободы, которые не могут быть обработаны одним процессорным ядром или одним вычислительным узлом за приемлемое время. Очень большие проблемы фактически не могут быть решены вообще на одном узле из-за ограничений памяти. Таким образом, существует принципиальная потребность в параллельных программных средствах, использующих современные мульти- и многоядерные системы, а также кластерные компьютеры. Как правило, измерения производительности в отношении распараллеливания фокусируются на так называемых сильных и слабых свойствах масштабирования, которые, однако, представляют несколько больший академический интерес. Поскольку разработка SAMG сосредоточена на промышленном использовании, нас скорее должен волновать вопрос: как

минимизировать время для решения на фиксированных и ограниченных (но произвольных) аппаратных ресурсах для небольшого диапазона размеров проблемы? Следовательно, мы должны обеспечить большую гибкость в SAMG, чтобы обеспечить компромисс между требованиями к памяти и параллельной масштабируемостью по сравнению с общим временем до решения для фиксированного размера проблемы с использованием конкретной конфигурации оборудования. Эта гибкость в сочетании с охватом практически всех алгоритмических направлений, которые являются современными в AMG [3], и их распараллеливанием (многоядерная, распределенное распараллеливание - это гибридная комбинация) являются уникальными точками продаж SAMG.

В работе [3] уже подчеркивалась важность сочетания НРС с масштабируемыми (оптимальными) численными решателями. Однако, когда НРС стал серьезной проблемой для разработчиков промышленного программного обеспечения в Европе (в середине 1990-х годов, когда практически все крупные европейские компании по моделированию программного обеспечения присоединились к проекту распараллеливания Europort [2]), численно масштабируемое программное обеспечение решателей еще не было доступно. Основной численно масштабируемой технологией решателя тех дней была геометрическая многосеточная, которая не подходила для разработки промышленного программного обеспечения из-за ее слабой надежности и ее не подключаемого характера [3]. Кроме того, существующие AMG-решатели были чисто исследовательскими кодами и далеко не были достаточно универсальными и надежными, чтобы быть пригодными для промышленного использования. В этой ситуации распараллеливание кодов промышленного моделирования неизбежно должно было опираться только на стандартные одноуровневые решатели. К счастью, с дальнейшим развитием технологии AMG появились промышленно



значимые масштабируемые решатели, комбинация которых с HPC тогда обещала настоящий прорыв в технологии промышленных решателей.

Насколько важно объединить HPC и масштабируемое численное решение, проиллюстрированное следующим упрощающим, несколько забавным сценарием в Таблице 1. Здесь мы сравниваем типичные параметры приложения первых AMG в начале 1980-х с параметрами SAMG в 2010-х в контексте моделирования CFD с коррекцией давления. Игнорируя другое аппаратное и программное обеспечение (например, операционную систему, компиляторы и т.д.), Увеличение вычислительной мощности в  $10^7$  раз больше, чем примерно в 20 раз больше, чем увеличение по требованию пользователя  $0.5 \cdot 10^6$ . Таким образом, мы хотим добиться улучшения в 20 раз в общем времени выполнения AMG, которое мы могли примерно наблюдать. Хотя это сравнение чрезвычайно упрощено, оно подчеркивает важнейшее преимущество AMG перед стандартными методами решения. AMG (в принципе) может идти в ногу с быстрым развитием аппаратного обеспечения только благодаря своей числовой масштабируемости; в то время как любой неиерархический решатель покажет рост времени выполнения с увеличением размера проблем.

Факторы роста в Таблице 1 в значительной степени соответствуют известному Закону Мура, который предсказывал уже в 1965 году экспоненциальный рост сложности интегральных схем с периодом удвоения всего 18–24 месяцев [1]. Эти факторы роста, однако, являются совместным результатом многих технологических разработок. Для простоты обозначения мы ссылаемся на все компоненты оборудования и их организацию как (вычислительную) архитектуру.



В настоящее время высокопроизводительные компьютеры, операционные системы и компиляторы имеют большой набор функций и компонентов для оптимизации времени выполнения приложений.

Таблица 1 Сравнение типичного варианта использования AMG в 1980-х и 2010-х годах

	AMG (1980s)	AMG (2010s)	Фактор
CPU clock	$\approx 1\text{MHz}$	$\approx 10\text{ THz}^a$	$10^7$
Main memory	$\approx 48\text{ KBytes}$	$\approx 100\text{TBytes}^b$	$>2 \cdot 10^9$
Customer demand	$N=256$	$N=130\,000\,000$	$0.5 \cdot 10^6$
Solution time	$O(1)\text{min}$	$O(1)\text{ s}$	$O(10)$

a Сумма частот по всем ядрам.

b Суммарное количество оперативной памяти по всем ядрам.

Однако чем сложнее архитектура, тем сложнее разработка программного обеспечения, чтобы добиться удовлетворительного использования вычислительной системы. Это особенно справедливо для AMG и его реализации для различных факторов параллелизма (многоядерный, многопроцессорный, распределенный механизм параллелизма). Помимо параллелизации необходимо эффективное использование архитектурных особенностей, тонкая настройка переходов от одинарной к двойной точности от типа `float` к типу `double` (оптимизация циклов, векторизация, использование кэша) является сложной задачей для операций над разреженной матрицей, которые доминируют над производительностью AMG. Однако обратите внимание, что практически невозможно достичь теоретического предела при параллельной производительности с AMG на современных вычислительных архитектурах из-за их сложности и задействованной алгоритмы AMG. Тем не менее, в SAMG мы прошли долгий путь, чтобы максимально приблизиться к оптимальной производительности

при сохранении необходимой надежности, необходимой промышленным пользователям. Значительное количество программных усилий, затраченных на чистую параллельную оптимизацию производительности, сегодня окупается нашими клиентами и обеспечивает прочную основу для будущих аппаратных разработок. В оставшейся части этой статьи мы представим различные проблемы и проблемы, возникающие при распараллеливании любого иерархического итеративного линейного решателя и AMG в частности. Затем мы кратко рассмотрим этапы параллельного внедрения SAMG, чтобы справиться с этими проблемами численного решения) и наметить наши подходы к успешному использованию функций HPC в этих компонентах. Наконец, мы представляем некоторые показатели производительности SAMG в реальных прикладных настройках и суммируем текущее состояние и философию параллельной SAMG, включая ссылку на специальную «авто параллельную» версию SAMG: XSAMG.

## **2 Проблемы, связанные с параллельными компьютерными архитектурами**

Ниже мы рассмотрим влияние трех фундаментальных аппаратных разработок на технологию линейных решателей и AMG в частности. Здесь мы сосредоточимся на одноядерной производительности с плавающей запятой, параллелизме многоядерной / общей памяти и параллелизме распределенной памяти.

### **2.1. Одноядерная производительность: тактовая частота процессора и частота памяти.**

До середины 2000-х гг. Тактовые частоты процессора быстро росли, в значительной степени на уровне закона Мура, и в итоге достигли тактовой частоты примерно 3-4 ГГц. Этот период был назван «эрой ГГц». Для приложений, интенсивно работающих, это обеспечивало «бесплатную езду»: не меняя ничего, производительность удваивалась по скорости каждые 2

года. Однако это еще не все: хотя тактовая частота процессора и размер основной памяти растут быстро, скорость доступа не увеличивается с той же скоростью. Если взглянуть на известные серии чипов, то можно обнаружить 50-кратное увеличение частоты ЦП между серединой 1990-х и серединой 2000-х годов (в соответствии с законом Мура), но только примерно 10-кратное увеличение частоты памяти. То есть отношение производительности ЦП и памяти, выраженное как  $\alpha$ , которое определяется как число циклов ЦП, прошедших при чтении в памяти одного слова, увеличилось в 5 раз. Таким образом, интенсивный доступ к памяти стал потенциальным узким местом.

Таблица 2 Влияние эффективности памяти на производительность AMG

	Base CPU frequency	$\alpha$ by "stream"	SAMG run time
Machine A	3.3 GHz	$\approx 7$	17.7
Machine B	2.6 GHz	$\approx 2$	13.7

Это оказывает непосредственное влияние на производительность последовательного SAMG. Основной тип операции в AMG - разреженное умножение матрицы на вектор или структурно аналогичные операции. Такие операции, в основном, связаны с памятью. Грубо говоря, вся матрица должна быть загружена из основной памяти, но только две операции в точке не выполняются для каждой записи. Поскольку загрузка памяти (намного) дороже, чем операции с плавающей точкой на современных архитектурах, в производительности AMG преобладает производительность памяти. Кроме того, типичные шаблоны доступа к памяти в разреженных матричных операциях не являются ни кеширующими, ни векторизованными. Таким образом, параметр  $\alpha$  определяет барьер производительности для разреженных операций матрицы и, следовательно, также для AMG. Мы подтверждаем это утверждение конкретным эталонным тестом на двух современных машинах, где одна из машин имеет более высокую частоту ЦП, но также и гораздо более медленный доступ к памяти, то есть значительно

выше  $\alpha$  см. Таблицу 2. Здесь мы даем время выполнения SAMG и результаты устоявшегося «потока» [6], которые прямо измеряют  $\alpha$ . Исходя из этих цифр, мы можем только заметить, что AMG (то есть в целом разреженные операции с матрицами) не может выиграть от увеличения тактовой частоты ЦП, если производительность памяти не улучшится соответствующим образом. Фактически, выполнение SAMG на более медленном процессоре с гораздо лучшей производительностью памяти происходит на 25% быстрее. Таким образом, AMG извлекла выгоду из эры ГГц, но не в полной мере.

## **2.2. Многоядерные процессоры и параллелизм разделяемой памяти**

С появлением «многоядерности» эпоха ГГц внезапно завершилась. Важными аргументами для остановки повышения частоты процессора были тепловыделение и энергоэффективность компьютеров. Физика диктует, например, Двухъядерный процессор, работающий на двух ядрах с тактовой частотой 80%, потребляет столько же энергии, сколько одноядерный процессор на полной скорости. Точно так же восьмиядерный процессор, работающий на 50% тактовой частоты, требует такой же энергии. Таким образом, многоядерные процессоры обладают потенциалом для достижения в значительной степени улучшенного баланса «производительность на ватт». Таким образом, многоядерные процессоры стали моделями, которые не смогли продлить срок действия. Как следствие, бесплатная эра ГГц закончилась для всех, и почти все коды должны были распараллеливаться с многоядерными, совместимыми с новыми программами для программирования улучшить производительность во время выполнения.

Это было сделано для SAMG с помощью парадигмы программирования на основе директив OpenMP [7], которая (обычно) охватывает так называемые параллельные потоки в многоядерной архитектуре. В SAMG почти 1000 алгоритмов и кодовых блоков, которые все оснащены параллелизмом OpenMP, причем большинство из них сложным способом, выходящим за

рамки простого неявного программирования OpenMP. Помимо этих усилий, многоядерный также привнес несколько новых трудностей в SAMG. Барьер памяти, обсуждаемый в разд. 2.1 остался или даже ухудшился, так что многоядерные ускорения (обязательно) ниже оптимальных (см. Раздел 2.2.1). Многие компоненты AMG по своей природе являются последовательными и должны быть заменены параллельными вариантами, которые, однако, обычно изменяют числовое поведение и приводят к не побитовым идентичным результатам с последовательной версией (см. Раздел 2.2.2). Кроме того, распараллеливание OpenMP (по определению в его стандарте) допускает так называемые условия гонки, которые могут сделать результаты невоспроизводимыми даже между двумя запусками с множеством потоков (раздел.2.2.3). Наконец, большинство HPP платформ из нескольких многоядерных ЦП (несколько «сокетов» ), Который все еще, с логической точки зрения, может обрабатываться OpenMP как однородная система с общей памятью. Это, однако, может привести к дополнительным потерям производительности, которые трудно обнаружить (см. Раздел 2.2.4). Для некоммерческих пакетов программного обеспечения и исследовательских кодов такие проблемы обычно принимаются и считаются незначительными. Но SAMG имеет иное позиционирование с ориентацией на промышленных пользователей. Отсутствие совершенных многоядерных ускорений трудно продать, условия гонки обычно не принимаются, в то время как результаты, зависящие от степени параллелизма, которые даются практически для всех нетривиальных алгоритмов, тем временем принимаются и многими промышленными пользователями.

### **2.2.1. Многоядерный доступ к памяти.**

Что касается воспоминаний о «горлышке» и факторе, описанном в предыдущем разделе, два эффекта объединяются в многоядерный. С одной стороны, производители оборудования существенно улучшили связи с памятью, так что абсолютная пропускная способность памяти увеличилась.

Однако эта пропускная способность не масштабируется с количеством ядер. Таким образом, устойчивая общая пропускная способность насыщается в определенной точке, как ясно показано «потоком» на типичном восьмиядерном ЦП сегодня, см. Таблицу 3. Это означает, что устойчивая  $\alpha$  значительно увеличивается при переходе от 1 до 8 ядер, то есть относительно производительность памяти ухудшается. Понятно, что этот эффект не влияет на производительность SAMG. Тем не менее, цифры, приведенные в таблице 4, ясно показывают, что SAMG демонстрирует почти такое же поведение при масштабировании, как и простая релаксация Якоби, т.е. Операции с разреженной матрицей. Таким образом, SAMG фактически выполняет практический предел, которого он может достичь из-за основных схем доступа к памяти.

Таблица 3 Пропускная способность памяти на типичном восьмиядерном процессоре

Number cores	Throughput (stream), GB/s	Speedup
1	14.1	1
2	24.4	1.73
4	36.2	2.57
8	37.3	2.65

Таблица 4 Сравнение параллельного ускорения SAMG и простого Якоби-сглаживания

Number cores	Speedup SAMG	Speedup Jacobi
1	1	1
2	1.84	1.85
4	3.3	3.4
8	5.1	5.5

### 2.2.2. Внутренний однопоточный компонент.

Многие ключевые блоки кода состоят из циклов, которые являются рекурсивными, то есть итерации зависят от результата предыдущих. Как уже говорилось, сглаживатель Гаусса-Зайделя или стандартный процесс огрубления Рунге Штюбена являются ключевыми примерами этой проблемы. Обрезать такие циклы простым неявным OpenMP просто неправильно. Нужно создавать распараллеливаемые варианты таких компонентов,

блокируя стратегии, в которых соответствующий алгоритм применяется параллельно каждому блоку, в сочетании с некоторой коррекцией для обработки зависимостей между блоками (например, с помощью методов декомпозиции данных или области). К сожалению, нет простых общих правил, чтобы выбрать эти блоки и соответствующие поправки для учета межблочных вычислений. Зависимости блоков. Фактически, большинство из этих стратегий блокирования, реализованных в SAMG, довольно сложны, и каждая из них предназначена для одной конкретной задачи. Укажем одно важное следствие введения таких параллельных вариантов или обобщений по существу последовательных алгоритмов: общее вычисление становится зависимым от числа параллельных ситуаций (как правило, в открытом случае). Таким образом, результаты вычислений подвержены отклонениям между различными прогонами одного и того же случая моделирования на разных количествах потоков. Это наблюдение справедливо для блочных вариантов любого по сути последовательного алгоритма, а не только для AMG.

### 2.2.3. Условие гонки.

Условие гонок возникает, когда несколько потоков, которые совместно используют общий сегмент памяти, объединяются в область чтения / записи с определенными распределениями памяти. Примечания, что такие события могут давать результаты, которые не могут быть воспроизведены: два разных запуска одного и того же исходного кода, ввода, числа потоков и вычислительную архитектуру могут давать разные результаты, в зависимости от того, как решено, зависит ли это от решения. Кроме того, OpenMP (в соответствии со стандартным определением) может сам вводить такие условия гонки, например, когда локальные данные с плавающей точкой потока суммируются по потокам (с помощью так называемых сокращений OpenMP (`omp reduction: +sum`)). Другой проблемой для библиотеки с поддержкой OpenMP является, конечно, то, что вызывающая



пользовательская программа сама может содержать условия гонки, которые могут влиять на ввод в библиотеку и тем самым вводить эти эффекты в библиотеку, не связанную с условиями гонки.

#### **2.2.4. Несколько процессорных сокетов.**

Неявный обмен данными между сокетом может повлиять на производительность, так как перемещение данных между сокетами обычно обходится дороже, чем в физической памяти на одном сокете. Такие эффекты (к сожалению) связаны с управлением памятью пользовательской программы (и операционной системы). Например, массив данных, выделенный и инициализированный на определенном сокете, будет постоянно помещаться в память этого сокета (по крайней мере, на компьютерах с Linux). Поэтому любой доступ к этому массиву через другой сокет приведет к таким неявным и нежелательным перемещениям данных. Чтобы избежать этой проблемы или, по крайней мере, уменьшить ее влияние на время выполнения, рекомендуется инициализировать расположение разделяемой памяти, используя в основном весь поток, который используется (так называемое правило «первое касание»).

#### **2.3. Распределенный параллелизм памяти.**

С конца 1980-х годов, задолго до многоядерности, параллелизм в распределенной памяти стал неизбежным средством сокращения чрезмерного времени выполнения. Однако этот тип параллелизма требует еще больших усилий по программированию, поскольку данные больше не распределяются между процессорами, а должны передаваться явно, обычно MPI [5]. В настоящее время промышленные клиенты используют SAMG в больших вычислительных кластерах, где многие «узлы» многоядерных многопоточных процессоров связаны друг с другом с помощью эффективных коммуникационных межсоединений. (SAMG на самом деле очень успешно используется нашими клиентами с более чем 3000 ядер). Это создает еще один сложный уровень сложности в программировании AMG, особенно



когда связь на основе MPI используется только для обмена межузловых данных, в то время как внутриузловой параллелизм достигается с помощью OpenMP. Примечания, что MPI-параллель AMG фактически является алгоритмическим вариантом AMG, а не точным распараллеливанием, сравните раздел. 2.2.2.2 (Тем не менее, обратите внимание, что из-за потребительского спроса параллельный SANG может работать в режиме (с низкой эффективностью), который очень близок к последовательному алгоритму) Новые параллелизуемые варианты существующих алгоритмов, также основанные на определенных стратегиях блокирования, должны быть реализованы для MPI. Обратите внимание, что для этого программное усилие является еще одним дополнением, поскольку реализации OpenMP и MPI являются отдельными и не могут приносить прибыль друг другу. По сравнению с OpenMP, где мы уже заявляли, что около 1000 кодовых блоков в SAMG оснащены операторами OpenMP, MPI-параллелизм гораздо более вовлечен. MPI-версия SAMG требует около 50% своих кодовых строк только для организации MPI-параллелизма.

Различия в производительности, налагаемые распределенным параллелизмом, аналогичны различиям в многоядерном контексте, см. Разд. 2.2. Ограниченное ускорение внутри узла, обсуждаемое в разд. Конечно, 2.2.1 нельзя удалить с помощью MPI, так что показатели производительности SAMG для каждого узла практически одинаковы для OpenMP и MPI. Преимущества MPI по сравнению с OpenMP состоят в том, что не существует условий гонки и что множественные сокеты не представляют дополнительной проблемы. К сожалению, «грубая задача сетки» становится намного сложнее для MPI-параллелизма, чем с чистым параллелизмом OpenMP. Часто в вычислениях MPI участвуют сотни или 1000 процессов, поэтому для очень грубых сеток едва ли остается достаточно работы для всех процессов. В такой ситуации решение о том, где прекратить грубое обращение и объявить данный уровень как самый грубый, становится

решающим. Напомним, что (как правило) самый грубый уровень должен быть решен (почти) точно, так как альтернативные грубые сеточные решения, такие как выполнение всего нескольких итераций итерационного решателя, могут серьезно повлиять на числовую устойчивость всего процесса AMG и, следовательно, должны использоваться с осторожностью. Таким образом, возникает проблема решения разреженной линейной системы умеренного и незначительного размера для многих процессов при слишком ранней остановке укрупнения. При продолжающемся укрупнении в течение слишком долгого времени происходит все больше и больше процессов на холостом ходу и ухудшается соотношение работы и связи. В любом случае, где-то AMG применяет «закон Амдала» для грубых сеток, которые не могут быть полностью исключены. Отметим, однако, что это явление полностью обусловлено распределенным параллелизмом и не может присутствовать в последовательном или чистом параллельном OpenMP AMG.

### **3. Как SAMG противостоит вызовам HPC.**

Теперь, когда мы представили фундаментальные проблемы, возникающие при распараллеливании в целом, давайте вкратце изложим стратегии, которые мы использовали при распараллеливании основных компонентов SAMG: сглаживание, огрубление и решение с грубой сеткой. Как указывалось ранее, это охватывает только небольшой раздел всей организации OpenMP и MPI, которая была реализована в SAMG, но предоставляет хорошие образцы для подражания для стратегий, используемых в SAMG. Прежде чем углубляться в детали, давайте подведем итоги общего эффекта от наших усилий по распараллеливанию:

- SAMG имеет почти 100% распараллеливание для OpenMP и MPI; то есть, в SAMG нет важных последовательных блоков кода, которые бы вызывали сильный «эффект Амдала». Кроме того, SAMG обеспечивает гибридное распараллеливание, сочетающее OpenMP и MPI-параллелизм, что дает пользователю большую гибкость в сокращении

времени решения на конкретной аппаратной конфигурации. Насколько нам известно, это уникальная особенность среди всех доступных пакетов AMG.

- Мы устранили все потенциальные условия гонки OpenMP в SAMG (за счет крошечных дополнительных накладных расходов). Затронутое сокращение OpenMP было исключено и перепрограммировано вручную. Это и целый ряд других мер повлекли за собой значительное количество дополнительной работы по программированию в SAMG, которая может служить примером того, что промышленные требования требуют дополнительного качества и программирования по сравнению с академическим программным обеспечением.

- Мы используем первую политику касания (см. Раздел 2.2.4) везде, где это целесообразно в SAMG, чтобы минимизировать вероятность потери производительности из нескольких сокетов.

- Наш фокус всегда был на стойкость и общую значимость для различных случаев промышленного использования. Таким образом, чрезвычайно настроенные варианты для очень особых случаев не включены в SAMG. В таких случаях мы жертвовали несколькими процентами производительности в пользу универсальности и в интересах наших клиентов, которые получают удовлетворительную параллельную производительность для всех используемых настроек параметров. Кроме того, мы оказываем существенную поддержку нашим клиентам в поиске оптимальных настроек, чтобы минимизировать время решения их соответствующих приложений и аппаратных ресурсов.

### **3.1. Настройка и распараллеливание сглаживания.**

Классические сглаживатели «Jacobi» и «Gauß – Seidel» состоят из, казалось бы, очень простых компактных шестистрочных двойных циклов. Однако в этих циклах (если они выбраны в качестве сглаживателя) легко тратится 50%

(если не больше) времени решения SAMG. Таким образом, оптимизация и распараллеливание этих блоков кода, то есть решение всех проблем НРС, описанных в разд. 2, является важной задачей для SAMG. Эти ошибки являются весьма полезными для всех этих сглаживателей. Таким образом, очень просто, все остается на усмотрение компилятора и тщательной проверки выбранных флагов. Тем не менее, можно пойти дальше. Стандартная формулировка использует стандартные структуры данных для разреженных матриц. Однако существует множество интересных альтернативных форматов для хранения разреженных матриц. При разработке SAMG мы достаточно тщательно проверили такие переформулировки и, наконец, пришли к выводу, что в определенных случаях стоит пожертвовать некоторой памятью и преобразовать пользовательскую матрицу в новый полезный формат. В этих случаях накладные расходы времени выполнения для преобразования и общее время выполнения сокращаются за счет дополнительного хранилища. В некоторых случаях использования от наших промышленных клиентов мы находим увеличение производительности примерно на 20% в общем времени выполнения, см. Таблицу 5, в то время как поведение параллельного масштабирования сохраняется. Еще лучше ситуация, когда пользовательская матрица имеет субструктурирование в виде  $2 \times 2$  или  $3 \times 3$  блоков, как это происходит в случае форматов случая, возникающих в приложениях упругости. Используя соответствующую блочную структуру данных для хранения системной матрицы, время выполнения одного этапа сглаживания Якоби увеличивается в 2–3 раза, см. Таблицу 6 и Рис. 1.

Таблица 5 Сравнение параллельного времени выполнения схем релаксации

Number cores	Jacobi	Gauß-Seidel (with race conditions)	Gauß-Seidel (optimized)
1	1.70	1.74	1.36
2	0.79	0.82	0.72
4	0.46	0.47	0.40
5	0.32	0.32	0.25
16	0.18	0.17	0.14

Во многих случаях внутренне последовательное сглаживание Гаусса-Зейделя обеспечивает существенно более короткое время решения, чем естественное параллельное сглаживание Якоби, благодаря его лучшему сглаживанию. Например, при использовании сглаживателя Якоби потребовалось 35 итераций SAMG для сходимости для задачи, рассматриваемой в таблице 5, тогда как при использовании сглаживания Гаусса-Зейделя (работающего на одном ядре) потребовалось всего 24 итерации.

Таблица 6 Сравнение параллельных времен прогона схем релаксации для матриц с внутренней блочной структурой.

Number cores	Jacobi (scalar)	Jacobi (block)
64	3.59	1.49
128	1.84	0.55
256	0.92	0.30
512	0.46	0.17
1024	0.23	0.09
2048	0.14	0.07

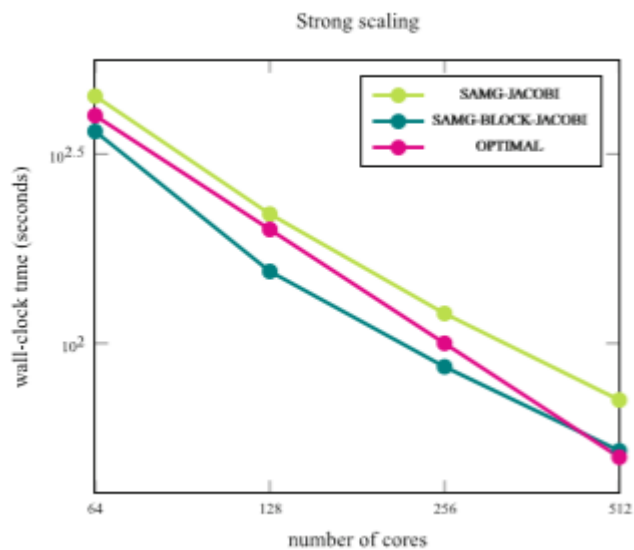


Рис. 1 Сильное масштабирование параллельного решателя SAMG в приложении сложной эластичности с  $N = 46\,875\,000$  степеней свободы

(Архитектура: Intel (R) E5-2650 v2 при 2,60 ГГц, 16 ядерных узлов, гибридный OpenMP / MPI-параллелизм)

Таким образом, наш оптимизированный сглаживатель Гаусса-Зейделя SAMG дает общее уменьшение времени при общем времени решения более чем на 50% по сравнению с SAMG с сглаживанием Якоби в этом примере. К сожалению, дополнительные усилия по программированию снова необходимы, чтобы поддерживать это улучшенное поведение сходимости также параллельно, по крайней мере, в значительной степени. Сглаживание Гаусса-Зейделя обновляет вектор решения один компонент за другим, причем любое обновление зависит от предыдущих. Этот процесс, как правило, не может быть распараллелен любой стратегией блокировки. Для разреженных матриц каждая из элементарных компонент обновления (т. Е. Внутренний цикл) включает в себя только небольшое количество данных, которые также не обеспечивают достаточную массу для распараллеливания. Таким образом, буквальное распараллеливание этого алгоритма практически невозможно. Стандартный подход к этому вопросу - это метод декомпозиции доменов: примените блочные локальные Гаусса-Зейделя и Якоби для межблочных связей. Однако этот подход может привести к потере некоторых улучшенных параметров сглаживания вдоль границ блоков и асимптотически приведет к сглаживающему поведению Якоби. В SAMG мы разработали несколько уточнений этого подхода, которые направлены на уменьшение «доли Якоби» процесса сглаживания при сохранении свободной реализации условия гонки в OpenMP. Благодаря этим усилиям наша система Гаусса-Зейделя сглаживания является более надежной в отношении количества ядер. Например, в то время как SAMG потребовалось 24 итерации, чтобы сойтись со сглаживанием Гаусса-Зейделя на одном ядре в предыдущем примере, ему потребовалось 29 итераций с 16 ядрами, что очень приемлемо

по сравнению с 35 итерациями, необходимыми для чистого сглаживания Якоби.

### 3.2. Огрубление Джон Руге и Клаус Штюбена.

Давайте теперь кратко укажем некоторые идеи распараллеливания для одной из стратегий укрупнения в SAMG: классического процесса укрупнения Руге-Штюбена. Оригинальный процесс рекурсивный и использует глобальное представление системной матрицы. Снова, ограничение укрупнения в подблоки является очевидной идеей. Однако, делая это простым методом грубой силы, полностью локализуя AMG и игнорируя все межблочные зависимости, можно завершить с помощью разложения в основной домен с локальным AMG, который не сходится в мульти-сеточном качестве. Ранние усовершенствования этого подхода, также в SAMG, заключались в методах «блокирования субдомена» (см. Также [4]), в которых обрабатывалось переблокирование блоков-блоков, и все-таки было лучше. Однако версия этого первоначального подхода блокирования субдомена, которая была действительно надежной, дружественной к памяти и эффективной, была не найден в то время. Интересно, что это был один из первых коммерческих пользователей SAMG, которые потребовали существенного улучшения здесь. В этот момент в SAMG началась значительная фаза разработки, которая привела к реализации набора стратегий, которые по существу исправляют эту межблоковую проблему. Эти обобщения блокирования субдомена позволяют торговать памятью или связывать требования с числовыми свойствами. Только при разумном повышении требований к связи одна из этих стратегий во многих случаях позволяет (почти) повторно захватить числовое поведение последовательного кода. Напомним, однако, что наша библиотека SAMG предоставляет практически все соответствующие эффекты AMG и процедур грубого огрубения, которые имеют свои специфические проблемы, когда речь идет о распараллеливании. Несмотря на то, что в течение многих лет мы вкладывали значительные средства в



исследования и программирование в распараллеливание всех этих методов, работа над параллельными схемами укрупнения продолжает еще больше улучшать масштабируемость и время решения.

### 3.3. Решение на самой грубой сетке.

Наконец, давайте более подробно рассмотрим печально известный «грубый вызов сетки», включая потенциальный «эффект Амдала», который мы уже упоминали в разделе. 2.3. Фундаментальная проблема здесь заключается в том, что после ряда этапов укрупнения получающиеся линейные системы становятся слишком маленькими по размеру, чтобы эффективно распределяться между участвующими ядрами; это справедливо для любой многосеточной схемы. Чтобы справиться с этой проблемой, SAMG предлагает ряд стратегий, поскольку не существует единого решения этой проблемы. Прежде всего, влияние качества решателя грубых сеток на поведение сходимости зависит от проблемы, и, таким образом, эта проблема может быть безопасно проигнорирована в некоторых приложениях, в то время как в других она будет доминировать в общей производительности. Отметим также, что доступные аппаратные ресурсы, например, Сеть связи и доступная память должны учитываться при разработке подходящего подхода с использованием грубой сетки, чтобы минимизировать общее время решения. Поэтому в этом трактате мы ограничимся демонстрацией эффективности наших контрмер в промышленном использовании, см. Также рис. 2. Напомним, что в серийном SAMG сетки укрупняются до незначительного размера, равного примерно 100 степеням свободы. Однако, используя аналогичную локальную грубую сеточную нагрузку параллельно, можно получить гораздо больший общий размер крупной сеточной системы. Например, когда мы используем 512 процессов, как указано в этом примере, общая проблема грубой сетки (примерно) представляет собой систему измерения  $50\,000 \times 50\,000$ , и вычислительные усилия, связанные с ее решением, больше не пренебрежимо малы. Если мы хотим выполнить



прямое решение этой крупнозернистой системы, как правило, на одном ядре, его раствор в этом конкретном примере должен составлять более 20% от общего времени выполнения. Таким образом, последовательное грубое сеточное решение может привести к жесткому барьеру для параллельного масштабирования, что неприемлемо для наших клиентов. Однако, используя одну из грубых сеточных стратегий, доступных в SAMG, которая учитывает имеющуюся аппаратную конфигурацию, мы реализуем вполне приемлемое поведение параллельного масштабирования, см. Рис. 2. Из графиков, показанных на фиг. 2 и 1 мы можем наблюдать почти оптимальное поведение масштабирования SAMG на 64–2048 ядрах. Соответствующие измерения получены из решения приложения потока жидкости клиента из автомобильной промышленности с  $N = 130$  млн степеней свободы и приложения эластичности с  $N = 46\,875\,000$  степеней свободы.

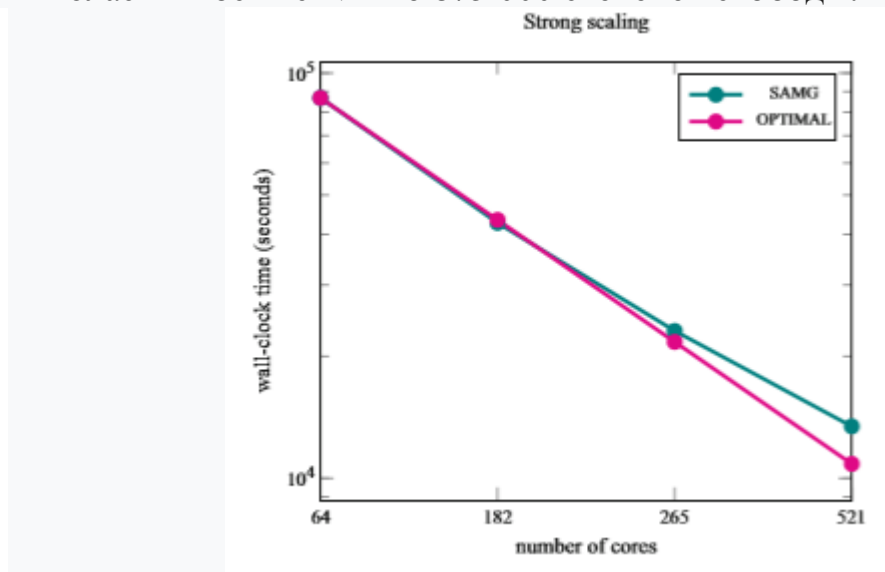


Рис. 2 Сильное масштабирование параллельного решателя SAMG в OpenFoam с  $N = 130\,000\,000$  степеней свободы (Архитектура: Intel (R) E5-2650 v2 @ 2.60 ГГц, 16 ядерных узлов, чистый MPI-параллелизм).

#### 4. SCAI's Parallel SAMG Solver Library

Таким образом, распараллеливание многосеточных методов и, в частности, AMG, ни в коем случае не является тривиальной задачей, и многие вопросы, представленные в этом документе, придется пересматривать снова и снова с

каждой новой разработкой оборудования. Более того, удобство использования AMG в промышленном контексте накладывает ряд дополнительных проблем, которые мы также затронули. Ключевым моментом является то, что единственными двумя важными мерами для промышленных пользователей являются надежность и время решения - с учетом их размера проблемы и аппаратных ресурсов. Таким образом, SAMG обеспечивает несколько направлений параллелизма: OpenMP, MPI и гибрид, что позволяет пользователю выбирать оптимальные настройки в отношении доступного оборудования. Наконец, мы хотим указать на специальную версию MPI-параллельной SAMG, XSAMG, которая предоставляет уникальную возможность продажи, оснащая полностью последовательные пользовательские приложения полноценным гибридным параллелизмом MPI-OpenMP - по крайней мере, для линейного решателя. Эта линейка продуктов SAMG используется клиентами, чей код моделирования не является MPI-параллельным, но использует большую долю времени выполнения в линейном решателе. В таком случае, когда используется ветвь SAMG в XSAMG, MPI-параллелизм создается «по оси Y». Внутри пользовательские данные перераспределяются между доступными процессами MPI, и выполняется параллельное MPI SAMG. В конце концов, распределенное решение собрано обратно. Обратите внимание, что вызывающая последовательность для XSAMG идентична последовательному интерфейсу - код пользователя не должен быть изменен (за исключением двух предельных вызовов инициализации и завершения). Конечно, распределение и сбор данных не являются (вообще) бесплатными, и ускорения времени выполнения ограничены только частью линейного решателя. Таким образом, нельзя ожидать идеального ускорения, и этот подход имеет смысл только тогда, когда значительная доля времени в линейном решателе является значительной.

XSAMG представляет особый интерес для пользователей, у которых есть последовательное приложение, но доступно несколько узлов кластера. С помощью XSAMG они могут охватывать решающую часть, состоящую из нескольких узлов кластера, особенность, которую невозможно реализовать при любом другом подходе. Например, один из наших клиентов XSAMG получает ускорение 3.3 в части линейного решателя при переходе от 1 к 4 узлам с использованием XSAMG, который явно превосходит параллелизм одиночного узла, см. Таблицу 7.

Таблица 7 Масштабирование XSAMG по нескольким узлам MPI

Number nodes	XSAMG runtime
1	24.3
2	12.2
4	7.3

В заключение мы кратко комментируем философию SAMG относительно специальные архитектуры. Помимо частичных успехов для отдельных модулей, попытка портировать SAMG в полном объеме на специальные чипы, такие как графические процессоры или многоядерные ускорители, до сих пор не удалась. Необходимо приложить огромные усилия для того, чтобы сделать даже простые структуры кода максимально эффективными для таких чипов. Особые версии AMG могут обходиться, но они никак не могут противостоять общности и надежности, необходимой для промышленного использования. Тем не менее, SAMG будет готов к новому поколению микросхем такого рода, как только их удобство и эффективность программирования будут улучшены.

#### Литература

1. G.E. Moore, Cramming more components onto integrated circuits. Electronics 8, 114–117 (1965)
2. K. Stüben, Europort-D: commercial benefits of using parallel technology, in Parallel Computing: Fundamentals, Applications and New Directions, ed. by E. D'Hollander, F. Peters, G. Joubert, U. Trottenberg, R. Völpe.

- Advances in Parallel Computing, vol. 12 (North-Holland, Amsterdam, 1998), pp. 61–78
3. K. Stüben, J.W. Ruge, T. Clees, S. Gries, Algebraic multigrid — from academia to industry, in Scientific Computing and Algorithms in Industrial Simulation — Projects and Products of Fraunhofer SCAI, ed. by M. Griebel, A. Schüller, M.A. Schweitzer (Springer, New York, 2017)
4. U.M. Yang, Parallel algebraic multigrid methods — high performance preconditioners, in Numerical Solution of Partial Differential Equations on Parallel Computers, ed. by A.M. Bruaset, A. Tveito (Springer, Berlin, Heidelberg, 2006), pp. 209–236
5. MPI Forum. <http://mpi-forum.org>. Accessed 31 Mar 2017
6. STREAM: Sustainable Memory Bandwidth in High Performance Computers. <https://www.cs.virginia.edu/stream/>. Accessed 31 Mar 2017
7. The OpenMP API Specification for Parallel Programming. <http://www.openmp.org>. Accessed 31 Mar 2017

Reducing complexity of algebraic multigrid by aggregation

Serge Gratton, Pascal Hénon, Pavel Jiránek and Xavier Vasseur

## Уменьшение операторной сложности алгебраического многосеточного метода путем агрегации

### Аннотация

Типичный подход к снижению вычислительных затрат и требований к памяти в классических алгебраических многосеточных методах заключается в замене консервативного алгоритма укрупнения и интерполяции на короткие расстояния на фиксированном количестве мелких уровней агрессивным укрупнением с интерполяцией на большие расстояния. Хотя качество получающегося в результате предобусловливателя алгебраической многосеточной сетки часто ухудшается с точки зрения скоростей сходимости

и количества итераций предварительно обработанного итерационного решателя, общая производительность может существенно улучшиться. Мы исследуем здесь, в качестве альтернативы, возможность заменить классическое агрессивное укрупнение агрегацией, которое мотивируется тем фактом, что сходимость методов агрегации может быть независимой от размера задачи при условии, что число уровней фиксировано. Относительная простота агрегации может привести к улучшению затрат на решение и настройку. Численные эксперименты показывают актуальность предлагаемой комбинации как для академических, так и для базовых задач при моделировании пласта в нефтяной промышленности.

## **Введение**

Алгебраические многосеточные (AMG) методы [1–4] являются одними из наиболее эффективных алгоритмов решения и предобуславливания для больших и разреженных систем линейных уравнений, которые возникают в широком диапазоне научных и технических приложений, управляемых эллиптическими уравнениями в частных производных (PDE). Многосеточные методы комбинируют эффекты сглаживания и грубосеточной коррекции. Операция сглаживания использует схемы релаксации, такие как итерации Якоби или Гаусса-Зейделя, и пытается подавить колебательные компоненты ошибки, в то время как грубая коррекция предназначена для устранения дополнительных гладких компонентов ошибки, которые эффективно не уменьшаются сглаживанием. Последний этап состоит из передачи сглаженной ошибки на более грубый уровень с меньшим количеством степеней свободы (DOF), решения связанного с этим уравнения и обратной интерполяции вычисленной поправки для обновления текущего решения на уровне мелких данных. Если

проблема грубого уровня слишком велика, ту же схему можно применять рекурсивно, что приводит к иерархии уровней уменьшающегося размера. В отличие от геометрических методов [2, 5], AMG строит многосеточную иерархию полностью алгебраическими средствами без первоочередной непрерывной задачи. Методы AMG могут, следовательно, использоваться в «черном ящике» и могут применяться даже для задач, определенных в сложных геометрических областях, где это может быть затруднительно - по возможности разработать подходящий геометрический многосеточный метод.

Большинство современных методов могут быть классифицированы в терминах схем укрупнения как классические и агрегационные AMG. В классическом AMG [1–4] определенное подмножество степеней свободы (узлов) с мелким уровнем идентифицируется как множество узлов грубого уровня, и оператор линейной интерполяции выводится из элементов матрицы. В методах агрегации [6–10] узлы грубого уровня ассоциируются с непрерывными непересекающимися группами узлов конечного уровня, называемыми агрегатами. Интерполяция определяется просто путем введения значений от грубого уровня до степенных степеней свободы в каждом агрегате. Мы напомним основные принципы этих методов в Разделе 2. Методы AMG, основанные на классических подходах с консервативными алгоритмами укрупнения и интерполяционными формулами на короткие расстояния, обычно демонстрируют очень хорошую производительность с точки зрения скоростей сходимости предобусловленных итерационных методов, отраженных малым числом итераций для достижения заданных критериев остановки. Однако цена, которую нужно заплатить, характеризуется относительно высокими вычислительными затратами и

требованиями к памяти, связанными с фазами установки и решения предварительного кондиционера. Альтернативно, более агрессивная схема укрупнения с интерполяцией на большие расстояния [4] может быть построена на фиксированном количестве мелких уровней, за которым следует консервативный метод на последующих грубых уровнях. Хотя скорость сходимости предварительно подготовленных итераций может ухудшиться, общая производительность обычно улучшается.

Методы агрегации AMG используют простые операторы интерполяции и часто предоставляют иерархии с низкой сложностью операторов в зависимости от алгоритма агрегации. Тем не менее, они редко использовались ранее в многосеточной среде из-за того, что скорость сходимости результирующего V-цикла часто зависит от размера задачи [4, 7, 11]; в частности, скорость сходимости ухудшается с увеличением количества уровней. Масштабируемость методов агрегации можно восстановить, например, улучшив операторы интерполяции [12–16] и, следовательно, эффективно увеличивая перекрытие агрегатов или используя определенное полиномиальное ускорение коррекции грубого уровня [7, 9, 17–19] вместо обычного V-цикла. Сходимость методов агрегации, с другой стороны, не зависит от размера задачи, если число уровней остается неизменным; см., например, [6, 20]. Сама по себе такая основа, конечно, не практична, поскольку может потребовать решения, возможно, больших грубых систем с высокой степенью точности, например, с помощью прямого решателя, что может быть неосуществимо на практике. В этой статье мы рассматриваем объединение агрегации и классических подходов AMG в рамках одной иерархической альтернативы для классического агрессивного огрубления (раздел 3). Таким образом, мы можем добиться значительного снижения



сложности оператора и снизить затраты на этапы настройки и решения получающегося предварительного кондиционера аналогично классическим агрессивным методам.

Наша работа мотивирована конкретным применением AMG в контексте решения линейных алгебраических систем, возникающих из моделей многофазного потока при моделировании пласта, которые используются в нефтяной промышленности для моделирования потоков нефти, газа и воды в нефти. резервуар подлежит добыче и закачке жидкости. Дискретизация и линеаризация модельных уравнений приводят к большой последовательности систем, которые объединяют неизвестные давления и насыщения или концентрации в вычислительных ячейках дискретизации базовой области и которые должны быть эффективно решены [21–23]. Возможный подход для предварительной подготовки таких систем основан на развязке зависимости давления от оставшихся неизвестных. В этом контексте популярным методом предварительной обработки является метод снижения давления (CPR) [24, 25], который состоит из двух этапов. На первом этапе подсистема давления решается приблизительно за одну или несколько итераций V-цикла метода AMG. На втором этапе приближенный глобальный предобусловливатель, обычно основанный на неполной факторизации всей матрицы Якоби, применяется к обновленному остаточному вектору. Настройка метода AMG часто представляет собой узкую горловину производительности, и поэтому наша цель - снизить затраты на установку AMG, чтобы улучшить общую производительность предобусловливателя CPR. Мы иллюстрируем наш подход на числовых примерах, включающих как простые академические проблемы, так и проблемы, возникающие при моделировании пласта в Разделе 4.



## Алгебраический многосеточный метод

Мы кратко напомним в этом разделе основные принципы (алгебраических) многосеточных методов; см., например, [3–5] для более подробной информации. Рассмотрим линейную алгебраическую систему  $Au = f$ , где  $A$  - действительная  $N \cdot N$  симметричная и положительно определенная матрица, а  $f$  и  $u$  - соответственно правая часть и неизвестные векторы решения. Многосеточный метод пытается построить аппроксимации решения исходной задачи на иерархии уровней, связанных с укрупненными пространствами, чтобы ускорить сходимость решений на мелких уровнях. Основная идея заключается в том, что сглаженные компоненты ошибок, которые не могут быть эффективно устранены с помощью схемы релаксации, удаляются с помощью коррекции грубого уровня, то есть путем решения приближенно-остаточного уравнения на уровне грубых ошибок с меньшим количеством неизвестных и интерполяции вычисленной коррекции обратно в точный уровень для того, чтобы получить улучшенное приближение решения. Такой метод может быть использован как в качестве решателя, так и в качестве предварительного условия для итеративного метода, такого как CG или GMRES. Многосетевая иерархия определяет последовательность сеток  $\Omega_\ell$ , операторы уровней  $A_\ell$ , межуровневые операторы передачи  $P_\ell$  (пролонгаторы) и  $R_\ell$  (ограничители) и операторы сглаживания  $S_{\ell, \text{pre}}$  и  $S_{\ell, \text{post}}$ . В геометрической многосетке сетки можно построить путем огрубления исходной тонкой сетки  $\Omega_1 := \Omega$ , возможно, путем удвоения размера сетки во всех пространственных направлениях или в ее подмножестве. Операторы  $A_\ell$  строятся с учетом дискретизации заданного ДУРЧП, не учитывая грубую сетку, и оператор интерполяции  $P_\ell$  определяется посредством естественного интерполяции между двумя отдельными функциональными

пространствами, связанными с  $\Omega_{\ell+1}$  и  $\Omega_\ell$ . Наконец, операторы сглаживания  $S_{\ell, \text{pre}}$  и  $S_{\ell, \text{post}}$  создаются для того, чтобы получить эффективный цикл, то есть, чтобы подавить компоненты ошибок, которые не могут быть эффективно удалены с помощью коррекции грубого уровня, которая, однако, может зависеть от проблемы (например, использовать линейные или плоские сглаживатели для задач с анизотропными коэффициентами). Традиционный многосеточный V-цикл может быть определен, как в алгоритме 1 (где  $L$  обозначает индекс самого грубого уровня).

---

**Algorithm 1** Recursive multigrid V-cycle  $\text{MG}_\ell(f_\ell, v_\ell)$

---

**Input:** right-hand side  $f_\ell$  and approximate solution  $v_\ell$  of  $A_\ell u_\ell = f_\ell$   
**Output:** updated approximate solution  $v_\ell$

**if**  $\ell = L$  **then**  
    solve  $A_\ell v_\ell = f_\ell$   
**else**  
     $v_\ell \leftarrow S_{\ell, \text{pre}}(f_\ell, v_\ell)$  (Pre-smoothing)  
     $r_{\ell+1} \leftarrow R_\ell(f_\ell - A_\ell v_\ell)$  (Residual restriction)  
     $c_{\ell+1} \leftarrow 0$  and do  $\text{MG}_{\ell+1}(r_{\ell+1}, c_{\ell+1})$  (Coarse-grid correction)  
     $v_\ell \leftarrow v_\ell + P_\ell c_{\ell+1}$  (Solution update)  
     $v_\ell \leftarrow S_{\ell, \text{post}}(f_\ell, v_\ell)$  (Post-smoothing)  
**end if**

---

Сглаживатели в AMG, с другой стороны, обычно фиксируются на простой схеме релаксации, такой как точечная итерация Якоби или Гаусса-Зейделя, и на этих значениях «сеток» (где сетка  $\Omega_\ell$  может быть связана просто с множеством индексов  $\{1; \dots; N_\ell\}$ ) и межуровневые операторы строятся из заданной матрицы точного уровня  $A_1 := A$ . Чтобы получить хороший предобусловливатель AMG, схема укрупнения и, возможно, интерполяции адаптируются таким образом, чтобы диапазон операторов интерполяции смог точно улавливать компоненты ошибок, которые неэффективно устраняются сглаживателем. Обычно это достигается с помощью концепции прочности соединения, которая количественно определяет зависимость от соседей и широко используется как эвристическая схема для построения схем укрупнения и интерполяции AMG.

---

**Algorithm 2** AMG setup

---

**Input:** matrix  $A \in \mathbb{R}^{N \times N}$ , grid  $\Omega = \{1, \dots, N\}$ , and maximum number of levels  $L_{\max}$   
**Output:** the prolongators  $P_\ell$ , restrictors  $R_\ell$ , the level operators  $A_{\ell+1}$ , and the grids  $\Omega_{\ell+1}$ ,  $\ell = 2, \dots, L$ , where  $L < L_{\max}$ .  
set  $\Omega_1 \leftarrow \Omega$ ,  $A_1 \leftarrow A$ ,  $\ell = 1$   
**while**  $\Omega_\ell$  is not small enough and  $\ell < L_{\max}$  **do**  
    Construct the coarse level  $\Omega_{\ell+1}$   
    Compute the interpolation operator  $P_\ell$   
    Set  $R_\ell = P_\ell^T$   
    Compute the coarse-level operator  $A_{\ell+1} = R_\ell A_\ell P_\ell$   
     $\ell \leftarrow \ell + 1$   
**end while**

---

Вычисление иерархии AMG выполняется во время фазы установки, которая обычно следует более или менее фиксированному шаблону, проиллюстрированному в алгоритме 2. Некоторые компоненты установки AMG часто фиксированы; в частности, ограничитель просто задается транспонированием продолжения ( $R_\ell = P_\ell^T$ ), а операторы грубого уровня задаются произведением Галеркина (тройное матричное произведение матрицы на матрицу)  $A_{\ell+1} = R_\ell A_\ell P_\ell$ . Первые два шага цикла настройки, в которых построены грубые узлы из множества  $\Omega_{\ell+1}$  и оператор интерполяции  $P_\ell$ , дают целые семейства методов AMG. Мы рассмотрим основные из них в оставшейся части этого раздела.

2.1. Процедура огрубления и интерполяция в алгебраическом многосеточном методе.

Мы кратко опишем классический и агрегирующий методы укрупнения (рис. 1) и некоторые типичные методы интерполяции. В этом разделе мы опускаем индексы уровня для простоты обозначения; мы предполагаем, что  $N \times N$  симметричная положительная определенная матрица  $A$  задана на сетке подробного уровня  $\Omega = \{1, \dots, N\}$ . Изложение предназначено для систем, возникающих из дискретных скалярных задач с постоянным ядром. В этом случае суммы строк в  $A$  в основном равны нулю, за исключением тех, которые соответствуют граничным условиям Дирихле,

то есть постоянные векторы представляют собой «близкое» к нулю пространство  $A$ . В более общем случае операторы продолжения настраиваются, чтобы сохранить ближнее нулевое пространство. через иерархию, которая может быть выполнена определенной модификацией правил интерполяции и, возможно, адаптивной оценкой базисных векторов ближнего пространства [26, 27].

Китайская статья 2019 года.

### **Параллельный алгебраический многосеточный алгоритм: Состояние и проблемы крупномасштабных компьютерных приложений.**

**Аннотация** Алгебраический многосеточный метод (AMG) является одним из наиболее эффективных алгоритмов решения дискретных линейных алгебраических уравнений в частных производных и широко используется при крупномасштабном численном моделировании практических задач. С непрерывным улучшением производительности суперкомпьютера масштаб вычислений и параллелизм фактического численного моделирования становится все больше и больше, и в то же время, функции приложений и особенности компьютерной архитектуры становятся все более и более сложными. Серьезные проблемы. В этом документе объединены тенденции развития крупномасштабных вычислений, особенно для предстоящих вычислений на экзафлопном уровне, проанализированы проблемы алгоритма AMG в этих трех аспектах, обобщены состояние и ход исследований и с нетерпением ожидаются будущие приоритеты исследований. Ключевые слова Научные и инженерные расчеты, численное моделирование, алгебраический многосеточный метод (AMG), предобуславливатели, параллельные алгоритмы, параллельные вычисления, вычисления на экзафлопном уровне.

## Введение.

В области научных и инженерных вычислений линейный решатель для решения разреженных линейных алгебраических уравнений  $Ax = b$  является основным узким местом производительности для многих приложений численного моделирования. Например, при численном моделировании многих основных приложений в Китае [mo06, rei09, wang09] время ЦП для решения таких уравнений составляет более 80% от общего времени численного моделирования. В последние годы благодаря значительно улучшенным характеристикам суперкомпьютеров стало возможным точное численное моделирование сложных физических систем, что привело к увеличению масштаба линейных алгебраических уравнений. На примере численного моделирования лазерного синтеза, имеющего большое значение в области национальной безопасности и энергетики, [rei09], для точного моделирования эволюции радиационной гидродинамической неустойчивости интерфейса во время имплозии, физическое моделирование должно быть модернизировано с однопоточного до многопоточного. В трехмерном случае масштаб сетки необходимо увеличить на 3 порядка по сравнению с существующей базой: со  $10^9$  до  $10^{12}$ . В настоящее время миллиарды компьютеров имеют миллионы процессорных ядер, что едва соответствует потребностям однопоточного моделирования. Для параллельного многопоточного моделирования в настоящее время это невыносимо для миллиардов компьютеров. Для такого рода приложений численного моделирования, с одной стороны, требуется крупномасштабный суперкомпьютер, а с другой стороны, требуется алгоритм линейного решателя с оптимальной вычислительной сложностью.

Для крупномасштабных разреженных линейных алгебраических уравнений, полученных путем дискретизации дифференциальных уравнений в частных производных, Multigrid (MG) [brandt77, hackbusch78] является быстрым алгоритмом с оптимальной вычислительной сложностью, то есть объемом

вычислений, требуемым алгоритмом. Существует линейная связь с масштабом проблемы. MG - это итерационный метод, который устраняет высокочастотные компоненты ошибки итерации посредством процесса сглаживания (называемого сглаживателем), а оставшиеся низкочастотные компоненты решаются путем решения меньшей линейной системы на грубой сетке. Исключение (называется коррекция на грубой сетке). Для линейных систем на грубых сетках тот же процесс выполняется рекурсивно, пока масштаб крупной сетки не станет достаточно маленьким. Сглаживатель обычно использует простые релаксационные итерации, такие как Якоби или Гаусс-Зайдель. В дополнение к гладкому процессу и процессу грубосеточной коррекции итерационные ошибки могут быть быстро сведены в допустимый диапазон. Для многих задач количество итераций, требуемых алгоритмом MG, не зависит от масштаба задачи, и теперь стало незаменимым алгоритмом ускорения для многих практических приложений численного моделирования [engquist01, givoli01, simon05].

В зависимости от того, как строятся грубые сетки, многосеточные методы можно условно разделить на геометрические многосеточные (GMG) и алгебраические многосеточные (AMG) методы. Для GMG грубые сетки обычно получаются непосредственно с использованием иерархий сетки, сформированных в процессе дискретизации, или послойного сгущения геометрической информации, основанной на подробных сетках. После более чем сорока лет разработки алгоритм GMG был относительно зрелым, а также более богатой и хорошо развитой математической теорией [hackbusch85, хујс92], которая имела большой успех во многих приложениях. Однако эффективность решения GMG зависит от геометрической информации дискретных сеток и свойств уравнений в частных производных, которые часто недоступны или труднодоступны во многих практических задачах, что в значительной степени ограничивает популярность и применение GMG. Таким образом, разработка многосеточного метода, который не зависит от

геометрической информации, становится целью практического применения. AMG принадлежит к этому классу методов, которые следуют тому же принципу решения, что и GMG, но его алгоритмические компоненты построены из чисто алгебраической информации о дискретной системе (например, размер матричных элементов), что обеспечивает оптимальную эффективность для нескольких сеток без геометрической информации. Таким образом, AMG сохраняет оптимальную вычислительную сложность GMG и может использоваться как решение черного ящика Plug-and-Play, а универсальность и надежность алгоритма значительно улучшены. В частности, методы AMG могут легко и эффективно решать многие проблемы со сложными областями, неструктурированными сетками и негладкими коэффициентами, даже некоторые из которых не имеют фона для уравнений в частных производных. И вообще приветствуется практическое применение.

Здесь мы кратко рассмотрим эволюцию метода AMG. AMG была представлена в начале 1980-х годов Брандтом, Маккормиком, Джоном Руге и другими [brandt82 / 84, ruge83, stuben83] и существует уже более трех десятилетий. Ruge и Stüben усовершенствовали его, сформировав то, что теперь называется классическим методом AMG (C-AMG) [ruge87], также известным как метод RS. В 1990-х годах C-AMG получил дальнейшее развитие в различных практических приложениях [huang91, zaslavsky93/95, fuhrmann95, chang96, griebel96, grauschopf97, fullenbach00, cleary00, huang01, kraus02], а адаптивность алгоритмов постоянно повышалась. В литературе [stuben01a/01b] Stüben делает хорошее резюме развития этого периода. В конце 90-х годов, движимые практическими приложениями, появились новые алгоритмы AMG, в которых были разработаны два типа методов. Классические-это метод, основанный на частичной геометрии и аналитической информации. Такие методы строят специальные грубые сетки и интерполяционные операторы с использованием информации о базисных функциях конечных элементов, матрицах единиц и жесткости, а также о



свойствах уравнений в частных производных, в основном: методы, основанные на гладких агрегациях (SA-AMG) [vanek96 / 01, mandel99, notay10/12], методы на основе слияния ячеек[chan98]и интерполяции ячеек (AMGe) [brezina00] и их варианты[henson01, chartier01/03], методы на основе минимаксной интерполяции энергии[wan99, xujc04, brannick06a, vassilevski06, vassilevski10, olson11], методы на основе информации о сетке[shu04/05/06b], методы на основе конечных элементов высокого порядка [heys05, shu04/06, sun05, huang06, lai11], методы для анизотропных задач[brannick12], методы для уравнений в частных производных[li01, stuben01a/01B, fullenbach02] и их комбинации[jones01, kolev06, chartier07, lashuk08, efendiev11, brezina11]и др. Другая категория-адаптивные методы AMG. Традиционные методы AMG обычно имеют неявное предположение о том, что аппроксимация нулевого пространства известна заранее, и на основе этого построены соответствующие компоненты алгоритма. Adaptive AMG не требует этого предположения, основная идея состоит в том, что компоненты алгоритма адаптивно настраиваются и оптимизируются в процессе решения. К таким методам относятся: адаптивная стратегия укрупнения на основе совместимости релаксации (CR) [brandt00, livne04, falgout05a, brannick06b, brannick10, методы адаптивной фильтрации (Adaptive Filtering) [wagner97, wagner00, nagel08], метод бутстрапа AMG[brandt02, manteuffel10, brandt11, bolten11, brannick13, brandt15, dambra19]и методы, основанные на адаптивной интерполяции сокращения[maclachlan06/07/12]и др. Существуют и другие методы, такие как: метод, основанный на локально оптимальных операторах интерполяции[dohrmann07], метод, основанный на вычислительных элементах[kraus06/08], Lean AMG[livne12] и недавний метод AMG (RN-AMG), основанный на корневом узле [manteuffel17], который сочетает C-AMG с SA-AMG. Эти алгоритмы улучшают адаптивность и надежность методов AMG с разных точек зрения.



Теоретически анализ сходимости AMG далек от зрелости алгоритма GMG. Текущий теоретический анализ в основном сосредоточен на двухслойном сеточном случае [brandt86, ruge87, braess95, stuben01, vanek01, chang01, mandel03, falgout04b, falgout05a, liu05, napov12, macLachlan14, hu16, brannick18, xuxf18]. Для алгоритма C-AMG теоретические результаты в основном сосредоточены на таких случаях, как M-матрица, положительно определенная матрица и симметричная положительно определенная матрица со слабой диагональной доминантой [ruge87, huang91, chang96, stuben99, chang02]. Брезина и др. построили двухслойную последовательную теорию сходимости AMGe [brezina00] на основе слабых аппроксимационных свойств интерполяционных операторов, но только для точечной релаксации и некоторых специальных типов грубых переменных. Для более общих гладких подмногообразий и грубых типов переменных Фальгоут и Василевский устанавливают общую двухслойную теоретическую структуру [falgout04b, falgout05a], может применяться к более широкому кругу задач и алгоритмов, таких как уравнения Максвелла. В последнее время Сюй Сюэфэн и Чжан ченсон дали заряды для интерполяционного оператора идеала AMG через теорию сходимости двухслойных сеток, что дало новые идеи для построения алгоритма AMG [xuxf18]. Для многослойного случая нет полной теоретической гарантии. Структура пространственной декомпозиции и коррекции подпространства, разработанная Сюй Цзином гиперпространством [xujc92/17b], предоставляет некоторые теоретические аналитические инструменты и информацию, но существует много пробелов для построения строгой теории сходимости многослойных сеток для различных алгоритмов. В статье [ruge87, stuben01a/b] Ruge и Stueben обобщают результаты классических алгоритмов и теорий AMG. Последние достижения в области методов AMG можно найти в недавно опубликованной обзорной статье Сюй Цзин Чао и Зикатанова [xujc17], которая

систематически обобщает и комментирует вышеупомянутые типы алгоритмов и теорий AMG.

С другой стороны, важным достижением в AMG-подходе является разработка параллельных алгоритмов AMG, на которых основное внимание уделяется подготовке этой статьи. В конце 1990-х годов, с появлением крупномасштабных параллельных вычислений, потребность в практических приложениях для параллельных вычислений росла, что привело к изучению параллельных алгоритмов AMG. Для большинства компонентов алгоритма, таких как сглаживание и интерполяция, параллельная реализация относительно проста, опираясь на связанные с GMG параллельные вычисления[jones97, mo97, falgout00]. Однако для процесса огрубления сетки разработка параллельных алгоритмов AMG представляет собой существенные трудности из-за присущей последовательной природе классических стратегий огрубления, и связанные с ними исследования проходят относительно долгий извилистый путь. В 1998 году Cleary и Falgout et al., национальная лаборатория Лоуренса Ливермора (LLNL), США, предложили по своей природе параллельную стратегию огрубления сетки на основе алгоритма CLJP [cleary98], которая вдохновила исследование параллельных вычислений AMG. С тех пор отечественные и зарубежные ученые предложили ряд параллельных алгоритмов огрубления сетки[haase00, tuminaro00, krechel01, henson02, haase02, haase03, joubert03, gallivan03, bank05, xuxw05, yang06, alber06, alber07, greibel06, griebel07, mo07, [xuxw07a / 07b], преодоление трудностей, связанных с последовательным характером стратегии огрубления сетки в классическом алгоритме AMG, делает алгоритм AMG применимым с большим успехом на сотнях и тысячах процессорных ядер в то время от триллионов до триллионов параллельных компьютеров. В то же время LLNL разработала первый в мире параллельный

пакет AMG BoomerAMG [henson02], который всегда был самым важным и наиболее характерным компонентом параллельной предобусловленной библиотеки Hypre[soft-hypre]. Однако с появлением более крупномасштабных высокопроизводительных компьютеров ограничения этих алгоритмов постепенно раскрываются. В 2005 году тесты на миллионном компьютере BlueGene/L LLNL показали [sterck06a], что алгоритмы AMG, которые хорошо работают на тысячах ядер, трудно масштабировать до более чем десяти миллионов ядер. BlueGene / L имеет 130 000 процессорных ядер, что является самым производительным параллельным компьютером в мире в то время и первым компьютером на международном уровне с более чем 100 000 процессорных ядер. Анализ показывает, что основной причиной невозможности масштабирования алгоритмов AMG является высокая операторная сложность классического принципа огрубления сетки RS (конкретное определение см. В разделе 2 этой статьи), что приводит к тому, что масштаб вычислений не масштабируется везде, где количество ядер процессора увеличивается, что приводит к тому что масштабируемость не проявляется на компьютерах с триллионами ядер до триллионов раз. Чтобы преодолеть это узкое место, De Sterck и Yang et al. предложили принцип огрубления сетки с низкой операторной сложностью, основанный на чрезвычайно независимых множествах, и соответствующий алгоритм PMIS стратегии параллельного огрубления сетки [sterck06b]. Эта стратегия преодолела узкое место в параллельной масштабируемости и вскоре расширила классический AMG до всей системы BlueGene / L[sterck06a / b], что стало важным прорывом в исследованиях параллельных вычислений AMG. Однако на дне нет бесплатного обеда, и хотя проблема параллельного масштабирования в некоторой степени решена, параллельная

эффективность не идеальна (см. раздел 4). Более того, конвергенция алгоритма страдает, поскольку PMIS использует стратегию быстрого грубой обработки (напоминает агрессивное огрубление), направленную на значительное снижение операторной сложности, но в результате чего связь с грубой сеткой уменьшается, что приводит к ослаблению сходимости. В частности, для практических задач со сложными прикладными характеристиками, такими как мультимедиа, мультимасштаб, мультифизическая связь (междисциплинарность) и сильные разрывы, скорость сходимости алгоритма AMG резко снижается по мере увеличения вычислительного масштаба и сильно отклоняется от оптимальной сходимости алгоритма с несколькими сетками. Чтобы смягчить противоречие между параллельной масштабируемостью и быстрой конвергенцией, за последние десять лет ученые проделали большую работу, предложив ряд решений [sterck08, xuxw09, baker11a, yang10, gahvari13, falgout14, vassilevski14, treister15, ballard16, bienz16, xuxw16/17, bienz19] с разной степенью успеха в разных приложениях. Однако для конкретного класса практических задач разработка алгоритмов AMG с хорошим параллелизмом и оптимальной сходимостью по-прежнему является довольно сложной открытой проблемой.

Проблемы не заканчиваются там. В настоящее время пиковая производительность суперкомпьютеров достигла миллиардного тысячного порядка  $10^{12}$ , и она находится на заключительной стадии ускорения до десяти миллиардов тысяч вычислений  $10^{13}$  (Exascale или E-level, Эксафлопный уровень). По сравнению с компьютером Petascale (Petascale, P-level) десять лет назад, в дополнение к дальнейшему расширению параллельного масштаба, компьютер Petascale демонстрирует много новых функций в

архитектуре, особенно внутриузловой архитектуре. Это усложняется, и особенности многоуровневого вложенного массивного параллелизма и ускоренного ускорения разнородных многих ядер становятся все более очевидными, что делает параллельную реализацию и оптимизацию производительности алгоритма AMG сталкиваются с беспрецедентными проблемами. В будущем эта проблема будет обостряться на десяти миллиардах компьютеров. Серия исследований показала, что [baker10b / 11b / 11c / 11d, gahvari11 / 12, doe-solver12, doe-examath14, park15, doe-ecp-st19], чтобы адаптироваться к эволюции современной и будущей архитектуры суперкомпьютеров, AMG Алгоритм переработан, и даже принципы построения алгоритма изменены.

Подводя итог, можно сказать, что для крупномасштабных вычислений для практических приложений текущий алгоритм AMG сталкивается с «непрерывным расширением параллельного масштаба, ведущим к уменьшению параллельной масштабируемости», «возрастающей сложностью функций приложений, приводящих к повышенной вычислительной сложности алгоритмов», и «сложными особенностями машинной архитектуры» Приводит к низкой эффективности вычислений с плавающей точкой. В этой статье анализируются эти проблемы и рассказывается о прогрессе, достигнутом учеными и приложениями за последние годы. Эта статья организована следующим образом: в разделе 2 представлен базовый алгоритм AMG, в разделе 3 представлен обзор приложений и программного обеспечения AMG, в разделе 4 рассматриваются три вышеуказанные задачи, представлены текущие результаты исследований, а в разделе 5 делаются обобщения и описываются перспективы.

## 2. Основной алгоритм AMG.

Этот раздел берет классический алгоритм AMG в качестве примера, чтобы кратко представить основные компоненты алгоритма метода AMG. Для получения подробной информации о построении различных алгоритмов AMG вы можете обратиться ко многим документам, упомянутым в предыдущем разделе, который ограничен в пространстве и не будет повторяться здесь. Китайское введение в алгоритм AMG см. В [xuhw07b] и главе 2 [gu15]. Рассмотрим систему разреженных линейных алгебраических уравнений  $Ax = b$ , где  $A$ ,  $x$  и  $b$  - разреженные матрицы, неизвестные векторы и правые части СЛАУ соответственно. Хотя метод AMG не опирается на информацию о сетке, в описании алгоритма все еще используется стандартная терминология с несколькими сетками. Фактически граф смежности разреженной матрицы можно рассматривать как виртуальную сетку. Игнорируя отношения между неизвестными, мы обычно непосредственно отмечаем неизвестное множество  $\Omega = \{x_1, x_2, \dots, x_n\}$  как сетку. Взяв в качестве примера двухслойную сетку, алгоритм 1 дает последовательность действий для решения СЛАУ с помощью классического алгоритма AMG [ruge87].

### 3. Обзор приложений и программного обеспечения AMG.

AMG стал одним из самых популярных алгоритмов для решения многих сложных вычислительных задач в практических приложениях, особенно для практических задач с такими функциями, как мультифизика, многомасштабность, мультимедиа, неструктурированные сетки и сложные области. Например, большая часть приложений в программе ASC США использует AMG в качестве основного алгоритма решателя [marinak01 / 11, gittings08, leininger11], многих крупных проектов Министерства энергетики

США, включая проект Advanced Computing for Scientific Discovery (SciDAC) [doe-scidac] и проект расчета десяти миллиардов миллиардов (ECP) [doe-ecp] развернули разработку более эффективных алгоритмов AMG в качестве важного контента для развертывания. Согласно соответствующим отчетам, алгоритм AMG играет ключевую роль в повышении вычислительной мощности смежных приложений Министерства энергетики [henson03, falgout05b, zosel07, gittings08, doe-bt09, springmeyer11, smith11, doe-solver12, yang13, doe-examath14, doe-ecp-ST19]. Фактически, каждая из трех основных лабораторий Национального управления по ядерной безопасности Министерства энергетики США (NNSA) имеет свое собственное программное обеспечение AMG, такое как BoomerAMG [henson02] от LLNL, MueLu [soft-ml] от SNL и LAMG от LANL [joubert03]. Среди них, BoomerAMG является основным решателем параллельной библиотеки предварительных условий HYPRE [soft-hypre], которая в основном сосредоточена на классическом алгоритме AMG, является первым в мире параллельным программным обеспечением AMG и одним из наиболее широко используемых в мире решателей AMG. MueLu является основным компонентом Trilinos [soft-trilinos], который фокусируется на алгоритме сглаженной агрегации AMG (SA-AMG) и в основном используется в инженерной механике, связанной с моделированием и анализом. LAMG - это непубличный программный пакет, в основном используемый для приложений, связанных с LANL [gittings08]. Ввиду важности алгоритма AMG в ключевых приложениях, решатель AMG также был определен в качестве программы тестирования для американской программы ASC на производительность в десять миллиардов миллиардов компьютерных тестов [coral-bench]. Кроме того, алгоритм AMG широко принят в промышленности



и широко используется в автомобилестроении [stueben01a / 01b], моделировании полупроводниковых приборов [fullenbach00 / 02, clees02], разведке нефти и газа [willien09, clees10, gries14] и моделировании потока подземных вод [stuben03] И другие отрасли промышленности. Также было разработано соответствующее коммерческое программное обеспечение, например, программное обеспечение SAMG [soft-samg], разработанное Немецким институтом научных вычислений и алгоритмов (SCAI), является первым коммерческим программным обеспечением AMG. Коммерческое программное обеспечение ANSYS для механического анализа было выпущено с версии 5.7. Решил и начал активно использовать [ansys01 / 03]. Позднее ANSYS сотрудничал с NVIDIA для запуска версии GPU решателя AMG AmgX [ansys14 / 15]. Оценки показывают, что AMG удовлетворяет потребности клиентов в более высокой вычислительной эффективности и значительно повышает вычислительную мощность ANSYS. В разделе 3.3 мы представим некоторые типичные приложения AMG.

В Китае алгоритм AMG широко используется в лазерном синтезе [xiao03, jiang07, xuhw09a / 09b, an09, zhou11 / 12, yue15, yue17 / 18a, xuhw16 / 17], инженерной механике [tian18], сложной электромагнитной среде [wangwj18], большой Крупномасштабные интегральные схемы [chen10 / 12, zhang16, zhouh18], резервуары и поток подземных вод [li01, hu11 / 13, cheng13, guan15, qiao17] и другие применения. В этих приложениях алгоритм AMG обычно интегрируется или упаковывается в виде сторонних библиотек в такие программные среды, как JASMIN [soft-jasmin, mo10], JAUMIN [soft-jaumin, liuqk16] и PHG [soft-phg, zhang16]. Ключевые алгоритмы, которые поддерживают эти приложения для крупномасштабных вычислений.



В Китае относительно мало программного обеспечения, связанного с AMG, в основном, в том числе JXPAMG [soft-jxramg], разработанного Пекинским институтом девяти и Университетом Сянтань, FASP [soft-fasp], разработанного Сюй Цзиньчао и Чжан Чэньсун. Среди них JXPAMG в основном используется для каркасных приложений JASMIN и JAUMIN, а FASP в основном используется для приложений моделирования пласта.

Хотя алгоритм AMG был впервые предложен для задач геодезических расчетов [brandt82], подобно GMG, решение эллиптических уравнений в частных производных в настоящее время является наиболее успешной прикладной задачей алгоритма AMG. Кроме того, алгоритм AMG широко применяется к другим типам уравнений с частными производными и даже к задачам без фона уравнений с частными производными. Например: вычислительная гидродинамика и гиперболические уравнения [robinson93, heikkola03, wabro04, desterck05], уравнения Навье-Стокса [lonsdale93, webster94, raw95], уравнения Гельмгольца и Максвелла [reitzinger02b, bochev03, adams07, hiptmair07, iraksonin07 / 12, aghabarati13 / 15, wangwj18], линейные уравнения упругости [ruge86, griebel03, xiao05 / 07a / 07b / 09a / 09b, xiao11a, xiao11b, xiao15, kraus08, baker10a / 15, augustin16], вычисления статического и статического магнитного поля [kaltenbac, reitzinger02a], расчет решетчатого материала [shu02 / 08], задача мантийной конвекции [rudi15], задача на собственные значения [han15], задача многофазного потока [bui17 / 18], дробное уравнение [yue18b], метод DG [bastian12] Цепь Маркова [desterck10, brandt11], анализ изображений и кластеризация [kimmel03, dambra19] и т. д.

В практических приложениях алгоритм AMG обычно используется в качестве предобуславливателя для итерационного метода Крылова. Для

некоторых сложных приложений прямое воздействие AMG в качестве предварительного кондиционера на дискретные системы не является идеальным или даже трудно сходящимся. Возможный метод решения проблемы этого типа - использовать физическую предварительную стратегию, сначала преобразовав исходное уравнение в одну или несколько относительно простых задач, которые AMG может эффективно решить. Типичным примером в этом отношении является предварительное условие NX [hiptmair07], предложенное Хиптмайром и Сюй Цзиньчао для решения уравнения Максвелла. Основная идея состоит в том, чтобы преобразовать уравнение Максвелла в четыре уравнения типа Пуассона с помощью стратегии вспомогательного подпространства [xujc96] (алгоритм AMG может Быстрое решение), что значительно повышает скорость сходимости решения исходной задачи. Этот алгоритм был быстро распараллелен Колевым и др. [Kolve09 / 12], и решатель AMS, основанный на предварительном кондиционере, был разработан и интегрирован в программное обеспечение HYPRE, достигнув большого успеха в практических приложениях. Научный офис Министерства энергетики США в 2009 году назвал решатель AMS одним из десяти прорывных достижений в области вычислительной техники за последние годы [doe-bt09]. В настоящее время, основанная на идее вспомогательного подпространства и алгоритме AMG для решения сложных задач, она стала общей стратегией, которая широко использовалась при практическом моделировании задач. Далее представлены несколько типичных случаев практического применения, когда алгоритм AMG успешно применяется. Эти примеры приходят из области научных и инженерных вычислений, промышленного моделирования и передовых научных открытий.

### 3.1. Механика радиационной жидкости.

В областях лазерного инерциального синтеза (ICF), астрофизики и других областях движение жидкости и передача энергии в состоянии с высокой плотностью энергии является очень сложным процессом, который описывается системой уравнений радиационной гидродинамики (RHD). Из-за влияния множества физических процессов, множества сред, множественных масштабов и сильной нелинейности решение этого типа уравнения является очень сложным и является типичной сложной вычислительной задачей [mo06, rei09]. Расчет RHD всегда был важным сценарием применения алгоритма AMG.

Нацелившись на двумерную программу RHD, Болдуин и др. Из LLNL провели оценку эффективности различных алгоритмов линейного решателя в литературе [baldwin99] и продемонстрировали потенциал многосеточного алгоритма для решения таких задач. За последние 20 лет ученые в стране и за рубежом предложили различные алгоритмы AMG для различных моделей RHD. В настоящее время AMG стал наиболее эффективным алгоритмом для решения проблем RHD. Поскольку решение таких уравнений имеет большое значение в приложениях национальной безопасности, исследования AMG по таким уравнениям в основном сосредоточены в соответствующих лабораториях Министерства энергетики США [henson02 / 03, howell03, falgout05b, brannick06c, zosel07, doe-bt09, doe-solver12, yang13, doe-ecp-st19] и Пекинский институт № 9 в Китае. Среди них исследовательская группа из девяти Пекинских институтов и Университета Сянтань провела большое количество совместных исследований двухмерных, двухмерных многогрупповых, трехмерных моделей [xiao03, jiang07, xuhw09a / 09b / 16/17, an09, zhou11 / 12, 15/17 / 18a]. Эти исследования расширили разрешающую

способность таких уравнений от однотемпературной теплопроводности до трехтемпературной теплопроводности и многогрупповой диффузии и от двумерной до трехмерной. В настоящее время алгоритм AMG для двумерной RHD является относительно зрелым [xixw18], и основная трудность заключается в трехмерной ситуации. Специально для трехмерной многогрупповой диффузионной модели с непрерывным улучшением разрешения сетки эффективность существующего алгоритма AMG все еще далека от фактических потребностей моделирования, и необходимы непрерывные и углубленные исследования.

### **3.2. Строительная механика.**

Структурно-механический анализ существует во многих практических приложениях. Этот тип приложения часто должен решать нелинейные мультифизические связанные уравнения в частных производных, и это также типичный тип вычислительно сложной задачи. Руге и Маккормик и др. Очень рано применили алгоритм AMG для решения таких задач [ruge86, mccormick89]. Адамс и др. При анализе механической структуры скелета человека, основанной на AMG, успешно завершил анализ механической структуры скелета позвоночника человека с более чем 500-миллионными степенями свободы на тысячах процессоров ASCI White и получил высокую вычислительную производительность. [Adams03 / 04a / 04b], опубликованный на Международной конференции по суперкомпьютерам, был одним из крупнейших практических приложений неявных вычислений в то время. Брезина и др. Использовали алгоритм AMG для реализации крупномасштабных расчетов задач структурной механики формирования кристаллов на тысячах процессоров [brezina06b]. В последнее время Тянь Ронг и др. Использовали задачи расчета теплового контакта и статического

анализа, связанные с оценкой безопасности плотин Тяньшэнцзяо и плотины Три ущелья в Китае, используя физические предварительные условия для преобразования исходной задачи в нескольких операторов, подобных Пуассону, и Используя алгоритм AMG для решения этой задачи, моделирование плотины с миллиардами степеней свободы было успешно реализовано на десятках тысяч процессорных ядер Tianhe II [tian18]. Специально для реальной модели плотины Три ущелья в этой работе были достигнуты расчеты с высоким разрешением, превышающие 5 миллиардов степеней свободы, и впервые в мире был проведен общий статический анализ плотины Три ущелья в Китае. Эти работы показывают эффективность и важную роль алгоритма AMG в практическом крупномасштабном численном моделировании.

### **3.3. Промышленные применения.**

Это приложение в основном приводит примеры из Германии. Условно говоря, одна из выдающихся особенностей Германии в области численного моделирования заключается в том, что она уделяет особое внимание исследованиям основных алгоритмов и сочетанию исследований и разработок программного обеспечения с промышленными приложениями, а также движению рынка и современным тенденциям. Эта особенность может быть хорошо отражена в разработке алгоритмов и программного обеспечения AMG. Можно сказать, что алгоритм AMG проник во все аспекты немецкого промышленного применения. Здесь мы в основном представляем пример Фраунгоферовского института научных вычислений и алгоритмов (SCAI) в Германии [scai].

На самом деле создание института напрямую связано с алгоритмом AMG. В 1980-х годах основатель института Троттенберг и его коллега Штюбен

раскрыли потенциал многосеточных алгоритмов, особенно AMG в промышленных приложениях, и привели команду к разработке первого в мире коммерческого программного обеспечения для алгоритмов AMG SAMG [soft-samg] и более поздние параллельные версии SAMGr и XAMG. Позже SAMG сотрудничал с промышленными отделами, такими как Mercedes-Benz, чтобы применить его к проектированию автомобильных двигателей, решая главным образом проблему чрезмерно длинных автомобильных циклов проектирования из-за низкой вычислительной эффективности в то время, и добился большого успеха [krechel99]. В 1992 году Троттенберг основал SCAI в Августине, Германия, первоначально как агентство Немецкой ассоциации математики и обработки данных (GMD), занимающееся производством базовых алгоритмов. В 2001 году SCAI был включен в Ассоциацию содействия прикладным исследованиям Фраунгофера, крупнейшего в Европе научно-исследовательского института прикладных исследований, и стал институтом Фраунгофера. В то же время SAMG как семя породило серию программных продуктов. В настоящее время SCAI разработала около 20 программных продуктов, которые широко используются в промышленных областях Германии и Европы, а также SCAI стала исследовательским учреждением, которое способствует преобразованию базовых алгоритмов в промышленные приложения.

Создание и развитие SCAI - это модель, объединяющая базовый алгоритм исследования с индустриализацией посредством программной продуктивизации. Одним из факторов, сыгравших важную роль в этом процессе, было внедрение алгоритма AMG в промышленности Троттенбергом и Штюбенем. Штюбен - один из первых ученых в мире, изучавший алгоритм AMG. Так называемый классический алгоритм AMG,

также известный как алгоритм RS, назван в честь американского ученого Руге и его имени. В 2001 году Троттенберг в сочетании с разработкой SCAI подытожил последние 20 лет опыта в области многосеточных алгоритмов и разработки программного обеспечения в своей монографии «Многосеточный» [trottenberg01]. В этой книге в определенной степени обобщены многосеточные алгоритмы, управляемые приложениями. Результаты. В 2017 году нынешний директор SCAI Грибель (также эксперт по алгоритмам AMG) и его коллеги подвели итоги 25-летнего опыта SCAI с момента его создания в монографии «Научные вычисления и алгоритмы в промышленном моделировании» [griebel17], объяснив, как SCAI интегрирует основные алгоритмы с промышленностью. Приложение отлично сочетается. Среди них, Штюбен и др. Подробно подытожили достижения и опыт исследований алгоритма AMG в промышленных приложениях за последние тридцать лет под названием «Алгебраическая многосетка - от академических наук к промышленности» [stuben17]. Сегодня, как одно из ведущих программных продуктов SCAI, SAMG вошла в бизнес-процессы, разработанные автомобильной промышленностью, такими как Volkswagen и Mercedes-Benz, и широко используется в разведке нефти и газа, подземных водотоков, электронных устройств и полупроводниковых приборов, литейной промышленности, энергосистем, электрохимических процессов и т. д. Коммерческие компании в отрасли стали важной частью основной конкурентоспособности этих отраслей. Для конкретного прогресса в этом отношении, пожалуйста, обратитесь к литературе [stuben01 / 17, griebel17] и веб-странице SAMG [soft-samg], которая содержит большое количество примеров промышленного применения и больше ссылок, которые ограничены в пространстве и не будут повторяться здесь.



### **3.4. Квантовая хромодинамика.**

Типичным примером успешного применения AMG в области передовых научных открытий является расчет квантовой хромодинамики (КХД). КХД является основной теорией, описывающей сильные взаимодействия в физике частиц, и может быть использована для объяснения взаимодействий между кварками и глюонами в адронах и глюонах через цветные заряды. В настоящее время это взаимодействие исследования обычно проводятся с помощью так называемой теории решетчатой калибровки (LGT). В теоретическом расчете основной задачей является численное решение уравнения Дирака, которое представляет собой серьезную проблему для существующего метода AMG. Основная причина заключается в том, что ошибка алгебраического сглаживания дискретной линейной системы уравнения Дирака не только колеблющаяся, но и на ее колебание влияют случайные факторы в калибровочном поле, что очень трудно для построения эффективных стратегий укрупнения и интерполяции. Чтобы справиться с этими трудностями, Брежина, Бранник и Каль разработали различные адаптивные алгоритмы AMG [brannick06c, brezina06a, kahl09, brannick13], которые могут фиксировать случайность ошибок алгебраического сглаживания. Они успешно используются в вычислениях КХД и показывают относительно Большой потенциал - более эффективный алгоритм решения таких уравнений.

### **4. Алгоритм AMG сталкивается с компьютерными проблемами на уровне E-эксафлопных вычислений.**

В предыдущем разделе мы видели, что AMG добилась больших успехов в практическом применении. Однако с непрерывным улучшением производительности суперкомпьютера моделирование практических задач



становится все более и более сложным, что приводит к увеличению масштаба вычислений практических приложений и сопровождается все более сложными характеристиками приложений, что оказывает значительное влияние на производительность алгоритмов AMG. Здесь возникает большая проблема. Производительность AMG зависит от трех аспектов: параллельная масштабируемость, масштабируемость алгоритма и производительность с плавающей запятой. Параллельная масштабируемость относится к способности параллельной реализации алгоритма использовать дополнительные вычислительные ресурсы (количество ядер ЦП), измеряемые параллельной эффективностью. Масштабируемость алгоритма относится к степени, в которой объем вычислений алгоритма увеличивается при увеличении размера задачи и обычно может измеряться количеством итераций. Производительность с плавающей запятой относится к доле компонентов алгоритма, которые используют пиковую производительность компьютера, и измеряется эффективностью вычислений с плавающей запятой. Для моделирования практических задач эти три аспекта относительно независимы, но связаны друг с другом, и вместе они определяют производительность крупномасштабных параллельных вычислений AMG.

Этот раздел посвящен вычислениям на десять миллиардов миллиардов (уровень E эксафлопных вычислений) и описывает проблемы метода AMG в параллельной масштабируемости, масштабируемости алгоритмов и оптимизации производительности с плавающей запятой, а также знакомит с прогрессом исследований и тенденциями развития.

#### **4.1. Параллельные масштабируемые задачи.**

На тысячах процессорных ядер De Sterck из Университета Ватерлоо, Канада, и команда HYPRE из LLNL впервые обнаружили важное влияние сложности оператора на параллельную масштабируемость AMG [desterck06a / 06b] и явно предположили что сложно высокую сложность оператора масштабировать за тысячи ядер процессора. Следовательно, низкая сложность оператора почти стала необходимым условием для крупномасштабных параллельных вычислений AMG, особенно для трехмерных задач. Они также дали принцип параллельного укрупнения для построения малой сложности оператора и предложили алгоритм огрубления **PMIS**, основанный на максимальном независимом наборе, который удовлетворяет этому принципу, расширяя уравнение Пуассона до 128 000 ядер ЦП Blue Gene / L. В прошлом мы достигли параллельной эффективности 30% [desterck06a / b], что было лучшим результатом в то время. Следует отметить, что этот результат получается из-за слабой масштабируемости, а масштаб вычисления одного ядра близок к 30000. Исследования показали, что [baker11a, хухw13], только когда размер одного ядра достаточно велик (крупнозернистый параллелизм), AMG может достичь лучшей параллельной эффективности. По мере увеличения числа ядер, чтобы поддерживать идеальную параллельную эффективность, размер одного ядра также должен увеличиваться.

Однако, с одной стороны, крупнозернистый параллелизм не может удовлетворить потребности практических приложений. В практических приложениях, когда количество ядер увеличивается, одно ядро с таким же размером может достичь высокой эффективности параллельной работы. С другой стороны, архитектурные характеристики систем E-Class ограничат возможность грубого параллелизма. Исходя из текущей тенденции развития,

количество узлов и ядер системы E-уровня, по оценкам, достигнет порядка 100 000 и 100 млрд. Соответственно, что в 10 и 1000 раз превышает количество узлов и ядер системы P-уровня. Следовательно, улучшение производительности системы E-класса в основном достигается за счет увеличения количества ядер в узле, так что вычислительная мощность одного узла может быть значительно улучшена. Тем не менее, в соответствии с текущей тенденцией развития, увеличение емкости хранилища и пропускной способности выборки одного узла является относительно медленным. Это означает, что в системе класса E средняя память и пропускная способность одного ядра также могут уменьшиться. Поэтому традиционная практика получения хорошей параллельной эффективности AMG за счет увеличения размера одноядерных вычислений больше не будет реалистичной, и необходимо учитывать масштабируемость AMG при мелкозернистом параллелизме [bell12, хихw13].

В ответ на эту проблему тестирование и анализ команды LLNL HYPRE на нескольких системах уровня P, таких как Jaguar, показывают, что [baker11a / 11b / 11c] принимает процессы между узлами и внутриузловой поток MPI / OpenMP параллельно. Правильное сопоставление с количеством потоков может улучшить способность алгоритма AMG адаптироваться к мелкозернистому параллелизму на 100 000 тысяч процессорных ядер. В литературе [baker15] Бейкер оценил масштабируемость классического алгоритма AMG на сотнях тысяч процессорных ядер для линейной эластичности. Однако, с одной стороны, это улучшение ограничено из-за потери производительности из-за сложной архитектуры внутри узла (обсуждается в разделе 1.4.3 ниже). С другой стороны, по сравнению с улучшением производительности одного узла задержка связи и улучшение

полосы пропускания между узлами относительно медленны, а разница между производительностью вычислений внутри узлов и производительностью связи между узлами имеет тенденцию к увеличению [vienne12]. Эта тенденция еще больше уменьшит способность AMG адаптироваться к мелкозернистому параллелизму, и на уровне алгоритма AMG необходимо предложить новые идеи.

Исходя из предположения о выравнивании нагрузки, производительность параллельной масштабируемости в основном зависит от отношения количества вычислений параллельного алгоритма к объему связи. Для алгоритма AMG это в основном относится к параллельной масштабируемости одного шага итерации на этапе решения и на этапе запуска. От точных сеток до грубых сеток размер матриц становится все меньше и меньше, а гранулярность параллелизма становится тоньше, в то же время матрицы становятся плотнее, а режим связи становится все более и более сложным, что делает расчет и коэффициент связи на грубых сетках очень низкими. Поэтому грубые сетки являются основной причиной того, что AMG испытывает трудности с адаптацией к мелкозернистому параллелизму. Кроме того, исследования показывают, что сложный режим связи на уровне грубой сетки может привести к большому количеству соревнований в коммуникационной сети суперкомпьютерной сетевой системы, что может привести к нестабильной работе системы и поведению [bienz14, wu15]. Как уменьшить сложность режима связи на грубой сетке, а также потери параллельной эффективности и вызванную ею нестабильность системы, является очень актуальной проблемой, которую необходимо решить. К последним разработкам в этой области относятся: Гахвари и др. Предложены многочисленные стратегии для уменьшения перемещения данных,

компромиссного обмена данными и вычислений для сокращения трафика на уровне грубых ячеек [gahvari12 / 14a / 14b], Фальгоут и др. Предложена не грубая стратегия грубого сгиба для грубых сеток. Разреженная обработка для улучшения режима связи грубого ячеистого слоя [falgout14, treister15, bienz16], Василевский и др. Предложена стратегия аддитивной грубой сетки для уменьшения накладных расходов на связь [vassilevski14], Баллард и др. Предложена разреженная матрица для вычисления грубой сетки в процессе установки. Алгоритм чередования продуктов для уменьшения накладных расходов на связь [ballard16]. Автор этой статьи и его сотрудники предлагают адаптивную стратегию настройки для решения последовательности уравнений, обычно существующих в реальных численных моделированиях, используя динамические изменения свойств матрицы в последовательности уравнений [xuhw16]. Согласно различным свойствам матрицы, эта стратегия адаптивно регулирует количество слоев грубой сетки AMG и время сглаживания, чтобы минимизировать влияние грубой сетки на производительность параллельной масштабируемости. Недавно Bienz и др. Предложили коммуникационную стратегию с учетом узлов, основанную на многошаговом обмене данными, основанную на сложности параллельной реализации крупной сетки AMG, которая может снизить накладные расходы на обмен данными между узлами в крупномасштабных сценариях [bienz19]. Эти стратегии нацелены на различные приложения и в разной степени смягчают параллельные потери производительности, вызванные грубыми сетками. Тем не менее, для крупномасштабного параллелизма систем уровня E, как эффективно уменьшить влияние грубых сеток на производительность параллельной масштабируемости, все еще остается очень сложной задачей. Необходимо разработать конкретные вычислительные архитектуры с

высоким коэффициентом связи для вычислений. Параллельный алгоритм AMG.

#### 4.2. Проблемы масштабируемости алгоритма.

23/01/2020 Требуется уменьшить операторную сложность с помощью агрессивного огрубления.

##### **Aggressive Coarsenings**

Опыт показал [49], что часто второй проход генерирует слишком много C узлов, вызывая большие сложности и неэффективность. Следовательно, условие (C1) был изменен на следующее.

(C10'): Каждая F-точка, в которой я нуждаюсь, должна сильно зависеть хотя бы от одной C-точки  $j$ .

Теперь только первый проход огрубления RS отвечает этому требованию. Эта метод приводит к лучшей сложности, но худшей скорости сходимости. Несмотря на то, что этот подход часто значительно снижает сложность, сложность может быть достаточно высокой и требовать больше памяти, чем хотелось бы. Если C-точки находятся еще дальше друг от друга, это приводит к агрессивному огрублению. Это достигается с помощью следующего нового определения сильной связи: переменная  $i$  сильно  $n$ -связанна вдоль пути длины  $\ell$  с переменной  $j$ , если существует последовательность переменных  $i_0, i_1, \dots, i_\ell$ , с  $i = i_0$  и  $j = i_\ell$  и  $i_k$  сильно связаны (как определено ранее) с  $i_{k+1}$  для  $k = 0, \dots, \ell - 1$ . Переменная  $i$  сильно  $n$ -связанный w.r.t.  $(p, \ell)$  к переменной  $j$ , если существует хотя бы  $p$  путей длины  $\leq \ell$  так что  $i$  сильно  $n$ -связан с  $j$  вдоль каждого из этих путей. Это может быть наиболее эффективно реализованным путем применения первого прохода огрубления RS в первый раз, как описано в предыдущем разделе, во

второй раз определение сильной  $n$ -связности w.r.t.  $(p, \ell)$  только между результирующими  $S$ -точками (через соседние  $F$ -точки). Подробнее см. [49]. Результат применения агрессивное огрубление  $A2$ , то есть выбор  $p = 2$  и  $\ell = 2$ , и агрессивное  $A1$  огрубление, то есть  $p = 1$  и  $\ell = 2$ , к 5-точечному лапласиану на сетке  $7 \times 7$  показано на рисунке 2.

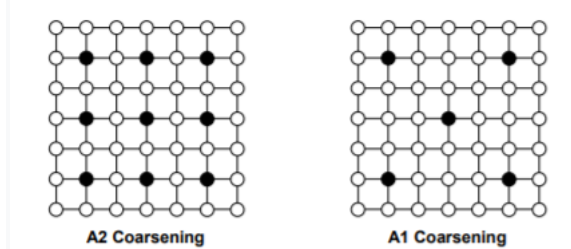


Рис. 2. Различные последовательные укрупнения для 5-точечного лапласиана.

### Aggressive Coarsenings

Используются расширенные прочные связи и соответствующий оператор интерполяции построить грубые пространства меньшей размерности. Эта процедура также известна как агрессивное огрубление. Напомним определение пути в графе, данное в §7.1 и все наши соображения на графе смежности  $G(S)$  матрицы прочности  $S \in \mathbb{R}^n \times n$  определено в (8.4). Агрессивное огрубление относится к выбору грубой сетки вершины как независимое множество в графе смежности, соответствующее оператору интерполяции на расстоянии более 2.

**Определение 9.6.** (Сильная связь вдоль пути) Вершина  $i$  называется сильно соединённой с вершиной  $j$  вдоль пути длины  $\ell$ , если существует путь  $(k_0, k_1, \dots, k_\ell)$  в  $G(S)$ , такой, что  $k_0 = i$ ,  $k_\ell = j$  и  $sc(k_m, k_{m+1}) \geq \theta$ ,  $m = 0, 1, \dots, \ell - 1$ .

Следующее определение связано с количеством вершин со строгой связью.

**Определение 9.7.**  $((m, \ell)$ -сильная связь) Для заданных целых чисел  $m > 0$  и  $\ell > 0$ , а вершина  $i$   $((m, \ell)$ -сильно связана с вершиной  $j$  тогда и только тогда,

когда  $i$  сильно соединяюсь с  $j$  по крайней мере  $m$  путей длины  $\ell$  (согласно определению 9.6).

Алгоритм агрессивного укрупнения генерирует MIS, используя алгоритм 5 для графа  $G_{m, \ell} = (V, E_{m, \ell})$  с множеством вершин  $V = \{1, \dots, n\}$  и множество ребер  $E_{m, \ell}$  определяется как  $E_{m, \ell} = \{(i, j) \mid (m, \ell) \text{ сильно связан с } j\}$  (9.9)

Как известно (Diestel 2010)  $(S^\ell)_{ij}$  отличен от нуля тогда и только тогда, когда существует путь длина  $\leq \ell$  между  $i$  и  $j$ . Агрессивное грубое использование эксплуатирует это свойство и является алгоритмом, который выбирает набор степеней свободы грубой сетки, соответствующих вершинам графа, которые находятся на расстоянии графа больше  $\ell$ . Он использует смежность график  $G(S^\ell)$  из  $S^\ell$  вместо  $G(S)$  в агрегации или алгоритме MIS.

В качестве примера рассмотрим агрессивное укрупнение при  $\ell = 2$  и  $m = 1$ . Множество грубых степеней свободы сетки получают, применяя стандарт MIS алгоритм 5 дважды: (1) мы находим MIS в  $G(S)$ , а затем получаем набор грубых степени свободы сетки  $C$  (они находятся на расстоянии не менее 2); (2) Применить MIS алгоритм второй раз на графе с вершинами  $C$ -точек и ребер между ними дан оператор силы, соответствующий  $S^2$ . Точно так же для агрегации агрессивное огрубление соответствует применению Алгоритм агрегации 6 рекурсивно, или применяя его непосредственно к графу, соответствующему  $S^\ell$  для данного  $\ell$ .

**Сделать однопоточный PMIS и интерполяцию с расстоянием более 2.**