

УДК 004.428

## АДАПТАЦИЯ РЕШАТЕЛЯ К ПОТОКУ СЛАУ

В. А. Ерзунов\*, Ю. Г. Бартенев  
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Дается описание алгоритмов и параметров адаптивного механизма решения потока СЛАУ, реализованного в библиотеке PMLP/ParSol. Исследования проведены на нескольких задачах, рассчитываемых по различным методикам. Результаты счета задач с адаптивным механизмом и без него показали, что благодаря ему можно существенно ускорить счет. Это достигается выбором наиболее подходящего метода решения СЛАУ на разных участках счета, отличающихся сложностью решения (обусловленностью матриц) СЛАУ. Использование адаптивного механизма позволяет освободить пользователя от подбора оптимального метода расчета произвольной задачи и повысить надежность решения СЛАУ в процессе компьютерного моделирования.

*Ключевые слова:* система линейных алгебраических уравнений (СЛАУ), разреженные матрицы, вычислительные системы с распределенной памятью, многоядерные процессоры с общей памятью, адаптивный механизм.

### Введение

Программа адаптивного механизма подбора решателя к потоку систем линейных алгебраических уравнений (СЛАУ)<sup>1</sup> реализована в библиотеке PMLP/ParSol [1] для использования в методиках расчета задач на многопроцессорных ЭВМ и применяется в РФЯЦ-ВНИИЭФ. Она предназначена для ускорения счета задач путем выбора наиболее подходящего метода решения СЛАУ (среди заданных) на разных участках счета, отличающихся сложностью решения СЛАУ и соответственно требующих разных методов для надежного и наискорейшего решения всего потока СЛАУ задачи. Таким образом, использование адаптивного механизма в значительной мере освобождает пользователя от необходимости подбора оптимального метода решения СЛАУ произвольной задачи.

Первая попытка реализации адаптивного механизма в библиотеке PMLP/ParSol была предпринята более десяти лет назад и опробована при моделировании теплопроводности в газодинамической методике МЕДУЗА [2]. Цель была та же — сократить время решения потока СЛАУ изменяющейся сложности (обусловленности) путем подбора наискорейшего метода среди заданных. Однако алгоритм выбора оптимального метода был довольно дорогостоящим и не очень надежным, что в дальнейшем привело к полной переработке адаптивного механизма и принципов его применения, как это представлено в настоящей статье. Среди других разработок с той же целью можно отметить адаптивное изменение параметров полиномиального сглаживателя в ходе итераций селективного решателя AMG при решении каждой СЛАУ [3], частичное построение селективного и агрегативного AMG для решения потока СЛАУ после решения СЛАУ с полным построением AMG [4, 5].

Кратко охарактеризуем методы PMLP/ParSol, которые приводятся в статье при описании адаптивного механизма. Итерационные решатели BiCGStab и CG [6] служат для решения СЛАУ соответственно с матрицей общего вида и симметричной положительно определенной матрицей в па-

\*Ерзунов Владимир Алексеевич, старший научный сотрудник,  
e-mail: VAErzunov@vniief.ru

<sup>1</sup>Для краткости будем называть эту программу просто адаптивным механизмом.

раллельном многопоточном режиме. В качестве предобусловливателей, служащих для ускорения итераций упомянутых решателей, здесь используются:

- 1) неполные разложения на портрете исходной матрицы IC0, ILU0 и с заполнением ILU<sub>t</sub> [1, 6] в последовательном режиме и в сочетании с блочным методом Якоби (JacobiBlock) или аддитивным методом Шварца (Schwartz) [1, 6] с корректировкой на грубой сетке CGC (Coarse Grid Correction) и согласованием решения в перекрытии областей по Капорину [7] в параллельном многопоточном режиме;
- 2) параллельные многопоточные Jacobi (обратная диагональ матрицы), алгебраические многосеточные предобусловливатели [1, 4] AMG(...), AMG2(...) селективного и AgAMG(...) агрегативного типов с различными параметрами огрубления, сглаживания и др.

## Описание алгоритма

Для адаптивного механизма пользователь задает набор методов решения потока СЛАУ (решатели и предобусловливатели) с параметрами, среди которых адаптивный механизм будет выбирать наиболее быстрый на предстоящем участке счета задачи в предположении, что сложность решения СЛАУ изменяется нечасто. Для выбора конкретных методов решения из большого числа имеющихся в библиотеке пользователь должен исходить из свойств СЛАУ, возникающих в процессе счета конкретной задачи. А именно, должны быть учтены блочность, размер, регулярность, симметричность матрицы или ее отсутствие, неизменность или изменяемость ее портрета, точность решения, критерий сходимости и характеристики вычислительного комплекса, т. е. возможность использования распределенной и общей памяти, наличие сопроцессоров и другие особенности. Для адаптивного механизма желательно задать разные по мощности методы решения на основании проведенных испытаний. В разных задачах, а также на разных участках счета одной задачи, отличающихся обусловленностью матриц, могут требоваться разные решатели и/или их параметры. Особенно это актуально для моделирования нестационарных процессов.

После оценки предположительного времени решения СЛАУ всеми методами из набора на последнем временном шаге (далее просто шаг) и времени решения СЛАУ текущим методом на этом шаге адаптивный механизм определяет метод, которым предположительно СЛАУ будет решаться наиболее быстро на следующем шаге. Отметим, что на шаге может решаться несколько СЛАУ, порождаемых применением метода Ньютона в случае нелинейности моделируемого процесса. Предположительное время решения СЛАУ каким-либо методом основывается на ранее проведенном его испытании на каком-то предыдущем шаге счета. Чаще всего оказывается, что на следующем шаге следует использовать текущий метод, что дополнительно может сократить весьма затратное время построения предобусловливателя.

Для задания параметров метода решения потока СЛАУ в задаче используется структура `parsol` библиотеки PMLP/ParSol, в которой задается метод построения и решения СЛАУ вне зависимости от того, включен адаптивный механизм в работу или нет. Без адаптивного механизма для каждого типа СЛАУ<sup>2</sup> один и тот же метод, указанный в этой структуре, используется от начала до конца задачи или до его замены на другой в ходе счета задачи с применением интерфейсных методов [1]. При использовании адаптивного механизма параметры в структуре `parsol` в ходе счета неявно меняются в соответствии с заданным набором методов и выбирается предположительно наискорейший метод (решатель и предобусловливатель) для решения следующих СЛАУ.

Можно выделить несколько этапов работы адаптивного механизма:

- чтение параметров из файла и подготовка к работе;
- испытание назначенных для выполнения методов;
- выбор оптимального метода для дальнейшего счета;
- обработка особой ситуации, т. е. реакция на нерешенность очередной СЛАУ.

<sup>2</sup>PMLP/ParSol допускает расслоение потока СЛАУ на подпотоки, соответствующие, например, разным физическим процессам, которые можно решать разными методами.

Файл `adapt_preconds.pmlp` с параметрами для адаптивного механизма должен находиться в той же папке, откуда производится запуск задачи. Чтение этого файла происходит при инициализации библиотеки в задаче. Его отсутствие означает не ошибку, а лишь отказ от использования адаптивного механизма.

После этого происходит подготовка к выполнению программы адаптивного механизма (в том числе, на каждом процессе — инициализация некоторых переменных и выделение памяти под рабочие массивы), но ее выполнение не начнется, пока не наступят специальные события, заданные параметрами. В памяти этой программы заводится массив структур по числу заданных методов, из которого в зависимости от алгоритма выбора метода решения та или иная структура будет выбираться и заноситься в структуру `parso1` библиотеки PMLP/ParSol для построения предобусловливателя и продолжения счета.

Проведение испытаний — это замеры времени полного и/или частичного (на основе ранее выполненного полного) построения предобусловливателя, времени решения СЛАУ, числа итераций и количества СЛАУ на шаге. На каждом шаге испытывается только один предобусловливатель из заданного набора. После решения каждой СЛАУ на шаге в ходе испытаний уточняются время построения предобусловливателя в зависимости от признака неизменности портрета исходной и предобусловливающих матриц и времени, затраченного на одну итерацию (цена итерации). После завершения решения всех СЛАУ испытываемым методом на закончившемся шаге выбирается следующий метод для испытания на следующем шаге. Испытания заканчиваются, когда собрана информация обо всех испытываемых методах (решателях и предобусловливателях). К дальнейшей работе не допускаются методы, заблокированные по результатам решения очередной СЛАУ (например, произошла ошибка при построении предобусловливателя или возвращен код *в ходе решения произошла нештатная ситуация*).

Если во время испытаний произошла нештатная ситуация, то испытанный метод отбраковывается, испытания считаются законченными неудачно и начнутся заново с началом очередного шага без отбракованного метода. Если во время испытаний какого-либо метода решателем возвращен код *заданная точность не достигнута*, то этот метод блокируется на несколько ближайших испытаний, после чего может быть восстановлен. При возникновении такого события решение СЛАУ повторится с любым другим незаблокированным методом из набора.

Испытания начинаются либо сразу после чтения параметров и окончания подготовки, либо после того, как суммарное число итераций решателя СЛАУ на шаге превысит заданное в параметрах значение. Испытания также могут начаться без достижения указанного суммарного числа итераций, если адаптивный механизм находится в состоянии *готов к включению* и метод, заданный в структуре `parso1`, не дал решения СЛАУ. После окончания испытаний адаптивный механизм приходит в состояние *в работе*. Если во время испытаний оказались заблокированы (отбракованы) все указанные методы, то работа адаптивного механизма прекращается и восстанавливаются первоначальные параметры метода решения, заданные в структуре `parso1`.

Следует отметить, что проведение испытаний почти не замедляет счета задачи, так как каждая СЛАУ решается всегда один раз, пусть даже неэффективным методом.

Для выбора метода решения перед началом очередного шага используются следующие данные, полученные в результате испытаний:

- число решенных СЛАУ на закончившемся (будем считать его текущим) шаге (`n_slaes`);
- суммарное число итераций на текущем шаге (`n_iters`);
- суммарное число итераций на предыдущем шаге (`prev_iters`);
- время полного построения предобусловливателя с новым портретом (`T_full`);
- время неполного построения предобусловливателя со старым портретом (`T_mini`);
- суммарное количество итераций на шаге во время испытаний (`test_iters`);
- время одной итерации (`t_one_iter`).

Результаты последних четырех пунктов получены во время ранее проведенных испытаний. Поэтому результат выбора зависит от того, насколько своевременно проведены испытания. То есть для точного сравнения методов испытания должны проходить на одном и том же шаге на одних

и тех же СЛАУ. Но поскольку это накладно, то предполагаем, что сложность решения СЛАУ на шагах во время испытаний примерно одинакова. Другое допущение заключается в том, что на следующих шагах после испытания будут решаться СЛАУ примерно такой же сложности, что и во время испытаний. Постепенное изменение сложности решения СЛАУ от шага к шагу в любую сторону приводит к тому, что в какой-то момент накопившихся изменений станет достаточно для того, чтобы адаптивный механизм сменил текущий метод на другой, более подходящий для данного этапа счета.

Конечно, вышеуказанные предположения и допущения в реальной задаче не всегда выполняются, поэтому в процессе счета есть возможность коррекции результатов испытаний для текущего метода. При скачкообразном изменении сложности потока СЛАУ во время испытаний возможны промахи в выборе самого эффективного метода решения для продолжения счета. Поэтому испытания должны проводиться периодически, как предписано параметром адаптивного механизма, или внепланово, например, при резком изменении сложности решаемых СЛАУ (условие определяется параметрами), а также изменении размера СЛАУ или заказе на перестроение адаптивного механизма. Поскольку цель адаптивного механизма — сокращение времени работы решателя СЛАУ, то это накладывает требование точности замеров времени, что обеспечивается даже в условиях *шума* ЭВМ.

Таким образом, перед началом очередного шага происходит вычисление предполагаемого времени  $T_s$  (см. далее) решения СЛАУ каждым методом из списка незаблокированных, в том числе текущим, как если бы эти методы (каждый в отдельности) использовались в ее решении на закончившемся шаге. А дальше осуществляется поиск метода с предположительно минимальным временем решения СЛАУ и выдача его номера для дальнейшего построения и использования в счете. Как уже отмечалось, чаще всего использованный на текущем шаге метод оказывается предположительно лучшим и на следующем шаге, до тех пор пока сложность СЛАУ не изменится существенно.

Обозначим порядковый номер предобусловливателя, выбранного ранее, через  $j$ , а очередного испытываемого — через  $i$ . Для каждого метода  $i \neq j$  сначала вычисляется предполагаемое число итераций для завершившегося (текущего) шага по формуле

$$p\_iters[i] = test\_iters[i]/test\_iters[j] \times n\_iters.$$

Затем вычисляется предполагаемое время выполнения итераций методом  $i$ , как если бы он использовался вместо метода  $j$  на текущем шаге:

$$T\_s[i] = p\_iters[i] \times t\_one\_iter[i].$$

К нему остается добавить время построения предобусловливателя либо при полном построении с постоянно изменяющимся портретом матрицы

$$T\_s[i] = T\_s[i] + T\_full[i] \times n\_slaes, \quad (1)$$

либо при изменении портрета матрицы только в начале шага

$$T\_s[i] = T\_s[i] + T\_full[i] + T\_mini[i] \times (n\_slaes - 1), \quad (2)$$

либо при неизменном портрете матрицы на всей задаче

$$T\_s[i] = T\_s[i] + T\_mini[i] \times n\_slaes. \quad (3)$$

Если подсчет предполагаемого времени по формулам (1) и (2) достаточно очевиден, то в формуле (3) есть некоторая некорректность, способная привести к ошибке при выборе наиболее оптимального предобусловливателя. Отметим, что (3) относится к нелинейным задачам, где на каждом шаге выполняются итерации Ньютона, порождающие последовательность СЛАУ обычно с одинаковым портретом матриц. В случае смены предобусловливателя все равно его придется строить полностью, т. е. правильной будет формула (2), но только на первом шаге, а на следующих шагах и при плавном изменении сложности СЛАУ в основном будет более правильной формула (3), которая и используется. При этом трудно угадать, произойдет ли полное построение предобусловливателя

на следующем шаге. Если при неизменном портрете матрицы полная перестройка предобусловливателя уже произошла, то на следующем шаге с большой вероятностью ее не будет, но она не исключена на последующих шагах. Таким образом, остается неясным, нужно ли учитывать в этом случае  $T\_full[i]$  — время полного построения предобусловливателя  $i$  — для оценки времени работы решателя. А для ряда предобусловливателей, например многосеточных, именно это время может быть в некоторых случаях основной долей времени решения СЛАУ.

После вычисления предполагаемого времени решения на шаге для всех доступных методов самый быстрый метод становится текущим.

Выбор оптимального метода решения СЛАУ происходит на каждом процессе и синхронизован, чтобы не допустить выбора разных решателей на разных процессах.

При решении потока СЛАУ с неполным построением предобусловливателя может произойти полная перестройка схемы использования предобусловливателя при превышении заданного в параметрах числа итераций. Так, при завершении решения с кодом *точность не достигнута* текущий предобусловливатель в дальнейшем строится полностью и СЛАУ решается заново. Если же ситуация повторится, то данный метод будет заблокирован до следующих испытаний и будет настраиваться следующий метод из набора.

Наступление очередного испытания ускорится, если разница числа итераций на текущем ( $n\_iters$ ) и предыдущем ( $prev\_iters$ ) шагах превысит заданное в параметрах число.

В параметрах можно также задать выполнение адаптивным механизмом замеров требуемых времен на каждом шаге для текущего метода. Эта возможность позволяет минимизировать последствия ошибок на испытаниях из-за чередующейся сложности СЛАУ в потоке.

Наиболее затратная часть программы — это проведение испытаний для заданных методов решения СЛАУ, некоторые из которых могут оказаться затратными по времени построения или решения СЛАУ. Поэтому после выбора оптимального метода можно заблокировать на несколько испытаний выбор затратных методов, для которых предполагается время решения СЛАУ намного больше, чем для оптимального. Это сокращает время определения наискорейшего метода. При этом не исключена ситуация, когда в процессе счета число претендентов на роль оптимального может сократиться до одного.

Включение в работу адаптивного механизма может быть полезным даже в том случае, если в его списке методов задан всего один. В случае, когда заданная точность не достигнута, адаптивный механизм пытается довести до нее решение, используя полное построение предобусловливателя, повторное решение с другим начальным приближением (см. далее) либо возвращаясь на время к методу, заданному в структуре `parso1`, или с помощью более мощного метода, указанного в качестве запасного, но не принимающего участия в решении до наступления нештатной ситуации.

При отключении программы адаптивного выбора (условие задается в параметрах) в использовании для решения СЛАУ остается последний выбранный наискорейший метод.

## Параметры программы

Файл `adapt_preconds.pmlp` с параметрами адаптивного механизма состоит из информационных строк, начинающихся с символа `#`, и собственно строк с параметрами. Параметры, указанные в этом файле, заменяют параметры в структуре `parso1` [1]. Таким образом, полный набор параметров для каждого метода состоит из указанных в файле и структуре `parso1`.

Все параметры из файла делятся на записи. Запись — набор параметров для одного из методов решения потока СЛАУ, используемого в адаптивном механизме или для управления. Каждая запись представляет собой одну строку файла параметров, в которой может стоять символ `#` — признак комментария, следующего до конца строки.

Запись для каждого метода решения СЛАУ может состоять из разделов параметров, соответствующих структуре `parso1`. Каждый раздел состоит из конкретных параметров раздела. Разделителями между параметрами могут служить любые символы, например пробел или знак `=`, кроме знака `#`. Имя раздела и имя параметра определяют расположение параметра в структуре `parso1`.

Таким образом, всю информацию из файла параметров адаптивного механизма можно разделить на комментарии, символьные (ключевые) параметры, обозначающие имена разделов или параметров в разделах, и численные значения указанных параметров в разделах.

Завершает информацию в файле строка `endfile`.

Приведем пример файла параметров для адаптивного механизма:

```
# parameters for adaptive
adapt=3 in_it=7 out_it=5 ignor=1 trial=10 x0=0 block=2
BiCGStab(max_iters=200, resrhs, scaling=2, rtol=1e-07)AgAMG
Schwartz (cross=4, join=4, ILUt_float(part=0.01))
CG(max_iters=400, resrhs, scaling=2, rtol=1e-010) AMG2_cheby
endfile
```

Первая строка с параметрами задает число методов, условия включения/выключения адаптивного механизма и его коэффициенты. Таким образом, в первой строке задано три метода для работы в адаптивном механизме, который включится, когда число итераций на шаге достигнет семи, и отключится при числе итераций на шаге менее пяти. Результаты решения СЛАУ, для которых требуется одна или менее итераций, будут проигнорированы.

При возникновении либо на испытаниях, либо на шаге ситуации завершения решения СЛАУ с кодом *заданная точность не достигнута* продолжение решения следующим методом в зависимости от параметра `x0` возможно со следующим вектором начального приближения:

- равным полученному вектору решения;
- нулевым;
- заданным пользователем.

Рекомендуется использовать первый вариант, имеющий место по умолчанию, а также явное задание, как в приведенном примере (`x0=0`).

Для испытаний в примере отводится не более 10% времени работы решателей. В первой строке также задан коэффициент для временной приостановки участия в испытаниях методов, которые в два раза медленнее оптимального.

Вторая строка приведенного примера задает решатель `BiCGstab` с максимальным числом итераций, равным 200, критерием сходимости `ResRhs`<sup>3</sup> и масштабированием. Точность решения задана равной  $10^{-7}$ . Предобусловливатель — `AgAMG`; параметры, необходимые для его построения, находятся в структуре `parso1` или используются по умолчанию.

Следующая строка задает предобусловливатель `Schwartz` с числом перекрытых строк, равным 4, согласованием в перекрытиях по Капорину [7] и остальными параметрами по умолчанию.

Параметры предобусловливателей, заменяющие значения, установленные по умолчанию, задаются в виде списка в круглых скобках, следующего за названием предобусловливателя, или в качестве расширения его названия. В данном случае `ILUt_float` задает предобусловливатель `ILUt` с разложением одинарной точности (точность разложения 0,01), который будет применяться в подобластях. В качестве предобусловливаемого будет использоваться решатель, определенный в структуре `parso1`.

В последнем указанном в примере методе будет использоваться решатель `CG` с максимальным числом итераций 400, критерием сходимости `ResRhs` и масштабированием. Точность решения задана равной  $10^{-10}$ . Задан предобусловливатель `AMG2` со сглаживателем Чебышёва. Параметры, необходимые для его построения, находятся в структуре `parso1`.

Последний метод в списке не участвует в испытаниях. Он является запасным и начнет применяться, когда остальные методы окажутся заблокированными по той или иной причине. Как правило, запасной метод выбирается из числа наиболее мощных и является наиболее затратным при построении. Этот метод никогда не блокируется. Если и с помощью запасного метода не удастся достичь заданной точности, то адаптивный механизм отключается, а пользователю возвращается соответствующий ненулевой код ответа решателя.

<sup>3</sup>Критерий сходимости `ResRhs`:  $\|Ax^{(k)} - b\| \leq atol + rtol \cdot \|b\|$ , где  $x^{(k)}$  — вектор решения на  $k$ -й итерации; `rtol` обычно задается в диапазоне от  $10^{-5}$  до  $10^{-9}$ , `atol` — равным нулю.

Ряд параметров для любых СЛАУ — возможное снижение точности решения на первой итерации Ньютона, возможная перенумерация для сужения ленты матрицы и др. — не зависит от метода решения и определяется значениями в структуре `parso1`.

Для использования адаптивного механизма в прикладной программе в начале каждого шага нужно вызвать коллективную интерфейсную функцию [1], так как именно в ней проводится анализ времен и числа сделанных итераций на завершившемся шаге для выбора оптимального метода решения на следующем шаге.

Поскольку анализ результатов счета может выполняться только после завершения счета на очередном шаге, то это накладывает ограничение на использование адаптивного механизма. Его не рекомендуется использовать в задачах с известным оптимальным методом на всех участках счета и в задачах, имеющих резко отличающийся по сложности решения единственный поток СЛАУ.<sup>4</sup> Наиболее эффективно использование адаптивного механизма для задач с постепенными изменениями сложности решения потока СЛАУ.

Для некоторых методик РФЯЦ-ВНИИЭФ, например "Логос Тепло" [10], нельзя использовать дополнительный файл `adapt_preconds.pmlp` для адаптивного механизма. Для них реализован интерфейсный метод передачи ограниченного числа параметров, который достаточен для всех классов задач. Кроме того, в процессе эксплуатации описываемой программы в разных методиках для каждой из них определен свой набор методов решения в адаптивном механизме по умолчанию. Поэтому для адаптивного решения потока СЛАУ достаточно указать номер методики РФЯЦ-ВНИИЭФ в структуре `parso1`.

Помимо основной задачи — наискорейшего решения потока СЛАУ — в программе адаптивного механизма реализованы блоки сбора статистической информации о работе, истории решения СЛАУ с временными характеристиками входящих методов и трассировкой участков программы. Эти блоки управляются отдельными параметрами и включаются при необходимости отладки задачи.

## Результаты тестирования

Тестирование программы адаптивного механизма выбора метода решения осуществлялось на наборах СЛАУ и задачах, рассчитываемых по разным методикам РФЯЦ-ВНИИЭФ. В тестовых задачах демонстрируется работа адаптивного механизма и приводится оценка быстродействия в расчетах с использованием и без использования этого механизма.

**Задача 1.** На рис. 1 показан график пошагового изменения сложности решения газодинамической задачи с теплопроводностью по методике МИМОЗА [8]. Размер СЛАУ — 72 200 944 строк, число MPI-процессов — 384. В качестве решателя СЛАУ теплопроводности задавался BiCGStab с ResRhs-критерием сходимости и точностью решения  $10^{-8}$ .

Для исследования использовались методы:

- 1) ILUp3 — ILU0 трехдиагональной части матрицы;
- 2) JacobiBlock(ILU0, CGC=600) — с двумя итерациями ILU0 и решением грубой СЛАУ, содержащей 600 уравнений;
- 3) AMG.

Для тестирования использовались распределенные СЛАУ, записанные на разных участках счета. На начальном и самом простом участке наискорейшим оказался решатель BiCGStab с предобуславлителем ILUp3, который не позволял решить СЛАУ на других участках. На сложных участках наискорейшим оказался BiCGStab с AMG, а на небольшом их числе — с JacobiBlock.

При счете задачи на сложных участках с включенным адаптивным механизмом для BiCGStab уверенно выбирался AMG как наискорейший при достаточно частом включении испытаний. На

<sup>4</sup>Последняя проблема преодолевается, если общий поток СЛАУ, например, для разных физических процессов разделить на несколько потоков примерно одинаковой сложности, создав для каждого потока СЛАУ отдельный объект линейной системы.

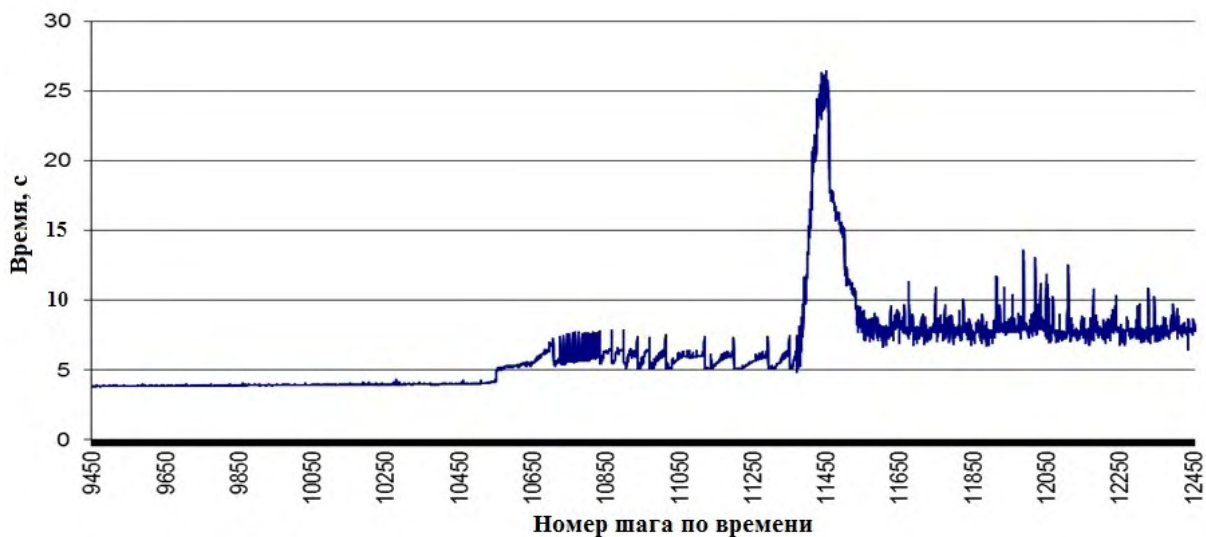


Рис. 1. График зависимости времени решения задачи от номера шага

первых шагах задачи применялся ILU<sub>r</sub>3, сокративший время ее решения примерно на 2%. Затем использовался второй метод и далее на сложном участке — AMG с неполным построением на нескольких шагах.

Без адаптивного механизма, например, при использовании метода AMG с полным построением время решения СЛАУ значительно возросло бы, судя по пикам кривой, которые относятся к шагам, где выполнялось полное построение AMG на первой СЛАУ шага. При использовании более слабого метода, например JacobiBlock(ILU0), заданная точность на сложном участке не была бы достигнута за приемлемое число итераций и задача или не решилась бы, или для ее решения потребовалось намного больше времени за счет уменьшения числа Куранта.

**Задача 2.** Для исследования поведения адаптивного механизма при решении СЛАУ теплопроводности двумерной газодинамической задачи по методике МЕДУЗА [9] задавались четыре предобусловливателя решателя BiCGStab:

- 1) Schwartz(ILUt);
- 2) JacobiBlock(ILUt, CGC=600);
- 3) AMG;
- 4) Schwartz(ILUt, CGC=600).

Адаптивный механизм включался после достижения на шаге суммарного числа итераций больше четырех при решении всех СЛАУ. Блокировка методов была отключена, игнорировались результаты решения СЛАУ с числом итераций меньше двух, испытания выполнялись каждые 400 шагов.

Запуски производились на 64 процессах без потоков. Размер СЛАУ — 957 696 строк. В качестве итерационного решателя использовался BiCGStab с ResRhs-критерием сходимости и точностью решения  $10^{-6}$ . Для оценки эффективности методов использовалось число выполненных задач шагов в течение одного часа.

При применении методов 1, 2 и 4 из приведенного выше списка без использования адаптивного механизма решение останавливалось из-за того, что даже при большом числе итераций заданная точность не достигалась. Были опробованы и другие методы, но они либо не давали результата, либо работали слишком медленно.

С предобусловливателем AMG\_mpr без использования программы адаптивного выбора были решены все СЛАУ задачи и получены следующие результаты за час счета: число шагов 2 479; доля времени решателя 40%. С использованием программы адаптивного выбора метода решения СЛАУ получено число шагов 3 186; доля времени решателя составила 30%.



Таким образом, использование адаптивного выбора метода решения повысило скорость счета этой задачи, сохранив надежность. При этом задача была досчитана до конца. Ускорение от использования адаптивного механизма составило 1,28 раза.

**Задача 3.** В задаче, рассчитываемой по методике "Логос Тепло" [10], решались СЛАУ диффузии тепла в твердом теле. В состав адаптивного механизма входили предобусловливатели BiCGStab:

- 1) JacobiBlock(ILU0);
- 2) AgAMG;
- 3) AMG2\_cheby.

Запуски производились на 108 процессах без потоков. Размер СЛАУ — 9 874 456 строк. Время решения с указанным составом методов адаптивного механизма составило 36,1 с. Без AMG2\_cheby время решения составило 86,3 с, без AMG2\_cheby и AgAMG — 212,1 с.

При применении самого простого и самого быстрого по построению метода JacobiBlock(ILU0) время возросло за счет сильного увеличения числа итераций BiCGStab на сложных участках. В этой задаче лучше всего использовались многосеточные предобусловливатели. В других задачах лучше проявил себя JacobiBlock(ILU0).

**Задача 4.** При решении задачи по газодинамической методике MPC [11] решатели СЛАУ теплопроводности использовались на 192 процессах в многопоточном режиме. Размер СЛАУ — 44 888 111 строк.

Для этой задачи задавался итерационный решатель CG с критерием сходимости ResRhs и точностью  $10^{-8}$ . Максимальное число итераций было равно 400. Время решения равнялось 3 часам. Число шагов задавалось равным 240.

Для исследования поведения адаптивного механизма задавались три метода:

- 1) ILUp3;
- 2) JacobiBlock(...,IC0);
- 3) Schwartz(IC0).

Адаптивный механизм включался после достижения на шаге суммарного числа итераций больше двух при решении всех СЛАУ.

Приведем статистические данные счета задачи:

- общее время работы решателей 3 924,7 с;
- время на итерациях 3 482,4 с;
- общее число итераций 26 268;
- общее число решенных СЛАУ 1 035;
- доля времени, занятая решателями, 51,5 %.

Результаты использования методов в составе программы адаптивного выбора метода решения СЛАУ показаны в таблице. Видно, что в процессе решения СЛАУ применялись всего лишь два метода — JacobiBlock и Schwartz. Третий — наиболее дешевый ILUp3 — использовался только в испытаниях. Включение всех этих методов в состав адаптивного набора обусловлено тем, что хотя неизвестно, какие из них окажутся наиболее эффективными для генерируемой задачей потока СЛАУ, но известно, что на других участках счета задачи они будут выбраны как наискорейшие.

Заметим, что до внедрения адаптивного механизма СЛАУ в задаче решалась только с предобусловливателем JacobiBlock, но медленнее.

## Результаты использования методов на примере задачи МРС

Метод решения	Число решенных СЛАУ	Время решения, с	Доля времени использования метода
ILUp3	88	457,2	13,1
JacobiBlock	496	1 573,3	45,2
Schwartz	451	1 451,9	41,7

## Заключение

Адаптивный механизм решения потока СЛАУ был опробован в применяемых в РФЯЦ-ВНИИЭФ методиках МИМОЗА, МЕДУЗА, "Логос Тепло" и МРС. Результаты счета задач с адаптивным механизмом и без него показали, что он может дать ускорение счета путем выбора наиболее быстрого (среди заданных) метода решения СЛАУ на разных участках счета, требующих разных методов решения. Использование адаптивного механизма может освободить пользователя от необходимости подбора оптимального метода решения СЛАУ произвольной задачи по ряду методик и повысить надежность решения СЛАУ.

## Список литературы

1. Алейников А. Ю., Барабанов Р. А., Бартенев Ю. Г., Ерзунов В. А., Карпов А. П., Кузнецов В. Ю., Петров Д. А., Резчиков В. Ю., Стаканов А. Н., Щаникова Е. Б. Применение параллельных решателей СЛАУ в пакетах программ инженерного анализа РФЯЦ-ВНИИЭФ // Супервычисления и математическое моделирование. Тр. XVI межд. конф. Саров: РФЯЦ-ВНИИЭФ, 2017. С. 28–36.  
*Aleynikov A. Yu., Barabanov R. A., Bartenev Yu. G., Erzunov V. A., Karpov A. P., Kuznetsov V. Yu., Petrov D. A., Rezchikov V. Yu., Stakanov A. N., Shchanikova E. B.* Primenenie paralelnykh reshatel'ey SLAU v paketakh programm inzhenernogo analiza RFYaTs-VNIIEF // Supervychisleniya i matematicheskoe modelirovanie. Tr. XVI mezhd. konf. Sarov: RFYaTs-VNIIEF, 2017. S. 28–36.
2. Ерзунов В. А., Горбунов А. А. Механизм адаптивного выбора решателя в библиотеке PMLP/ParSol // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2009. Вып. 1. С. 55–62.  
*Erzunov V. A., Gorbunov A. A.* Mekhanizm adaptivnogo vybora reshatelya v biblioteke PMLP/ParSol // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsesov. 2009. Vyp. 1. S. 55–62.
3. Жуков В. Т., Краснов М. М., Новикова Н. Д., Феодоритова О. Б. Алгебраический многосеточный метод с адаптивными сглаживателями на основе многочленов Чебышёва: Препринт № 113. М.: ИПМ им. М. В. Келдыша РАН, 2016. doi:10.20948/prepr-2016-113.  
*Zhukov V. T., Krasnov M. M., Novikova N. D., Feodoritova O. B.* Algebraicheskiy mnogosetochnyy metod s adaptivnymi sglazhivatel'nyami na osnove mnogochlenov Chebysheva: Preprint № 113. M.: IPM im. M. V. Keldysha RAS, 2016. doi:10.20948/prepr-2016-113.
4. Бартенев Ю. Г., Карпов А. П. Многократное использование многоуровневой структуры данных в алгебраическом многосеточном решателе СЛАУ // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2019. Вып. 4. С. 66–77.  
*Bartenev Yu. G., Karpov A. P.* Mnogokratnoe ispolzovanie mnogourovnevoy struktury dannykh v algebraicheskom mnogosetochnom reshatele SLAU // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsesov. 2019. Vyp. 4. S. 66–77.
5. Trilinos. <http://trilinos.org>.

6. Дж. Саад. Итерационные методы для разреженных линейных систем. 2-е изд. в 2 томах. М.: МГУ, 2014.  
*J.Saad. Iteratsionnye metody dlya razrezhennykh lineynykh algebraicheskikh uravneniy. 2-e izd. v 2 tomakh. M.: MGU, 2014.*
7. Капорин И. Е., Миллокова О. Ю. Массивно-параллельный алгоритм предобусловленного метода сопряженных градиентов для численного решения систем линейных алгебраических уравнений // Оптимизация и приложения / Под ред. В. Г. Жадана. М.: ВЦ им. А. А. Дородницына РАН, 2011. С. 127–152.  
*Kaporin I. E., Milyukova O. Yu. Massivno-parallelny algoritm predobuslovlennogo metoda sopryazhyennykh gradientov dlya chislennogo resheniya system lineynykh algebraicheskikh uravneniy // Optimizatsiya i prilozheniya. / Pod red. V.G. Zhdana. M.: VTs im. A.A. Dorodnitsyna RAN, 2011. S. 127–152.*
8. Софронов И. Д., Винокуров О. А., Змушко В. В., Плетенёв Ф. А., Измайлова Т. Б., Потapkина Л. Ф., Рыбаченко П. В., Рябикина Н. А., Соколова Н. В., Трофимова Л. Я. Комплекс программ МИМОЗА-99 // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 1999. Вып. 4. С. 37–41.  
*Sofronov I. D., Vinokurov O. A., Zmushko V. V., Pletenyev F. A., Izmaylova T. B., Potapkina L. F., Rybachenko P. V., Ryabikina N. A., Sokolova N. V., Trofimova L. Ya. Kompleks program MIMOZA-99. // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsesov. 1999. Vyp. 4. S. 37–41.*
9. Горбунов А. А. Метод решения уравнения теплопроводности на нерегулярных сетках в параллельном режиме в методике МЕДУЗА // Там же. 2008. Вып. 3. С. 32–46.  
*Gorbunov A. A. Metod resheniya uravneniya teploprovodnosti na neregulyarnykh setkakh v parallelnom rezhime v metodike MEDUZA // Tam zhe. 2008. Vyp. 3. S. 32–46.*
10. Вишняков А. Ю., Дерюгин Ю. Н., Глазунов В. А., Чистякова И. Н. Пакет программ ЛОГОС. Модуль расчета сопряженных и связанных задач теплопереноса // Супервычисления и математическое моделирование. Тр. XIV межд. конф. Саров: РФЯЦ-ВНИИЭФ, 2013. С. 154–161.  
*Vishnyakov A. Yu., Deryugin Yu. N., Glazunov V. A., Chistyakova I. N. Paket programm LOGOS. Modul raschyeta sopryazhyennykh i svyazannykh zadach teploperenosa // Supervychisleniya i matematicheskoe modelirovanie. Tr. XIV mezd. konf. Sarov: RFYaTs-VNIIEF, 2013. S. 154–161.*
11. Дерюгин Ю. Н., Полищук С. Н., Тихомиров В. П. Расчет лучистой теплопроводности в методике МРС с использованием неточных методов Ньютона // Супервычисления и математическое моделирование. Тр. XV межд. конф. Саров: РФЯЦ-ВНИИЭФ, 2015. С. 198–205.  
*Deryugin Yu. N., Polishchuk S. N., Tikhomirov V. P. Raschet luchistoy teploprovodnosti v metodike MRS s ispolzovaniem setochnykh metodov Nyutona // Supervychisleniya i matematicheskoe modelirovanie. Tr. XV mezhd. konf. Sarov: RFYaTs-VNIIEF, 2015. S. 198–205.*

Статья поступила в редакцию 08.07.20.

SOLVER ADAPTATION TO A SLAE FLUX / / V. A. Erzunov\*, Yu. G. Bartenev (FSUE "RFNC-VNIIEF", Sarov, N. Novgorod region).

The paper describes algorithms and parameters of an adaptive mechanism used to solve SLAEs and implemented in the PMLP/ParSol library. Studies were performed by solving several problems using different codes. Results obtained with and without the adaptive mechanism demonstrate that it may significantly increase the speed of computations. This can be achieved by selecting an optimum SLAE solution method in various stages of the compute process differing from each other in the SLAE solution complexity (matrix conditioning). With the use of the adaptive mechanism there is no need in selecting an optimum method to solve an arbitrary problem and also the SLAE solution reliability may be enhanced in the computer simulation process.

*Keywords:* system of linear algebraic equations (SLAE), sparse matrices, distributed-memory computing systems, multicore shared-memory processors, adaptive mechanism.

---

\*Erzunov Vladimir Alekseevich, senior scientist,  
*e-mail:* VAErzunov@vniief.ru

---