

Оценка применимости предобусловливателя AIPS в качестве альтернативы ILU(0) при гидродинамическом моделировании нефтегазовых месторождений на графических процессорах

Добровольцев Александр Сергеевич, Сохатский Михаил Александрович,
Юлдашев Артур Владимирович
Институт компьютерных исследований, УУНИТ



Время решения СЛАУ при численном решении задачи моделирования фильтрационных течений углеводородов в пористой среде может достигать от 60 до 90 % от всего времени расчета.

При численном решении задачи многофазной фильтрации в пористой среде:

- после дискретизации (метод конечных объемов + неявная схема) получаем нелинейную систему:

$$R(x) = 0$$

- для линеаризации используется метод Ньютона:

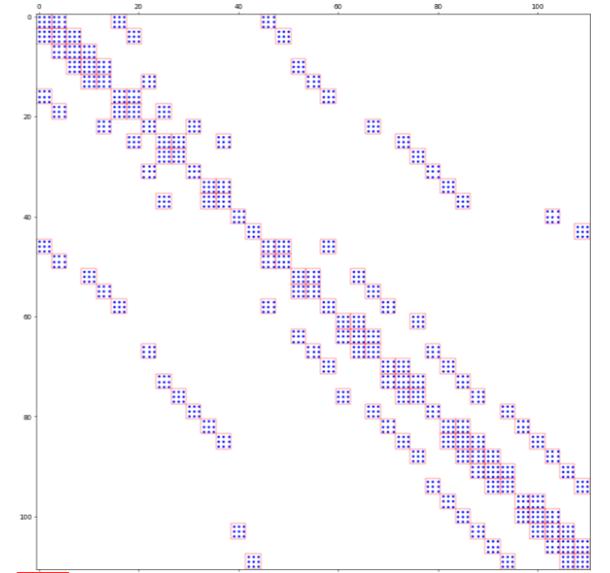
$$J(x^n)\delta x^n = -R(x^n),$$

$$x^{n+1} = x^n + \delta x^n,$$

где $J(x^n)$ – матрица Якоби, $R(x^n)$ – вектор остатка.

Матрица Якоби представляет собой несимметричную разреженную матрицу с мелкоблочной структурой, характеризующаяся плотными подматрицами размера $k \times k$, где k – количество фаз в гидродинамической модели.

Предобусловливатель AIPS [1], основанный на аппроксимации обратной матрицы с использованием ряда Неймана, обладает большим ресурсом параллелизма, но не учитывает мелкоблочную структуру матриц.



 – подматрица размером $k \times k$

Особенности матриц:

- размерность $10^6 - 10^8$
- заполненность менее 0,01 %
- плохая обусловленность
- несимметричность по значениям
- наличие нерегулярной части
- **мелкоблочная структура**, характеризующаяся плотными подматрицами размера $k \times k$, где k – количество фаз в гидродинамической модели



Цель работы – исследование применимости предобусловливателя AIPS, адаптированного к мелкоблочной структуре матриц, в качестве альтернативы ILU(0) при решении СЛАУ, возникающих при гидродинамическом моделировании нефтегазовых месторождений, на гибридных вычислительных системах с графическими процессорами.

Задачи исследования

1. Реализация модификации предобусловливателя AIPS с учетом мелкоблочной структуры матрицы СЛАУ для многопроцессорных систем с распределенной памятью.
2. Реализация модификации предобусловливателя AIPS, оптимизированной для вычислений на графических процессорах.
3. Исследование эффективности разработанной модификации AIPS относительно ILU(0) на графических процессорах.
4. Исследование эффективности разработанной модификации AIPS относительно ILU(0) в качестве второй ступени предобусловливателя CPR на графических процессорах.

Предобусловливатель AIPS



Ряд Неймана: $(E - B)^{-1} = E + B + B^2 + B^3 + \dots, \rho(B) < 1$,

где E – единичная матрица, $\rho(B)$ – спектральный радиус матрицы B .

Если представить матрицу в виде:

$$A = P + R = P(E + P^{-1}R) = P(E - (-1)(P^{-1}R))$$

то при выполнении условия $\rho(P^{-1}R) < 1$ можно получить следующую формулу для построения предобусловливателя, аппроксимирующего обратную матрицу при заданном N :

$$A^{-1} \approx M_N^{-1} = \left(\sum_{k=0}^N (-1)^k (P^{-1}R)^k \right) P^{-1}, \text{ при условии } \rho(P^{-1}R) < 1.$$

$$Ax = b \quad \longrightarrow \quad x \approx M_N^{-1}b = P^{-1}b + \sum_{k=1}^N (-1)^k (P^{-1}R)^k P^{-1}b$$

В [1] в качестве P использовалась трехдиагональная часть матрицы A (рис. 1), для которой был предложен параллельный алгоритм пакетного решения линейных систем с трехдиагональными матрицами с использованием скалярной прогонки на GPU.

В данной работе предлагается в качестве P использовать мелкоблочную трехдиагональную часть матрицы A (рис. 2) с применением для её решения блочной прогонки в пакетном режиме.

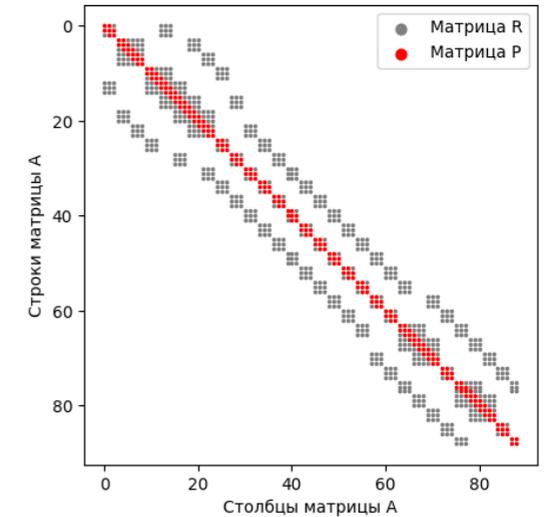


Рис. 1

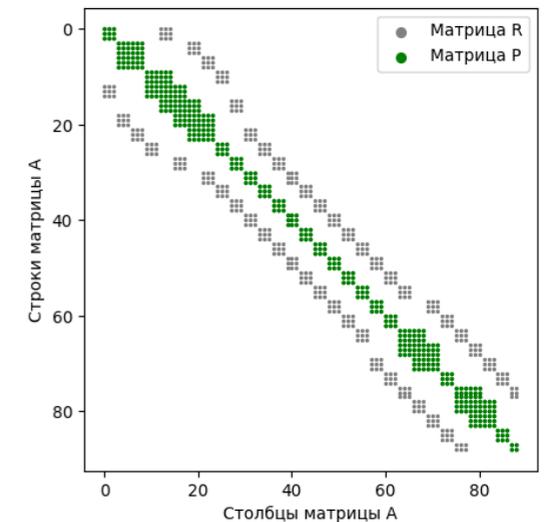
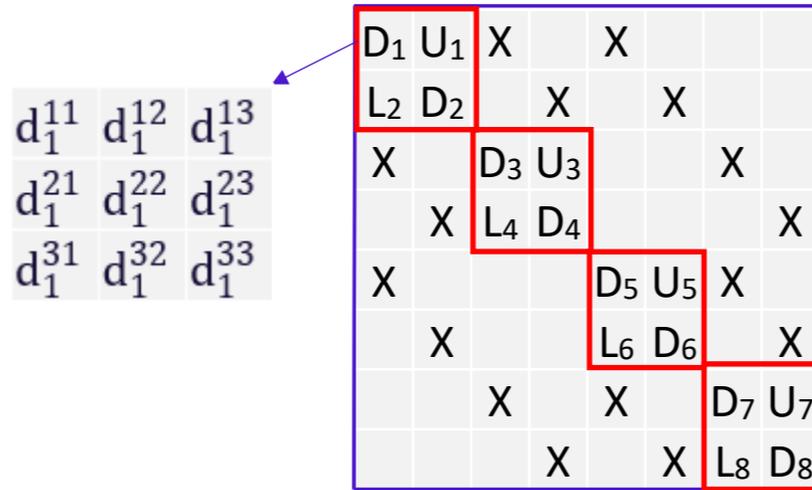
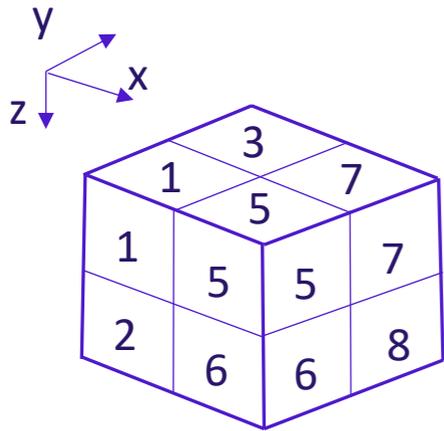


Рис. 2

[1] Юлдашев А.В., Репин Н.В., Спеле В.В. Параллельный предобусловливатель на основе степенного разложения обратной матрицы для решения разреженных линейных систем на графических процессорах // Вычислительные методы и программирование. 2019. Т. 20, № 4. С. 444–456.



Матрица A

D – мелкоблочные элементы главной диагонали
 U – мелкоблочные элементы верхней диагонали
 L – мелкоблочные элементы нижней диагонали
 X – мелкоблочные элементы вне трехдиагональной части

В мелкоблочную трехдиагональную часть матрицы СЛАУ входят коэффициенты, характеризующие связь соседних ячеек по оси Oz. Тогда мелкоблочная трехдиагональная часть матрицы СЛАУ будет состоять из $Nx * Nz$ независимых подчастей размера $Nz \times Nz$.

Алгоритм метода блочной прогонки решения мелкоблочных трехдиагональных СЛАУ

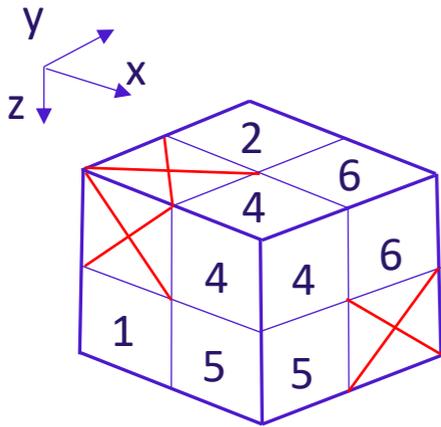
Прямой ход метода:

1. $U_1^* = D_1^{-1} U_1;$
2. $b_1^* = D_1^{-1} b_1$
3. $U_i^* = (D_i - L_i U_{i-1}^*)^{-1} U_i, i = 2, \dots, n;$
4. $b_i^* = (D_i - L_i U_{i-1}^*)^{-1} (b_i - L_i b_{i-1}^*), i = 2, \dots, n;$

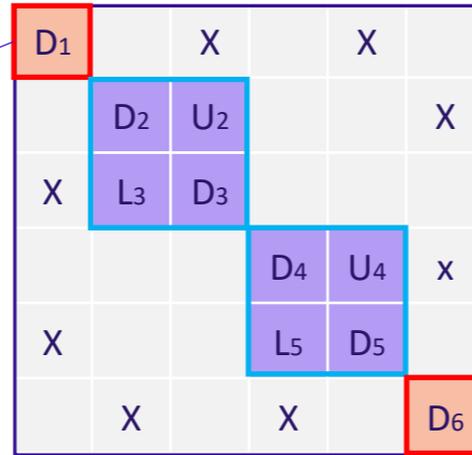
Обратный ход метода

1. $X_n = b_n^*;$
2. $X_i = b_i^* - U_i^* X_{i+1}, i = n - 1, n - 3, \dots, 1;$

Пакетное решение мелкоблочных трехдиагональных СЛАУ на GPU



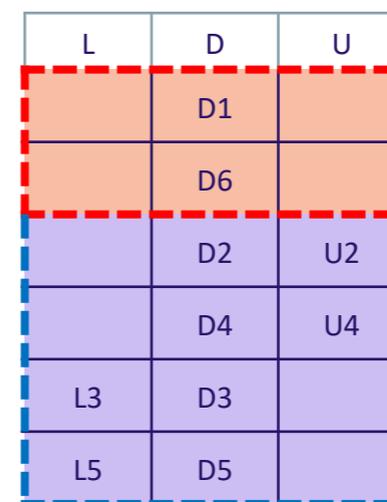
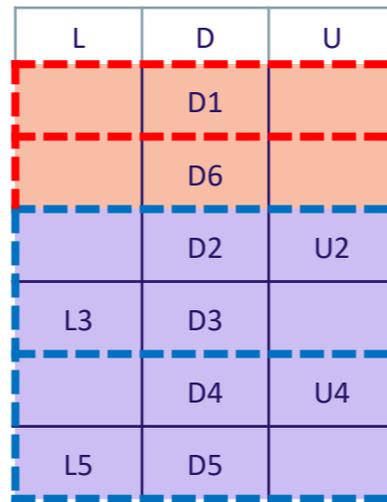
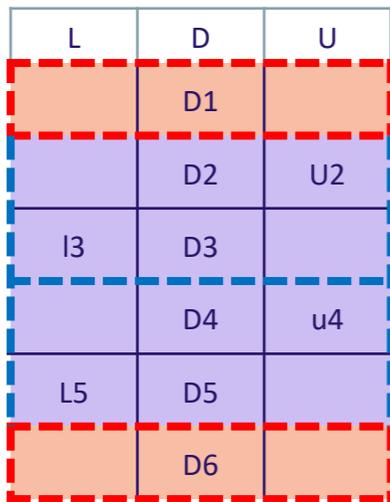
$$\begin{matrix} d_1^{11} & d_1^{12} & d_1^{13} \\ d_1^{21} & d_1^{22} & d_1^{23} \\ d_1^{31} & d_1^{32} & d_1^{33} \end{matrix}$$



Матрица A

D – мелкоблочные элементы главной диагонали
 U – мелкоблочные элементы верхней диагонали
 L – мелкоблочные элементы нижней диагонали
 X – мелкоблочные элементы вне трехдиагональной части

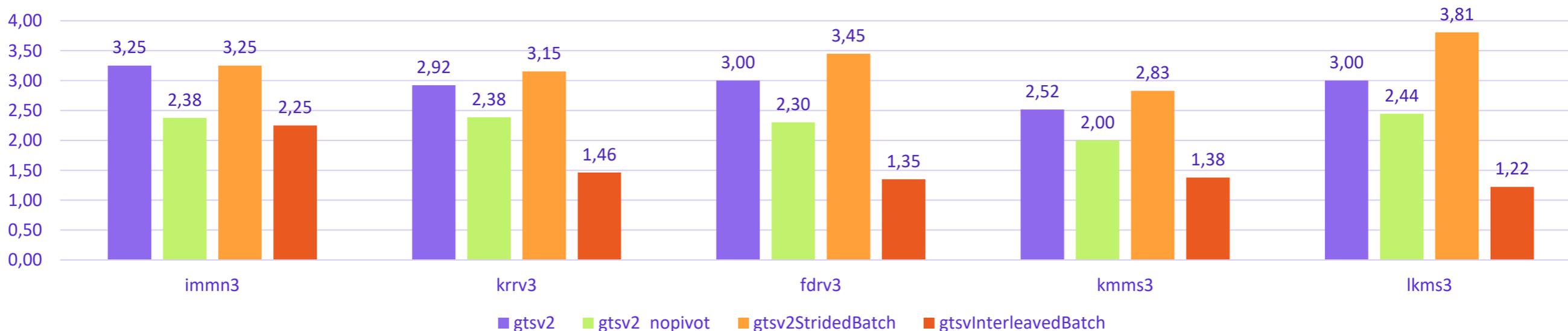
В целях балансировки нагрузки блоки переупорядочиваются согласно их размерности.



Ранее [4] было проведено сравнение быстродействия предложенного ранее алгоритма пакетного решения трехдиагональных СЛАУ на GPU A100 с алгоритмами, реализованными в библиотеке cuSPARSE: `cusparselt>gtsv2` и `cusparselt>gtsv2_nopivot` – функции, предназначенные для решения трехдиагональных СЛАУ общего вида, а также `cusparselt>gtsv2StridedBatch` и `cusparselt>gtsvInterleavedBatch` – функции для пакетного решения множества трехдиагональных СЛАУ одинаковой размерности.

Название матрицы	Размерность матрицы	Минимальный размер блока	Максимальный размер блока	Количество независимых блоков
imn3	768 034	1	34	221 402
krrv3	1 440 307	1	39	233 866
fdrv3	2 203 421	1	55	633 202
kmms3	2 876 965	1	121	389 199
lkms3	4 555 235	1	27	1 167 013

Ускорение предложенного алгоритма относительно cuSPARSE



Среднее ускорение относительно cuSPARSE: 2,94 (gtsv2) / 2,3 (gtsv2_nopivot) / 3,3 (gtsvStridedBatch) / 1,53 (gtsvInterleavedBatch).



Характеристики матриц

Название матрицы	Размерность матрицы	Количество ненулевых элементов	Среднее число ненулевых элементов в строке	Модифицированный AIPS $\rho(P^{-1}R)$
immn	2 304 102	42 859 314	18,6	0,918
mmnt	5 637 747	109 595 799	19,4	0,993
fdrv	6 610 263	118 221 633	17,9	0,806
kmms	8 630 895	167 332 329	19,4	0,995

Число итераций с AIPS при различном числе членов ряда Неймана

N	1	2	3	4	5	6	7	8	9	10
immn	10,5	13	9	12,5	9	12	7,5	9,5	7	10
mmnt	5	4	3,5	3	2,5	2,5	2,5	2,5	2,5	2,5
fdrv	1,5	1	1	1	1	1	0,5	0,5	0,5	0,5
kmms	13	15	9	10,5	12,5	8	7	7,5	5,5	7,5

Время расчета с AIPS при различном числе членов ряда Неймана, мс

N	1	2	3	4	5	6	7	8	9	10
immn	63,4	104,3	94,9	154,7	134,1	201,5	145,8	202,5	167,2	255,7
mmnt	82,6	91	99,1	103,1	102,5	114,7	126,4	138,3	150,1	161,7
fdrv	50,7	49	54,8	59,7	64,6	69,7	53,6	56	58,7	61,5
kmms	265,6	411,8	330,1	457,3	630,1	476,7	476,6	564,4	467,1	674,8

Численные результаты получены на высокопроизводительном сегменте вычислительного кластера УУНиТ на основе многопроцессорных узлов с 2 x CPU Intel Xeon Gold 6326 и GPU NVIDIA A100, а также облачной платформе immers.cloud с видеокартами GPU NVIDIA H100, GPU NVIDIA H200.

Расчеты проводились с итерационным методом BiCGStab и следующими предобусловливателями: ILU(0), AIPS. Условие остановки итерационного процесса – достижение относительной невязки величины $\varepsilon = 10^{-4}$.

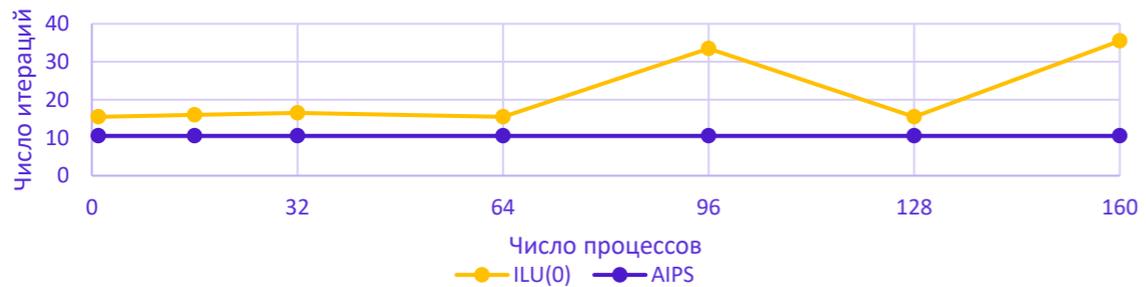
Оптимальное число членов ряда Неймана для AIPS установлено равным $N = 1$, дальнейшие расчеты выполнены с этим значением.

Сравнение эффективности ILU(0) и AIPS на CPU

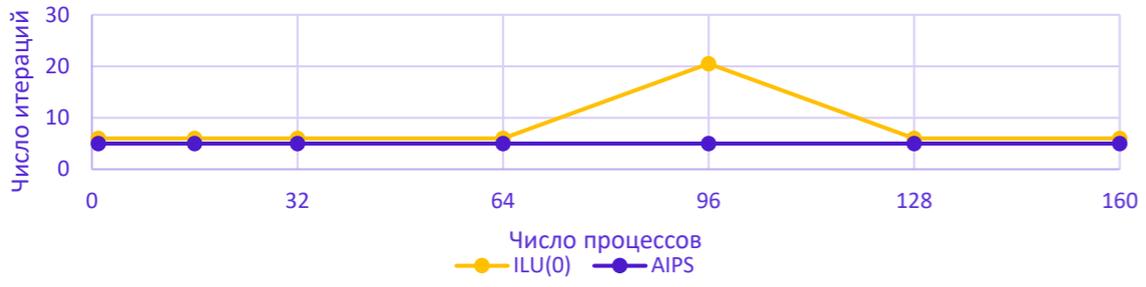


Число итераций при использовании ILU(0) и AIPS

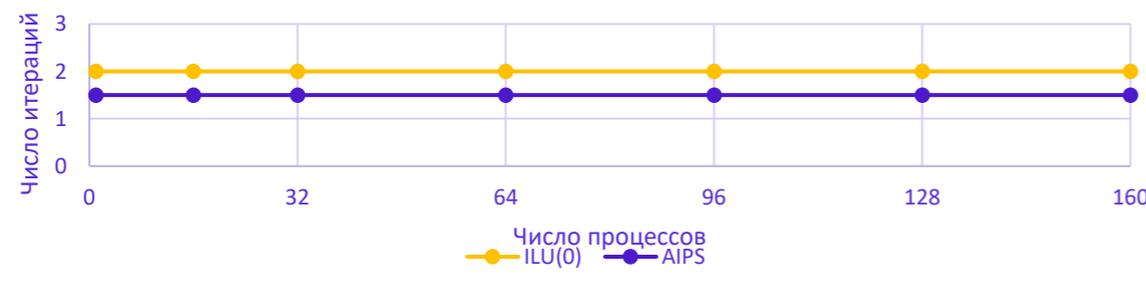
immn



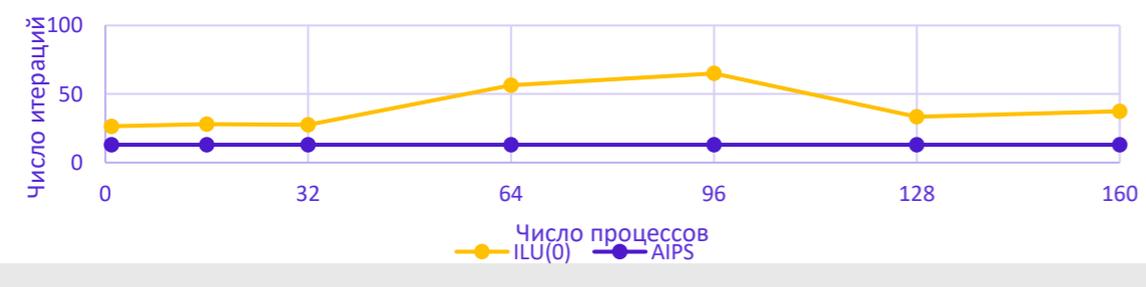
mmnt



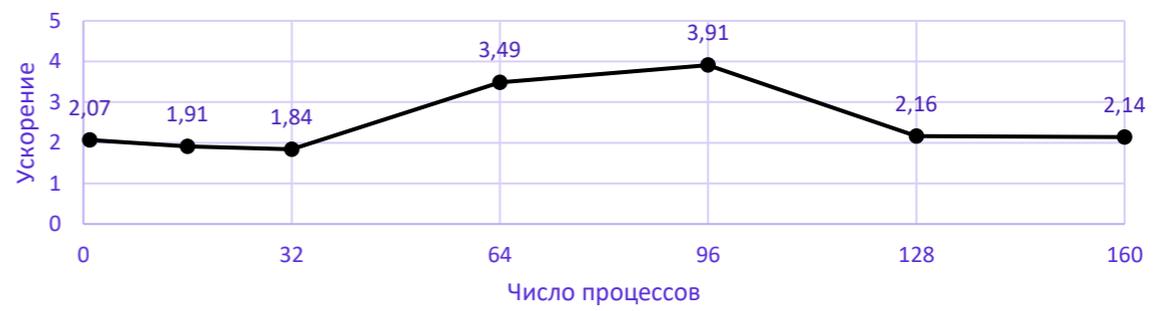
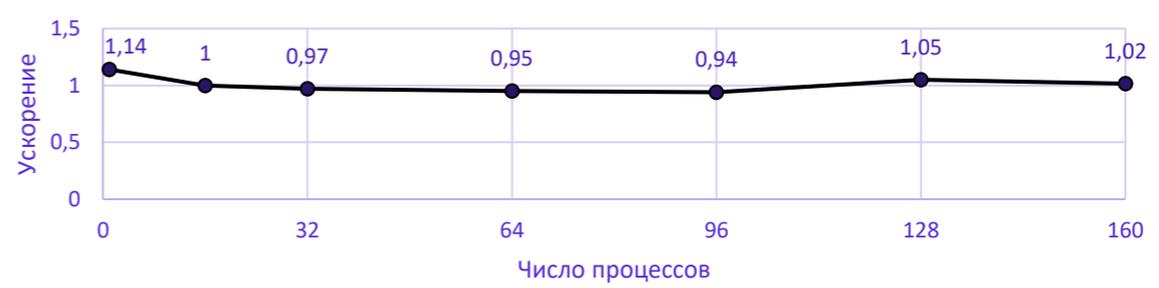
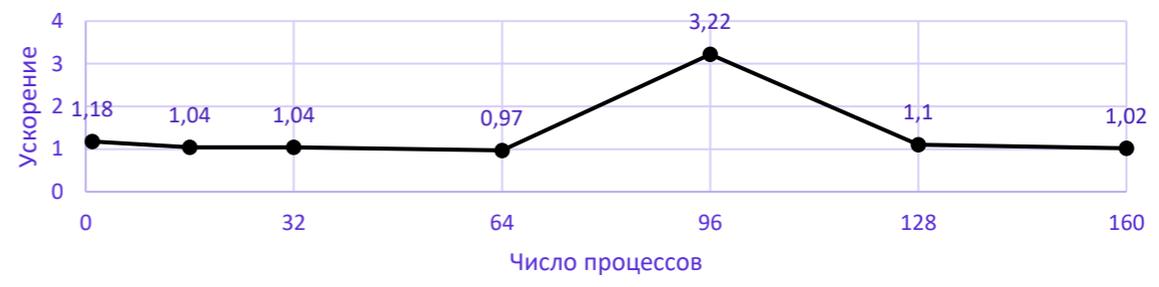
fdrv



kmms



Ускорение расчета с AIPS относительно расчета с ILU(0)



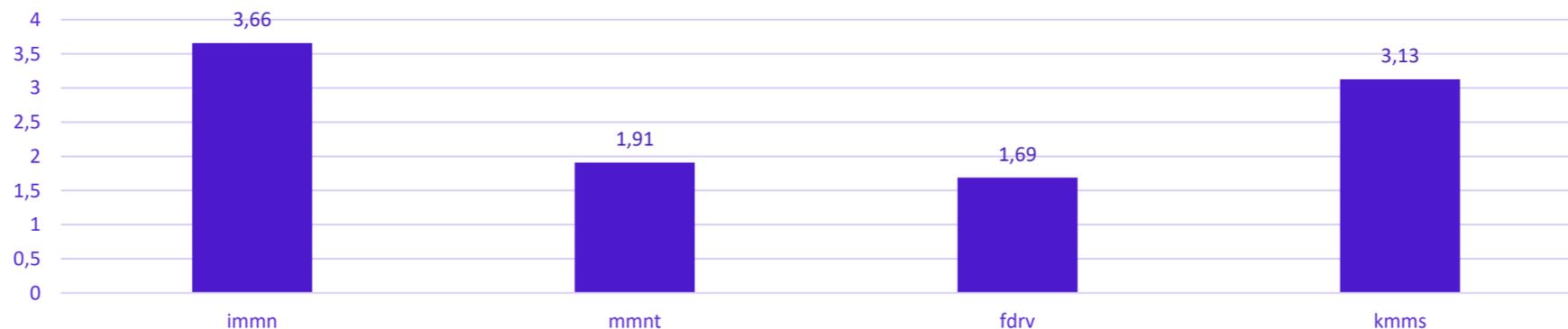
Среднее ускорение при использовании AIPS относительно ILU(0) – 1,59 раза

Сравнение эффективности ILU(0) и AIPS на GPU A100



Название матрицы	ILU(0)			AIPS		
	Кол-во итераций	Время расчета, мс	Сред. время 1 итерации, мс	Кол-во итераций	Время расчета, мс	Сред. время 1 итерации, мс
immn	15,5	232,1	14,97	10,5	63,36	6,03
mmnt	6	157,55	26,26	5	82,58	16,52
fdrv	2	85,95	42,98	1,5	50,75	33,83
kmms	26,5	829,75	31,31	13	265,26	20,40

Ускорение расчета с AIPS относительно расчета с ILU(0) на GPU



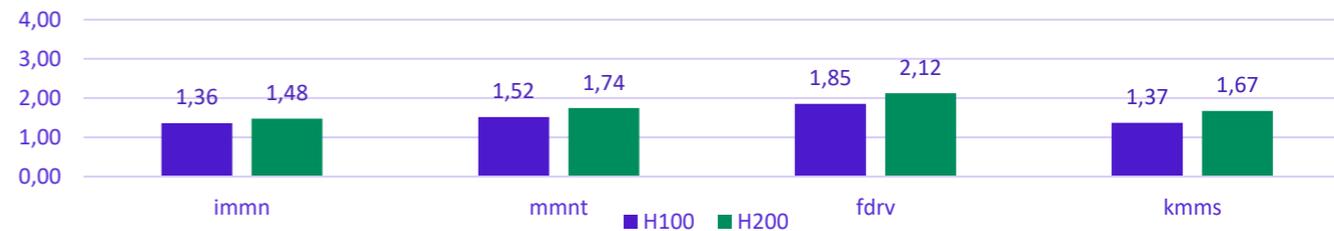
Среднее ускорение при использовании AIPS относительно ILU(0) – 2,6 раза

Сравнение эффективности ILU(0) и AIPS на GPU A100, H100 и H200

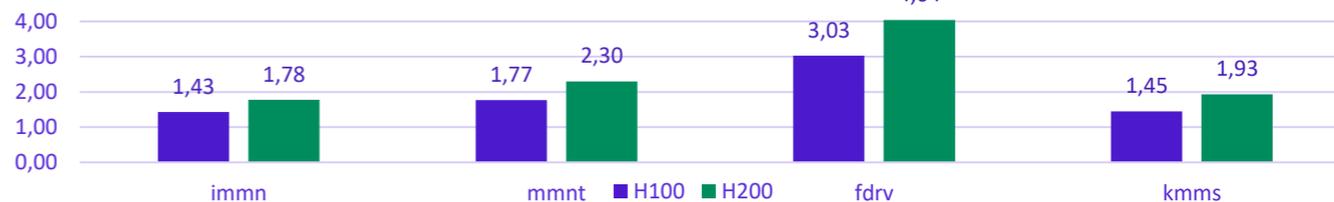


Название матрицы	GPU	ILU(0)	AIPS	Ускорение AIPS относительно ILU(0)
		Время расчета, мс	Время расчета, мс	
immn	A100	232,1	63,36	3,66
	H100	170,58	44,19	3,86
	H200	157,14	35,62	4,41
mmnt	A100	157,55	82,58	1,91
	H100	103,76	46,69	2,22
	H200	90,36	35,87	2,52
fdrv	A100	85,95	50,75	1,69
	H100	46,4	16,76	2,77
	H200	40,5	12,55	3,23
kmms	A100	829,75	265,26	3,13
	H100	603,82	183,09	3,30
	H200	496,79	137,22	3,62

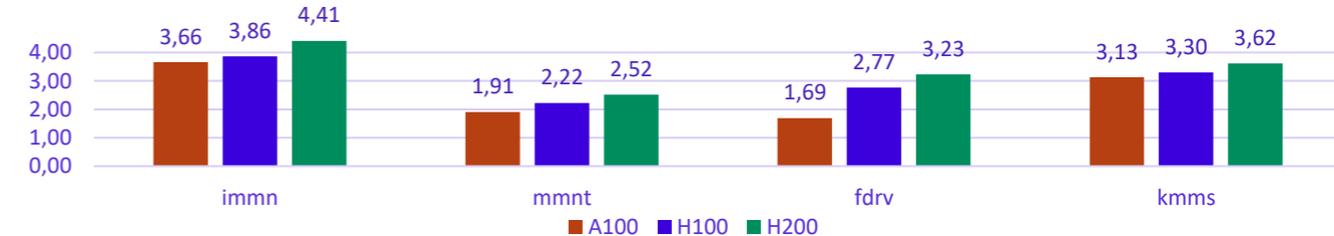
Ускорение расчетов с ILU(0) на различных GPU относительно A100



Ускорение расчетов с AIPS на различных GPU относительно A100



Ускорение расчетов с AIPS относительно расчета с ILU(0)



Среднее ускорение расчетов с ILU(0) относительно A100 – 1,53 раза на H100 и 1,75 раза на H200.

Среднее ускорение расчетов с AIPS относительно A100 – 1,92 раза на H100 и 2,51 раза на H200.

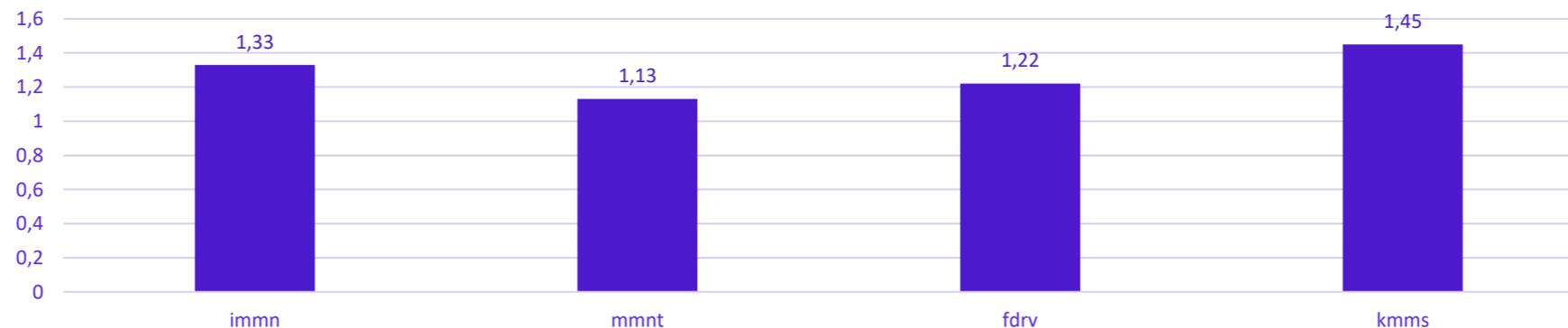
Среднее ускорение при использовании AIPS относительно ILU(0) – 3,04 раза на H100 и 3,45 раза на H200

Сравнение эффективности ILU(0) и AIPS в составе CPR на GPU A100



Название матрицы	CPR-AMG-ILU(0)			CPR-AMG-AIPS		
	Кол-во итераций	Время расчета, мс	Сред. время 1 итерации, мс	Кол-во итераций	Время расчета, мс	Сред. время 1 итерации, мс
immn	2,5	99,21	39,68	2,5	74,51	29,80
mmnt	2	147,74	73,87	1,5	131,05	87,37
fdrv	1	135,04	135,04	1	110,8	110,8
kmms	3,5	288,66	82,47	2	199,17	99,59

Ускорение расчета с CPR-AMG-AIPS относительно расчета с CPR-AMG-ILU(0)



Среднее ускорение при использовании CPR-AMG-AIPS относительно CPR-AMG-ILU(0) – 1,28 раза



1. Предложена модификация алгоритмов предобусловливателя AIPS, позволяющая учесть мелкоблочную структуру матриц, характерную для задачи моделирования многофазной фильтрации углеводородов в пористой среде.
2. Использование предобусловливателя AIPS на CPU обеспечивает стабильную скорость сходимости итерационного метода вне зависимости от числа процессов, снижая среднее число итераций в 1,99 раза по сравнению с ILU(0) и обеспечивая ускорение в 1,59 раза.
3. Оптимизированная для графических процессоров модификация предобусловливателя AIPS позволяет ускорить решение СЛАУ относительно ILU(0) в среднем в 2,6 раза на GPU A100. При переходе на более современные GPU среднее ускорение возрастает: до 3,04 раза на H100 и 3,44 раза на H200.
4. Проведено сравнение эффективности AIPS и ILU(0) в качестве второй ступени предобусловливателя CPR. Получено, что среднее ускорение CPR-AMG-AIPS относительно CPR-AMG-ILU(0) составляет 1,28 раза на GPU A100.



Всероссийская научная конференция с международным участием

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ ТЕХНОЛОГИИ 2026

<https://ПaBT.PФ>



Уфа, 7–9 апреля 2026 г.

Уфимский университет науки и технологий



СПАСИБО ЗА ВНИМАНИЕ!