

ПАРАЛЛЕЛЬНЫЕ ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СЛАУ В ЗАДАЧАХ ФИЛЬТРАЦИИ

Игорь Н. Коньшин
ВЦ, ИВМ, ИБРАЭ РАН

Абрау-Дюрсо

17.09.2013

План

- Методы решения линейных систем
- Последовательные методы (или решение подзадач)
 - Треугольные факторизации
 - Многосеточные методы
 - Многоуровневые методы
- Параллельные методы решения
 - Блочный метод Якоби
 - Аддитивный метод Шварца
 - ВИЛУ
- Комбинации решателей
- *Численные эксперименты*

СЛАУ и их решатели

Типы линейных систем $Ax = b$:

- Плотные и *разреженные*
- Структурированные и *неструктурированные*
- Симметричные и *несимметричные*
- Комплексные и *действительные*
 - *Размерность СЛАУ*
 - *Сложность решения СЛАУ*

Типы решателей линейных систем:

- Прямые и *итерационные*
- Последовательные и *параллельные*
- *In-core* и out-of-core

Последовательные методы решения СЛАУ

Треугольные факторизации

$$A x = b$$

- Точные LU-факторизации (разложения): $A = L U$
- Неполные (и приближенные) факторизации: $A \approx L U$

Искусство построения неполных факторизаций → фильтрация элементов по структуре или по значению...

- ILU – **неполнота** при фильтрации по **структуре**
- ILU – неточность (**приближенность**) факторизации при фильтрации по **значениям** элементов

Неполная $ILU(k)$ факторизация

- $ILU0$ – факторизация в структуру исходной матрицы A
- $ILU(k)$ – расширение структуры k раз (структура матрицы A^{k+1} внутри профиля A)
- *Как выбрать k для заданного качества разложения?*
- *Как зависит скорость сходимости от k ?*
- *Как зависит скорость сходимости от размерности?*
- *Затраты памяти на хранение A и LU (объем памяти)*
- *Затраты арифметические на вычисление LU (время)*
- Хотелось бы: $M(L+U) \sim M(A)$, а заодно $AC(LU) \sim c \cdot M(A)$

Приближенная $ILU(\tau)$ факторизация

- Фильтрация элементов по значению (оставляем элементы $|a_{ij}| > \tau \geq 0$)
- Точность **приближенной** факторизации $ILU(\tau)$ для разложения $A=LU+E$ составляет $O(\tau)$, т.е. это метод первого порядка точности по τ , т.к. $|e_{ij}| < \tau$
- Точность **неполной** факторизации $ILU(k)$ составляет $O(1)$, т.е. это метод нулевого порядка точности
- *Объем требуемой памяти для хранения LU заранее непредсказуем*
- *Способы повышения устойчивости и робастности...*

ILU2(τ) факторизация (И.Е.Капорин'94)

- Фильтрация элементов по значению (оставляем элементы $|a_{ij}| > \tau \geq 0$)

- В вычислениях участвуют элементы $|a_{ij}| > \tau^2$

- Точность разложения

$$A=(L+E_1)(U+E_2)=(LU+LE_2+UE_1)+E_1 \quad E_2=L'U'+E$$

составляет $O(\tau^2)$, т.е. это метод **второго порядка точности** по τ , т.к. $|e_{ij}| < \tau^2$

- Точность разложения $ILU2(\tau)$ соответствует $ILU(\tau^2)$, а память для хранения сомножителей – разложению $ILU(\tau)$

- Не путать: $ILU2(\tau) = ILU2 \neq ILU(2) = ILU(k=2)$

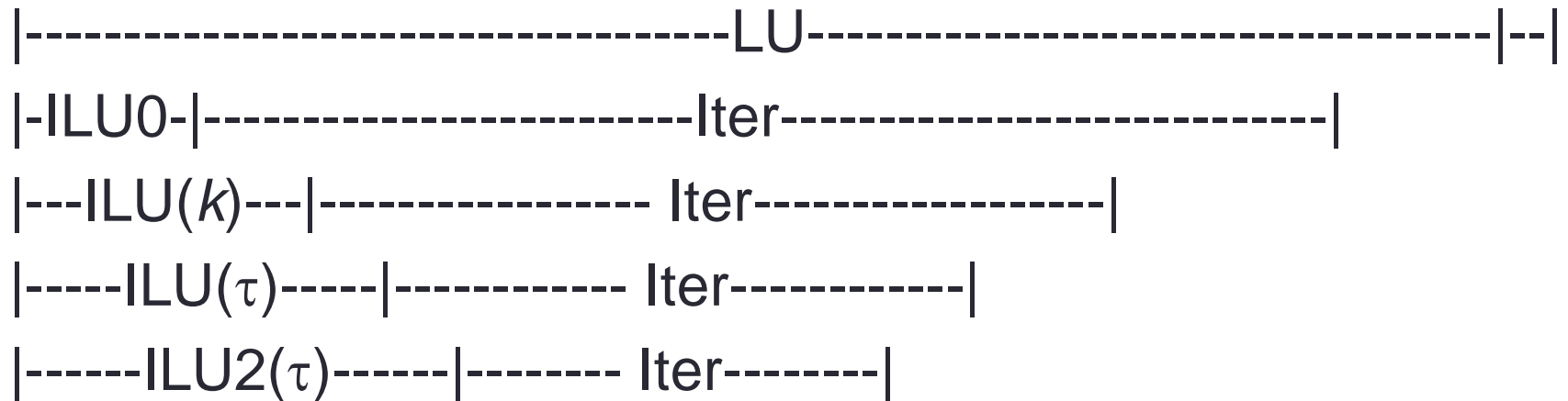
MLILU (И.Е.Капорин'09)

- MLILU – многоуровневое разложение – усовершенствованный аналог выбора ведущего элемента по всей матрице (самого точного варианта треугольного разложения)
- Двустороннее итерационное масштабирование для выравнивания норм A_L и A_U
- Выбор ведущей подматрицы с диагональным преобладанием
- Факторизация ведущего блока и вычисление дополнения по Шуру
- Рекурсивное применение этой же процедуры к дополнению по Шуру...

Точность факторизаций

Факторизация	Точность	Назначение
ILU0	$O(1)$	Простые линейные системы
ILU(k)	$O(1)$	Умеренно сложные
ILU(τ)	$O(\tau)$	Посложнее
ILU2(τ)	$O(\tau^2)$	Сложные и очень сложные
LU	$O(\delta)$	До очень сложных, но небольшого размера
MLILU	Выше чем $O(\tau)$	Самые сложные, включая нули на главной диагонали
MG	хорошая	Не самые сложные, и желательно для эллиптических задач

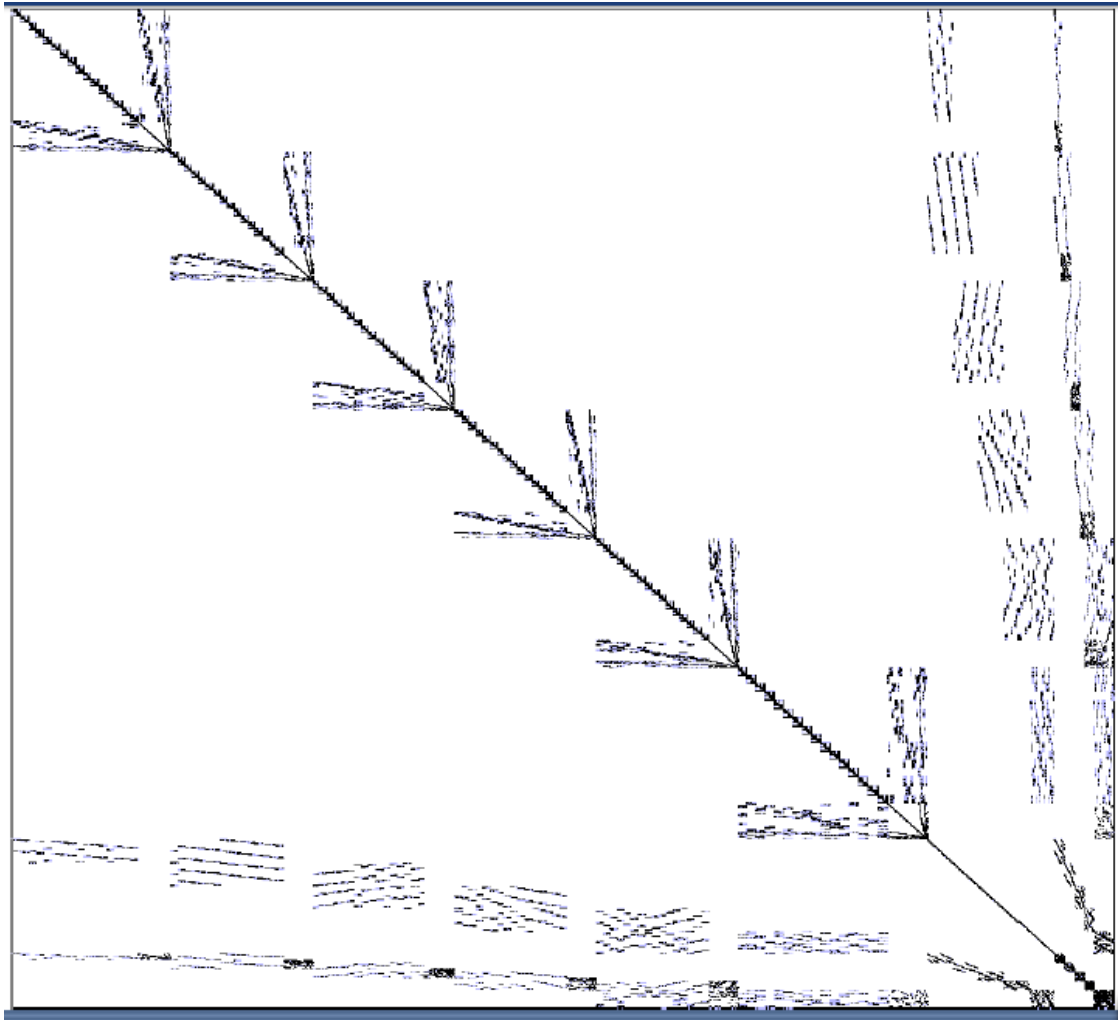
Затраты на факторизацию и итерации



Характерное время вычисления и применения различных типов факторизации

Параллельные методы решения СЛАУ

Явное распараллеливание с помощью иерархического упорядочивания



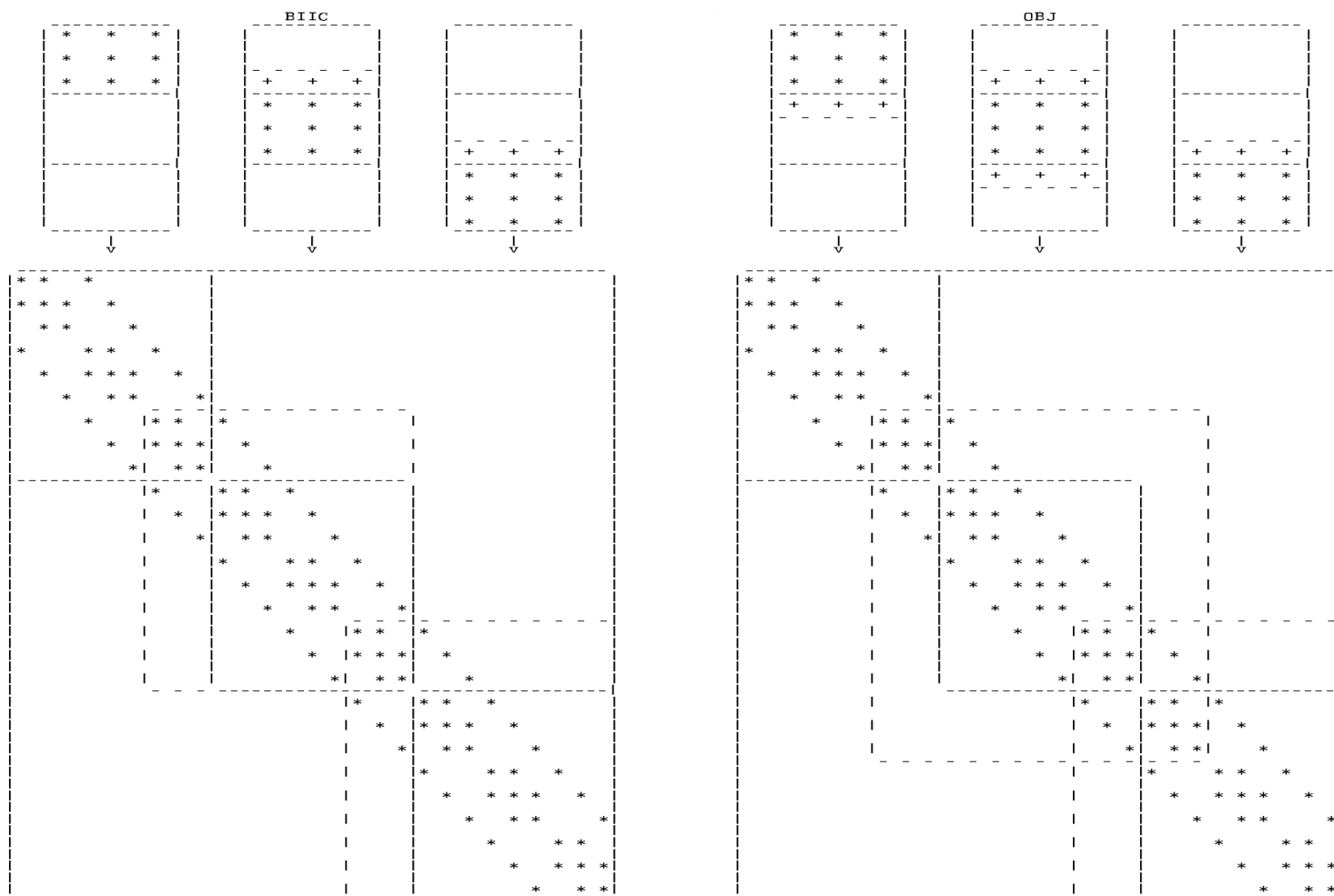
VJ

- VJ – блочный метод Якоби (вычисление и применение переобуславливателя на разных процессорах полностью независимы)
- (+) «идеальный» переобуславливатель с точки зрения параллельности (не требует обменов)
- (-) возможно резкое увеличение количества итераций по сравнению с последовательной (одноблочной) версией
- *Чтобы сделать блоки независимыми, лишние связи, которые связывают блоки, **просто** отбрасываются*
- *VJ – это самый «**простой**» параллельный переобуславливатель*

ASM

- ASM – аддитивный метод Шварца с перекрытием подобластей (блоков) в q слоев (вычисление переобуславливателя на разных процессорах независимы, но применение его требует небольших пересылок)
- (+) малые обмены при умеренных значениях q (хорошая параллельность)
- (-) возможно некоторое увеличение количества итераций, но гораздо мягче чем для BJ
- $ASM(q=0) \equiv BJ$
- *ASM реализован в общедоступных пакетах PETSc и Trilinos*

Разбиение области и матрицы



BIILU (И.Е.Капорин'90,94)

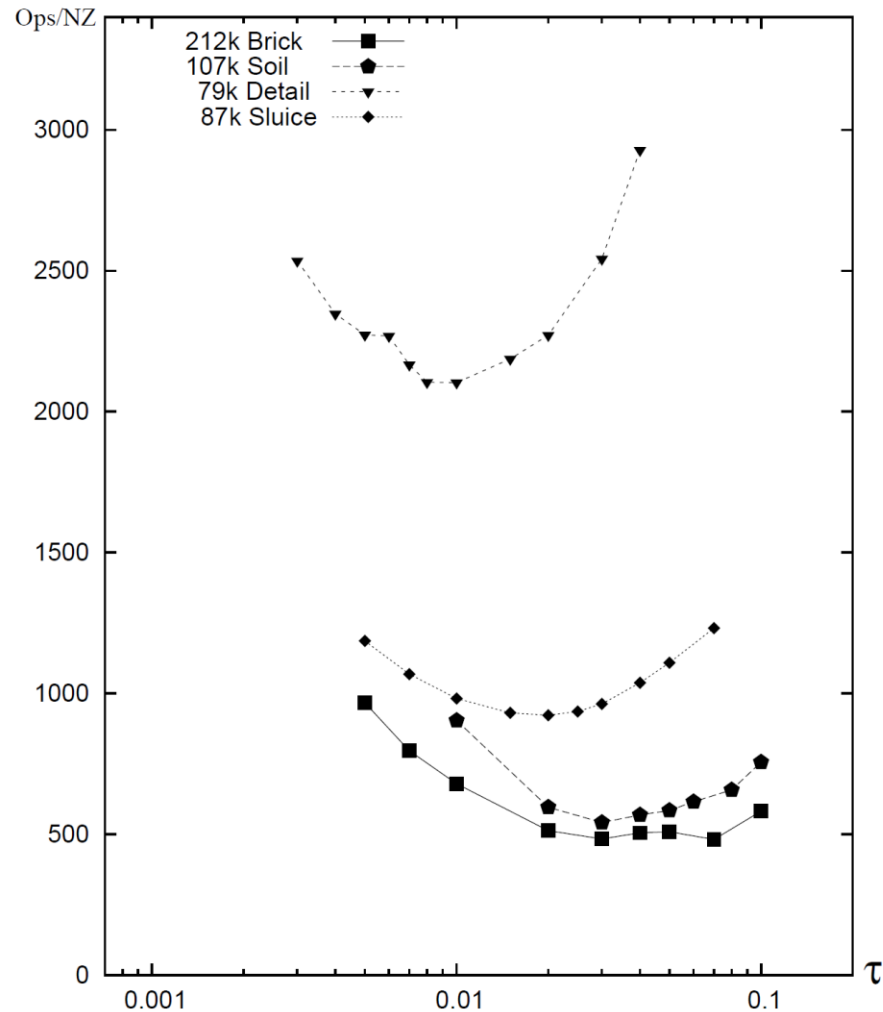
- BIILU – также аддитивный переобуславливатель, разработанный специально для треугольного разложения (улучшенный аналог метода Шварца ASM) также с перекрытием подобластей в q слоев (но перекрытия «односторонние» только вверх матрицы)
- (+) Количество итераций меньше чем у ASM
- (+) И арифметические и коммуникационные затраты меньше чем у ASM, т.к. само окаймление меньше

- $BIILU(q=0) \equiv BJ$
- *BIILU нет ни в PETSc ни в Trilinos*
- *Возможно применение BIILU для «внешней» схемы распараллеливания в виде комбинации методов:
 $BIILU2=BIILU+ILU2$*

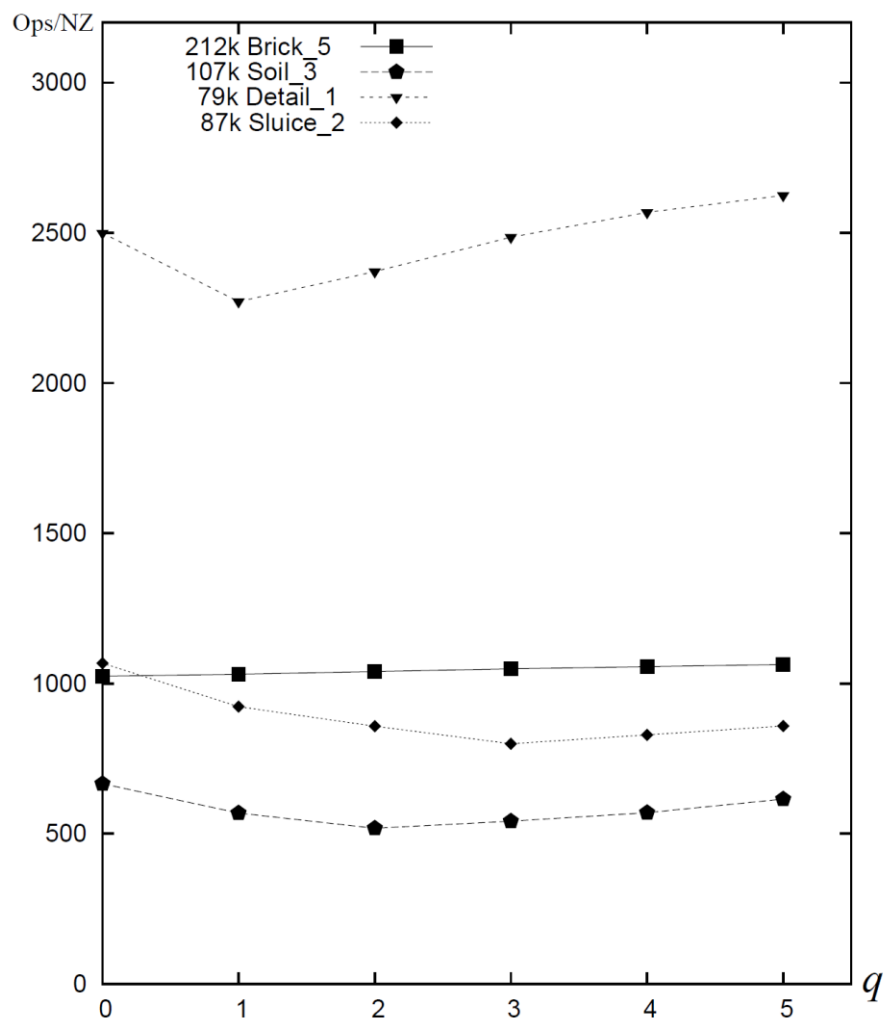
Численные эксперименты

для некоторых комбинаций решателей

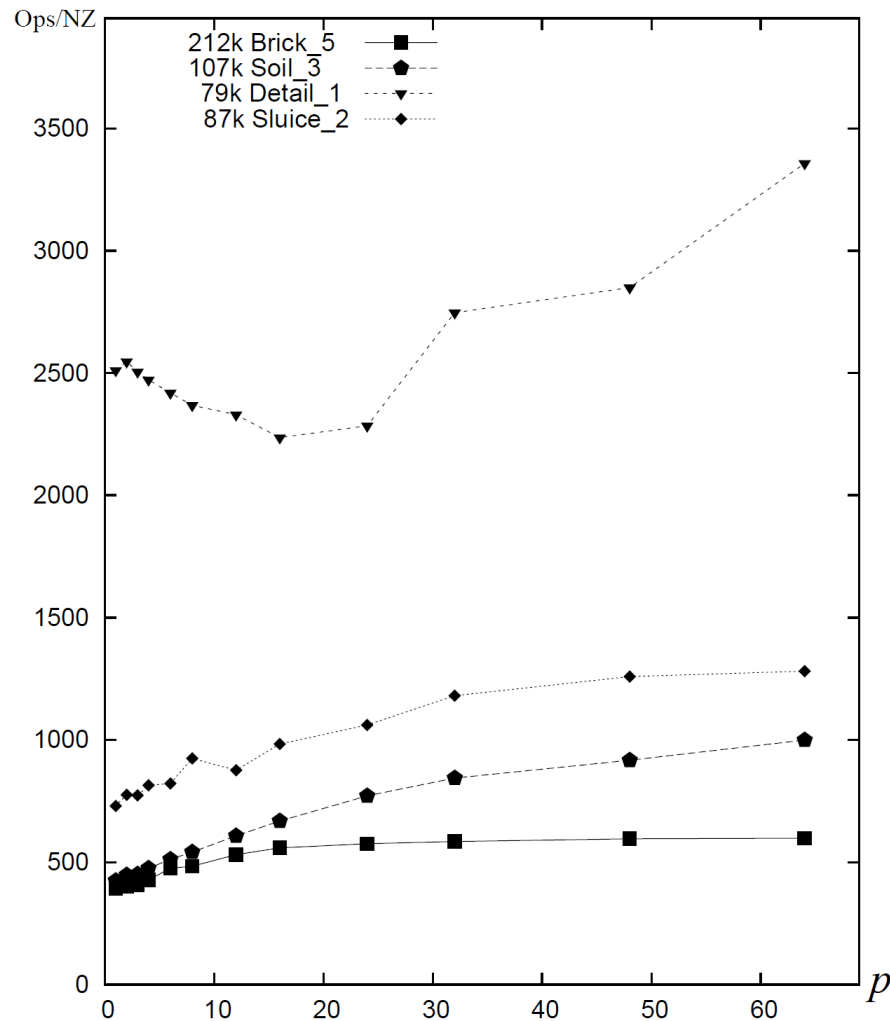
Зависимость арифметических затрат AC(VIIC2) от порога фильтрации τ



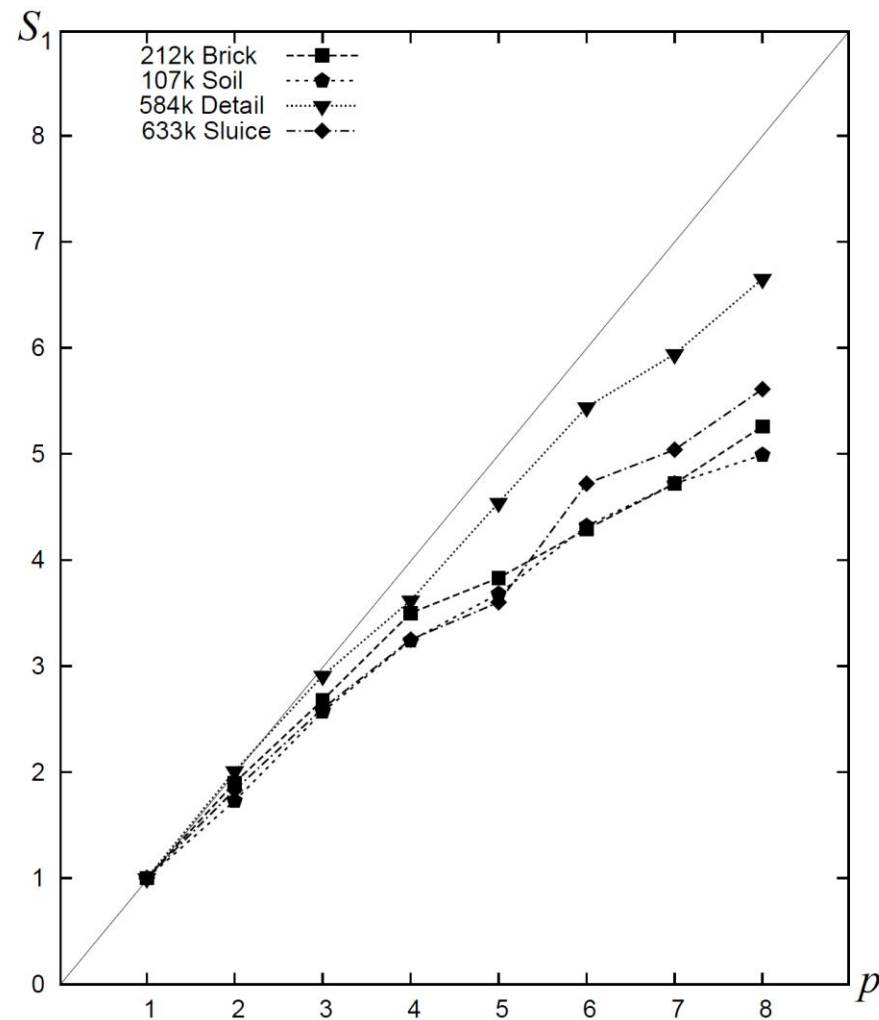
Зависимость АС(ВІІС2) от параметра перекрытия q



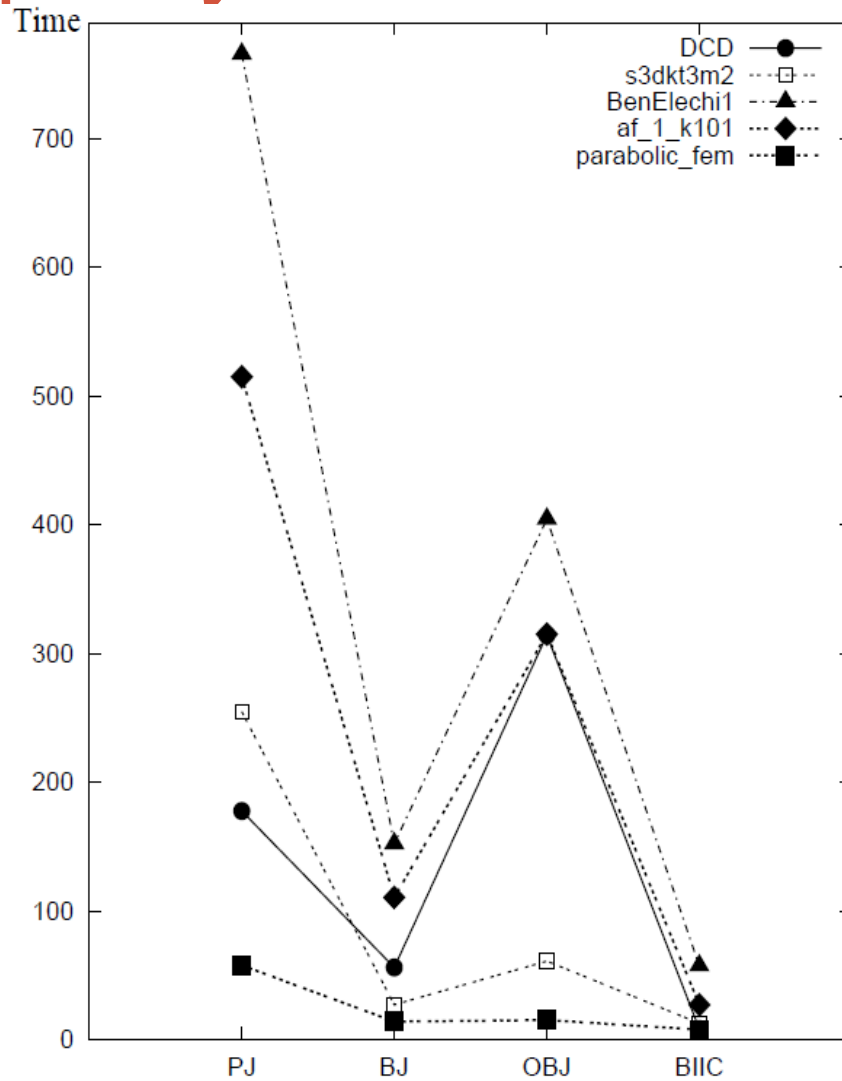
Зависимость АС(ВІІС2) от количества блоков p



Зависимость ускорения от числа процессоров p



Сравнение параллельных переобуславливателей



PJ (Point Jacobi)
BJ (Block Jacobi)
OBJ = ASM
BIIC

+IC2 as subdomain solver
+CG iterations
8 PEs

Решение СЛАУ для задач фильтрации

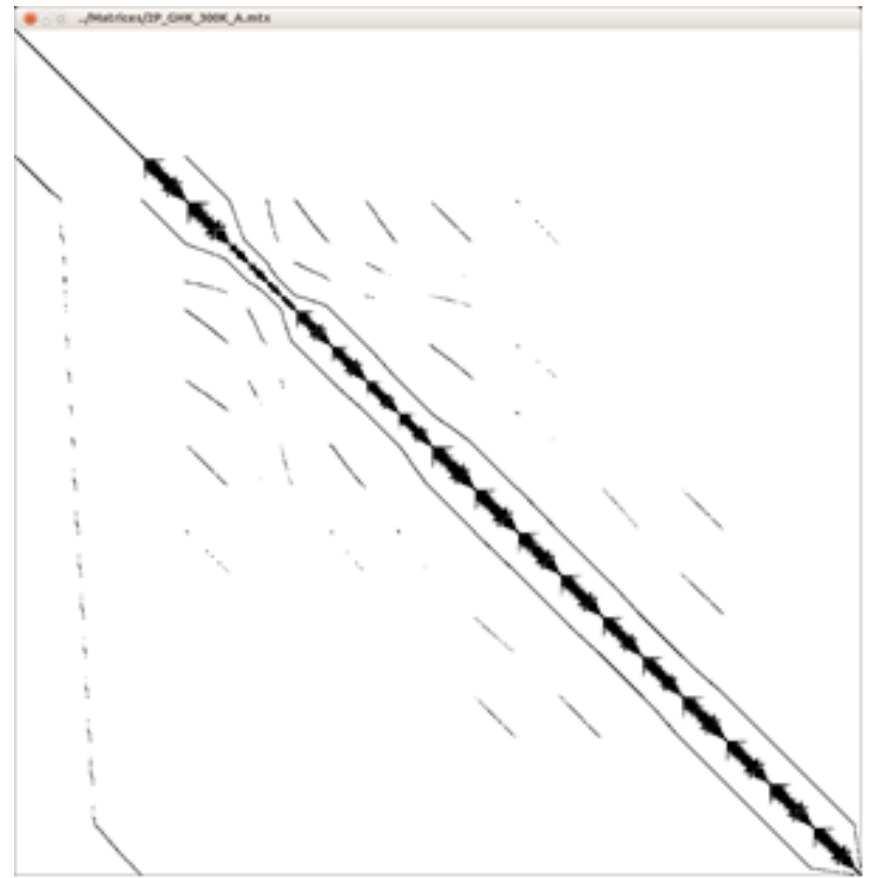
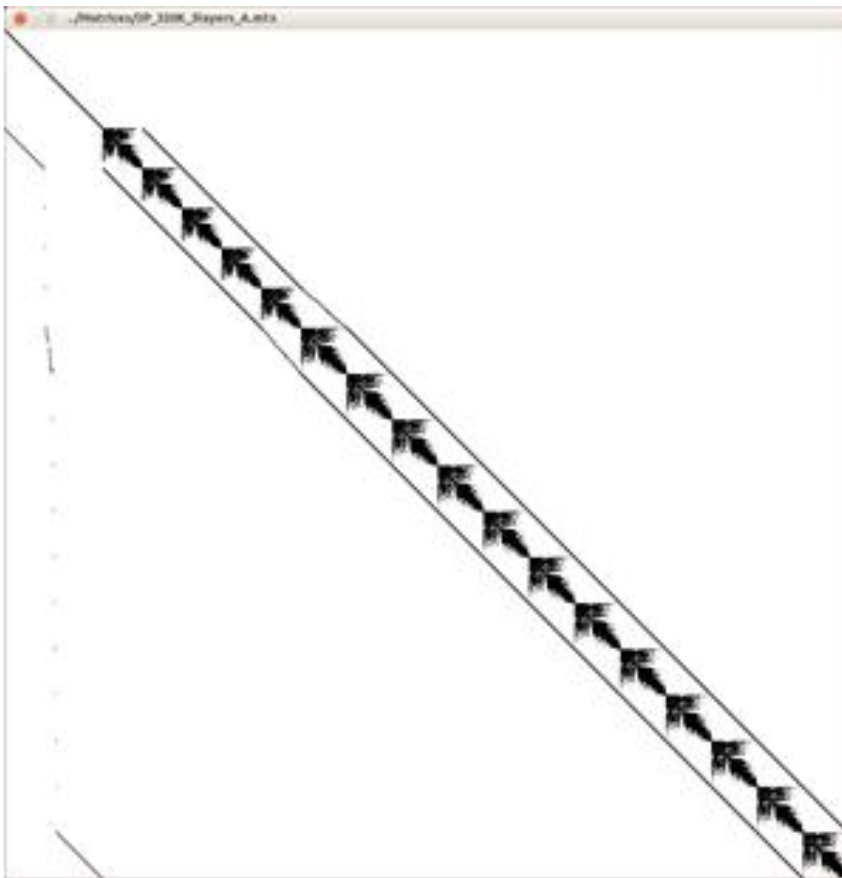
Дискретизация задачи фильтрации

- Двухточечная схема (+И.Капырин)
- Монотонная схема (+К.Никитин)
- О-схема (+И.Капырин)

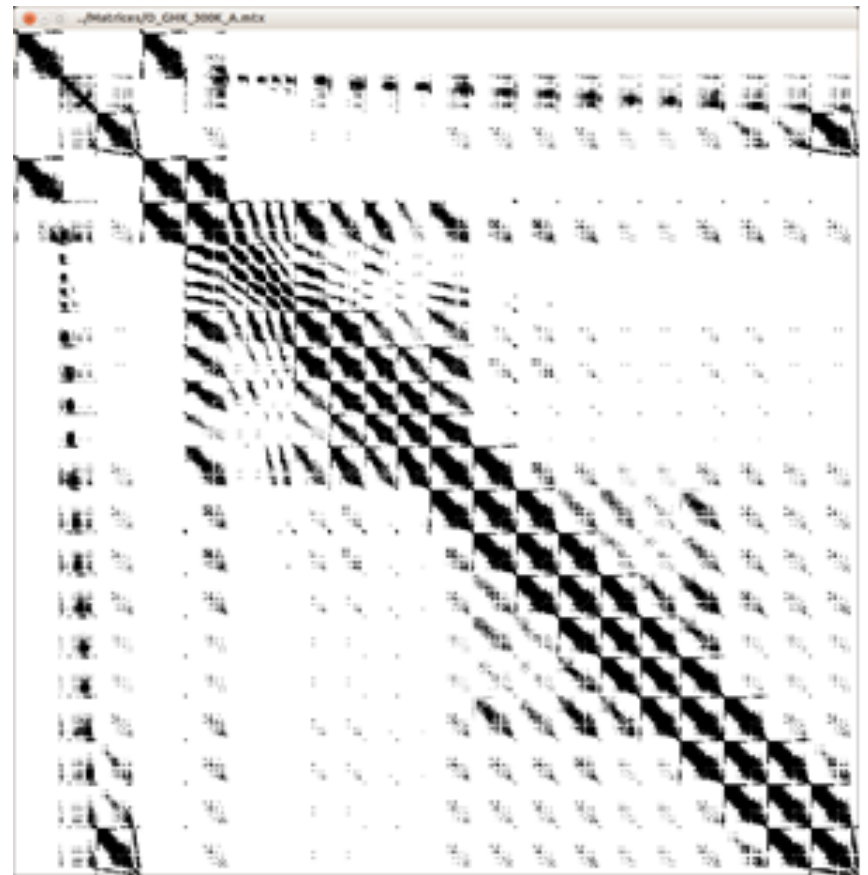
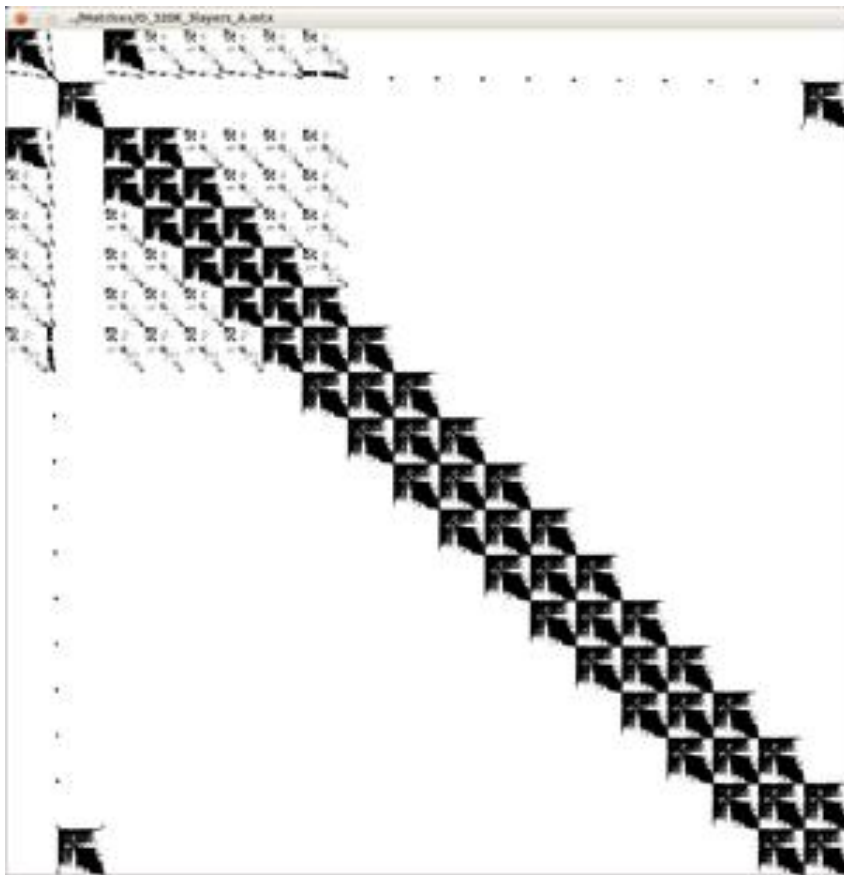
Решение СЛАУ

- методы PETSc (+К.Терехов)

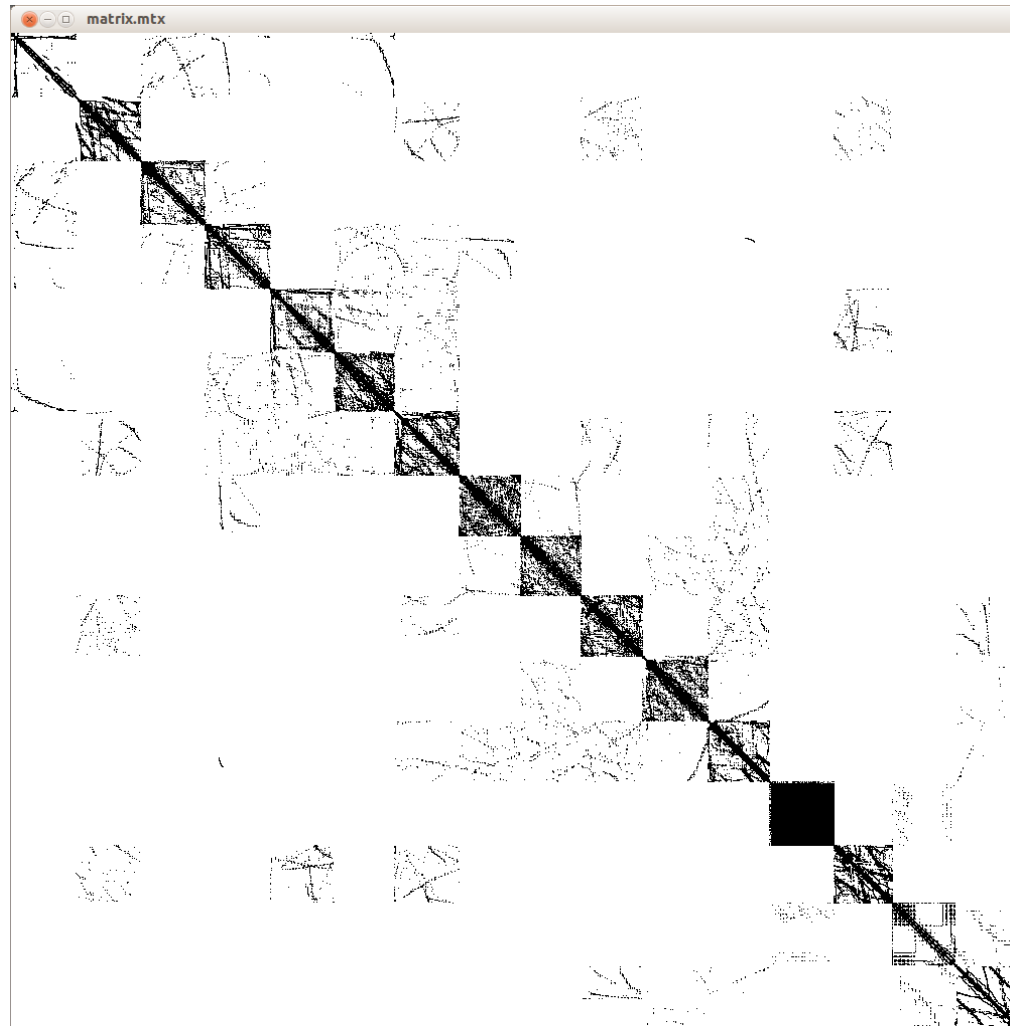
Структура матриц A для двух задач при двухточечной схеме дискретизации



Структура матриц A для O-схемы дискретизации



Структура матрицы A после упорядочивания ParMetis ($p=16$)



Решение СЛАУ для нелинейной схемы дискретизации

Матрица	Метод	ρ	Кол-во итер.	Время, сек.
A-002	asm1-ilu0	1	365	28.7209
		2	639	31.890
		4	620	24.4349
		8	1026	42.794
		16	916	29.5863
		32	1202	24.6534
		64	1388	16.7267
A-002	asm2-ilu2	1	146	15.2629
		2	212	15.8048
		4	215	12.9985
		8	285	21.1654
		16	260	18.4594
		32	303	15.7822
		64	369	12.551

Методы решения СЛАУ для $p=8;64$

Матрица	Метод	p	Кол-во итер.	Время, сек.
A-002	asm1_ilu0	8	1026	42.794
	asm2_ilu1	8	484	35.9021
	asm2-ilu2	8	285	21.1654
	boomeramg	8	138	25.3125
	euclid_lev2	8	168	32.5063
	gamg	8	154	26.5907
	mumps	8	1	10.8936
A-002	asm1_ilu0	64	1388	16.7267
	asm2_ilu1	64	409	12.0731
	asm2-ilu2	64	369	12.551
	boomeramg	64	105	45.8315
	euclid_lev2	64	592	17.1614
	gamg	64	99	7.34684
	mumps	64	1	10.1299

Методы решения СЛАУ для $p=8;64$

Матрица	Метод	p	Кол-во итер.	Время, сек.
2P_320K_3layers	asm1_ilu0	8	217	9.71741
	asm2_ilu1	8	166	12.2034
	asm2-ilu2	8	136	12.3795
	boomeramg	8	7	12.3033
	euclid_lev2	8	130	27.3097
	gamg	8	27	7.30627
	mumps	8	1	18.7316
2P_320K_3layers	asm1_ilu0	64	335	3.36935
	asm2_ilu1	64	216	5.37287
	asm2-ilu2	64	177	5.16936
	boomeramg	64	8	16.1341
	euclid_lev2	64	172	6.84347
	gamg	64	29	3.76336
	mumps	64	1	8.92752

Решение СЛАУ методами ASM-ILU для $p=64$

Матрица	Метод	p	Кол-во итер.	Время, сек.
A-002	asm1_ilu0	64	1388	16.7267
A-010	asm1_ilu0	64	1803	21.6618
A-050	asm1_ilu0	64	2096	25.3305
2P_320K_3layers	asm1_ilu0	64	335	3.36935
2P_G_300K	asm1_ilu0	64	1784	20.5135
A-002	asm2_ilu1	64	409	12.0731
A-010	asm2_ilu1	64	532	15.5482
A-050	asm2_ilu1	64	446	13.1291
2P_320K_3layers	asm2_ilu1	64	216	5.37287
2P_G_300K	asm2_ilu1	64	467	14.6743
O_320K_3layers	asm2_ilu1	64	253	47.308
A-002	asm2_ilu2	64	369	12.551
A-010	asm2_ilu2	64	466	15.7976
A-050	asm2_ilu2	64	537	18.0852
2P_320K_3layers	asm2_ilu2	64	177	5.16936
2P_G_300K	asm2_ilu2	64	435	16.8921
O_320K_3layers	asm2_ilu2	64	137	46.4674

Решение СЛАУ прямым методом

Матрица	Метод	ρ	Кол-во итер.	Время, сек.
A-002	mumps	1	1	28.192
	mumps	8	1	10.8936
	mumps	64	1	10.1299
A-010	mumps	1	1	26.4306
	mumps	8	1	11.4453
	mumps	64	1	10.306
A-050	mumps	1	1	26.4207
	mumps	8	1	11.168
	mumps	64	1	10.3162
2P_320K_3layers	mumps	1	1	69.0514
	mumps	8	1	18.7316
	mumps	64	1	8.92752
O_320K_3layers	mumps	1	1	1169.33
	mumps	8	1	327.999
	mumps	64	1	36.8708
2P_G_300K	mumps	1	1	28.3546
	mumps	8	1	11.0181
	mumps	64	1	7.57703
O_G_300K	mumps	1	1	517.444
	mumps	8	1	142.096
	mumps	64	1	25.4494

Решение СЛАУ многосеточными методами

Матрица	Метод	p	Кол-во итер.	Время, сек.
A-002	boomeramg	64	105	45.8315
A-010	boomeramg	64	10	9.26362
A-050	boomeramg	64	8	8.57821
2P_320K_3layers	boomeramg	64	8	16.1341
2P_G_300K	boomeramg	64	8	9.57112
A-002	euclid_lev2	64	592	17.1614
A-010	euclid_lev2	64	2536	54.0938
A-050	euclid_lev2	64	1752	37.6875
2P_320K_3layers	euclid_lev2	64	172	6.84347
2P_G_300K	euclid_lev2	64	1284	21.9181
O_320K_3layers	euclid_lev2	64	156	1085.24
A-002	gamg	64	99	7.34684
A-010	gamg	64	117	8.5614
A-050	gamg	64	121	8.12799
2P_320K_3layers	gamg	64	29	3.76336

Литература

- [1] I.E.Kaporin. High quality preconditioning of a general symmetric positive definite matrix based on its $U^T U + U^T R + R^T U$ decomposition. Numer. Linear Algebra Appl., 5 (1998) 483-509
- [2] I.Kaporin, I.Konshin. A parallel block overlap preconditioning with inexact submatrix inversion for linear elasticity problems. Numer. Linear Algebra Appl., 9 (2002) 141-162
- [3] I.Kaporin, I.Konshin. Postfiltering of triangular factors in IC2-factorization for balancing of parallel preconditioners. Comp. Math. Math. Phys. 49 (2009) 940-957

Appendix

Решение СЛАУ для двухточечной схемы дискретизации

Матрица	Метод	ρ	Кол-во итер.	Время, сек.
2P_320K_3layers	asm1-ilu0	1	200	14.2605
		2	261	11.2265
		4	277	10.0593
		8	217	9.71741
		16	201	6.34487
		32	263	4.41591
		64	335	3.36935
2P_320K_3layers	asm2-ilu2	1	42	54.4459
		2	81	79.8464
		4	93	56.7688
		8	101	76.0497
		16	106	67.9825
		32	132	62.9921
		64	137	46.4674

Structural mechanics applications

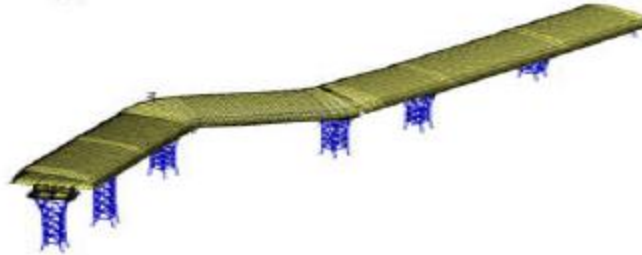
#	n	b	DShift	T_1	T_2	Time, c.
1	40 020	3	1e-6	1e-3	1e-6	0.42
2	88 782	3	1e-5	3e-3	1e-5	3.65
3	98 650	2	1e-6	1e-3	1e-6	8.59
4	112 560	3	1e-6	1e-3	1e-6	4.10
5	195 024	3	1e-6	1e-3	1e-6	10.4
6	247 542	3	1e-5	3e-3	1e-5	17.0
7	256 572	3	1e-5	3e-3	1e-5	16.8
8	397 950	3	1e-4	1e-2	1e-4	53.6
9	427 014	3	1e-6	1e-3	1e-6	41.6
10	755 592	3	1e-7	3e-4	1e-7	101.
11	798 230	2	3e-4	3e-3	1e-5	242.
12	889 890	3	1e-5	3e-3	1e-5	72.
13	1 623 600	3	1e-10	1e-5	1e-10	113.
14	2 534 446	2	1e-6	1e-3	1e-6	431.
15	3 247 200	3	1e-10	1e-5	1e-10	351.
16	4 870 800	3	1e-12	1e-6	1e-12	823. (*)
17	8 033 520	3	1e-8	1e-6	1e-10	766. (*)

Problems origin

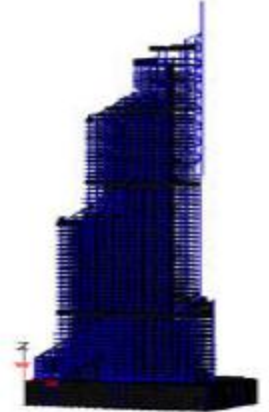
модель 1



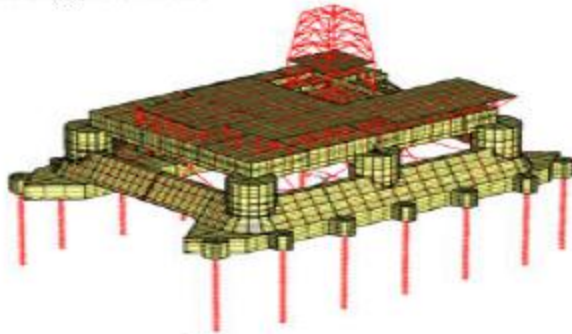
модель 4



модель 10



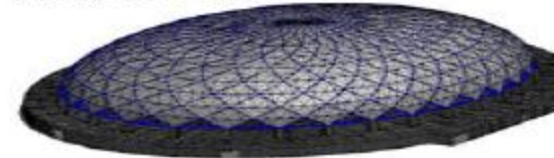
модель 3



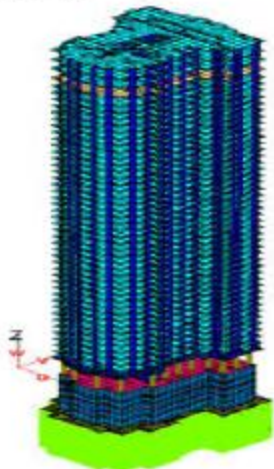
модель 14



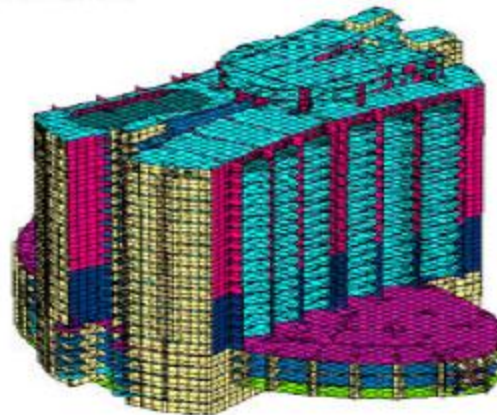
модель 12



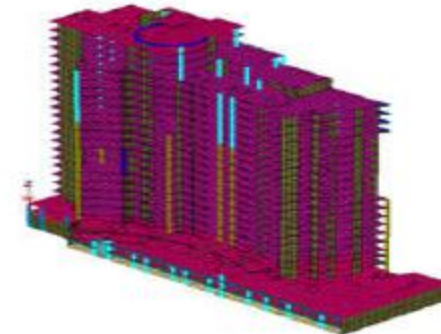
модель 9



модель 7



модель 5



Condition numbers of scaled matrix

#	N	Cond
9	427 014	3×10^9
13	1 623 600	2×10^{10}
15	3 247 200	8×10^{11}
16	4 870 800	4×10^{12}
17	8 033 520	3×10^{13}

Sample histogram of eigenvalues:

e.v.:	10^{-8}	10^{-7}	10^{-6}	...	1
a number of e.v.:	10^1	10^2	10^3	...	10^6

Block size b for Problem 13

b	$\frac{\text{nzA}(b)}{\text{nzA}(1)}$	$\frac{\text{nzU}(b)}{\text{nzAU}(b)}$	# Iter	Time (Prec), c.	Time (Iter), c.	Time, c.
1	1.00	7.40	58	393.3	146.0	539.3
2	1.60	7.47	55	83.8	81.9	165.7
>3<	2.05	7.28	60	46.0	77.0	>123.0<
4	2.62	7.24	61	51.3	84.7	136.0
5	3.20	7.05	60	58.6	79.6	138.2
6	2.16	7.46	65	60.1	96.1	156.2
8	3.87	6.37	54	119.3	96.2	215.5
10	4.42	6.25	55	129.7	107.0	236.7

Solution time for MPI+threads

Prec+Iter (#Iter)

#	1 (1x1)	2 (1x2)	4 (1x4)	8 (2x4)	16 (4x4)	32 (8x4)
1	0.45+0.73 (38)	0.23+0.83 (66)	0.13+0.51 (66)	0.08+0.97 (187)	-	-
2	2.5+4.0 (62)	1.2+5.1 (126)	0.7+3.3 (126)	0.3+2.9 (197)	-	-
3	7.9+11.5 (107)	3.8+15.3 (248)	2.0+10.0 (248)	-	-	-
4	3.1+7.5 (90)	1.6+6.3 (123)	0.8+4.3 (124)	0.5+4.0 (209)	-	-
5	11+8 (48)	6+15 (150)	3+10 (150)	2+12 (243)	1+9(363)	0.4+7.8(47)
6	13+17 (88)	8+13 (117)	5+9 (117)	2+9 (200)	-	-
7	20+16 (82)	10+11 (96)	6+7 (96)	3+6 (139)	1+7(221)	-
8	35+74 (198)	18+55 (198)	10+34 (198)	5+18 (204)	-	-
9	30+55 (160)	19+53 (218)	12+31 (218)	-	-	-

Solution time for MPI+threads

Prec+Iter (#Iter)

#	1 (1x1)	2 (1x2)	4 (1x4)	8 (2x4)	16 (4x4)	32 (8x4)
10	135+168(180)	75+140(240)	49+100(240)	23+92(400)	11+91(630)	6+78(840)
11	174+191(173)	88+135(177)	116+126(197)	-	-	-
12	39+75(98)	19+52(99)	12+42(101)	8+20(99)	-	-
13	165+60(30)	86+63(46)	44+56(46)	21+37(46)	12+23(45)	-
14	443+476(141)	256+536(181)	132+349(206)	-	-	-
15	332+754(160)	169+580(160)	90+508(164)	48+287(176)	36+157(208)	23+151(36)
16	-	-	137+686(139)	-	-	-
17	-	-	-	-	96+670(350)	-

Solution time for RHS #1 and #2 (Problem 13)

Iterative method	# right-hand side	# Iter	Time, c.
CG	1	400	199.3
	2	402	199.4
SOFGMRES	1	337	348.6
	2 (subspace saved from #1)	45	48.5

Аддитивное приближенное разложение (ВІІС)

Разложение ВІІС (Block Incomplete Inverse Cholesky)
[Капорин'90,94].

Пусть матрица A переупорядочена и разбита на p блоков:

$$\sum_{t=1}^p n_t = n.$$

“Базисное” множество индексов $\{k_{t-1} + 1, \dots, k_t\}$, где $k_{t-1} = n_1 + \dots + n_{t-1}$ ($k_0 = 0$, $k_p = n$) соответствует разбиению на блоки.

“Перекрывающееся” множество индексов $\{j_t(1), \dots, j_t(m_t - n_t)\}$ где $j_t(p) \leq k_{t-1}$, $m_t \geq n_t$, $m_1 = n_1$, строится на основе графа связности разреженной матрицы A .

ВІІС-предобусловливатель записывается в **аддитивном** виде:

$$H = \sum_{t=1}^p V_t U_t^{-1} \begin{bmatrix} 0 & 0 \\ 0 & I_{n_t} \end{bmatrix} U_t^{-T} V_t^T,$$

где V_t – прямоугольные матрицы:

$$V_t = [e_{j_t(1)} | \dots | e_{j_t(m_t - n_t)} | e_{k_{t-1}+1} | \dots | e_{k_t}], \quad t = 1, 2, \dots, p,$$

а U_t – правый треугольный множитель t -й “расширенной” диагональной $(m_t \times m_t)$ -подматрицы

$$A_t \equiv V_t^T A V_t = U_t^T U_t.$$