Контроль точности с использованием ball-арифметики в методе сопряженных градиентов на примере задачи безусловной квадратичной оптимизации

С. В. Розинов

Институт систем энергетики СО РАН, Иркутск, Россия

25 сентября 2023 г.

Мотивация

Встроенная процессорная арифметика накапливает ошибки, что приводит к недоказательным результатам.

- Потребность в арифметике повышенной (лучше произвольной) точности;
- Скорость, наличие быстрых арифметических примитивов;
- Компилируемость, исполнение в машинном коде;
- Показатель точности вычисляемых данных.

Безусловная квадратичная оптимизация

$$F(x) = \frac{1}{2}\langle Qx, x\rangle - \langle c, x\rangle \to min, \quad x \in R^n$$

- $Q \succ 0$ - симметричная положительно определённая матрица размера $n \times n$.
- В данном случае $F'(x) = Qx - c$. Условие экстремума функции $F'(x) = 0$ эквивалентно системе $Qx - c = 0$.
- Функция $F(x)$ достигает своей нижней грани в единственной точке x^*, определяемой уравнением $Qx^* = c$.
- Данная задача оптимизации сводится к решению системы линейных уравнений $Qx = c$.

Безусловная квадратичная оптимизация: использование

- Решение линейных систем:
  - Определенных
  - Недоопределенных
  - Переопределенных
- Квадратичная выпуклая оптимизация с условиями типа равенства
- В качестве вспомогательной задачи в более сложных алгоритмах оптимизации

Почему метод сопряженных градиентов?

- На практике часто сходится быстрее и за меньшее количество шагов, чем градиентный спуск или метод Гаусса
- Прост в реализации ⟹ меньше накладных расходов на итерацию
- Экономный расход ресурсов памяти и процессора для больших разреженных задач (используется только скалярное произведение векторов и умножение матрицы на вектор)

Классический МСГ

$$r_1 = c - Qx_0$$

$$p_1 = r_1$$

$$\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle Qp_k, p_k \rangle}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k Q p_k$$

$$\beta_k = \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k$$

Критерии останова:

- На практике обычно используется $\|r_k\| < \varepsilon$
- Не более I_{max} итераций без прогресса

Теоретическая сходимость метода:

- Если вычисления точны
- Если матрица $\boldsymbol{Q_{n \times n}}$ положительно (отрицательно) определена

Гарантирована сходимость
к единственной точке минимума (максимума)
не более чем за n итераций.

Инструмент: ball-арифметика (ARB library)

- Разновидность интервальной арифметики, предложенная Йорисом ван дер Хувеном в работе

  Ball arithmetic. Technical Report, HAL 00432152. 2011.

- Реализация: Фредрик Йоханссон в 2012 г. разработал библиотеку ARB, которая в 2023 г. вошла в проект FLINT – высокопроизводительную библиотеку арифметики многочленов, точной линейной алгебры и пр.
- Язык реализации: C.

Инструмент: ball-арифметика (ARB library)

- Развивается более 10 лет ⇒ зрелый код.
- Открытый исходный код ⇒ расширяемость, интегрируемость с другими языками программирования.
- Широкий набор оптимизированных примитивов:
  - Действительные и комплексные числа произвольной точности с автоматическим отслеживанием погрешности;
  - Рациональные числа;
  - Векторы и матрицы;
  - Корни, тригонометрия, специальные функции, многочлены;
  - …
- Встроенное распараллеливание («умное» умножение матриц).

Почему ball-арифметика?

- Представление действительного числа в ball-арифметике:

$$[m \pm r] = [m - r, m + r],$$

m – центр интервала, r – радиус.

- m, r – двоичные числа с плавающей точкой:

$$u2^v : u, v \in \mathbb{Z}.$$

- m – число с мантиссой и экспонентой произвольной точности
- r – число с фиксированной (30 бит) мантиссой и экспонентой произвольной точности

- Экономия памяти в 2 раза ☺
- Кратный рост производительности арифметических операций ☺ ☺

Почему ball-арифметика?

TABLE 2
Time to perform a basic operation on intervals with MPFI and Arb, normalized by the time to perform the same operation on floating-point numbers (i.e. just the midpoints) with MPFR. As operands, we take intervals for $x = \sqrt{3}, y = \sqrt{5}$ computed to full precision.

prec	MPFI	Arb	MPFI	Arb	MPFI	Arb
	add		mul		fma	
64	2.58	1.08	2.06	1.03	1.42	0.56
128	2.15	1.03	2.16	1.09	1.62	0.68
256	2.20	1.48	2.14	1.23	1.65	0.70
1024	2.22	1.39	2.05	0.99	1.49	0.76
4096	2.10	1.70	2.02	1.05	1.63	0.95
32768	2.11	1.65	2.02	1.02	1.78	1.00
	div		sqrt		pow	
64	2.96	1.72	2.02	1.78	0.97	0.09
128	2.81	1.79	2.01	1.50	1.21	0.11
256	2.56	1.38	2.15	1.31	1.40	0.13
1024	2.23	0.92	2.03	1.09	1.68	0.29
4096	2.09	0.82	2.03	1.04	1.94	0.67
32768	1.98	1.01	2.02	1.04	1.95	0.79

TABLE 4
Time in seconds to compute recursive factorial product with $N = 10^5$.

prec	MPFR	MPFI	Arb
64	0.0129	0.0271	0.00315
128	0.0137	0.0285	0.00303
256	0.0165	0.0345	0.00396
1024	0.0417	0.0852	0.00441
4096	0.0309	0.0617	0.00543
32768	0.109	0.234	0.00883

Johansson F. Arb: efficient arbitrary-precision midpoint-radius interval arithmetic. IEEE Transactions on Computers. 2017; 66(8):1281-1292.

Генерация тестовых задач

- Вычислим плохо обусловленную матрицу $\boldsymbol{Q_{n\times n}}$.
- Зададим случайный вектор $x^* \in \mathrm{R}^n$ (решение задачи) с компонентами из интервала границ значений $[-b, b]$, $b \in \mathrm{R}$.
- Вычислим свободный вектор $c = Qx^*$.
- На этапе генерации уже вносятся ошибки. Как и в практических данных.

Генерация плохо обусловленных матриц Q

- Спектральный способ построения $Q_{n\times n}$:

$$Q = V\Lambda V^T,$$

V - состоит по строкам из ортонормированной системы векторов v^i, полученных из случайных векторов $u^i \in \mathrm{R}^n$ в результате процесса Грама-Шмидта,

Λ – диагональная матрица из случайных чисел λ_i : $\Lambda_{ii} = \lambda_i$, $i = 1, \ldots, n$.

- v^i - собственные векторы, а λ_i - собственные числа матрицы Q.
- Q - всегда симметрична;
- Q - положительно определенная, если все $\lambda_i > 0$;
- Q - положительно полуопределенная, если все $\lambda_i \geqslant 0$;
- Q - знаконеопределенная, если среди λ_i есть числа с разными знаками.

Генерация плохо обусловленных матриц *Q*

- Матрица Гильберта: в качестве матрицы $Q_{n\times n}$ используем $H_{n\times n}$:

$$H_{n\times n} = \frac{1}{i+j-1}, i, j = 1, 2, ..., n$$

- Матрица Гильберта положительно определена
- Число обусловленности возрастает очень быстро:

$$\text{cond}(H_{n\times n}) = O((1+\sqrt{2})^{4n}/\sqrt{n})$$

Показатель точности ball-чисел

Количество точных битов ball-числа q:

$$P_2(q) = B_m(q) - B_r(q) - 1$$

$B_m(q)$, $B_r(q)$ - позиции старших битов центра и радиуса числа q, выравненные по порядку.

Пример (аналогия в десятичной системе):

***q* = 335.42e4 ± 2.71e1** Выравнивание порядка: ***q* = 3354200e0 ± 27.1e0**

***m* = 3354200e0** – позиция старшего разряда: 6

***r* = 27.1e0** – позиция старшего разряда: 1

$P_{10}(q)$ = 6 – 1 – 1 = **4 точных разряда**.

Количество точных десятичных цифр для двоичных чисел:

$$P_{10}(q) = \lfloor round\left(\frac{P_2(q)}{log_2 10}\right) - 1 \rfloor.$$

Показатель точности ball-чисел

- В особых случаях:

$$P_{10}(m \pm 0) = M$$
$$P_{10}(0 \pm r) = P_{10}(1 \pm r).$$

M – заданная длина мантиссы *m.*

- Для составных числовых объектов:

b – вектор длины *n*:

$$P_{10}(b) = min(P_{10}(b_i),\ i = 1, \ldots, n),$$

A – матрица *n* × *m*:

$$P_{10}(A) = min(P_{10}(a_{ij}),\ i = 1, \ldots, n,\ j = 1, \ldots, m).$$

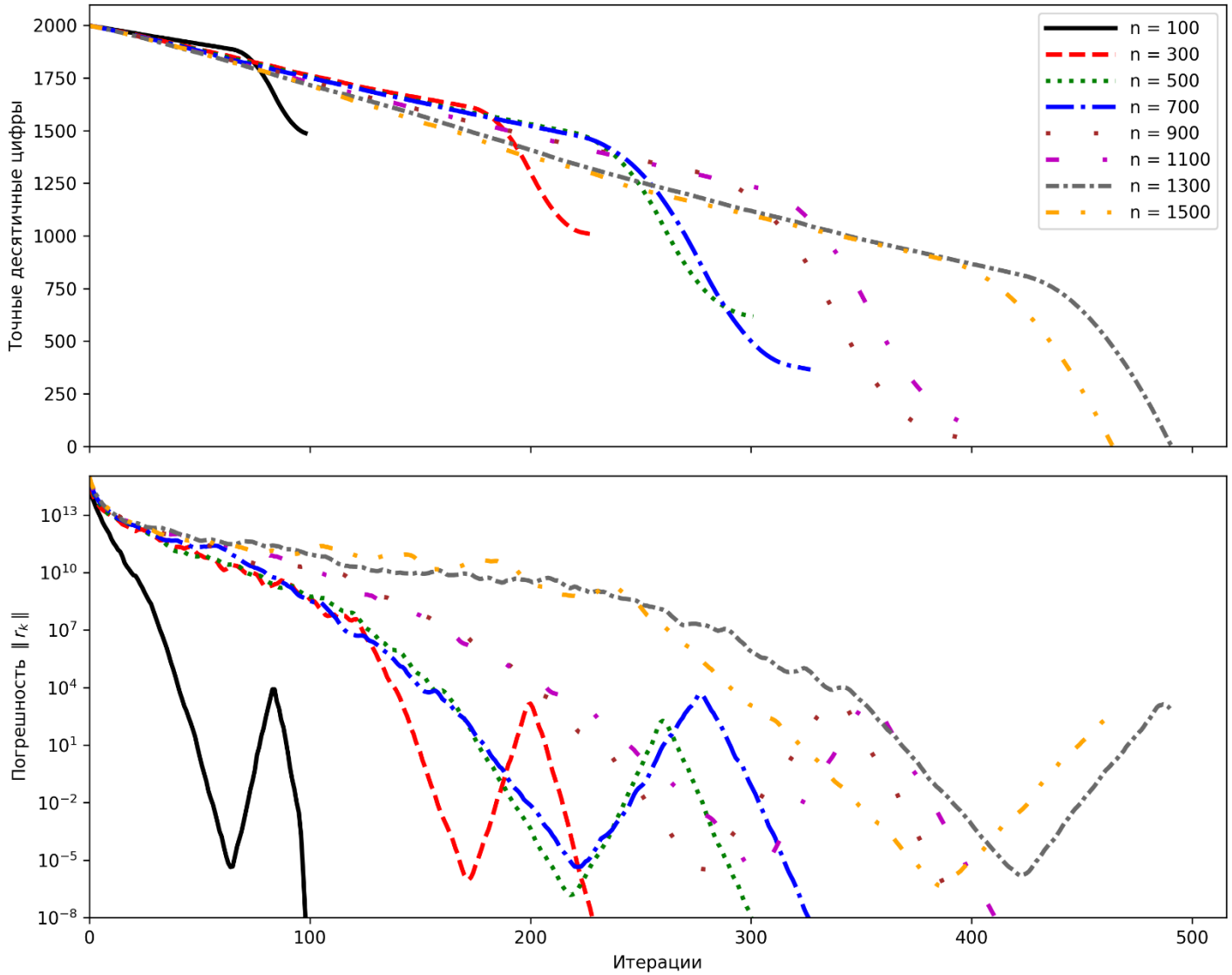
Вычислительный эксперимент № 1

Параметры эксперимента

Размерность задачи n	$100...1500$
Разброс компонентов x^*	$-3 \times 10^4 ... 3 \times 10^4$
Разброс собственных чисел $\lvert\lambda_i\rvert$	$10^{-10} ... 10^{10}$
Допустимая погрешность ε	10^{-8} для спектральной Q 10^{-50} для гильбертовой Q
Начальная точность M, цифр	2000

Вычисления в арифметике *float64* не дают результата: итерации продолжаются без прогресса.

Вычислительный эксперимент № 1

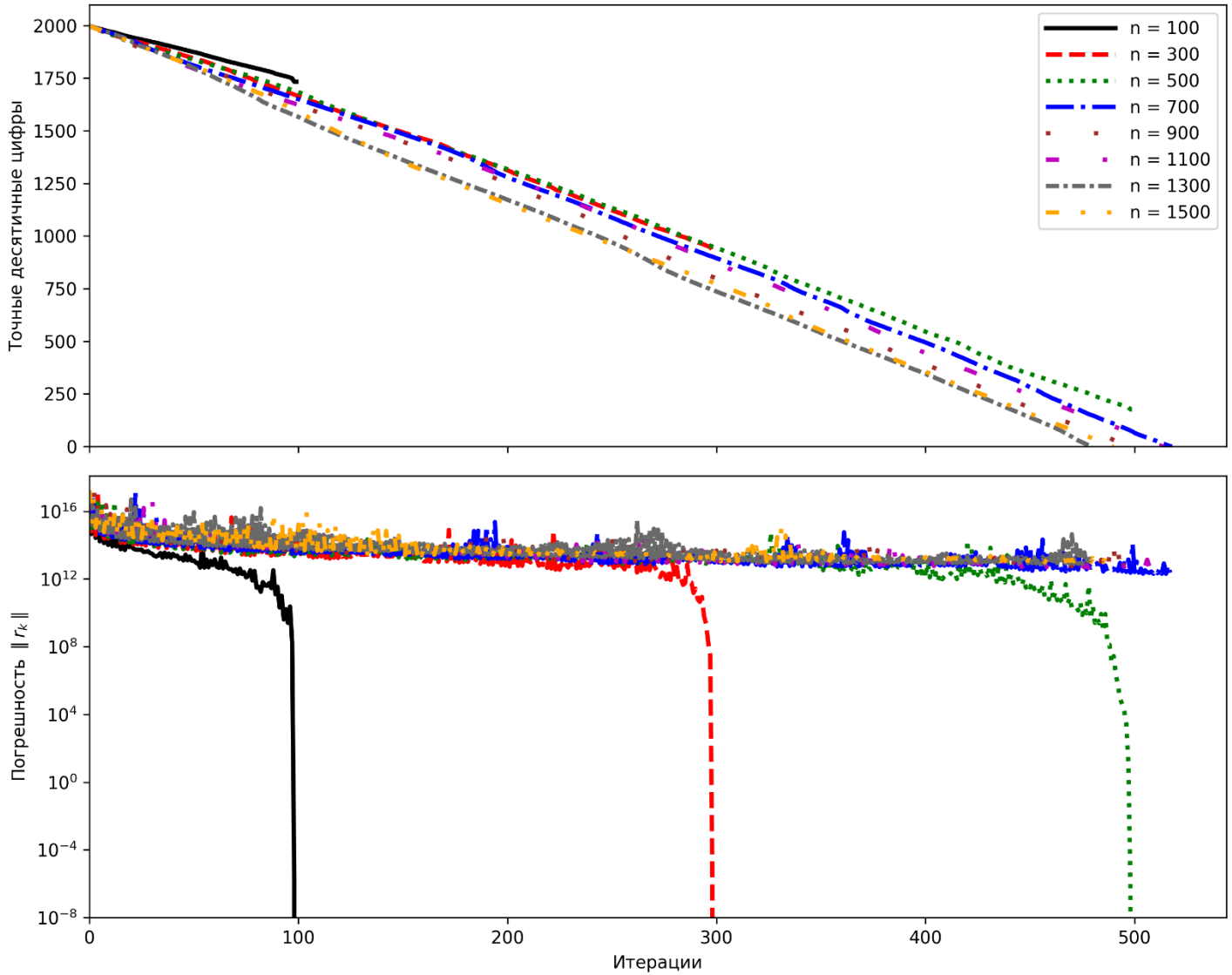


Деградация точности β_k
Положительно определенная матрица Q

Спектральный способ построения

Поиск минимума

Вычислительный эксперимент № 1

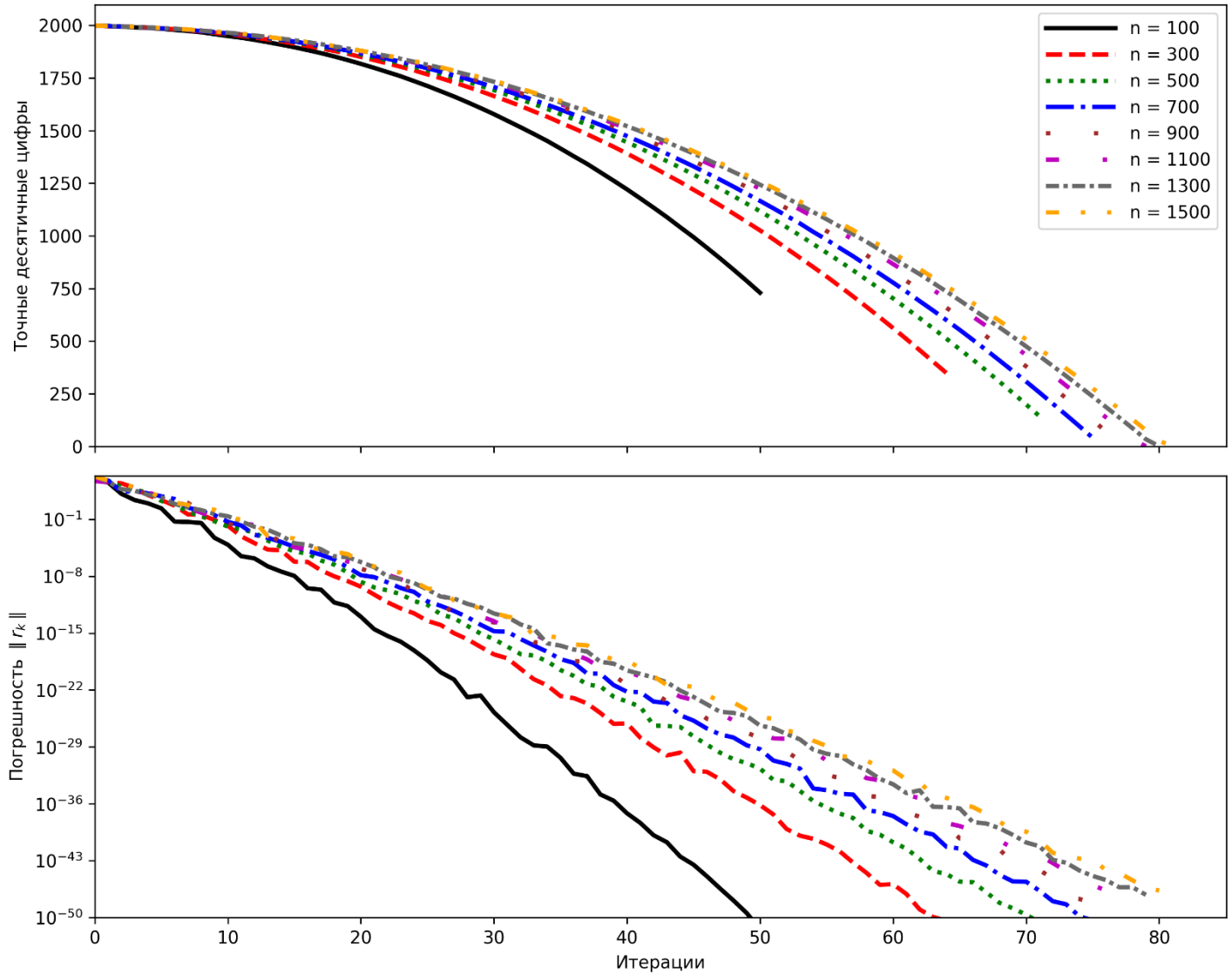


Деградация точности β_k
Знаконеопределенная матрица Q

Спектральный способ построения

Поиск седловой точки

Вычислительный эксперимент № 1



Деградация точности β_k
Положительно определенная матрица Q

Способ построения Гильберта

Поиск минимума

Наблюдения по результатам

- Деградация точности в процессе вычислений не хаотична. Характер убывания точности связан с природой решаемой задачи.

- На протяжении итерационного процесса темп деградации точности может как возрастать, так и убывать.

- При начальной длине мантиссы, достаточной для компенсации потерь точности, алгоритм КМСГ, как правило, достигает точки ε - оптимального решения генерируемых тестовых задач.

- На практике до начала итерационного процесса КМСГ оценка темпа убывания точности и необходимого запаса мантиссы затруднительна – мы можем не знать свойств матрицы Q.

Что делать при исчерпании точности?

- Перезапуск алгоритма:
  - С увеличением длины мантиссы (линейно, экспоненциально);
  - Попытаться предсказать достаточную длину мантиссы по результатам неудачного раунда (раундов);
- Использовать центрирование
- Комбинация подходов

Процедура центрирования

Замена: $q = m \pm r \longrightarrow \hat{q} = m \pm \hat{r}(M, m),$

$\hat{r}(M, m)$ - радиус двоичного представления центра *m* в мантиссе длины *M*.

- Восстановление показателя точности к начальному $P_{10}(\hat{q}) = M$
- Возможность продолжить вычисления с контролем точности. Искажение исправляется последующими итерациями.
- Сохранение контроля над потерей информации о значении *q* - ограничение отклонения не более чем в десятичной позиции P_{10min} :

$$|q - \hat{q}| \leqslant r \leqslant 10^{-P_{10min}}$$

Процедура центрирования

- Составные числовые объекты центрируются покомпонентно:

$$\hat{b} = [\hat{b}_i], i = 1, ..., n$$
$$\hat{A} = [\hat{a}_{ij}], i = 1, ..., n, j = 1, ..., m.$$

- Как выбрать P_{10min}? Оценка отклонения для векторов.

Для ball-вектора

$$b = [b_i], b_i = m_i \pm r_i,$$
$$i = 1, ..., n.$$

Оценка отклонения

$$\|\hat{b} - b\|_1 \leqslant n \cdot max(r_i, i = 1, ..., n) \leqslant n \cdot 10^{-P_{10min}}$$
$$\|\hat{b} - b\|_2 \leqslant \sqrt{n} \cdot max(r_i, i = 1, ..., n) \leqslant \sqrt{n} \cdot 10^{-P_{10min}}.$$

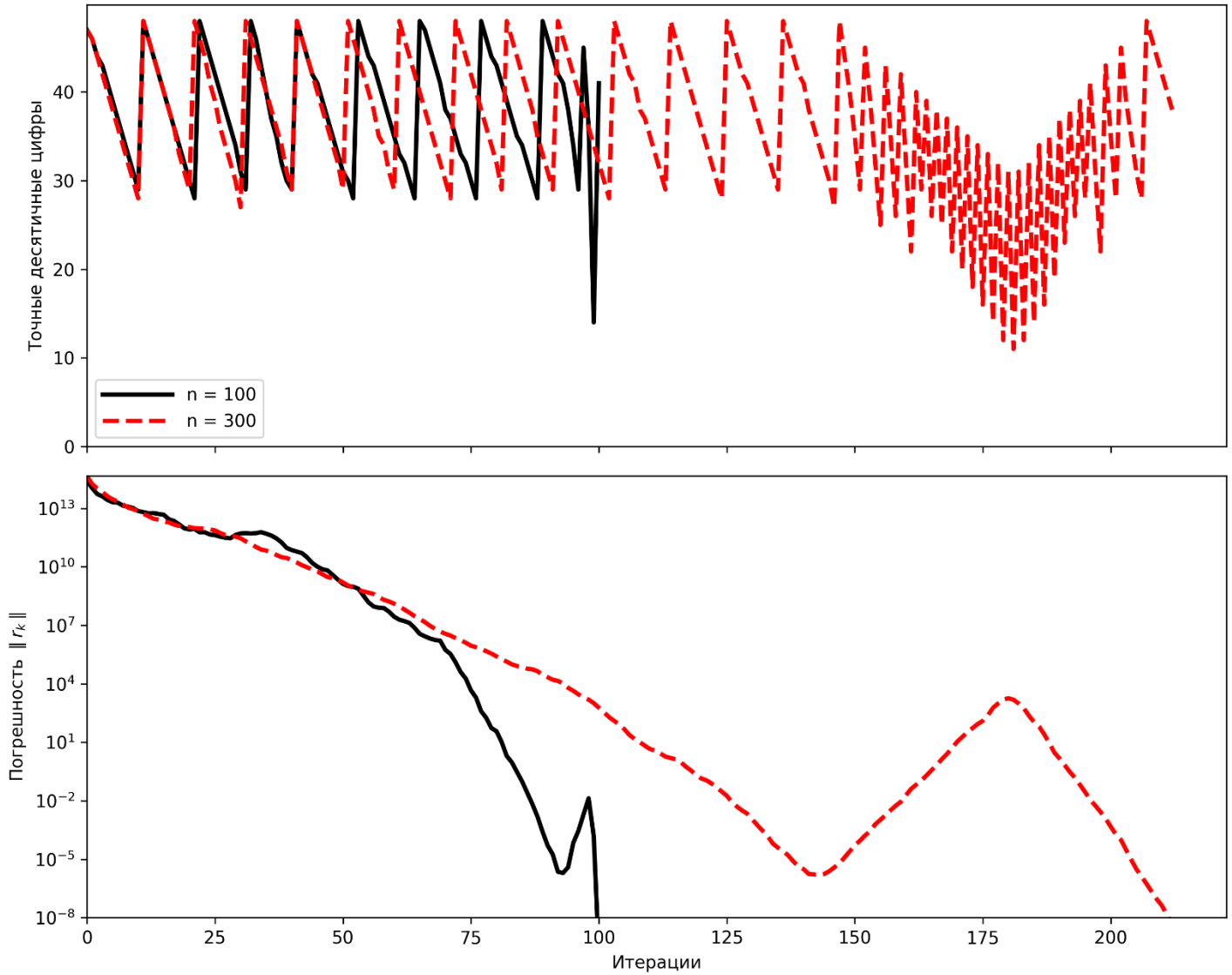
$$10^{-P_{10min}} \ll \varepsilon$$

Процедура центрирования

Примеры для длины мантиссы $M = 50$

M	q	$P_{10}(q)$	$\hat{q}$	$P_{10}(\hat{q})$	Замечание
50	$0.3 \pm 1e{-10}$	8	$0.3 \pm 1.34e{-52}$	50	Радиус положителен, поскольку мантисса 0.3 не представляется точно конечной двоичной дробью.
50	$0.25 \pm 1e{-10}$	8	0.25 ± 0	50	Радиус равен нулю, так как мантисса 0.25 - точная конечная двоичная дробь.
50	$0 \pm 1e{-4}$	3	0 ± 0	50	Аналогично предыдущему примеру.

Вычислительный эксперимент № 2

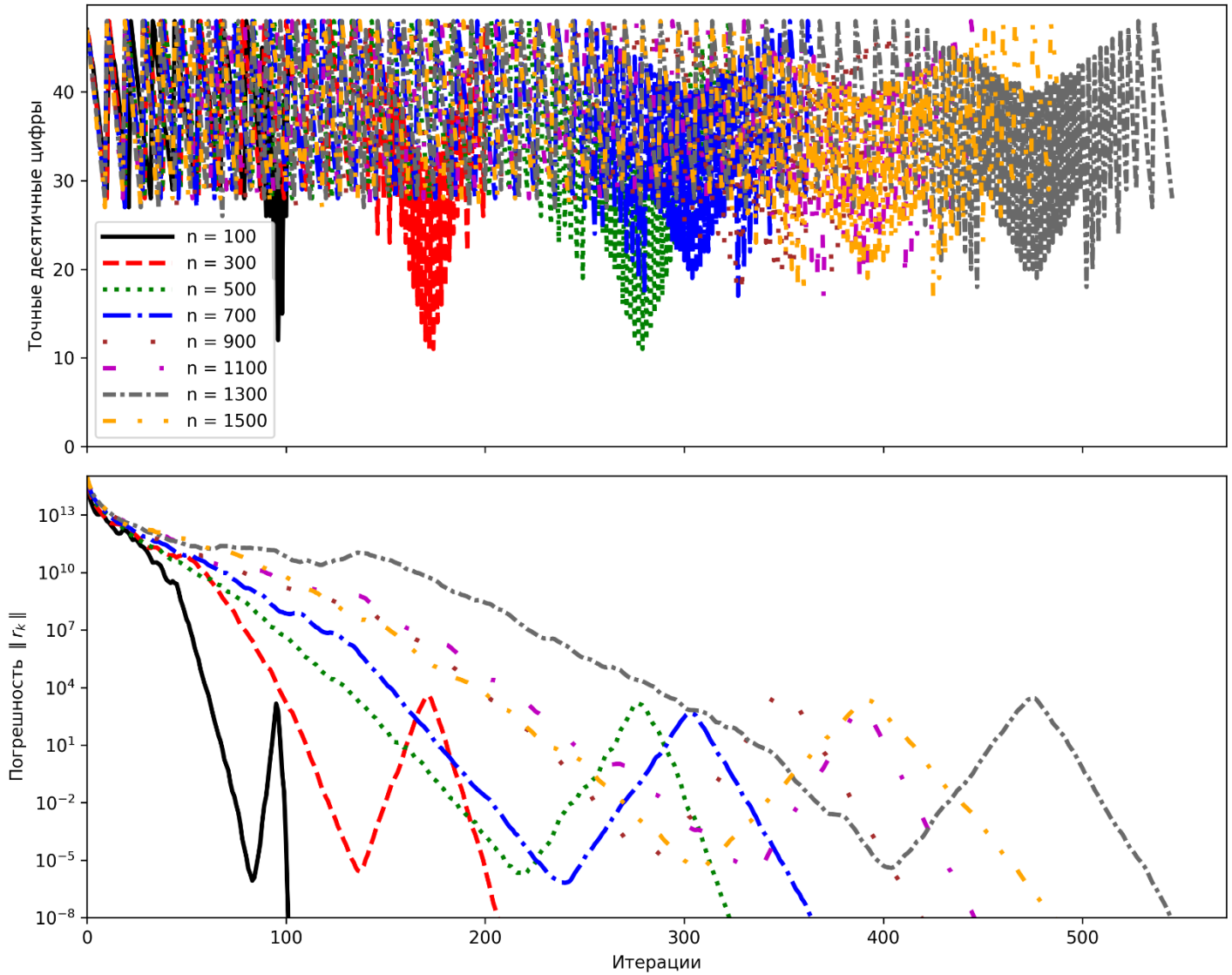


Деградация точности β_k
Положительно определенная матрица Q

Спектральный способ построения

Поиск минимума

Вычислительный эксперимент № 2

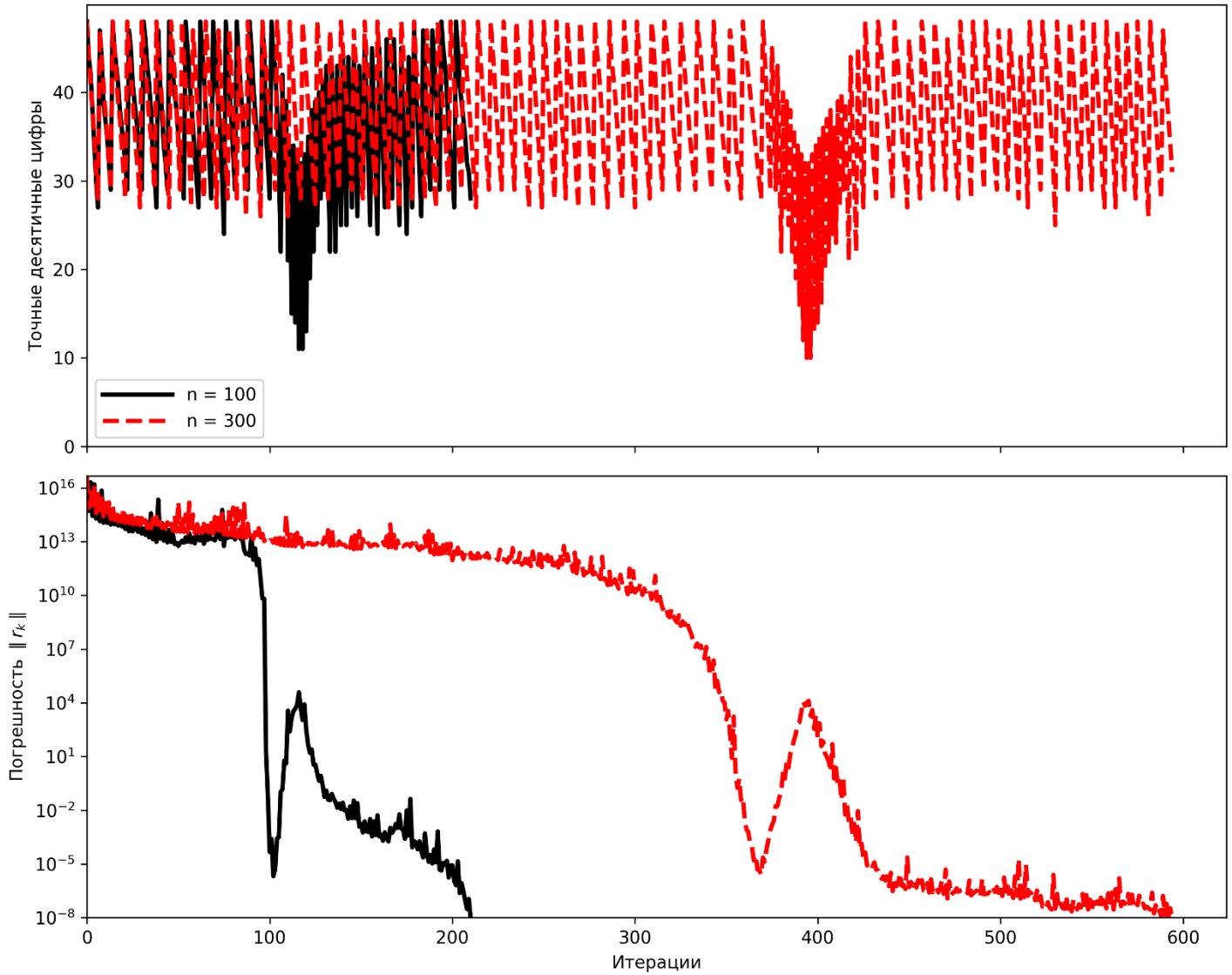


Деградация точности β_k
Положительно определенная матрица Q

Спектральный способ построения

Поиск минимума

Вычислительный эксперимент № 2

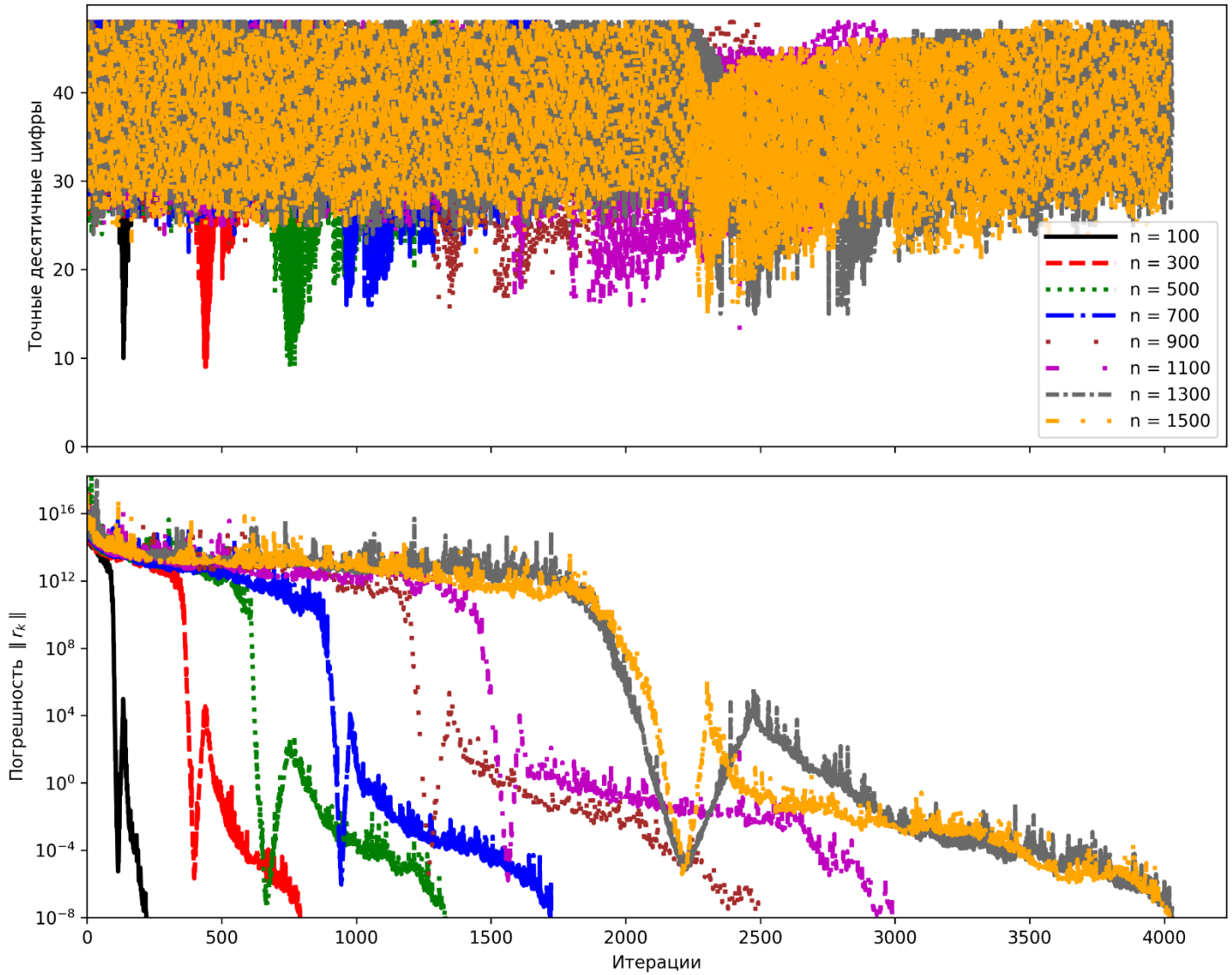


Деградация точности β_k
Знаконеопределенная матрица Q

Спектральный способ построения

Поиск седловой точки

Вычислительный эксперимент № 2



Деградация точности β_k
Знаконеопределенная матрица Q

Спектральный способ построения

Поиск седловой точки

Сравнительный эксперимент

Параметры эксперимента

Размерность задачи n	100...1500
Разброс компонентов x^*	$-3 \times 10^4 ... 3 \times 10^4$
Разброс собственных чисел $\lvert\lambda_i\rvert$	$10^{-10} ... 10^{10}$
Допустимая погрешность ε	10^{-8} для спектральной Q 10^{-15} для гильбертовой Q
Количество расчетов	10
Начальная точность M	Без центрирования: подбирается динамически С центрированием: фиксирована для каждого вида задачи
Порог точности P_{10min}	С центрированием: фиксирован для каждого вида задачи

Сравнительный эксперимент: результаты

Положительно определенная матрица;

спектральный способ построения.

		Без центрирования			С центрированием				
n	ε	M	Итерации	Время, с	M	P_{10min}	Итерации	Время, с	Изм. времени, %
100	10^{-8}	365	100	0,078	30	20	118	0,053	−32
300	10^{-8}	1093	232	1,367	30	20	262	0,338	−75
500	10^{-8}	1450	309	5,114	30	20	353	1,007	−80
700	10^{-8}	1730	340	11,699	30	20	377	1,887	−84
900	10^{-8}	2073	409	26,341	30	20	470	3,752	−86
1100	10^{-8}	2345	468	48,151	30	20	664	7,872	−84
1300	10^{-8}	2568	503	77,556	30	20	816	14,820	−81
1500	10^{-8}	2759	549	113,739	30	20	870	19,137	−83

Сравнительный эксперимент: результаты

Знаконеопределенная матрица;

спектральный способ построения.

		Без центрирования			С центрированием				
n	ε	M	Итерации	Время, с	M	P_{10min}	Итерации	Время, с	Изм. времени, %
100	10^{-8}	350	100	0.084	50	30	213	0.112	37
300	10^{-8}	1113	300	1.530	50	30	749	1.246	−18
500	10^{-8}	1910	500	9.142	50	30	1275	4.962	−46
700	10^{-8}	2784	700	31.427	50	30	1727	11.242	−64
900	10^{-8}	3727	900	81.469	50	30	2270	22.910	−72
1100	10^{-8}	4775	1100	180.061	50	30	3103	45.605	−75
1300	10^{-8}	5683	1300	347.431	50	30	3607	76.783	−78
1500	10^{-8}	6707	1500	615.525	50	30	4141	118.311	−81

Сравнительный эксперимент: результаты

Положительно определенная матрица;

способ построения Гильберта.

		Без центрирования			С центрированием				
n	ε	M	Итерации	Время, с	M	P_{10min}	Итерации	Время, с	Изм. времени, %
100	10^{-15}	300	23	0.026	50	30	111	0.063	160
300	10^{-15}	281	28	0.105	50	30	176	0.329	216
500	10^{-15}	312	31	0.292	50	30	228	0.941	232
700	10^{-15}	324	32	0.461	50	30	272	1.826	297
900	10^{-15}	336	33	0.786	50	30	306	3.375	329
1100	10^{-15}	350	34	1.287	50	30	323	5.584	334
1300	10^{-15}	390	35	1.823	50	30	352	7.131	294
1500	10^{-15}	370	36	2.208	50	30	380	10.149	359

Наблюдения по результатам

- Центрирование может и не давать выигрыша.
- Целесообразно комбинировать центрирование и увеличение длины мантиссы (при отсутствии улучшения решения).
- Оба механизма могут быть реализованы в виде отключаемых опций для более гибкой подстройки под конкретный класс практических задач.

Комбинированный алгоритм МСГ с контролем точности (упрощенная схема)

1. Инициализация МСГ.
2. Итерация МСГ.
3. Если $\|r_k\| < \varepsilon$ - СТОП: Найдено ε-оптимальное решение задачи.
4. Если превышен предел I_{max} по итерациям без прогресса – на шаг 9.
5. Проверка исчерпания точности. Если $P_{10}(\beta_k)$ >= P_{10min} - продолжить итерации; на шаг 2.
6. Если центрирование отключено – на шаг 9.
7. Центрировать x_{k+1}, r_{k+1}, p_{k+1}.
8. Продолжить итерации; на шаг 2.
9. Увеличить длину мантиссы (линейно или экспоненциально).
10. Если превышен предел M_{max} по длине мантиссы – СТОП: Неудача, решение не найдено.
11. Перезапуск МСГ; на шаг 1.

Заключение

- Возможность повышения надежности данных в КМСГ.

- Средство рефлексии в отношении точности данных в процессе работы алгоритмов.

- Дальнейшее развитие:
  - Блочные и разреженные системы;
  - Предсказание (регрессия) требуемой длины мантиссы для расчетов без центрирования;
  - Использование в составе более сложных моделей.

Спасибо за внимание