

# IMPLEMENTATION AND CONVERGENCE EVALUATION OF MULTIPLE-PRECISION ITERATIVE CG AND PCG SOLVERS FOR GPUS

K. Isupov, V. Knyazkov\*, A. Korzhavina

Vyatka State University,  
Kirov 610000, Russia

\*Penza State University,  
Penza 440026, Russia

---

DOI: 10.24412/2073-0667-2022-1-17-27

To solve systems of linear equations with sparse coefficient matrices, iterative Krylov subspace methods such as conjugate gradients (CG) and preconditioned conjugate gradients (PCG) are widely used. The convergence rate and numerical reliability of these methods may suffer from rounding errors. One way to reduce the impact of rounding errors is to use arithmetic over numbers with increased word length, which is called multiple-precision arithmetic. In the paper, we consider multiple-precision iterative CG and PCG solvers for graphics processing units (GPUs) and evaluate their convergence rate. The preconditioned implementation uses a diagonal preconditioner, which is well suited for parallel computing. The residue number system is employed to support multiple-precision floating-point arithmetic. The matrix-vector product is implemented as a hybrid kernel, in which a double-precision matrix, represented in the CSR storage format, is multiplied by a dense multiple-precision vector. A two-stage algorithm is used to compute the parallel multiple-precision dot product. Experimental results with sparse matrices of different sizes show that higher arithmetic precision improves the convergence rate of iterative methods.

**Key words:** sparse linear algebra, conjugate gradient method, multiple-precision arithmetic, residue number system, CUDA.

## References

1. Saad Y. Iterative Methods for Sparse Linear Systems, 2nd ed. Philadelphia: SIAM, 2003. 547 p.
2. Barrett R. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (2nd Edition) / R. Barrett et al. Philadelphia: SIAM, 1994. 107 p.
3. Saito T., Ishiwata E., Hasegawa H. Analysis of the GCR method with mixed precision arithmetic using QuPAT // Journal of Computational Science. 2012. Vol. 3, N 3. P. 87–91. DOI: 10.1016/j.jocs.2011.05.001.
4. Cools S., Yetkin E.F., Agullo E., Giraud L., Vanroose W. Analyzing the Effect of Local Rounding Error Propagation on the Maximal Attainable Accuracy of the Pipelined Conjugate Gradient Method // SIAM Journal on Matrix Analysis and Applications. 2018. Vol. 39, N 1. P. 426–450. DOI: 10.1137/17M1117872.
5. Kouya T. A highly efficient implementation of multiple precision sparse matrix-vector multiplication and its application to product-type Krylov subspace methods // International Journal of Numerical Methods and Applications. 2012. Vol. 7, N 2. P. 107–119.

---

This research was funded by the Russian Science Foundation grant number 20-71-00046.

- 
6. Hishinuma T., Hasegawa H., Tanaka T. SIMD Parallel Sparse Matrix-Vector and Transposed-Matrix-Vector Multiplication in DD Precision // Lecture Notes in Computer Science. 2017. Vol. 10150. P. 21–34.
  7. Hasegawa H. Utilizing the quadruple-precision floating-point arithmetic operation for the Krylov subspace methods. 2003. [Electron. Res.]: <http://www.slis.tsukuba.ac.jp/~hasegawa.hidehiko ga/GYOSEKI/hasegaw.pdf> (accessed 17 January 2022).
  8. Masui K. Ogino M. Research on the Convergence of Iterative Method Using Mixed Precision Calculation Solving Complex Symmetric Linear Equation // IEEE Transactions on Magnetics. 2020. Vol. 56, N 1. Article no. 7503604. DOI: 10.1109/TMAG.2019.2951280.
  9. Mukunoki D., Takahashi D. Using Quadruple Precision Arithmetic to Accelerate Krylov Subspace Methods on GPUs // Lecture Notes in Computer Science. 2014. Vol. 8384. P. 632–642. DOI: 10.1007/978-3-642-55224-3\_59.
  10. Isupov K., Knyazkov V. Multiple-Precision BLAS Library for Graphics Processing Units // Communications in Computer and Information Science. 2020. Vol. 1331. P. 37–49. DOI: 10.1007/978-3-030-64616-5\_4.
  11. Omondi A., Premkumar B. Residue Number Systems: Theory and Implementation. Imperial College Press, 2007. 402 p.



# РЕАЛИЗАЦИЯ И ОЦЕНКА СХОДИМОСТИ ИТЕРАЦИОННЫХ CG И PCG РЕШАТЕЛЕЙ МНОГОКРАТНОЙ ТОЧНОСТИ ДЛЯ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

К. С. Исупов, В. С. Князьков\*, А. С. Коржавина

Вятский государственный университет,  
610000, Киров, Россия

\*Пензенский государственный университет,  
440026, Пенза, Россия

---

УДК 004.222+004.272.25

DOI: 10.24412/2073-0667-2022-1-17-27

Для решения систем линейных алгебраических уравнений с разреженными матрицами коэффициентов широко применяются итерационные методы подпространства Крылова. Однако сходимость этих методов может ухудшаться из-за ошибок округления, возникающих при выполнении вычислений в арифметике фиксированной разрядности. Снизить влияние ошибок округления позволяет использование арифметики многократной точности, обеспечивающей выполнение операций с числами повышенной разрядности. В статье представлены реализации итерационных решателей многократной точности на базе метода сопряженных градиентов без предобуславливания и с диагональным предобуславливанием для графических процессоров видеокарт. Для поддержки вычислений с числами повышенной разрядности используется система остаточных классов. Матрично-векторное произведение реализовано в виде гибридного ядра, в котором матрица двойной точности, представленная в формате CSR, умножается на вектор многократной точности. Параллельное скалярное произведение вычисляется с использованием двухэтапного алгоритма. Результаты экспериментов с разреженными матрицами различных размеров показывают, что повышенная точность арифметики позволяет ускорить сходимость итерационных методов.

**Ключевые слова:** разреженная линейная алгебра, метод сопряженных градиентов, арифметика многократной точности, система остаточных классов, CUDA.

**Введение.** Системы линейных алгебраических уравнений (СЛАУ) часто возникают из дискретизации дифференциальных уравнений в частных производных методами конечных элементов, конечных разностей или конечных объемов. Матрицы из этих приложений обычно большие и разреженные — существенная часть их элементов представлена нулями. В настоящее время одними из самых успешных методов решения разреженных линейных систем являются итерационные методы подпространства Крылова, такие как метод сопряженных градиентов (CG, PCG), обобщенный метод минимальных невязок (GMRES), BiCG, BiCGSTAB, CGS и MINRES [1]. Эти методы состоят в выполнении ряда матрично-векторных и векторных операций, таких как  $Ax$ ,  $A^T x$ ,  $r^T r$ ,  $r + \beta r$  и др., а их сходимость

---

Исследование выполнено за счет гранта Российского научного фонда (проект № 20-71-00046).

зависит от спектральных свойств матрицы коэффициентов  $A$ . Часто свойства матрицы  $A$  не обеспечивают приемлемую скорость сходимости итераций, поэтому вместо исходной линейной системы  $Ax = b$  решают систему  $M^{-1}Ax = M^{-1}b$ , где  $M$  — предобуславливатель. Выбор эффективного предобуславливателя играет важнейшую роль в применении итерационных методов [2].

С другой стороны, даже при тщательном выборе предобуславливателя сходимость и надежность итерационных методов может быть довольно плохой из-за ошибок округления, возникающих при выполнении вычислительных операций в арифметике фиксированной разрядности [3]. В частности, в [4] рассмотрены вопросы влияния погрешностей вычислений на эффективность итерационных линейных решателей на базе CG метода и показано, что в ряде случаев локальные ошибки округления могут существенно замедлять сходимость итераций, в особенности в алгоритмах, оптимизированных для параллельных архитектур. Иногда проблема сходимости не решается даже тщательным подбором предобуславливателя. Более того, многие обладающие хорошими математическими свойствами предобуславливатели (например, основанные на неполном LU разложении) включают существенную часть последовательных вычислений, что затрудняет их эффективную реализацию на параллельных системах. В связи с этим важную роль приобретают подходы, позволяющие снизить влияние ошибок округления на результат вычислений. Одним из таких подходов является увеличение точности (разрядности) компьютерной арифметики. В настоящее время имеется ряд работ в данном направлении:

1) В [5] представлены реализации высокоточных итерационных решателей на базе различных вариантов метода сопряженных градиентов: BiCG, CGS, BiCGSTAB и GPBiCG. Показано, что для обеспечения сходимости итерационных методов в ряде случаев требуется точность более 2000 бит (даже при использовании предобуславливателей).

2) В [6] реализовано оптимизированное умножение разреженной матрицы двойной точности на вектор четверной точности с использованием набора расширений AVX2. Для достижения четверной точности использован арифметический формат double-double, в котором число хранится в виде невычисленной суммы двух стандартных (double) чисел с плавающей точкой, что обеспечивает приблизительно 106 бит мантиссы.

3) В [7] исследована сходимость BiCG метода при решении плохо обусловленных линейных систем с матрицами Теплица. Показано, что реализация решателя без предобуславливателя, но с использованием расширенной (четверной) точности позволяет добиться сходящегося итерационного процесса, в отличие от реализации на основе стандартной арифметики.

4) В [8] реализован итерационный CG решатель четверной и смешанной (double + double-double) точности, использованный при решении симметричных линейных систем, возникающих в анализе высокочастотных электромагнитных полей. Представленные результаты показывают, что применение смешанной точности позволяет сократить число итераций и время решения по сравнению со стандартной арифметикой двойной точности.

5) В [9] показано, что CUDA-реализации CG и BiCGSTAB методов в double-double арифметике без предобуславливания оказываются быстрее и точнее при решении некоторых задач, чем реализации в стандартной арифметике двойной точности с предобуславливанием на базе неполного LU разложения.

В данной статье исследуется влияние точности компьютерной арифметики на сходимость итерационных методов на примере параллельной реализации CG и PCG решателей многократной точности для графических процессоров видеокарт (GPU). Рассматривае-

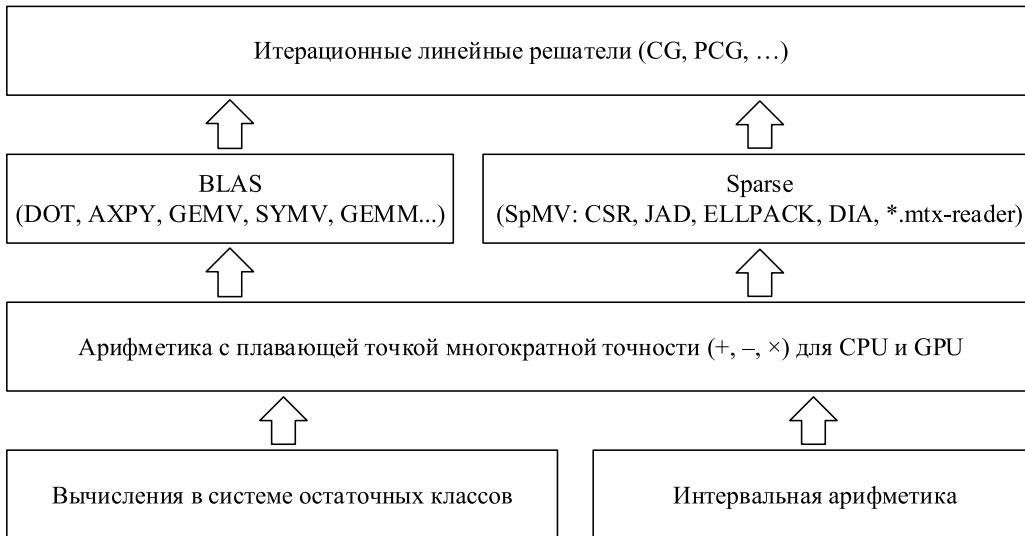


Рис. 1. Обобщенная структура библиотеки MPRES-BLAS

мые решатели входят в состав библиотеки подпрограмм линейной алгебры MPRES-BLAS [10].

**1. Библиотека MPRES-BLAS.** MPRES-BLAS — это библиотека операций линейной алгебры многократной точности для GPU с поддержкой CUDA. Для представления чисел с плавающей точкой библиотека использует систему остаточных классов (СОК) [11]. СОК обладает рядом преимуществ перед позиционными системами счисления, главным из которых является отсутствие переносов между цифрами числа, что позволяет заменить многоразрядные операции группами покомпонентных операций небольшой разрядности, которые выполняются независимо друг от друга. Общая структура MPRES-BLAS представлена на рис. 1.

Библиотека реализует ряд плотных матричных, матрично-векторных и векторных операций линейной алгебры, подобных BLAS, а также поддерживает разреженное матрично-векторное умножение в форматах CSR, ELLPACK, JAD и DIA. Кроме этого, MPRES-BLAS может выступать в качестве универсальной арифметической библиотеки, так как предоставляет базовые арифметические операции многократной точности на CPU и GPU.

При использовании MPRES-BLAS точность вычислений определяется набором модулей  $\{m_1, m_2, \dots, m_n\}$ . Модули должны быть попарно взаимно простыми целыми числами. Если  $M$  — произведение всех модулей, то точность арифметических операций в битах определяется значением  $p = \lfloor \log_2 \sqrt{M} \rfloor - 1$ . В библиотеке имеется несколько предопределенных наборов модулей, обеспечивающих различные уровни точности, но пользователь также может самостоятельно задать новый набор.

Для представления чисел с плавающей точкой в MPRES-BLAS определен тип данных `mp_float_t`. Переменные этого типа могут быть объявлены и определены как на стороне хоста, так и непосредственно из кода устройства. Обмен данными между хостом и устройством осуществляется путем вызова стандартных функций CUDA API. Помимо типа данных `mp_float_t`, в библиотеке также определены два векторных типа, `mp_array_t` и `mp_collection_t`, предназначенные для эффективных вычислений с массивами многоразрядных чисел на GPU. Тип `mp_array_t` хранит массив в разложенной форме, то есть как набор массивов, представляющих отдельные части чисел многократной точности. В свою

очередь, `mp_collection_t` можно рассматривать как „облегченный“ вариант `mp_array_t`. Векторные типы хорошо подходят для плотных ядер линейной алгебры с регулярным доступом к памяти, но не дают существенных преимуществ при реализации разреженных ядер с нерегулярным паттерном доступа. Поэтому в рассматриваемых далее реализациях линейных решателей CG и PCG используется тип данных `mp_float_t`.

**2. Решатели CG и PCG многократной точности.** В алгоритмах 1 и 2 представлен псевдокод итерационного метода сопряженных градиентов без предобуславливания (CG) и с диагональным предобуславлением (PCG) для решения симметричных положительно определенных СЛАУ. Предобуславливатель  $M = \text{diag}(1/a_{11}, 1/a_{22}, \dots, 1/a_{nn})$  предварительно вычисляется на хосте и копируется в память GPU перед выполнением метода. Несмотря на то, что выбранный предобуславливатель обычно уступает по числу итераций более классическим вариантам, таким как неполное разложение Холецкого, он обладает рядом достоинств. Прежде всего, он очень просто вычисляется и подходит для параллельной реализации.

---

Алгоритм 1. CG решатель многократной точности.

```

1:  $k \leftarrow 1$ 
2:  $r \leftarrow b - Ax_0$                                 ▷ mp_spmv_csr, mp_diff (GPU)
3:  $\|r_0\| \leftarrow \|r\|$                             ▷ mp_norm2 (CPU + GPU)
4: while  $\|r\| > tol \cdot \|r_0\|$  and  $k \leq N_{\max}$  do    ▷ mp_norm2 (CPU + GPU)
5:    $\rho_k \leftarrow r^T r$                                 ▷ mp_dot (GPU)
6:   if  $k = 1$  then
7:      $p \leftarrow r$ 
8:   else
9:      $\beta = \rho_k / \rho_{k-1}$                           ▷ mp_div (CPU)
10:     $p \leftarrow r + \beta p$                             ▷ mp_axpy (GPU)
11:   end if
12:    $q \leftarrow Ap$                                     ▷ mp_spmv_csr (GPU)
13:    $\alpha \leftarrow \rho_k / p^T q$                       ▷ mp_dot, mp_div (CPU + GPU)
14:    $x \leftarrow x + \alpha p$                            ▷ mp_axpy (GPU)
15:    $r \leftarrow r - \alpha q$                            ▷ mp_axpy (GPU)
16: end while
```

---

В комментариях к псевдокоду указаны используемые функции библиотеки MPRES-BLAS, а также платформы на которых они выполняются. Описание функций представлено в таблице 1. Поскольку ряд операций, таких как деление и извлечение квадратного корня, являются сложными для СОК, в настоящее время они реализованы в виде обертки над функциями библиотеки MPFR<sup>1</sup> и выполняются на CPU.

Операция разреженного матрично-векторного умножения (SpMV) является ключевым вычислительным ядром итерационных методов. Рассматриваемые реализации используют ядро смешанной точности `mp_spmv_csr`, которое умножает матрицу двойной точности (double) на вектор многократной точности. Для хранения матрицы используется формат сжатых строк (CSR), и каждый поток GPU обрабатывает одну строку, генерируя один элемент выходного вектора. Для эффективного доступа к промежуточным результатам используется разделяемая память.

---

<sup>1</sup><https://www.mpfr.org/>

---

Алгоритм 2. PCG решатель многократной точности.

```

1:  $k \leftarrow 1$ 
2:  $r \leftarrow b - Ax_0$                                 ▷ mp_spmv_csr, mp_diff (GPU)
3:  $\|r_0\| \leftarrow \|r\|$                             ▷ mp_norm2 (CPU + GPU)
4: while  $\|r\| > tol \cdot \|r_0\|$  and  $k \leq N_{\max}$  do    ▷ mp_norm2 (CPU + GPU)
5:    $z \leftarrow Mr$                                     ▷ mp_prod_d (GPU)
6:    $\rho_k \leftarrow r^T z$                             ▷ mp_dot (GPU)
7:   if  $k = 1$  then
8:      $p \leftarrow z$                                 ▷ mp_div (CPU)
9:   else
10:     $\beta = \rho_k / \rho_{k-1}$                       ▷ mp_axpy (GPU)
11:     $p \leftarrow z + \beta p$ 
12:   end if
13:    $q \leftarrow Ap$                                 ▷ mp_spmv_csr (GPU)
14:    $\alpha \leftarrow \rho_k / p^T q$                     ▷ mp_dot, mp_div (CPU + GPU)
15:    $x \leftarrow x + \alpha p$                         ▷ mp_axpy (GPU)
16:    $r \leftarrow r - \alpha q$                           ▷ mp_maxpy (GPU)
17:    $k \leftarrow k + 1$ 
18: end while

```

---

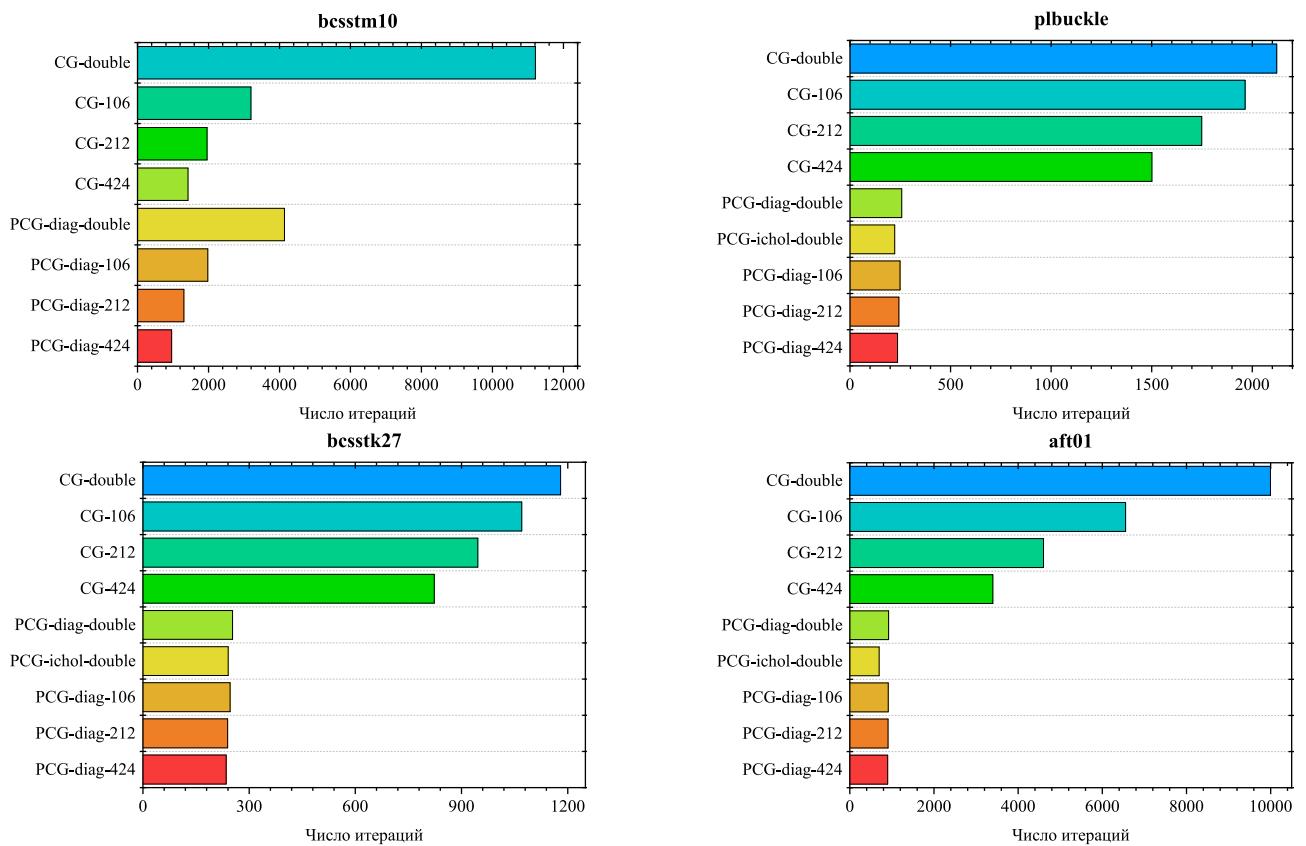


Рис. 2. Число итераций различных реализаций CG и PCG решателей

Таблица 1

Используемые функции многократной точности из библиотеки MPRES-BLAS

Функция	Описание	
<code>mp_spmv_csr</code>	Разреженное матрично-векторное умножение (SpMV)	Ненулевые элементы матрицы представлены в формате двойной точности, входной и выходной векторы — в формате многократной точности. Каждая строка матрицы обрабатывается одним потоком GPU.
<code>mp_diff</code>	Вычитание векторов	Каждый поток GPU выполняет вычисление одного элемента вектора-разности. Операция полностью параллельна.
<code>mp_norm2</code>	Евклидова норма вектора	Сумма квадратов элементов вектора вычисляется на GPU в арифметике многократной точности с использованием двухэтапного алгоритма суммирования. Вычисленная сумма далее копируется в RAM, и квадратный корень вычисляется на CPU с использованием высокоточной библиотеки MPFR. Результатная норма преобразуется в формат двойной точности.
<code>mp_prod_d</code>	Умножение вектора двойной точности на вектор многократной точности	Каждый поток GPU выполняет вычисление одного элемента вектора-произведения. Операция полностью параллельна.
<code>mp_dot</code>	Скалярное произведение векторов	Используется двухэтапный алгоритм на GPU. Первое ядро выполняет вычисления с множеством блоков потоков, сохраняя промежуточные результаты в глобальную память. Второе ядро использует один блок потоков для суммирования промежуточных результатов.
<code>mp_div</code>	Деление двух чисел с плавающей точкой	Операторы копируются в RAM, выполняется деление с использованием высокоточной библиотеки MPFR, результат копируется обратно в память GPU.
<code>mp_axpy</code>	Масштабированное сложение векторов	Каждый поток GPU выполняет вычисление одного элемента результата вектора многократной точности. Операция полностью параллельна.

Таблица 2

Рассматриваемые реализации итерационных решателей

Обозначение	Описание
<code>CG-double</code>	Двойная точность (double), без предобуславливания (GPU)
<code>CG-106</code>	Точность 106 бит, без предобуславливания (GPU)
<code>CG-212</code>	Точность 212 бит, без предобуславливания (GPU)
<code>CG-424</code>	Точность 424 бит, без предобуславливания (GPU)
<code>PCG-diag-double</code>	Двойная точность (double), диагональное предобуславливание (GPU)
<code>PCG-diag-106</code>	Точность 106 бит, диагональное предобуславливание (GPU)
<code>PCG-diag-212</code>	Точность 212 бит, диагональное предобуславливание (GPU)
<code>PCG-diag-424</code>	Точность 424 бит, диагональное предобуславливание (GPU)
<code>PCG-ichol-double</code>	Двойная точность (double), неполное разложение Холецкого (MATLAB pcg)

Таблица 3  
Разреженные матрицы для экспериментов

Матрица	Строк	Ненулевых	Приложение
bcsstm10	1 086	22 092	Динамический анализ в проектировании конструкций
plbuckle	1 282	30 644	Структурная механика
bcsstk27	1 224	56 126	Динамический анализ в проектировании конструкций
aft01	8 205	125 567	Моделирование акустического излучения

**3. Эксперименты.** Мы оценили влияние точности на скорость сходимости итерационных CG и PCG решателей. В таблице 2 перечислены рассмотренные реализации.

Эксперименты проводились на системе, оснащенной NVIDIA RTX 2080 GPU (46 потоковых мультипроцессоров, 8 Гб памяти GDDR6, аппаратная совместимость 7.5), Intel Core i5 7500 (3.40 GHz), 32 Гб памяти DDR4. Программное обеспечение: Ubuntu 20.04.2 LTS, CUDA Toolkit 11.1 / NVIDIA Driver 455.32.00.

Для экспериментов использовались симметричные положительно определенные тестовые матрицы из коллекции SuiteSparse (<https://sparse.tamu.edu/>). Описание матриц представлено в таблице 3.

Параметры эксперимента:

- Начальное приближение:  $x_0 = (0, 0, 0, \dots, 0)^T$ ;
- Правая часть:  $b = (1, 1, 1, \dots, 1)^T$ ;
- Точность:  $tol = 10^{-8}$ ;
- Максимальное число итераций: 15 000.

Результаты экспериментов представлены на рис. 2. Мы видим, что для решателей без предобуславливания наблюдается ярко выраженная зависимость числа итераций от точности. Например, для матрицы bcsstm10 решателю двойной точности CG-double потребовалось 11 210 итераций, в то время как решателю CG-424 потребовалось только 1 420 итераций. Таким образом, увеличение точности в 8 раз привело к пропорциональному снижению числа итераций, необходимых для решения системы с заданной точностью. Для решателей с диагональным предобуславливанием наиболее существенное влияние точности на скорость сходимости наблюдается также в случае матрицы bcsstm10, а для других матриц оно выражено в меньшей степени. В целом эффективность диагонального предобуславливания сравнима с эффективностью предобуславливателя Холецкого для всех рассмотренных примеров.

Для матрицы aft01 фактическая погрешность решения ( $\|b - Ax\|/\|b\|$ ), полученного с использованием стандартной арифметики двойной точности (CG-double), составила  $1.02 \times 10^{-8}$ , и таким образом оказалась выше заданного предела  $tol = 10^{-8}$ . Это вызвано ошибками округления, которые накапливаются при вычислении невязки  $r = r - \alpha q$  на каждой итерации метода. В свою очередь, в реализациях многократной точности фактическая ошибка решения находится в допустимых пределах для всех тестовых случаев. Таким образом, использование арифметики многократной точности не только улучшает сходимость итерационных методов, но также повышает их надежность.

**Заключение.** В работе рассмотрены итерационные CG и PCG решатели многократной точности для графических процессоров видеокарт, являющиеся частью библиотеки высокоточных подпрограмм линейной алгебры MPRES-BLAS. Результаты экспериментов показывают, что влияние точности арифметики на скорость сходимости итерационных методов зависит от решаемой задачи и может быть весьма существенным. Кроме этого, при повышенной точности улучшается надежность решателей.

Важную роль играет подбор оптимального предобуславливателя. Для рассмотренных тестовых матриц неполное разложение Холецкого обеспечивает наименьшее число итераций (за исключением матрицы `bcsstm10`). Вместе с тем, его построение и применение включает существенную часть последовательных вычислений, и поэтому реализация на параллельных системах оказывается затруднительной. В этом отношении мы полагаем, что использование арифметики многократной точности совместно с „мягким“ параллельным предобуславливателем, таким как диагональный предобуславливатель, может оказаться более эффективным подходом для некоторых классов задач.

## Список литературы

1. Saad Y. Iterative Methods for Sparse Linear Systems, 2nd ed. Philadelphia: SIAM, 2003. 547 p.
2. Barrett R. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods (2nd Edition) / R. Barrett et al. Philadelphia: SIAM, 1994. 107 p.
3. Saito T., Ishiwata E., Hasegawa H. Analysis of the GCR method with mixed precision arithmetic using QuPAT // Journal of Computational Science. 2012. Vol. 3, N 3. P. 87–91. DOI: 10.1016/j.jocs.2011.05.001.
4. Cools S., Yetkin E.F., Agullo E., Giraud L., Vanroose W. Analyzing the Effect of Local Rounding Error Propagation on the Maximal Attainable Accuracy of the Pipelined Conjugate Gradient Method // SIAM Journal on Matrix Analysis and Applications. 2018. Vol. 39, N 1. P. 426–450. DOI: 10.1137/17M1117872.
5. Kouya T. A highly efficient implementation of multiple precision sparse matrix-vector multiplication and its application to product-type Krylov subspace methods // International Journal of Numerical Methods and Applications. 2012. Vol. 7, N 2. P. 107–119.
6. Hishinuma T., Hasegawa H., Tanaka T. SIMD Parallel Sparse Matrix-Vector and Transposed-Matrix-Vector Multiplication in DD Precision // Lecture Notes in Computer Science. 2017. Vol. 10150. P. 21–34.
7. Hasegawa H. Utilizing the quadruple-precision floating-point arithmetic operation for the Krylov subspace methods. 2003. [Electron. Res.]: <http://www.slis.tsukuba.ac.jp/~hasegawa.hidehiko/ga/GYSEKI/hasegaw.pdf> (accessed 17 January 2022).
8. Masui K. Ogino M. Research on the Convergence of Iterative Method Using Mixed Precision Calculation Solving Complex Symmetric Linear Equation // IEEE Transactions on Magnetics. 2020. Vol. 56, N 1. Article no. 7503604. DOI: 10.1109/TMAG.2019.2951280.
9. Mukunoki D., Takahashi D. Using Quadruple Precision Arithmetic to Accelerate Krylov Subspace Methods on GPUs // Lecture Notes in Computer Science. 2014. Vol. 8384. P. 632–642. DOI: 10.1007/978-3-642-55224-3\_59.
10. Isupov K., Knyazkov V. Multiple-Precision BLAS Library for Graphics Processing Units // Communications in Computer and Information Science. 2020. Vol. 1331. P. 37–49. DOI: 10.1007/978-3-030-64616-5\_4.
11. Omondi A., Premkumar B. Residue Number Systems: Theory and Implementation. Imperial College Press, 2007. 402 p.

**Исупов Константин Сергеевич** — канд. техн. наук, доцент кафедры электронных вычислительных машин Вятского государственного университета, e-mail: [ks\\_isupov@vyatsu.ru](mailto:ks_isupov@vyatsu.ru). Область профессиональных интересов: вычисления многократной точности, система остаточ-

ных классов, компьютерная арифметика, параллельные алгоритмы, программирование для графических процессоров, CUDA. Является автором и соавтором более 30 научных работ в указанных областях.



**Konstantin Isupov** — PhD in Computer Science, assistant professor at the Department of Electronic Computing Machines, Vyatka State University, e-mail: [ks\\_isupov@vyatsu.ru](mailto:ks_isupov@vyatsu.ru). His research interests include high-precision computation, residue number system, computer arithmetic, parallel algorithms, GPU programming, CUDA. He has authored or coauthored more than 30 papers in these areas.



**Князьков Владимир Сергеевич** — д-р техн. наук, профессор, главный научный сотрудник научно-исследовательского института фундаментальных и прикладных исследований Пензенского государственного университета, e-mail: [kniakov@pnzgu.ru](mailto:kniakov@pnzgu.ru).

Является автором более 50 свидетельств на изобретения и патентов, опубликовал более 100 научных работ в области параллельных архитектур, высокопроизводительных вычислений и реконфигурируемых вычислительных систем.

**Vladimir Knyazkov** — Dr. Sci. in Computer Sciences and Engineering, principal research

scientist at the Research Institute of Fundamental and Applied Studies, Penza State University, e-mail: [kniakov@pnzgu.ru](mailto:kniakov@pnzgu.ru). He is the author of 50 invention certificates and patents, and published over 100 papers in the fields of parallel architectures, high-performance computing and reconfigurable computing systems.



**Коржавина Анастасия Сергеевна** — канд. техн. наук, доцент кафедры электронных вычислительных машин Вятского государственного университета, e-mail: [as\\_korzhavina@vyatsu.ru](mailto:as_korzhavina@vyatsu.ru). Область научных интересов: высокоточные вычисления, компьютерная арифметика, реконфигурируемые вычислительные системы. Является автором и соавтором более 20 научных работ. **Контактный номер телефона:** +7(953)674-39-02

**Anastasia Korzhavina** — PhD in Computer Science, assistant professor at the Department of Electronic Computing Machines, Vyatka State University, e-mail: [as\\_korzhavina@vyatsu.ru](mailto:as_korzhavina@vyatsu.ru). Her research interests include high-precision computation, computer arithmetic, reconfigurable computer systems. She has authored or coauthored more than 20 papers.

*Дата поступления — 01.02.2022*