ЧИСЛЕННЫЕ МЕТОДЫ И АЛГОРИТМЫ

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ РАНДОМИЗИРОВАННОГО РЕГУЛЯРИЗОВАННОГО АЛГОРИТМА КАЧМАЖА

А.И. Жданов [1], Ю.В. Сидоров [2]
[1] Самарский государственный технический университет (СамГТУ), Самара, Россия,
[2] ЧОУ ВО Самарский институт управления, Самара, Россия

Аннотация

В статье рассмотрена параллельная реализация рандомизированного регуляризованного алгоритма Качмажа. Приведён пример использования параллельной рандомизированной версии алгоритма для решения интегрального уравнения Фредгольма первого рода с возмущённой правой частью, и показано, что в этом случае удаётся ускорить вычисления до 4 раз по сравнению с последовательной рандомизированной версией.

Ключевые слова: итерационные методы, регуляризованные решения, параллельные вычисления, обработка сигналов.

Цитирование: **Жданов, А.И.** Параллельная реализация рандомизированного регуляризированного алгоритма Качмажа / А.И. Жданов, Ю.В. Сидоров // Компьютерная оптика. – 2015. – Т. 39, № 4. – С. 536-541. – DOI: 10.18287/0134-2452-2015-39-4-536-541.

Введение

Итерационный алгоритм Качмажа [1] и его модификации [2, 3, 4] применяются при решении различных прикладных задач: томографии [5, 6], обработки сигналов [7], обработки изображений и в адаптивном управлении [8, 9, 10]. Для повышения скорости сходимости алгоритма Качмажа в работе [11] предложен его рандомизированный вариант.

Для решения плохо обусловленных и некорректных задач в работе [12] предложена регуляризованная форма алгоритма Качмажа.

Этот алгоритм основан на применении алгоритма Качмажа к системе расширенных регуляризованных нормальных уравнений [12, 13]. Регуляризированная форма алгоритма Качмажа [12] успешно применяется при решении прикладных задач [13,14].

Системы уравнений, возникающие при решении многих прикладных задач, часто имеют большую или сверхбольшую размерность. Для решения таких задач эффективно применение проекционного алгоритма Качмажа и высокопроизводительной вычислительной техники и современных параллельных технологий [15], [16] – [18]. Для классического проекционного алгоритма Качмажа существуют его параллельные реализации, которые ускоряют процесс решения прикладных задач [19, 20].

В настоящее время для регуляризованной формы алгоритма Качмажа, который используется для решения некорректных и плохо обусловленных линейных систем большой размерности, отсутствуют параллельные реализации.

Предлагаемая работа посвящена разработке и использованию параллельной реализации рандомизированной регуляризованной формы алгоритма Качмажа [13].

1. Регуляризованный алгоритм Качмажа

Рассмотрим задачу регуляризации А. Н. Тихонова

$$\min_{u\in\mathbb{R}^n}\left\{\left\|Au-f\right\|^2+\alpha\left\|u\right\|^2\right\}, \quad (1)$$

где $A\in\mathbb{R}^{m\times n}$, $u\in\mathbb{R}^n$, $f\in\mathbb{R}^m$, $(m\geq n)$, α – параметр регуляризации, $\|\cdot\|=\|\cdot\|_2$ – евклидова векторная норма.

Регуляризованный алгоритм Качмажа [13] для решения задачи (1) имеет вид

$$\theta_k=\theta_{k+1}-\rho_{k-1}\begin{pmatrix}a_{j(k)}\\ e_{j(k)}\end{pmatrix},\ \rho_{k-1}=\frac{\left\langle\left(a^T_{j(k)},-\alpha e^T_{j(k)}\right),\theta_{k-1}\right\rangle}{\left\|a_{j(k)}\right\|^2+\alpha}, \quad (2)$$

где $\theta_k=\left(r_k^T,u_k^T\right)^T$, $r=f-Au$,

$A=\left(a_1,\ \ a_2,\ \ \ldots,\ \ a_n\right), a_i\in\mathbb{R}^m$,

$\left(e_1,\ \ e_2\ \ \ldots,\ \ e_n\right)=I_n$, $\langle\bullet,\bullet\rangle$ – скалярное произведение, последовательность $j(k)=(k \bmod n)+1$, $k=1,2,\ldots,$ т.е. $\left\{j(k)\right\}_1^\infty$ – периодическая последовательность вида $1,2,\ldots n,1,2,\ldots n,\ldots$.

Если начальный вектор θ_0 удовлетворяет условию

$$\left(\sqrt{\alpha}I_m \vdots A\right)\theta_0=f,$$

где $\left(\sqrt{\alpha}I_m \vdots A\right)$ – блочная матрица, то, как показано в [13], итерационный процесс сходится к решению задачи (1).

Для повышения скорости сходимости алгоритма (2) используется рандомизированная форма алгоритма

$$\theta_k=\theta_{k+1}-\rho_{k-1}\begin{pmatrix}a_{i(k)}\\ e_{i(k)}\end{pmatrix},\ \rho_{k-1}=\frac{\left\langle\left(a^T_{i(k)},-\alpha e^T_{i(k)}\right),\theta_{k-1}\right\rangle}{\left\|a_{i(k)}\right\|^2+\alpha}, \quad (3)$$

здесь последовательность $i(k)$ – выбирается случайным образом из множества столбцов $\{1,2,\ldots,n\}$ пропорционально вероятностям $\|a_{i(k)}\|^2$ в соответствии с [11].

Применение рандомизированной формы алгоритма (3) при решении определённого класса задач [21] позволяет значительно уменьшить число итераций [13].

Далее будем рассматривать рандомизированную форму алгоритма (3).

Итерационный процесс (3) можно представить в следующем виде:

Алгоритм 1

$A \in \mathbb{R}^{m\times n},\ f \in \mathbb{R}^m, u \in \mathbb{R}^n, u_0 = \{0,\dots,0\}'$

$r_0 = f - A\cdot u_0, k = 1, \theta_0 = \left(r_0^T, u_0^T\right)^T$, α

while $\|\theta_k - \theta_{k-1}\| < eps$ *do*

$j = i(k)$

$ro = \left\langle \left(a_j^T, -\alpha\cdot e_j^T\right), \theta_{k-1}\right\rangle$

$$ro = \frac{ro}{\|a_j\|^2 + \alpha}$$

$$\theta_k = \theta_{k-1} - ro\cdot\begin{pmatrix} a_j \\ -e_j \end{pmatrix}$$

$k = k + 1$

end while

Из анализа алгоритма 1 можно сделать вывод, что при использовании компьютерных библиотек, выполняющих основные операции линейной алгебры, рассматриваемый алгоритм сводится к следующим стандартным операциям (BLAS Level 1) – евклидова норма $nrm2$, скалярное произведение dot и умножение вектора на скаляр и добавление результата к вектору $saxpy$.

Далее будет рассмотрена параллельная реализация рандомизированной регуляризованной формы алгоритма Качмажа.

2. Параллельная реализация рандомизированной регуляризованной формы алгоритма Качмажа

При создании параллельной версии рандомизированной регуляризованной формы алгоритма Качмажа (3) упор делался на ускорение наиболее трудоёмких операций алгоритма.

Основными операциями, требующими ускорения, являются:

- скалярное произведение;
- умножение вектора на скаляр и добавление результата к вектору;
- вычисление евклидовой нормы вектора.

В качестве средства разработки была выбрана оптимизированная математическая библиотека, ориентированная на многоядерные системы с общей памятью Intel® Math Kernel Library (Intel® MKL)[22], как многократно зарекомендовавшая себя при решении аналогичных задач [23].

Библиотека Intel® MKL содержит все необходимые функции, реализующие перечисленные выше операции – функции BLAS Level 1.

При использовании библиотеки Intel® MKL параллельную программу можно разделить на следующие этапы.

1. Распределение памяти, загрузка исходных данных, уточнение параметров параллелизации.
2. Построение функции распределения вероятностей столбцов.
3. Генерация дискретной случайной величины, выбор номера столбца.
4. Вычисление θ_k.
5. Сохранение результатов.

Рассмотрим подробно 1-й, 2-й и 3-й этапы программной реализации.

Исходные данные задачи формулируются следующим образом:

1. $A_1 = \begin{pmatrix} A \\ -I_n \end{pmatrix} \in \mathbb{R}^{(m+n)\times n}$.
2. $A_2 = \begin{pmatrix} A \\ -\alpha\cdot I_n \end{pmatrix} \in \mathbb{R}^{(m+n)\times n}$.
3. $theta = \begin{pmatrix} r_0 \\ u_0 \end{pmatrix} \in \mathbb{R}^{(m+n)}$.

Матрицы A_1 и A_2 рассматриваются как разреженные матрицы и хранятся в следующем формате:

- массив $vals(nnz)$ содержит ненулевые элементы матрицы по столбцам, nnz – число ненулевых элементов матрицы;
- массив $indx(nnz)$ содержит номера строк, соответствующих элементам $vals$ – $0,1,\dots,(m+n)$;
- массив $pos(n)$ содержит указатели на начало столбцов матрицы;
- массив $len(n)$ содержит число ненулевых элементов матрицы в каждом столбце.

При данном подходе хранения матрицам A_1 и A_2 соответствуют следующие массивы:

- для матрицы A_1 – $vals\_1(nnz)$, $indx(nnz)$, $pos(n)$, $len(n)$;
- для матрицы A_2 – $vals\_2(nnz)$, $indx(nnz)$, $pos(n)$, $len(n)$.

Предложенный формат хранения разреженных матриц A_1 и A_2 позволяет использовать функции Sparse BLAS Level 1 библиотеки Intel® MKL.

Для построения функции распределения вероятностей и выбора номера столбцов $i(k)$ использовались параллельные версии функции стандартной библиотеки C++ (libstdc++ parallel mode). Генерация дискретной случайной величины осуществлялась с помощью стандартных функций MKL.

Для эффективного использования параллельных версий функций стандартной библиотеки C++ и

Intel® MKL, с помощью функций omp_set_dynamic и mkl_set_dynamic отключена динамическая корректировка числа потоков, максимальное число потоков задаётся для всех функций с помощью omp_set_num_threads(max_threads).

Программную реализацию рандомизированной регуляризованной формы алгоритма Качмажа (3) формально можно представить в следующем виде:

Алгоритм 2

$void\ cumul(\ )\{$

$\_\_\mathrm{gnu\_parallel::partial\_sum}(\ )$

$\}$

$\mathrm{int}\ main()\{$

$/*$ *уточнение параметров параллелизации* $*/$

$\mathrm{max\_}threads = mkl\_get\_\mathrm{max\_threads}();$

omp_set_dynamic(false);

omp_set_dynamic(0);

omp_set_num_threads(max_threads);

$/*$построение

функции распределения

вероятностей столбцов$*/$

$cumul();$

$vslNewStream();$

$\mathrm{while\, cblas\_nrm2()} < \mathrm{eps\, do}$

$vdRngUniform();$

$/*$выбор номера столбца$*/$

$j = \_\_\mathrm{gnu\_parallel::find\_if}();$

$/*$вычисление $ro = \left\langle \left(a_j^T, -\alpha \cdot e_j^T \right), \theta_{k-1} \right\rangle */$

$cblas\_ddoti(vals\_2, theta);$

$/*$вычисление $\theta_k = \theta_{k-1} - ro \cdot \begin{pmatrix} a_j \\ -e_j \end{pmatrix} */$

$cblas\_daxpyi(vals\_1, theta);$

$end\ while$

3. Вычислительный эксперимент

Вычислительный эксперимент проводился на доступной рядовому пользователю конфигурации:

- процессор: Intel® Core i7-3770 3.4 ГГц (4 ядра);
- материнская плата : ASUSTeK P8Z77-V LX;
- оперативная память 32 Гб (non ECC);
- операционная система : Fedora 19;
- Intel MKL 11.3;
- компилятор Intel® C/C++.

Все вычисления проводились с двойной точностью.

В качестве тестовой задачи рассматривалась дискретизация интегрального уравнения Фредгольма первого рода из пакета программ Regularization Tools [24] (Phillips' "famous" problem) – подобные задачи возникают в томографии, восстановлении изображений, при решении обратной задачи рассеяния [25]. Пакет программного обеспечения Regularization Tools состоит из набора функций MATLAB для анализа и решения дискретных некорректных задач. Алгоритм, рассматриваемый в данной статье, не реализован в [24].

Задача решения интегрального уравнения сводится к решению системы линейных алгебраических уравнений, полученной в результате дискретизации соответствующего интегрального уравнения

$$Au = \tilde{f},\ A \in \mathbb{R}^{n\times n}, u \in \mathbb{R}^n,\ \tilde{f} = \left(f + \varepsilon \right) \in \mathbb{R}^n. \qquad (4)$$

Здесь f – точная правая часть, ε – гауссовский шум $\left(N\left(0,\sigma^2\right) \right)$. Параметр регуляризации выбирался в соответствии с методикой, предложенной в [13]:

$$\alpha = \frac{\delta \cdot \sigma_{\max}^2(A)}{\left\| \tilde{f} \right\| + \delta},\ \delta = \frac{1}{n} \sum_{i=1}^{n} \left\| f - \tilde{f} \right\|^2,$$

где $\sigma_{\max}^2(A)$ – максимальное сингулярное значение матрицы A. Уровень шума ε задавался в пределах от 0,015 до 0,02.

Так как функции BLAS первого уровня библиотеки Intel® MKL не параллелятся для векторов размерностью меньше 1000 [22], размерность матрицы A задачи (4) выбиралась в пределах от 1024 до 40960.

Вычислительный эксперимент проводился для последовательного варианта рандомизированной регуляризованной формы алгоритма Качмажа и параллельной версии (число потоков max_threads = 4).

Сравнение времени выполнения версий алгоритма для различных размеров задачи представлено в табл. 1 (критерий остановки $\left\| \theta_{k+1} - \theta_k \right\| \le 10^{-8}$, время выполнения находилось как среднее из 100 запусков).

Табл. 1. Время выполнения рандомизированного регуляризованного алгоритма Качмажа для задачи Филлипса [21]

	Var_1	Var_2	Ускорение
m	t, сек	t, сек	
1024	0,13	0,068	1,91
2048	0,48	0,19	2,41
5120	2,60	0,91	2,85
10240	10,32	3,15	3,27
15360	23,18	6,51	3,56
20480	45,19	11,74	3,85
25600	62,45	15,99	3,90
40960	175,21	43,80	4,00

В табл. 1 *Var*_1 – последовательный вариант, *Var*_2 – параллельный вариант $\left(\mathrm{max\_}threads = 4 \right)$, $m=n$ – размер задачи, t – время выполнения итераций.

Параллельный вариант рандомизированной регуляризованной формы алгоритма Качмажа позволяет снизить время выполнения от 2 до 4 раз по сравнению с последовательным вариантом.

Заключение

В работе представлена параллельная реализация рандомизированного варианта регуляризированного алгоритма Качмажа [13]. Данная реализация позволяет снизить время решения задач типа (4) от 2 до 4 раз по сравнению с последовательной версией рандомизированного алгоритма.

Как и в работе [20], использование современных многоядерных (многопроцессорных) систем и соответствующих им параллельных библиотек (например, Intel® Math Kernel Library) позволяют ускорить вычисления для регуляризированного алгоритма Качмажа (3).

Дальнейшие работы планируется проводить как в направлении оптимизации разработанной программы, так и в реализации рассматриваемого алгоритма на GPU с помощью технологии NVIDIA CUDA® [26].

Благодарности

Работа выполнена при поддержке гранта РФФИ «ОФИ-М» 13-01-12014.

Литература

1. **Kaczmarz, S.** Angenäherte Auflösung von Systemen linearer Gleichungen / S. Kaczmarz // Bulletin International de l'Académie Polonaise des Sciences et des Lettres. – 1937. – Vol. 35. – P. 355-257.
2. **Zouzias, A.** Randomized Extended Kaczmarz for Solving Least Squares / A. Zouzias, N.M. Freris // SIAM Journal on Matrix Analysis and Applications. – 2013. – Vol. 34, N 2. – P. 773-793. – ISSN 0895-4798.
3. **Thoppe, G.A.** Stochastic Kaczmarz algorithm for network tomography / G.A. Thoppe, V. Borkar, D. Manjunath // Automatica. – 2014. – Vol. 50, N 3. – P. 910-914.
4. **Needell, D.** Randomized Kaczmarz solver for noisy linear systems / D. Needell // BIT Numerical Mathematics. – 2010. – Vol. 50, N 2. – P. 395-403. – ISSN 0005-1098.
5. **Popa, C.** Kaczmarz extended algorithm for tomographic image reconstruction from limited-data / C. Popa, R. Zdunek // Mathematics and Computers in Simulation. – 2004. – Vol. 65, N 6. – P. 579-598. – ISSN: 0378-4754
6. **Natterer, F.** The Mathematics of Computerized Tomography / F. Natterer. – Society for Industrial and Applied Mathematics, 2001. – 226 p.
7. **Feichtinger, H.G.** New Variants of the POCS Method using Affine Subspaces of Finite Codimension with Applications to Irregular Sampling / H.G. Feichtinger, C. Cenker, M. Mayer, H. Steier, T. Strohmer [Электронный ресурс]. – URL: https://www.math.ucdavis.edu/~strohmer/papers/1992/cfm1292.pdf (дата обращения 05.07.2015).
8. **Åström, K.** Theory and applications of adaptive control – A survey / K. Åström // Automatica. – 1983. – Vol. 19, N 5. – P. 471-486. – ISSN 0005-1098.
9. **Parks, P.C.** A comparison of five algorithms for the training of CMAC memories for learning control systems / P.C. Parks, J. Militzer // Automatica. – 1992. – Vol. 28, N 5. – P. 1027-1035. – ISSN 0005-1098.
10. **Richalet, J.** Model predictive heuristic control: Applications to industrial processes / J. Richalet, A. Rault, J.L. Testud, J. Papon // Automatica. – 1978. – Vol. 14, N 5. – P. 413-428. – ISSN 0005-1098.
11. **Strohmer, T.** A Randomized Kaczmarz Algorithm with Exponential Convergence / T. Strohmer, R. Vershynin // Journal of Fourier Analysis and Applications. – 2009. – Vol. 15, N 2. – P. 262-278. – ISSN 1069-5869.
12. **Жданов, А.И.** Проекционный регуляризирующий алгоритм для решения некорректных линейных алгебраических систем большой размерности / А.И. Жданов, А.А. Иванов // Вестник Самарского государственного технического университетата. Серия Физ.-мат. науки. – 2010. – Т. 5, № 21. – С. 309-312.
13. **Ivanov, A.A.** Kaczmarz Algorithm For Tikhonov Regularization Problem / A.A. Ivanov, A.I. Zhdanov // Applied Mathematics. E-Notes. – 2013. – Vol. 13. – P. 270-276.
14. **Sinha, L.** Quantification of the binding potential of cell-surface receptors in fresh excised specimens via dual-probe modeling of SERS nanoparticles / L. Sinha, Y. Wang, C. Yang, Al. Khan, J.G. Brankov, JT.C. Liu, K.M. Tichauer [Электронный ресурс]. – 2015. – URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4341215/ (дата обращения 05.07.2015).
15. **Ортега, Дж.М.** Введение в параллельные и векторные методы решения линейных систем / Дж. Ортега; пер. с анг. – М.: Мир, 1991. – 367 с. (J.M. Ortega. Introduction to Parallel and Vector Solution of Linear Systems. – N.Y.: Plenum Press, 1988).
16. **Алексеев, В.А.** Векторизация метода распространяющегося пучка и его реализация по технологии cuda / В.А. Алексеев, Д.Л. Головашкин // Компьютерная оптика. – 2010. – Т. 34, № 2. – С. 225-230.
17. **Якимов, П.Ю.** Предварительная обработка цифровых изображений в системах локализации и распознавания дорожных знаков / П.Ю. Якимов // Компьютерная оптика. – 2013. – Т. 37, № 3. – С. 401-405.
18. **Изотов, П.Ю.** Технология реализации нейросетевого алгоритма в среде cuda на примере распознавания рукописных цифр / П.Ю. Изотов, С.В. Суханов, Д.Л. Головашкин // Компьютерная оптика. – 2010. – Т. 34, № 2. – С. 243-251.
19. **Elble, J.M.** GPU computing with Kaczmarz's and other iterative algorithms for linear systems / J.M. Elble, N.V. Sahinidis, Panagiotis Vouzisb [Электронный ресурс]. – 2010. – URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2879082/ (дата обращения 05.07.2015).
20. **Liu, Ji**. An Asynchronous Parallel Randomized Kaczmarz Algorithm / Ji. Liu, S.J. Wright, S. Sridhar [Электронный ресурс]. – 2014. – URL: http://arxiv.org/pdf/1401.4780v2.pdf (дата обращения 05.07.2015).
21. **Phillips, D.L.** A technique for the numerical solution of certain integral equations of the first kind / D.L. Phillips // Journal of the ACM. – 1962. – Vol. 9, Issue 1. – P. 84-97.
22. Intel® MKL [Электронный ресурс]. – URL: https://software.intel.com/en-us/intel-mkl (дата обращения 05.07.2015).
23. **Ильин, В.П.** О вопросах распараллеливания крыловских итерационных методов / В.П. Ильин // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. – 2013. – Т. 2, № 3. – С. 48-62.
24. Regularization Tools Version 4.0 for Matlab 7.3 / P.Ch. Hansen // Numerical Algorithms. – 2007. – Vol. 46, Issue 2. – P. 189-194.
25. **Старк, Г.** Реконструкция изображений/ Г. Старк; пер. с англ. – М.: Мир, 1992. – 336 с. (H. Stark, ed. Image Recovery, Theory and Application . – N.Y.: Academic Press, 1987).
26. NVIDIA CUDA® [Электронный ресурс]. – URL: http://www.nvidia.ru/object/cuda-parallelcomputing-ru.html (дата обращения 05.07.2015).

References

[1] Kaczmarz S. Angenäherte Auflösung von Systemen linearer Gleichungen. Bulletin International de l'Académie Polonaise des Sciences et des Lettres 1937; 35: 355-57.

[2] Zouzias A, Freris NM. Randomized Extended Kaczmarz for Solving Least Squares. SIAM Journal on Matrix Analysis and Applications 2013; 34(2): 773-93.
[3] Thoppe G, Borkar V, Manjunath D. A stochastic Kaczmarz algorithm for network tomography. Automatica 2014; 50(3): 910-14.
[4] Needell D. Randomized Kaczmarz solver for noisy linear systems. BIT Numerical Mathematics 2010; 50(2): 395-403.
[5] Popa C, Zdunek R. Kaczmarz extended algorithm for tomographic image reconstruction from limited-data. Mathematics and Computers in Simulation 2004; 65(6): 579-98.
[6] Natterer F. The Mathematics of Computerized Tomography. Society for Industrial and Applied Mathematics;2001.
[7] Feichtinger HG, Cenker C, Mayer M, Steier H, Strohmer T. New Variants of the POCS Method using Affine Subspaces of Finite Codimension with Applications to Irregular Sampling. Source: ⟨https://www.math.ucdavis.edu/~strohmer/papers/1992/cfm1292.pdf⟩.
[8] Åström K. Theory and applications of adaptive control–A survey. Automatica 1983; 19(5): 471-86.
[9] Parks PC., Militzer J. Model predictive heuristic control: Applications to industrial processes. Automatica 1978; 14(5): 413-28.
[10] Richalet J, Rault A, Testud JL, Papon J. Model predictive heuristic control: Applications to industrial processes. Automatica 1992; 28(5): 1027-103.
[11] Strohmer T, Vershynin R. A Randomized Kaczmarz Algorithm with Exponential Convergence. Journal of Fourier Analysis and Applications 2009; 15(2): 262-78.
[12] Zhdanov AI, Ivanov AA. Projection regularization algorithm for solving linear algebraic system of large dimension. “Vestnic Samarskogo gosudarstvennogo tehnicheskogo Universiteta” Publisher 2010; 5(21): 309-12.
[13] Ivanov AA, Zhdanov AI. Kaczmarz Algorithm For Tikhonov Regularization. Problem. Appl. Math. E-Notes 2013; 13: 270-6.
[14] Sinha L, Wang Yu, Yang C, Khan Al, Brankov JG, Jonathan T, Liu C, Tichauer K.M. Quantification of the binding potential of cell-surface receptors in fresh excised specimens via dual-probe modeling of SERS nanoparticles. Source: ⟨http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4341215/⟩
[15] Ortega JM. Introduction to Parallel and Vector Solution of Linear Systems. New York: Plenum Press; 1988.
[16] Alekseev PY, Golovashkin DL. Vectoring propagating beam method and its realization technology cuda [In Russian]. Computer Optics 2010; 34(2): 225-230.
[17] Yakimov PY. Pre-processing of digital images in the systems of localization and recognition of road signs. Computer Optics 2013; 37(3): 401-405.
[18] Izotov PY, Sukhanov SV, Golovashkin DL. Technology implementation of neural network algorithm in an environment cuda an example of recognition of handwritten digits [In Russian]. Computer Optics 2010; 34(2): 243-251.
27. Elble JM, Sahinidis NV, Vouzisb P. GPU computing with Kaczmarz’s and other iterative algorithms for linear systems. Source: ⟨http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2879082/⟩.
28. Liu Ji, Wright SJ, Sridhar S. An Asynchronous Parallel Randomized Kaczmarz Algorithm. Source: ⟨http://arxiv.org/pdf/1401.4780v2.pdf⟩.
[19] Phillips DL. A technique for the numerical solution of certain integral equations of the first kind. Journal of the ACM 1962; 9(1): 84-97.
[20] Intel® MKL. Source: ⟨ https://software.intel.com/en-us/intel-mkl⟩
[21] Ilyin VP. On issues of parallelizing Krylov iterative methods. [In Russian]. Bulletin of the South Ural State University. Series: Computational Mathematics and Informatics 2013; 2(3): 48-62.
[22] Hansen PC. Regularization Tools Version 4.0 for Matlab 7.3. Numerical Algorithms 2007. 46(2): 189-194.
[23] H. Stark, ed. Image Recovery, Theory and Application. N.Y.: Academic Press; 1987.
[24] NVIDIA CUDA®. Source: ⟨http://www.nvidia.ru/object/cuda-parallelcomputing-ru.html⟩.

PARALLEL IMPLEMENTATION OF A RANDOMIZED REGULARIZED KACZMARZ'S ALGORITHM

A.I. Zhdanov [1], Y.V. Sidorov [2]
[1] Samara State Technical University (SamSTU),
[2] PIHE the Samara Institute of Management

Abstract

The article describes the parallel implementation of a randomized regularized Kaczmarz's algorithm. By way of illustration, the randomized parallel version of the algorithm is used for solving the Fredholm integral equation of the first kind with a perturbed right-hand side, showing that in this way the computation speed can be increased up to 4 times as compared to the sequential randomized version.

Keywords: iterative methods, regularized solutions, parallel computing, signal processing.
Citation: Zhdanov AI, Sidorov YV. Parallel implementation of a randomized regularized Kaczmarz's algorithm. Computer Optics 2015; 39(4): 536-41. DOI: 10.18287/0134-2452-2015-39-4-536-541.

Сведения об авторах

Жданов Александр Иванович, в 1975 г. окончил факультет автоматики и измерительной техники Куйбышевского политехнического института, в 1983 г. – аспирантуру Института проблем управления АН СССР (г. Москва) и там же защитил кандидатскую диссертацию по специальности «Техническая кибернетика и теория информации», в 1992 г. защитил в Санкт-Петербургском техническом университете диссертацию на соискание учёной степени доктора физико-математических наук. С 1993 г. профессор по кафедре прикладной математики. С 1992 г. по 2012 г. – заведующий кафедрой прикладной математики СГАУ. С декабря 2012 г. по настоящее время – декан факультета дистанционного и дополнительного образования и заведующий кафедрой высшей

математики и прикладной информатики СамГТУ. Научные интересы: вычислительная линейная алгебра, некорректные и плохо обусловленные задачи, анализ данных, математическое моделирование.

E-mail: *zhdanovaleksan@yandex.ru* .

Alexander Ivanovich Zhdanov, in 1975 graduated from Automatic Equipment and Measuring Technique Faculty of Kuibyshev Polytechnical Institute, in 1983 postgraduate study of Institute of Problems of Control of Academy of Sciences of the USSR (Moscow) and in the same place defended the master's thesis in "An engineering cybernetics and the information theory", in 1992 defended at the St. Petersburg Technical University the dissertation on competition of an academic degree of the doctor of physical and mathematical sciences. Since 1993 professor on department of applied Mathematics. From 1992 to 2012 – the head of Applied Mathematics department of SGAU. Since December, 2012 to the present – the dean of distant and additional education faculty and the head of Higher Mathematics and Application-oriented Informatics department of SAMGTU. Scientific interests: computing linear algebra, the incorrect and ill-posed problems, data analysis, mathematical simulation.

Сидоров Юрий Вячеславович, в 2002 году окончил магистратуру Самарской государственной архитектурно-строительной академии (СамГАСА, ныне – Самарский государственный архитектурно-строительной университет – СГАСА), работает старшим преподавателем НОУ ВПО Самарский институт управления. Область научных интересов: вычислительная линейная алгебра, параллельные вычисления, математическое моделирование.

E-mail: *linuxboy2007@gmail.com* .

Yury Vyacheslavovich Sidorov in 2002 graduated from magistracy of Samara State Architectural and Construction Academy (SAMGASA, now – Samara State Architectural and Construction University – SGASA), works as the high teacher of NIHPE the Samara Institute of Management. Area of scientific interests: computing linear algebra, parallel computing, mathematical simulation.

Поступила в редакцию 20 марта 2015 г.
Окончательный вариант – 2 августа 2015 г.