

Концепция и архитектура базы знаний систем линейных алгебраических уравнений

В. П. Ильин

Институт вычислительной математики и математической геофизики СО РАН

Рассматриваются мотивация разработки, концепция, архитектура и технологические аспекты базы знаний систем линейных алгебраических уравнений (СЛАУ), включающей библиотеки алгоритмов, коллекции и генераторы матриц, классифицированную библиографию и глоссарий, документацию и примеры заданий, архивы расчетных данных, средства принятия решений и машинного обучения, а также инструменты поддержки активного жизненного цикла системы.

Ключевые слова: классификация матриц, прямые и итерационные методы, распараллеливание алгоритмов, архивы данных, машинное обучение.

1 Введение

Решение систем линейных алгебраических уравнений (СЛАУ) представляет актуальную проблему вычислительной математики и суперкомпьютерного моделирования по двум основным причинам. Первое - всеобщая распространенность данной расчетной стадии, поскольку при исследовании фундаментальных и прикладных проблем, описываемых системами дифференциальными и/или интегральными уравнений, а также сетевыми и оптимизационными постановками, включая нестационарные и нелинейные задачи, практически неизбежно приходится решать СЛАУ высоких порядков. Вторым аспектом – это большая ресурсоемкость данного вычислительного этапа, поскольку объемы арифметических операций и требуемой памяти растут нелинейно с увеличением размерности задачи, когда число неизвестных доходит до 10^9 - 10^{12} . При этом доля машинного времени может достигать 80 - 90%, а экономические вопросы крупномасштабных машинных экспериментов становятся весьма насущными, см. [1], [2].

Несмотря на свою внешнюю простоту, СЛАУ представляет собой очень содержательный математический объект, определяемый главным образом свойствами своей матрицы, которая может быть вещественной и комплексной, симметричной и несимметричной, вырожденной и невырожденной, плотной и разреженной, и т.д. Еще более богатым является класс методов решения СЛАУ, которые делятся на прямые и итерационные. Среди последних уже более полувека основными являются предобусловленные алгоритмы в подпространствах Крылова. Программные реализации для широких классов алгебраических систем и их решателей представлены в многочисленных общедоступных или коммерческих библиотеках, количество и качество которых растет одновременно с бурным развитием компьютерных платформ.

Особую востребованность здесь приобретают распараллеливание алгоритмов и их отображение на архитектуры многопроцессорных вычислительных систем (МВС) с распределенной и иерархической общей памятью. Математическое и программное обеспечение по решению СЛАУ является одной из продвинутых областей вычислительных наук, а теоретические и экспериментальные исследования в этих направлениях активно продолжают учеными многих стран, о чем свидетельствует стабильный рост научных публикаций и конференций [3]-[5].

Глобализация математического моделирования постоянно приводит к появлению новых типов СЛАУ, зачастую плохо обусловленных, неустойчивых и экстремально ресурсоемких, требующих специальных подходов к их решению с гарантированной точностью. Несмотря на значительный прогресс в данной области, решение алгебраических систем отнюдь не является рутинной областью, а требует глубоких знаний и практического опыта, что неизбежно приводит к довольно узкой специализации математика-программиста, фактически недоступной для широкого круга пользователей с различной профессиональной подготовкой.

В этих условиях, когда исходные данные в реальных условиях могут оказываться неточными, неполными и противоречивыми, машинный эксперимент становится на грани между наукой и искусством, что требует в конечном итоге интеллектуализации вычислительной деятельности, направленной на качественное повышение производительности труда как разработчиков, так и конечных пользователей математического инструментария. Это принуждает к более широкому взгляду на создание программного продукта нового поколения как экосистемы или интегрированного вычислительного окружения (ИВО), содержащего не только реализации алгоритмов, но и весь необходимый багаж знаний. В применении к рассматриваемой нами предметной области это включает, например, коллекции и генераторы матриц для характерных методических и практических задач, примеры формирования исходных данных, архивы расчетных результатов, анализатор свойств матриц и СЛАУ, классифицированная библиография и глоссарий, гибкие средства расширения задач и применяемых алгоритмов, а также адаптации к эволюции компьютерных платформ, использование внешних программных продуктов, инструменты машинного обучения, в том числе выбор оптимального метода в конкретной ситуации, и т.д., см. [6]-[23] и цитируемые там работы.

В п.2 данной статьи мы на содержательном уровне представляем предметную область СЛАУ. Пункт 3 содержит описание общей структуры и функциональности базы знаний (БАЗ), а в заключении обсуждаются дальнейшие планы исследований.

2 Описание предметной области СЛАУ

Рассматриваемая проблема может быть в общем случае записана в комплексной форме:

$$Au = f, \quad A = \{a_{i,j}\} \in \mathbb{C}^{N_r, N_c}, \quad u = \{u_j\} \in \mathbb{C}^{N_c}, \quad f = \{f_i\} \in \mathbb{C}^{N_r}, \quad (1)$$

где N_r и N_c - число строк и столбцов матрицы A соответственно. Наиболее частые на практике СЛАУ, которые мы и будем рассматривать в дальнейшем, имеют квадратные вещественные матрицы $A \in \mathbb{R}^{N,N}$. Классификацию исследуемой предметной области мы рассмотрим кратко по следующим характеристикам, имея целью только обозначить многообразие объектов и операций над ними: свойства алгебраических систем, методы их решения, распараллеливание алгоритмов, программные технологии реализации.

2.1 Классификация и свойства СЛАУ

Актуальные на практике алгебраические системы можно характеризовать по таким признакам, как их происхождение, структурные и спектральные свойства, а также способы представления. Наибольший интерес представляют "большие" СЛАУ с порядками $N \approx 10^9 - 10^{12}$ и более, при этом средними на текущий период развития ЭВМ можно считать $N \approx 10^5 - 10^8$.

Практически все матрицы делятся на два типа – разреженные и плотные. В первом случае число ненулевых элементов мало, т.е. $nz \ll N^2$, и, как правило, их количество n_r в одной строке не зависит от N . Такие системы возникают при сеточных дискретизациях многомерных краевых задач, когда величина n_r составляет единицы или несколько десятков, причем она растет с увеличением порядка аппроксимации. Плотные матрицы возникают при численном решении интегральных уравнений, и в этих случаях величина nz равна или примерно пропорциональна N^2 .

Наиболее богатый и практически важный класс составляют большие разреженные СЛАУ, имеющие разнообразные блочные структуры: ленточные, окаймленные, седловые и др., которые зависят, естественно, от упорядоченности векторных компонент. Для однородных (единообразных) сеточных СЛАУ важное значение имеет конфигурация шаблона, то есть совокупности ненулевых элементов в строке матрицы. При этом можно определить граф матрицы, в котором ее ненулевой элемент $a_{i,j}$ соответствует ребру между i -й и j -й вершинами.

Важнейшими характеристиками матриц являются также их спектральные свойства, определяемые собственными и сингулярными числами, а также соответствующими векторами, в зависимости от которых СЛАУ может иметь или не иметь единственного решения. Актуальной количественной спецификацией невырожденной матрицы служит ее число обусловленности в одной из гильбертовских норм

$$\text{cond}_r(A) = \|A\|_p \|A^{-1}\|_p, \quad p = 1, 2, \infty, \quad (2)$$

определяющее свойство устойчивости численного решения СЛАУ к погрешностям исходных данных задачи или к арифметическим ошибкам округления. С последним вопросом тесно связан способ представления векторно-матричных компонент, где общепринятыми стандартами являются простая

или двойная точность с 32 и 64 битами в машинном слове. Для экстремально плохо обусловленных СЛАУ ($\text{cond}(A) \approx 10^{15}$) может потребоваться повышенная точность, что значительно увеличивает вычислительную сложность, и ее необходимость должна быть обоснована дополнительным априорным или апостериорным анализом.

С технологической точки зрения существенным является формат хранения матрицы. Наиболее сложный случай – когда СЛАУ генерируется из аппроксимации уравнений на неструктурированной сетке, при этом для каждого узла номера его соседей можно указать только перечислением. В таких случаях вынужденно применяются разреженно-сжатые форматы (например, CSR - Compressed Sparse Row), позволяющие хранить только ненулевые матричные элементы, но существенно замедляющие доступ к их значениям.

2.2 Краткий системный анализ методов решения СЛАУ

Существует два различных подхода к решению алгебраических систем – прямые и итерационные алгоритмы. Первые основаны на методах исключения Гаусса (LU-разложения), требующих выполнения $O(N^3)$, $\gamma \leq 3$ арифметических действий и значительных объемов памяти, делающих нецелесообразным их применение для больших матриц. Существующие стратегии оптимизации упорядоченности векторных компонент, выбора ведущих матричных элементов и уточнения численных решений, делают эти методы надежным (робастным) средством решения средних по размеру СЛАУ (имеющиеся исключения только подтверждают это правило).

Для больших разреженных систем безальтернативным является применение итерационных процессов вида

$$B(u^{n+1} - u^n) = r^n \equiv f - Au^n, \quad n = 0, 1, \dots, \quad (3)$$

где число итераций определяется по условию

$$\|r^n\| \leq \varepsilon \|f\|, \quad \varepsilon \ll 1 \quad (4)$$

и зависит от количества обусловленности матричного произведения $B^{-1}A$, где B – предобуславливающая матрица, которая должна экономично обрабатываться и хорошо аппроксимировать исходную матрицу A .

Современные итерационные подходы основаны на итерационных методах в подпространствах Крылова, реализуемых с помощью рекуррентных формул

$$\begin{aligned} u^{n+1} &= u^n + \alpha_n p^n, \quad r^{n+1} = r^n - \alpha_n A p^n, \\ p^{n+1} &= \beta_{1,n}^{-1} r^{n+1} + \sum_{k=0}^n \beta_{n,k} p^{n-k}, \quad n = 0, 1, \dots, \end{aligned} \quad (5)$$

где $\alpha_n, \beta_{n,k}$ и p^n – итерационные параметры и направляющие векторы, вычисляемые из различных ортогональных, вариационных и проекционных условий. Здесь существенно различаются алгоритмы с короткими ($m = 0$) и

длинными ($m = n$) рекурсиями, возникающими при решении симметричных (эрмитовых в комплексном случае) и несимметричных СЛАУ и определяющими соответственно методы сопряженных или полусопряженных направлений (примером последних служит популярный GMRES – обобщенный метод минимальных невязок). Наличие длинных рекурсий при большом числе итераций сильно увеличивает ресурсоемкость алгоритма, и тогда искусственно ограничивают значение m в (5). Это, в свою очередь, замедляет скорость сходимости итераций, что приводит к необходимости дополнительных исследований для поиска "золотой середины".

Имеется универсальный подход с трансформацией Гаусса, связанный с умножением матрицы A на транспонированную к ней A^T . Это приводит к симметризации СЛАУ, но ухудшает ее обусловленность и требует дополнительного матричного умножения на каждой итерации. Существуют также методы решения несимметричных систем, основанные на биортогонализации векторов (примеры - алгоритмы типа BiCGStab или IDR) или bidiagonalизации матриц Голуба-Кахана.

Другие интересные развития крыловских процессов связаны с дополнением подпространств на основе использования апостериорной информации. Такие разработки активно проводятся при многократном решении СЛАУ с различными последовательно определяемыми правыми частями, что является актуальными в нелинейных, нестационарных и оптимизационных задачах математического моделирования.

Для предобуславливания СЛАУ имеются три основные стратегии: приближенные треугольные разложения матриц (ILU - Incomplete LU-factorization), многосеточные алгоритмы (MG - Multi-Grid), асимптотически оптимальные по порядку, т.е. число операций пропорционально количеству неизвестных, и методы декомпозиции областей (DDM – Domain Decomposition Methods), которые являются главным инструментом распараллеливания алгоритмов на МВС с распределенной и общей иерархической памятью. Все итерационные алгоритмы наиболее эффективно реализуются с учётом их «геометрической предыстории», использующей информационные связи между соседними узлами сетки и соответствующие структурные свойства матриц. Однако зачастую задача конструирования решателя формулируется наоборот, в «алгебраическом ключе», когда алгоритм требуется реализовать только с использованием, например, CSR-формата матрицы. На его основе изучить блочную структуру матрицы в принципе можно, но решение этой проблемы требует серьёзного вовлечения искусственного интеллекта, поскольку в идеале вычислительный модуль должен функционировать как «чёрный ящик», без вмешательства человека.

2.3 Стратегии и тактики распараллеливания алгоритмов

Основными количественными показателями качества распараллеливания являются ускорение вычислений и эффективность использования процессоров

$$S_p = T_1/T_p, E_p = S_p/P, \quad T_p = T_p^a + T_p^c \quad (6)$$

где T_p^a и T_p^c суть времена выполнения арифметических операций и коммуникаций при решении задачи на p процессорах. Эти характеристики алгоритмов для конкретных конфигураций МВС можно оценить только очень грубо, и главный инструмент исследований здесь – это экспериментальные исследования со сравнительным анализом быстродействия различных алгоритмов для характерных типов СЛАУ. Фактически оптимизация вычислительных моделей заключается в минимизации обменов, так как передачи данных не только замедляют работу компьютера, но и значительно увеличивают энергозатраты, что существенно сказывается на эксплуатационных расходах.

Одним из основных средств распараллеливания на современных кластерных архитектурах МВС является гибридное программирование путём передачи сообщений между соединёнными шинами, одновременно работающими многоядерными вычислительными узлами с помощью библиотеки MPI (Message Passing Interface), а также проведением многопоточковых расчётов на каждом узле с помощью системы Open MP на общей памяти. Кроме того, можно осуществлять векторизацию операций, используя систему команд AVX, учитывающей свойства иерархической структуры памяти CPU (Central Processor Unit). Один из общих принципов ускорения заключается в совмещении арифметических действий с информационными обменами, что требует тонкого управления вычислительным процессом. Здесь могут быть использованы эффективные вычислительные инструментари, такие как библиотека матрично-векторных операций Sparse Blas со встроенными параллельными режимами исполнения. Существует поговорка, что «наиболее быстрые методы плохо распараллеливаются», которая может быть права в том, что если мы на каком-то вычислительном этапе сократим число арифметических операций, то это приведёт к увеличению относительной доли времени коммуникаций. Заключительный вывод данных качественных рассуждений состоит в том, что оптимизация распараллеливания – это поиск алгоритмов со сбалансированными обменами вычислений и обменов данными.

2.4 Общая структура и функциональность базы знаний СЛАУ

В целом БАЗ для алгебраических систем можно определить как интеллектуальное вычислительно-информационное окружение, призванное обеспечить высокую производительность программного кода, а также его разработчиков и пользователей. Главную целевую функциональность такой инструментальной экосистемы можно определить следующим образом: пусть библиотека содержит представительный набор решателей для широкого класса алгебраических систем, а также архив расчётных данных с анализом и рекомендациями по предыдущим экспериментам и набор методических руководств. Требуется для решения заданной СЛАУ или их набора подобрать конкретный код, обеспечивающий максимальную производительность на применяемой МВС.

Чисто теоретически поставить, а тем более решить такую задачу чрезвычайно сложно. Однако с привлечением большого объема экспериментальных данных в данном случае можно поставить проблему так называемой суррогатной оптимизации с привлечением эмпирических и статистических подходов. Другими словами, здесь применима идеология машинного обучения, в данном случае программно-информационной системы, представляющей собой базу математических знаний. Такая предметная область, как решение СЛАУ, является удобным объектом для построения концепции и методологии БАЗ, поскольку при всей кажущейся простоте алгебраической системы она обладает и актуальностью, и глубоким теоретическим содержанием, и включает технологические проблемы, затрагивающие профессиональные интересы широкого круга пользователей.

Современный уровень научно-технического прогресса порождает глубокие специализации в огромном количестве областей. Инженеру-авиаконструктору, например, невозможно, да и нецелесообразно разбираться в тонкостях решения СЛАУ. То же самое можно сказать и о программисте, который пишет код для моделирования управления самолетом, когда могут возникнуть нештатные алгебраические ситуации. И тут на помощь человеку должен прийти искусственный разум (в форме БАЗ), в котором заложены соответствующие знания и средства помощи в принятии решения.

Под базой знаний СЛАУ можно понимать хранилище всех возможных знаний и инструментов, которые могут оказаться полезными для решения рассматриваемого круга задач. С этой точки зрения в БАЗ включаются следующие разделы: библиотека алгебраических решателей, коллекции и генераторы матриц, архивы расчетных данных, анализатор СЛАУ, классифицированная библиография и глоссарий, пользовательская документация с примерами вычислительных сеансов, маркетинговая информация, различные инструментарии системной поддержки.

Для обеспечения длительного жизненного цикла с успешной широкой востребованностью такого большого проекта он должен удовлетворять следующим техническим требованиям: гибкое расширение состава алгоритмов и типов решаемых задач, адаптация к эволюции компьютерных платформ, переиспользование внешних программных продуктов, согласованное участие различных групп разработчиков, высокая производительность кода, интеллектуальные внешние и внутренние интерфейсы.

Состав решаемых СЛАУ может классифицироваться не только по первичным матричным характеристикам, но и по крупноблочным структурам, связанным с такими актуальными приложениями, как задачи электромагнетизма, прочности, гидрогазодинамики, многофазной фильтрации и т. д.

Набор алгебраических решателей включает как оригинальные реализации, так и возможности использования программ из других библиотек (например, EIGEN, PETSc, HYPRE и др.). Библиография по методам решения СЛАУ насчитывает тысячи наименований с многочисленными теоретическими разделами, в том числе следующие: CHE – чебышевские алгоритмы, KRY – методы в подпространствах Крылова, AMG – алгебраические мно-

госеточные методы, DDM – методы декомпозиции областей, REC – многократное решение СЛАУ, LLU – алгоритмы неполной факторизации, ADI – неявные методы переменных направлений, PRO – проекционные подходы, TRE – решатели на графах, PAR – параллельные алгоритмы, AI – искусственный интеллект, SAD – седловые СЛАУ и т. д. Данная компонента БАЗ совместно с глоссарием является основой онтологии СЛАУ, на которой должны формироваться инструменты машинного обучения разрабатываемой интеллектуальной системы (рука не поднимается написать модное слово "нейросети"), а также средства принятия решений.

3 Заключение. Обсуждение результатов и планов дальнейших исследований

В работе предложены концепция, целевой функционал и общая структура базы знаний СЛАУ. Проведен содержательный анализ рассматриваемой предметной области, состав функционального и системного наполнения БАЗ, технические требования к ее разработке, а также принципы реализации технологической поддержки машинного обучения и системы принятия решений. Планы дальнейших исследований включают построение онтологии и когнитивных инструментариев для создания интеллектуальной базы знаний СЛАУ.

3.1 Список литература

1. Dongarra J., Grigori, L., Higham, N. J. Numerical algorithms for high performance computational science, 2020. V. 378, iss. 2166 <https://doi.org/10.1098/rsta.2019.0066>
2. Il'in V.P. Iterative Preconditioned Methods in Krylov Spaces: Trends of the XXI Century. Computational Mathematics and Mathematical Physics, 2021, Vol. 61, No. 11, pp. 1750–1775. © Pleiades Publishing, Ltd., 2021. DOI: 10.1134/S0965542521110099
3. Y. Saad Y. Iterative Methods for Sparse Linear Systems, 2nd ed. (SIAM, 2003)
4. Dongarra J. List of freely available software for linear algebra on the web (2006) <http://netlib.org/utk/people/JackDongarra/la-sw.html>
5. Il'in V. Parallel intelligent computing in algebraic problems, Sokolinsky, L., Zymbler, M. (eds.) Parallel Computational Technologies. PCT 2021. Communications in Computer and Information Science. 2021. V. 1437. Springer, Cham. doi.org/10.1007/978-3-030-81691-9_8
6. Ильин В.П. Математическое моделирование Часть 1. Непрерывные и дискретные модели, Новосибирск, изд. СО РАН, 2017
7. Forrester A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modeling: A Practical Guide. Wiley, New York (2008)
8. Kleppe A. Software language engineering: Creating domain-specific language using metamodels. NY: Addison-Wesley, 2008
9. Il'in V. The integrated computational environment for optimization of complex systems, Proceedings of the 15-th International Asian School-Seminar "Optimization

- Problems of Complex Systems (OPCS 2019). P. 65–67 doi.org/10.1109/opcs.2019.888015.
10. Il'in V. Artificial intelligence problems in mathematical modeling, Voevodin V., Sobolev S. (eds.) RuSCDays 2019. CCIS. 2019. V. 1129. P. 505–516. Springer, Cham. doi.org/10.1007/978-3-030-36592-9_41
 11. Mikony S.V.: Formalization of the cognitive process using the basis of models. *Ontol. Des.* 8(1(27)), P. 35–48 (2018) (in Russian)
 12. LeCun Y., Bengio Y., Hinton G. Deep learning, *Nature*, 2015. V. 521. P. 436–444 doi.org/10.1038/nature14539
 13. Weinan, E.: Machine learning and computational mathematics // *Commun. Comput. Phys.* 2020. V. 28. P. 1639–1670 <https://doi.org/10.4208/cicp.oa-2020-0185>.
 14. Borgest N.M. Key terms of ontology of designing: review, analysis, generalizations [In Russian]. *Ontology of designing.* 2013. V. 3(9). P. 9–31
 15. Luccioni A., Bengio Y. On the Morality of Artificial Intelligence. arXiv:1912.11945 [cs.CY] /doi.org/10.48550/arXiv.1912.11945
 16. Antonov A., Dongarra J., Voevodin V. AlgoWiki Project as an Extension of the Top500 Methodology, *Supercomput.* *Frontiers and Innova.* 2018. V. 5, N. 1. P. 4–10
 17. Feoktistov A., Kostromin, R., Sidorov, I. A., Gorsky, S. A. Development of distributed subject-oriented applications for cloud computing through the integration of conceptual and modular programming, *Proc. of the 41st Intern. Conf. on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2018).* P. 234–239
 18. Загоруйко Ю.А.; Загоруйко Г. Б. Онтологический подход к разработке системы поддержки принятия решений на нефтегазодобывающем предприятии, *Вестн. НГУ. Сер.: Информационные технологии.* 2012. Т. 10, вып. 1. С. 121–128
 19. Zagorulko Y., Zagorulko G. Architecture of extensible tools for development of intelligent decision support systems, *New Trends in Software Methodologies, Tools and Techniques. Proc. of the 10-th SoMeT_11.* Hamido Fujita (Eds.). Amsterdam: IOS Press. 2011. P. 457–466
 20. Il'in V.P., Skopin I. N. About performance and intellectuality of supercomputer modeling, *Programming and Computer Software.* 2016. V. 42, iss. 1. P. 5–16
 21. Glin'skiy B., Kulikov I., Sapetina, A., Zagorulko, Y., Zagorulko, G. The creation of intelligent support methods for solving mathematical physics problems on supercomputers, *Communications in Computer and Information Science.* 2019. V. 1129. P. 427–438
 22. Gorodnichev M., Lebedev D. Semantic tools for development of high-level interactive applications for supercomputers, *J. Supercomp.* 2021. V. 77, iss. 10. P. 11866–11880 doi: 10.1007/s11227-021-03731-6
 23. Бурнаев Е.В., Бернштейн А.В., Оселедец И.В. и др. Фундаментальные исследования и разработки в области прикладного искусственного интеллекта, *Докл. РАН. Математика, информатика, процессы управления.* 2022. Т. 508. С. 19–27