Письма в ЭЧАЯ. 2025. Т. 22, № 3(260). С. 511–518

КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ФИЗИКЕ

РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ СХЕМЫ, ОСНОВАННОЙ НА ДВУМЕРНОМ DVR-БАЗИСЕ, ДЛЯ ИНТЕГРИРОВАНИЯ НЕСТАЦИОНАРНОГО ТРЕХМЕРНОГО УРАВНЕНИЯ ШРЕДИНГЕРА

А. С. Айриян [а, б], *Я. Буша (мл.)* [а, б], *В. С. Мележик* [а, в, 1]

[а] Объединенный институт ядерных исследований, Дубна

[б] Национальная научная лаборатория им. А. И. Алиханяна, Ереван

[в] Государственный университет «Дубна», Дубна, Россия

Предложена и исследована схема распараллеливания вычислений при интегрировании нестационарного трехмерного уравнения Шредингера в представлении дискретной переменной (DVR). При разложении по двумерному базису DVR исходное трехмерное уравнение сводится к системе обыкновенных дифференциальных уравнений шредингеровского типа. Для его численного решения применяется метод покомпонентного расщепления, обеспечивающий эффективность вычислений благодаря разбиению задачи на независимые подзадачи. В рамках проведенного исследования показано, что наиболее затратные этапы вычислительной процедуры, преобразование вектора искомых решений из представления DVR в представление сферических функций и обратно на каждом шаге по времени, могут быть эффективно распараллелены. Приведены результаты численного моделирования, демонстрирующие существенное сокращение времени вычислений для задач, требующих большого количества узлов сетки при аппроксимации исходной задачи по временной и пространственным переменным.

We have proposed and investigated a parallelization of the computational scheme for integrating the time-dependent three-dimensional Schrödinger equation in the discrete variable representation (DVR). When expanded in a two-dimensional DVR basis, the original three-dimensional equation is reduced to a system of ordinary differential equations of the Schrödinger type. The method of component-by-component splitting is applied for its numerical integration, ensuring computational efficiency by separating the problem into independent subproblems. The study demonstrates that the most computationally expensive steps of the procedure, namely, transforming the vector of the sought solutions from the DVR to the spherical function representation and vice versa at each time-step, can be efficiently parallelized. Results of the numerical simulation are presented, showing a significant reduction in computation time for problems requiring a large number of grid points when approximating the original problem in both temporal and spatial variables.

PACS: 03.65.–w; 02.60.Cb

[1] E-mail: melezhik@theor.jinr.ru

ВВЕДЕНИЕ

В работах [1–3] предложен эффективный метод интегрирования нестационарного трехмерного уравнения Шредингера (здесь и далее $\hbar = 1$):

$$i\frac{\partial}{\partial t}\psi(r,\Omega,t) = \left[-\frac{1}{2M}\frac{\partial^2}{\partial r^2} + \frac{L^2(\Omega)}{2Mr^2} + V_0(r) + V(r,\Omega,t)\right]\psi(r,\Omega,t), \tag{1}$$

где r — радиальная переменная, а $\Omega = \{\theta,\phi\}$ — угловые переменные; M — эффективная масса системы; $L^2(\Omega)$ — оператор квадрата углового момента системы. В этом подходе решение (1) ищется в виде разложения по базису Discrete-Variable Representation (DVR) [1–4]

$$f_j(\Omega) = \sum_{\nu=1}^{N_\Omega} Y_\nu(\Omega)(Y^{-1})_{\nu j}, \tag{2}$$

где $Y_\nu(\Omega)$ — линейная комбинация сферических функций, а матрица $(Y^{-1})_{\nu j}$ является обратной к квадратной матрице $Y_{j\nu} = Y_\nu(\Omega_j)$ (размерности $N_\Omega \times N_\Omega$), построенной из значений функций $Y_\nu(\Omega)$ в узловых точках Ω_j сетки на единичной гиперсфере Ω. В этом базисе исходное трехмерное уравнение (1) сводится к системе N_Ω обыкновенных дифференциальных уравнений

$$i\frac{\partial}{\partial t}\,\mathbf{u}(r,t) = [\hat{H}^{(0)}(r) + \hat{V}(r,t)]\,\mathbf{u}(r,t), \tag{3}$$

где

$$H^{(0)}_{jj'}(r) = -\frac{1}{2M}\left[\delta_{jj'}\frac{d^2}{dr^2} - \frac{L^2_{jj'}}{r^2}\right] + V_0(r)\delta_{jj'}, \tag{4}$$

$$V_{jj'}(r,t) = V(r,\Omega_j,t)\delta_{jj'}. \tag{5}$$

Здесь $\mathbf{u}(r,t) = \{u_j(r,t)\}_1^{N_\Omega}$ — вектор размерности N_Ω, а

$$L^2_{jj'} = \sum_{\nu=1}^{N_\Omega} Y_{j\nu}\, l(l+1)(Y^{-1})_{\nu j'}$$

— плотная матрица $N_\Omega \times N_\Omega$ оператора квадрата углового момента системы в представлении DVR.

В работе предложена и исследована схема распараллеливания вычислений при численном интегрирования задачи (3)–(5) методом покомпонентного расщепления [1–3].

1. ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМЫ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ШРЕДИНГЕРОВСКОГО ТИПА МЕТОДОМ ПОКОМПОНЕНТНОГО РАСЩЕПЛЕНИЯ

Для численного интегрирования системы дифференциальных уравнений (3) применяется метод покомпонентного расщепления [1–3]:

$$\mathbf{u}(r,t_n+\Delta t)=\exp\left[-\frac{i}{2}\Delta t\hat{V}(r,t_n)\right]\exp\left[-i\Delta t\hat{H}^{(0)}(r)\right]\exp\left[-\frac{i}{2}\Delta t\hat{V}(r,t_n)\right]\mathbf{u}(r,t_n). \tag{6}$$

В этой вычислительной схеме первая и третья операторные экспоненты в правой части выражения (6) представляют собой диагональную матрицу

$$\exp\left[-\frac{i}{2}\,\Delta t\,V(r,\Omega_j,t_n)\right]\delta_{jj'}, \tag{7}$$

не порождающую существенных вычислительных затрат, с числом операции $\sim N_\Omega$. А промежуточная экспонента является наиболее затратной частью всей вычислительной процедуры (6). Для ее выполнения сначала матрица $\hat{H}^{(0)}_{jj'}(r)$ диагонализуется:

$$\bar{\mathbf{u}}=\hat{S}\mathbf{u}, \tag{8}$$

$$S_{j\nu}=\sqrt{\omega_j}Y_\nu(\Omega_j), \tag{9}$$

$$\bar{H}^{(0)}_{\nu\nu'}=(\hat{S}\hat{H}^{(0)}\hat{S}^{+})_{\nu\nu'}=h^{(0)}_\nu(r)\delta_{\nu\nu'}=-\frac{1}{2M}\left[\frac{d^2}{dr^2}-\frac{l_\nu(l_\nu+1)}{r^2}+V_0(r)\right]\delta_{\nu\nu'}, \tag{10}$$

где ω_j — веса квадратурной формулы для численного интегрирования по узловым точкам Ω_j на единичной сфере Ω. В результате диагонализации (9), (10) процедура умножения вектора $\bar{\mathbf{u}}$ на $\exp\left[-i\Delta t h^{(0)}_\nu(r)\right]$ сводится к решению N_Ω независимых краевых задач:

$$\left[1+i\frac{\Delta t}{2}h^{(0)}_\nu(r)\right]\bar{u}_\nu\left(r,t_n+\frac{3}{4}\Delta t\right)=$$

$$=\left[1-i\frac{\Delta t}{2}\,h^{(0)}_\nu(r)\right]\bar{u}_{nu}\left(r,t_n+\frac{1}{4}\Delta t\right),\quad \nu=1,2,\ldots,N_\Omega, \tag{11}$$

$$\bar{u}_\nu\left(r=0,t_n+\frac{3}{4}\Delta t\right)=\bar{u}_\nu\left(r=r_m,t_n+\frac{3}{4}\Delta t\right)=0,\quad r_m\to+\infty. \tag{12}$$

Здесь использована аппроксимация Кранка–Николсона:

$$\exp\left[-i\Delta t h^{(0)}_\nu(r)\right]=\left[1+i\frac{\Delta t}{2}h^{(0)}_\nu(r)\right]^{-1}\left[1-i\frac{\Delta t}{2}h^{(0)}_\nu(r)\right]. \tag{13}$$

В результате интегрирования N_Ω независимых краевых задач (11), (12) находится вектор $\bar{\mathbf{u}}(r,t_n+3/4\,\Delta t)$, который с помощью матрицы $\hat{S}^{+}$ (обратной к $\hat{S}$) трансформируется в искомый на этом шаге процедуры (6) вектор

$$\mathbf{u}\left(r,t_n+\frac{3}{4}\Delta t\right)=\hat{S}^{+}\bar{\mathbf{u}}\left(r,t_n+\frac{3}{4}\Delta t\right). \tag{14}$$

Краевая задача (11), (12) аппроксимируется по радиальной переменной r методом конечных разностей 6-го порядка точности на квазиравномерной сетке $\{r_k\}_1^N$ и решается для каждого ν методом прогонки [2, 3]. При этом число операций вычислительной процедуры $\sim N$. А суммарное число операций при численной реализации всей вычислительной схемы (6) пропорционально величине

$$N_T N(2N_\Omega^2 + \alpha^2 N_\Omega + 2N_\Omega), \tag{15}$$

где $\alpha = 7$ — ширина ленты ленточной матрицы $h_\nu^{(0)}(r_k)$, аппроксимирующей исходный оператор $h_\nu^{(0)}(r)$ в (10) при семиточечной конечно-разностной аппроксимации в нем дифференцирования по r, а N_T — суммарное число шагов $t_n \to t_{n+1} = t_n + \Delta t$ по временной переменной. Квадратичная зависимость N_Ω^2 числа операций в формуле (15) от N_Ω возникает при прямом (8) и обратном (14) преобразованиях от координатного представления (DVR) к представлению собственных функций оператора углового момента и обратно.

2. РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ ПРОЦЕДУРЫ

При решении задач из различных областей квантовой физики, где требуется интегрировать нестационарное уравнение Шредингера (1) с заданной точностью с помощью разложения по DVR-базису (2), возникает необходимость решать задачу на большом числе узлов разностных сеток по временной (N_T) и пространственным (N и N_Ω) переменным в рамках вычислительной процедуры (6). В реальных расчетах [1–3] требуемое число узлов достигает величин

$$N_T \sim 10^4, \quad N \sim 2 \cdot 10^4, \quad N_\Omega \sim 10^3, \tag{16}$$

а время счета на современных персональных вычислительных устройствах при этом варьируется от нескольких десятков часов до нескольких рабочих дней. В этой связи становится актуальным вопрос об оптиматизации рассмотренной выше вычислительной схемы (6)–(14).

В данной работе мы исследовали возможность ускорения вычислительной схемы за счет распараллеливания вычислений. Нетрудно заметить, что численное решение задачи (11), (12) на одном временном шаге легко поддается распараллеливанию, поскольку оно состоит в интегрировании N_Ω независимых краевых задач для каждого ν, и каждую из них можно решать независимо на отдельном вычислительном устройстве. Однако с помощью профилировщика программ [1] было установлено, что численное решение задачи (11), (12) занимает не более 4 % всего расчетного времени, тогда как выполнение процедур (8) и (14) занимает около 95 % времени из-за квадратичной зависимости от N_Ω матричных умножений (8) и (14). Поэтому эти элементы вычислительной схемы были оптимизированы с помощью распараллеливания вычислений (рис. 1), в то время как распараллеливание самой краевой задачи (11), (12) в данной ситуации не представлялось целесообразным.

[1] https://valgrind.org

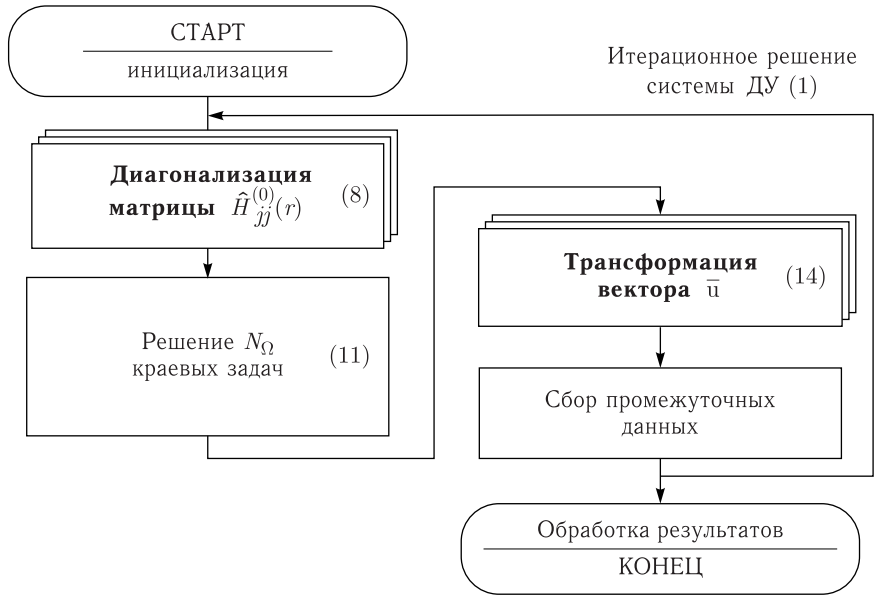


Рис. 1. Блок-схема алгоритма. Жирным шрифтом выделены распараллеленные элементы вычислительной схемы (диагонализация матрицы и трансформация вектора)

Для распараллеливания вычислений были применены две технологии программирования на системах с разделяемой памятью (shared memory): OpenMP [1] и технология вычислений на графических ускорителях CUDA [2].

Выбор этих технологий обусловлен возможностью выполнения параллельных вычислений на многоядерных и многопоточных процессорах, которые в настоящее время широко используются, в том числе в персональных компьютерах и ноутбуках. Таким образом, не требуется обязательного привлечения внешних вычислительных ресурсов (например, кластеров). Кроме того, OpenMP поддерживается всеми распространенными компиляторами. В нашем случае использовался компилятор GCC версии 11.2 [3]. Более того, предлагаемая реализация параллельных вычислений с использованием директив компилятора позволяет при необходимости, без изменений в тексте программы, выполнять расчеты без параллелизма в случае отсутствия параллельных вычислительных ресурсов. Сама параллелизация вычислений состоит в простой параллелизации циклов (параллельном выполнении инструкций тела цикла) для процедур (8) и (14) как на OpenMP (с помощью стандартной директивы «OMP parallel for»), так и на CUDA. В последнем случае входные данные предварительно, на стадии инициализации, переносятся в память графического процессора, после чего на каждом временном шаге вектор решений возвращается из памяти графического процессора.

[1] https://www.openmp.org
[2] https://developer.nvidia.com/cuda-toolkit
[3] https://gcc.gnu.org/onlinedocs/11.2.0

3. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Исследование свойств разработанных алгоритмов с использованием распараллеливания проводилось на гетерогенной вычислительной платформе «HybriLIT» ЛИТ ОИЯИ [5]. Расчеты выполнялись на двухпроцессорных серверах с Intel Xeon E5-2695 v2[1], каждый из которых содержит 12 физических ядер. Результаты исследований приведены в таблице и на рис. 2 и 3, где показаны время расчетов и ускорение параллельных вычислений в зависимости от количества задействованных вычислительных нитей (потоков), при этом гиперпоточность была отключена.

Результаты, полученные с использованием технологии OpenMP, показывают ускорение вычислений до $\sim 300\,\%$, однако эффективность параллельных вычислений при этом остается невысокой — около $25\,\%$. Под эффективностью параллельных вычислений понимается степень использования вычислительных ресурсов, которая определяется как отношение ускорения к количеству параллельных устройств. Наблюдается насыщение ускорения при 12 вычислительных потоках. При увеличении числа потоков выше этого значения начинает использоваться второй процессор, что уже не приводит к дополнительному ускорению из-за накладных затрат на синхронизацию передачи данных между процессорами. Эти накладные затраты существенно снижают эффект от автоматического кэширования данных внутри центрального процессора. Следует отметить, что современные, широко распространенные многоядерные процессоры в персональных устройствах, как правило, содержат не более 12 физических вычислительных ядер.

Время расчета и ускорение параллельных вычислений в зависимости от количества вычислительных потоков (нитей)

Количество нитей	$N_\Omega = 374$		$N_\Omega = 750$	
	Время, мин	Величина ускорения	Время, мин	Величина ускорения
Оригинальное	196	—	721	—
1	173	1,13	669	1,08
2	117	1,68	429	1,68
4	91	2,15	308	2,34
8	76	2,58	241	2,99
12	69	2,84	222	3,25
16	78	2,51	274	2,63
20	73	2,68	269	2,68
24	70	2,80	238	3,03
CUDA	21	9,33	55	13,11

Примечание. Расчеты проводились при фиксированном числе узлов по временной ($N_T = 10^4$) и радиальной ($N = 2 \cdot 10^4$) переменным для двух наборов узловых точек Ω_j на единичной сфере Ω с числом узлов $N_\Omega = 374$ и 750. В строке «оригинальное» приведены результаты, полученные с использованием оригинальной версии программы [3] на языке Fortran без распараллеливания и дополнительной оптимизации.

[1] https://ark.intel.com/content/www/us/en/ark/products/75281/intel-xeon-processor-e5-2695-v2-30m-cache-2-40-ghz.html

Результаты, представленные в таблице и на рис. 2 и 3, показывают, что использование графического процессора CUDA позволяет ускорить решение задачи в 9–13 раз. Такой эффект достигается главным образом благодаря быстрой разделяемой памяти между графическими потоками с возможностью задействования до 1024 потоков на разделяемую память.

Отметим, что ускорение параллельных вычислений оценивалось по сравнению с временем выполнения оригинальной программы без распараллеливания [3], написанной на языке Fortran. Параллельные версии, помимо самой параллелизации, также включали дополнительную оптимизацию. В частности, были затабулированы значения функций, аргументы которых многократно повторяются в ходе выполнения вычислительной процедуры. Эта оптимизация позволила дополнительно сократить время расчетов примерно на 10%.

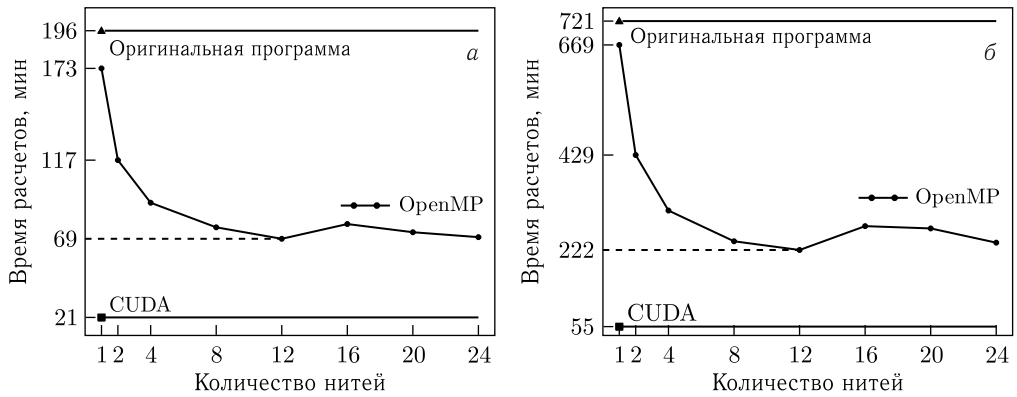


Рис. 2. Время расчетов в зависимости от количества потоков OpenMP. Расчеты проводились при фиксированном числе узлов по временной ($N_T = 10^4$) и радиальной ($N = 2 \cdot 10^4$) переменным для двух наборов узловых точек Ω_j на единичной сфере Ω с числом узлов $N_\Omega = 374$ (*а*) и 750 (*б*). CUDA — расчетное время, полученное с использованием графического ускорителя

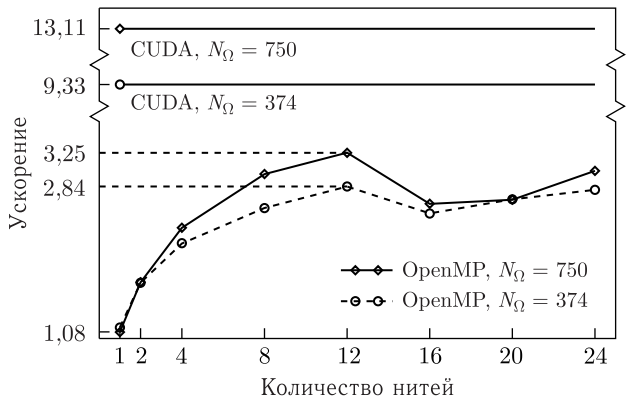


Рис. 3. Зависимость ускорения расчетов от количества потоков OpenMP. Расчеты проводились при фиксированном числе узлов по временной ($N_T = 10^4$) и радиальной ($N = 2 \cdot 10^4$) переменным для двух наборов узловых точек Ω_j на единичной сфере Ω с числом узлов $N_\Omega = 374$ и 750

ЗАКЛЮЧЕНИЕ

В работе предложен и численно исследован алгоритм параллельных вычислений для интегрирования нестационарного трехмерного уравнения Шредингера (1) в представлении дискретной переменной (DVR) (2). Показано, что наиболее затратные этапы вычислительной процедуры, преобразование вектора искомых решений из представления DVR в представление сферических функций и обратно на каждом шаге по времени, могут быть эффективно распараллелены. На примере расчета вероятности возбуждения и ионизации атома водорода в сильном лазерном поле [3] продемонстрировано, что использование графических процессоров позволяет сократить время вычислений с одного рабочего дня до 1 ч, что значительно способствует ускорению научно-исследовательского процесса. Кроме того, отметим, что при увеличении размерности интегрируемой численно задачи в данном подходе наблюдается ускорение параллельных вычислений за счет повышения загрузки параллельных вычислительных устройств. В силу этого обстоятельства предложенная вычислительная схема распараллеливания может оказаться наиболее «выгодной» при интегрировании задач больших размерностей, где отмеченный эффект проявляется наиболее выпукло.

Исследование выполнено за счет гранта Российского научного фонда № 20-11-20257.

СПИСОК ЛИТЕРАТУРЫ

1. *Melezhik V. S.* Polarization of Harmonics Generated from a Hydrogen Atom in a Strong Laser Field // Phys. Lett. A. 1997. V. 230, No. 3. P. 203–208; https://www.sciencedirect.com/science/article/pii/S0375960197002508.
2. *Melezhik V. S., Baye D.* Nonperturbative Time-Dependent Approach to Breakup of Halo Nuclei // Phys. Rev. C. 1999. V. 59. P. 3232–3239.
3. *Shadmehri S., Melezhik V. S.* A Hydrogen Atom in Strong Elliptically Polarized Laser Fields within Discrete Variable Representation // Laser Phys. 2023. V. 33, No. 2. P. 026001; arXiv:2111.06588.
4. *Melezhik V. S.* New Method for Solving Multidimensional Scattering Problem // J. Comput. Phys. 1991. V. 92, No. 1. P. 67–81; https://www.sciencedirect.com/science/article/pii/002199919190292S.
5. *Adam G. et al.* IT-Ecosystem of the HybriLIT Heterogeneous Platform for High-Performance Computing and Training of IT-Specialists // Selected Papers of the 8th Intern. Conf. “Distributed Computing and Grid-Technologies in Science and Education”. Aachen, 2018. V. 2267 of CEUR Workshop Proc. P. 638–644; https://ceur-ws.org/Vol-2267/638-644-paper-122.pdf.

Получено 13 ноября 2024 г.