0,0005 (натрий-калиевый механизм) и 0,007 (натрий-хлор) мМ, что согласуется с имеющейся в литературе экспериментальной информацией (0,001 мМ в [1]).

Построенная модель в первом приближении качественно описывает процесс закачки нейромедиатора обратно после прохождения нервного импульса и может быть использована для математического описания полного цикла движения медиаторов в нейронах. В дальнейшем планируется смоделировать процесс синтеза нейромедиаторов и процесс их движения к рецепторам на постсинаптическом нейроне.

1. Nicholls J.G., Martin A.R., Wallace B.G., Fuchs P.A., From Neuron To Brain (4th ed), Sinauer Associates, Inc (2001).
2. Melkikh A.V., Seleznev V.D. J.Theor.Bio, **248**, 350 (2007).
3. Melkikh A.V., Sutormina M.I., Developing Synthetic Transport Systems, Springer Netherlands (2013).

РЕШЕНИЕ КОНЕЧНОЙ МОДЕЛИ ХАББАРДА НА СИСТЕМАХ С ОБЩЕЙ ПАМЯТЬЮ

Данилов М.Ю.*, Искаков С.Н.

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, г. Екатеринбург, Россия

*E-mail:mike.d.ft402@gmail.com

Одной из актуальных в настоящее время задач является исследование свойств парамагнетиков. В данной работе проводится исследование электронной структуры молекулярного магнетика V_{15} при помощи конечной модели Хаббарда, гамильтониан которой имеет следующий вид:

$$\hat{H} = -\sum_{i \neq j} t_{ij} c^{\dagger}_{i\sigma} c_{j\sigma} + \sum_{i} \hat{U} n_{i\uparrow} n_{i\downarrow} - \sum_{i} \mu n_{i\uparrow} n_{i\downarrow} \qquad (1)$$

Решение модели Хаббарда методом точной диагонализации предусматривает нахождение собственных значений и собственных векторов матрицы гамильтониана, используя метод Арнольди.[1] Для описания реальных низкоразмерных систем данный метод требует организации эффективной обработки матриц большой размерности, что приводит к необходимости реализации дополнительных методов эффективного хранения данных.

В данной работе решение модели Хаббарда предлагается проводить на системах с общей памятью. При этом, ввиду ограниченного объема необходимо минимизировать память, требуемую при решении задачи. Для этих целей было предложено два метода.

Первый заключается в том, чтобы хранить симметричную матрицу гамильтониана без дублирования, воспользовавшись форматом Symmetric Sparse Skyline (SSS).[2] Это должно давать почти двукратное уменьшение требуемого объема памяти по сравнению с форматом Compressed Row Storage (CRS). Однако, при параллелизации умножения матрица-вектор требуется дополнительное выделение памяти на редукцию.

В рамках второго способа предлагается хранить только самые трудоемкие для воспроизведения составляющие: диагональ и знаки недиагональных элементов. Матрица знаков занимает в 8 раз меньше места по сравнению с готовыми значениями. Недостатком является то, что процедура умножения затрачивает дополнительное время на получение элементов.

Ниже приведено сравнение интересующих нас характеристик. Время выполнения указано для 143 тыс. состояний; объем памяти – для задачи V_{15}, с учетом редукции при 4 потоках.

Формат	Время выполнения, с	Требуемый объем памяти, ГБ
Symmetric Sparse Skyline	57,4	4,23
диагональ + знаки	96,9	2,31

Так как приоритетным в нашем случае является снижение требований к памяти, второй способ является более подходящим для дальнейшей работы.

1. Sorensen D., J. Matrix Anal. Appl., 13**,** 357 (1992).
2. Gkountouvas T. et al., 27th IPDPS, 273 (2013).

РАЗВИТИЕ БИРЖЕВЫХ АВТОМАТИЗИРОВАННЫХ ТОРГОВЫХ СИСТЕМ

Тищенко А.Л.*, Томашевич В.Г.

Уральский федеральный университет имени первого Президента России Б.Н. Ельцина, г. Екатеринбург, Россия

*E-mail: away2@mail.ru

Одним из важнейших элементов финансовой системы любой страны является биржа. Для ее нормального функционирования крайне важно поддерживать ликвидность, а значит, возникает необходимость в регулярном совершении сделок, помимо этого каждый участник торгов имеет намерение получать выгоду в результате таких операций. Таким образом, создаются предпосылки для разработки торговых систем, которые путем совершения сделок на бирже будут при этом извлекать прибыль. Совершенно очевидно, что для разработки таких