

УДК 539.3

DOI: 10.32326/1814-9146-2025-87-1-54-69

АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ ОБНАРУЖЕНИЯ КОНТАКТОВ В МЕТОДЕ ДИСКРЕТНЫХ ЭЛЕМЕНТОВ*

© 2025 г. **Горохов В.А., Кибец А.И., Нагорных Е.В.**

*Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского, Нижний Новгород, Российская Федерация*

kibec@mech.unn.ru

Поступила в редакцию 02.10.2024

Приведен краткий обзор публикаций, посвященных решению задач механики гранулированных сред методом дискретных элементов. В этих публикациях обсуждаются условия успешного применения алгоритмов и программного обеспечения, разрабатываемых для эффективного функционирования вычислительных систем, реализующих метод дискретных элементов. Отмечается, что наиболее трудоемкими в моделировании процессов деформирования методом дискретных элементов считаются вычисления, связанные с определением контактных сил. Ключевой проблемой является эволюционный характер контактов: во время деформирования исследуемой среды дискретные элементы могут вступать в контакт, который в дальнейшем может прерываться в зависимости от условий нагружения. В связи с этим необходима тщательная проработка организации и управления оперативной памятью связанных с компьютерным моделированием контактов. Основное внимание уделяется алгоритмам поиска дискретных элементов, вступающих в контакт. Эти алгоритмы должны быть робастными (то есть надежными), простыми в реализации и эффективными с точки зрения быстродействия и использования оперативной памяти. Сопоставляются два варианта алгоритма, основанные на применении вспомогательных (экранирующих) сеток: скрининг-алгоритм обнаружения контактов (Screening Contact Detection Algorithm) и алгоритм обнаружения контактов Munjiza-NBS (Munjiza-NBS Contact Detection Algorithm). Излагается краткое описание этих алгоритмов и их программной реализации. Приведены результаты верификации алгоритмов на тестовых задачах и сравнительного анализа их быстродействия. Показано, что выбранные алгоритмы обеспечивают достоверность информации о контактирующих в процессе движения дискретных элементах и могут успешно применяться в программной реализации метода дискретных элементов в задачах динамики гранулированных сред.

Ключевые слова: нелинейный, нестационарный, деформирование, метод дискретных элементов, контакт, поиск контактирующих элементов.

* Выполнено в рамках научной программы Национального центра физики и математики, направление №2 «Математическое моделирование на супер-ЭВМ экса- и зеттапроизводительности. Этап 2023–2025».

Введение

Основополагающей публикацией, посвященной методу дискретных элементов (Discrete Element Method – DEM), принято считать статью P.A. Cundall, O.D.L. Strack [1]. Согласно [1], в DEM взаимодействие дискретных элементов происходит в конечном интервале времени, в течение которого контактные силы меняются в зависимости от расстояния между их центрами. Дискретные элементы считались соединенными линейными пружинами в нормальном и тангенциальном направлениях, нормальные и тангенциальные контактные силы рассчитывались как произведение жесткостей пружины и приращений относительного смещения двух контактирующих элементов.

Информация о DEM отражена в монографиях L. Jing, O. Stephansson [2], C. O'Sullivan [3], F. Radjai, F. Dubois [4], M. Jebahi et al. [5], C. Thornton [6], S. Ji, L. Liu [7], а также в диссертационных работах R.M. O'Connor [8], S. Van Baars [9], H. Ouafdel [10], A.O. Raji [11], Y. Wan [12], C. Wellmann [13], J. Chen [14], K. Samiei [15]. В них представлены фундаментальные концепции, лежащие в основе метода дискретных элементов, их историческое развитие и широкий спектр приложений. Указаны области эффективного применения: а) имитация экспериментов; б) изучение пространственно-временной зависимости параметров процессов деформирования, включая предельные значения; в) выявление новых, ранее не известных явлений на микроуровне. В приложениях DEM акцент делается на обнаружении новой информации, расширяющей понимание поведения систем частиц, что может быть более важным, чем разработка новой континуальной модели, охватывающей все микро-структурные аспекты, которая, скорее всего, окажется слишком сложной для практической реализации (C. Thornton [6]).

Классическая модель дискретных элементов может быть представлена в виде трехфазной схемы, которая включает в себя обнаружение контакта, силовое взаимодействие и интегрирование по времени [12]. Фундаментальным компонентом моделирования DEM является обнаружение контактов [8]. Различные аспекты метода, связанные с обнаружением контактов, рассмотрены в статьях L. Vu-Quoc, X. Zhang, O.R. Walton [16], J. Ghaboussi, R. Barbosa [17], X. Lin, T.T. Ng [18], N. Bićanić [19], G. Nezami et al. [20], D. Zhao et al. [21], A. Wachs et al. [22], Z. Zheng et al. [23], L.J.H. Seelen, J.T. Padding, J.A.M. Kuipers [24], A.D. Rakotonirina et al. [25], Z. Lai, Q. Chen, L. Huang [26], P. Zhang et al. [27], F. Alonso-Marroquín, Y. Wang. [28], Munjiza A. [29] и T. Pöschel, T. Schwager [30], C. Hogue [31].

Ключевой проблемой в DEM является эволюционный характер контактов: во время моделирования контакты будут создаваться и удаляться. Поэтому необходимо тщательно продумать организацию и управление памятью, связанной с моделированием контактов. Выявление контактирующих элементов и определение контактных сил являются наиболее трудоемкими при моделировании DEM. По оценкам G. Suttman [32], эти расчеты составляют 90% времени моделирования DEM.

Самое простое, что можно сделать с точки зрения программной реализации, – это сверять каждый дискретный элемент со всеми другими на каждом шаге по времени. Такой подход прост, но он чрезвычайно трудоемкий. В [18] разработана трехмерная дискретно-элементная модель, включающая в себя алгоритм обнаружения контакта, основанный на концепции геометрического потенциала. В статье [22] стратегия обнаружения контакта основана на использовании алгоритма Гилберта – Джонсона – Кирти для вычисления расстояния между двумя выпуклыми телами.

В статье [23] разработан алгоритм, который содержит улучшенную глобальную стадию для нахождения потенциальных сегментов, а не ближайшего узла для дискретных элементов в качестве возможного контакта. В статье [24] для быстрой идентификации пар частиц, находящихся в контакте, используется стратегия обнаружения контакта широкой фазы и узкой фазы. В статье [25] используется стратегия обнаружения контакта, основанная на алгоритме Гилберта – Джонсона – Кирти и пространственной сортировке связанных ячеек, которая ускоряет моделирование контакта. В статье [27] предложен DEM на основе метаболы, где уравнение метаболы используется для описания формы частиц. Выявление контактирующих частиц осуществляется путем решения задачи оптимизации. Разработан алгоритм нахождения ближайших точек для DEM метаболы на основе метода Ньютона – Рафсона.

Как подчеркнуто в [29], алгоритм обнаружения контактов должен быть робастным (то есть надежным), простым в реализации, эффективным с точки зрения быстродействия и оперативной памяти. В статье [29] рассматриваются алгоритмы на основе сеток, которые относительно просты в реализации:

- а) алгоритм обнаружения контактов на основе двоичного дерева (Binary Tree Based Contact Detection Algorithm);
- б) алгоритм прямого отображения (Direct Mapping Algorithm);
- в) скрининг-алгоритм обнаружения контакта (Screening Contact Detection Algorithm);
- г) алгоритм обнаружения контактов с сортировкой (Sorting Contact Detection Algorithm);
- д) алгоритм обнаружения контактов Munjiza-NBS в 2D (Munjiza-NBS Contact Detection Algorithm in 2D);
- е) алгоритм C-grid Вильямса (Williams C-Grid Algorithm).

В [29] показано, что алгоритм обнаружения контактов Munjiza-NBS занимает меньше места в оперативной памяти, чем алгоритм обнаружения контактов на основе двоичного поиска и алгоритм обнаружения контактов с сортировкой. В настоящей статье алгоритм обнаружения контактов Munjiza-NBS сопоставляется со скрининг-алгоритмом. Кратко изложена их программная реализация. Приведены результаты верификации и сравнительного анализа их быстродействия.

1. Скрининг-алгоритм обнаружения контакта для дискретных элементов одинакового размера в двумерных задачах

Самый простой способ представления отображения дискретных элементов на сетку из одинаковых ячеек – это использование одного целого числа в каждой ячейке. Общее количество целых чисел равно общему количеству ячеек. Все эти целые числа лучше всего представить в виде двумерного массива целых чисел (так называемого массива экранирования):

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,j} & \dots & c_{1,n_{cel}} \\ c_{2,1} & c_{2,2} & \dots & c_{2,j} & \dots & c_{2,n_{cel}} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{i,1} & c_{i,2} & \dots & c_{i,j} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ c_{n_{cel},1} & c_{n_{cel},2} & \dots & c_{n_{cel},j} & \dots & c_{n_{cel},n_{cel}} \end{bmatrix}. \quad (1)$$

Распределение дискретных элементов по ячейкам (рис. 1) представлено в виде привязки к каждой ячейке односвязного списка дискретных элементов. В соотношении (1) первый дискретный элемент в списке (заголовок списка) представлен в ячейке массива C , а остальная часть списка представлена одномерным целочисленным массивом размера N :

$$E = [e_1 \quad e_2 \quad e_3 \quad \dots \quad e_i \quad \dots \quad e_N]. \quad (2)$$

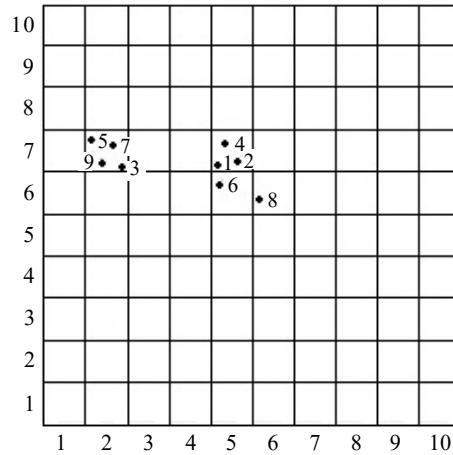


Рис. 1. Обнаружение контактов – отображение дискретных элементов на ячейки

Для дискретных элементов, показанных на рис. 1, представление отображения дискретных элементов на ячейки достигается установкой массива C на заголовки односвязных списков для каждой из ячеек, то есть

$$C = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 9 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 6 & 4 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 8 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}. \quad (3)$$

Остальные списки представлены массивом E , указывающим на следующий дискретный элемент в списке, как показано на рис. 2.

Стоит отметить, что большинство ячеек на рис. 1 не имеют дискретных элементов, отображенных на них. Такие ячейки называются пустыми. Пустые ячейки обозначаются путем установки значения соответствующей ячейки массива C числом -1 . Число -1 также используется для обозначения конца односвязного списка, то есть конец каждого односвязного непустого списка обозначается путем установки значения соответствующей ячейки массива E числом -1 . Массив E , показанный на

рис. 2, представляет собой четыре односвязных списка. В то же время массив C представляет 100 односвязных списков, из которых 96 пусты.

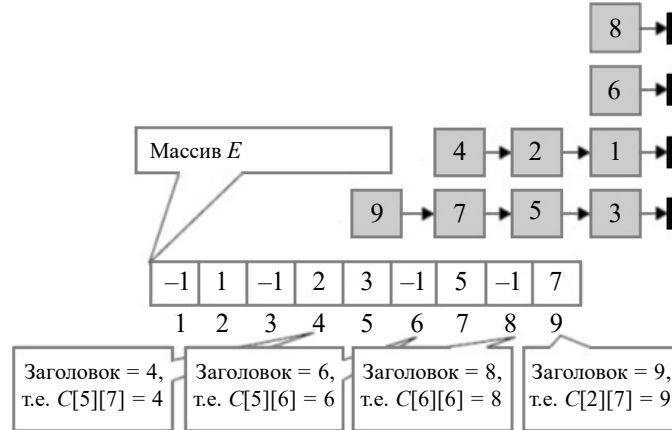


Рис. 2. Односвязные списки для системы дискретных элементов, показанных на рис. 1

Таким образом, для пространства, мало заполненного дискретными элементами, массив C может быть очень большим. Однако в ситуациях, когда дискретные элементы тесно связаны друг с другом, например в задачах упаковки, большинство ячеек не пусты и этот алгоритм имеет смысл. В противном случае пространство оперативной памяти, необходимое для хранения массива C , может быть не всегда доступно или, наоборот, размер задачи (общее количество дискретных элементов) может быть ограничен доступным пространством оперативной памяти. В двумерных задачах массив C – это двумерный массив. В трехмерном пространстве – это трехмерный массив.

Отображение осуществляется следующим образом.

Шаг 0: Начальная установка. Все ячейки массивов C и E устанавливаются числом -1 (пустые списки и/или концы списков).

Шаг 1: Распределение дискретных элементов по ячейкам. Эта операция выполняется на каждом временном шаге. Она включает в себя построение односвязного списка для каждой ячейки. Это достигается путем перебора всех дискретных элементов, целочисленного определения их координат и помещения номера дискретного элемента в соответствующий односвязный список:

$$i = \text{int } x_k = 1 + \text{Int} \left(\frac{x_k - x_{\min}}{d} \right), \quad j = \text{int } y_k = 1 + \text{Int} \left(\frac{y_k - y_{\min}}{d} \right), \quad (4)$$

где x_k, y_k – координаты центра дискретного элемента; $x_{\min}, y_{\min}$ – минимальные значения координат ячеек; d – размер ячейки; Int – целая часть числа.

Удаление дискретных элементов из односвязных списков производится путем перебора всех дискретных элементов, определения текущих координат их центров и установки соответствующего элемента массива C , а также соответствующего элемента массива E числом -1 , что указывает на пустоту списков.

После того как дискретные элементы были сопоставлены с ячейками, выполняется определение контактов. Для этого необходимо перебрать все дискретные элементы и выяснить, на какую ячейку отображен конкретный дискретный элемент. После этого выполняется прямая проверка всех дискретных элементов из центральной и соседних ячеек. Цикл над ячейками при этом не используется. Таким образом, общее

количество алгебраических операций не зависит от общего числа ячеек, то есть не зависит от размера массива C .

2. Алгоритм обнаружения контакта Munjiza-NBS

Алгоритм обнаружения контакта Munjiza-NBS [29] основан на декомпозиции пространства – разбиении его на одинаковые квадратные ячейки размером d (рис. 3), где точками отмечены центры дискретных элементов. Каждому дискретному элементу присваивается целое число $1, 2, 3, \dots, N$. Аналогично каждой ячейке присваивается пара целых чисел (i_x, i_y) , где $i_x = 1, 2, 3, \dots, n_x$, $i_y = 1, 2, 3, \dots, n_y$; n_x и n_y – общее количество ячеек в направлениях x и y соответственно:

$$n_x = \frac{x_{\max} - x_{\min}}{d}, \quad n_y = \frac{y_{\max} - y_{\min}}{d}. \quad (5)$$

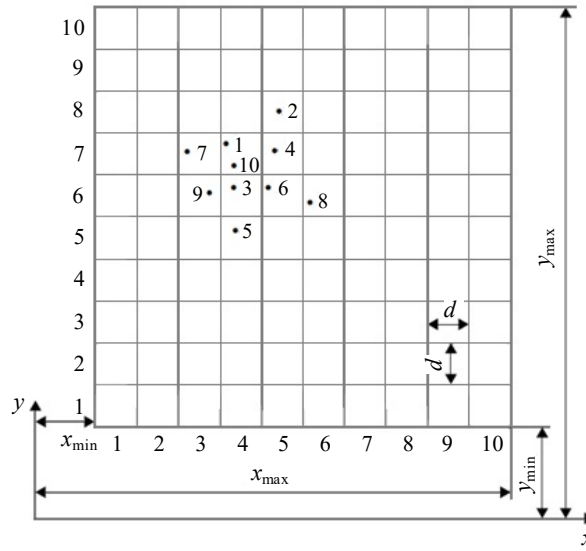


Рис. 3. Пространство, разделенное на одинаковые ячейки

Отображение множества дискретных элементов

$$E = \{1, 2, 3, 4, 5, \dots, N\} \quad (6)$$

на множество ячеек

$$C = \left\{ \begin{array}{cccccc} (1,1) & (1,2) & (1,3) & \dots & (1,n_y) \\ (2,1) & (2,2) & (2,3) & \dots & (2,n_y) \\ (3,1) & (3,2) & (3,3) & \dots & (3,n_y) \\ \dots & \dots & \dots & \dots & \dots \\ (n_x,1) & (n_x,2) & (n_x,3) & \dots & (n_x,n_y) \end{array} \right\} \quad (7)$$

определяется таким образом, чтобы каждый дискретный элемент был отнесен к одной и только одной ячейке. Для этого каждый дискретный элемент с координатами (x, y) присваивается ячейке (i_x, i_y) , где

$$i_x = \text{Int}\left(\frac{x - x_{\min}}{d}\right), \quad i_y = \text{Int}\left(\frac{y - y_{\min}}{d}\right) \quad (8)$$

– целочисленные относительные координаты центра ограничивающей окружности для каждого дискретного элемента, которые называются целочисленными координатами.

Помимо отображения дискретных элементов на ячейки также вводится отображение дискретных элементов на столбцы и строки ячеек. Считается, что дискретный элемент соответствует определенному ряду ячеек, если он соответствует любой ячейке из этого ряда. Аналогично дискретный элемент считается сопоставленным с определенным столбцом ячеек, если он соответствует любой ячейке из этого столбца.

В случае алгоритма Munjiza-NBS для экономии ресурсов процессора используются односвязные списки. Представление отображения выполняется в два этапа.

Этап 1. Производится отображение всех дискретных элементов на строки ячеек (по направлению y) и формируется односвязный список дискретных элементов для каждой строки i_y , где $i_y = 1, 2, 3, \dots, n_y$, (рис. 4). Всего существует n_y таких списков. Некоторые из списков пусты, другие содержат только один дискретный элемент и т.д. Эти списки называются y -списками, причем каждый y -список идентифицируется со строкой ячеек, для которой он сформирован, и называется y_{iy} -списком.

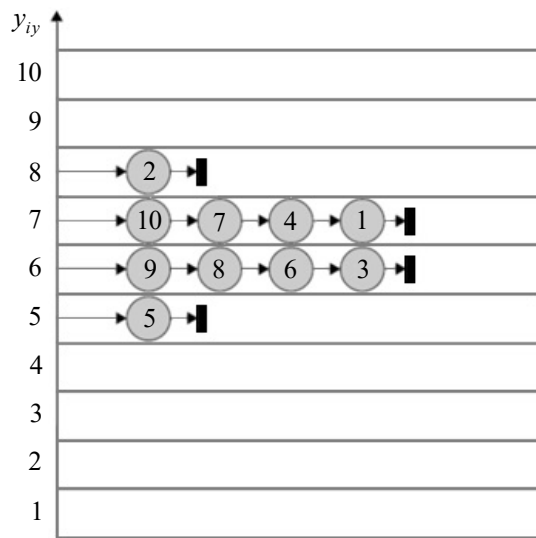


Рис. 4. Односвязные списки для рядов ячеек

Дискретные элементы в каждом из списков следуют друг за другом в убывающем числовом порядке (см. рис. 4). Это связано с тем, что дискретные элементы отображаются с помощью цикла над дискретными элементами в порядке возрастания их числа. Цифровое представление всех y -списков (пустых и непустых) достигается с помощью двух целочисленных массивов.

Первый массив B содержит номер последнего дискретного элемента, сопоставленного с каждой строкой, то есть заголовок односвязного списка для каждого ряда ячеек. Массив B – это одномерный массив размером n_y , где n_y – общее количество ячеек в направлении y , то есть общее количество рядов ячеек. Второй массив Y – это

одномерный массив размером N , где N – общее количество дискретных элементов. Для каждого дискретного элемента массив Y содержит следующий дискретный элемент в односвязном списке.

Для обоих массивов отрицательное число используется как завершение односвязного списка. Таким образом, если в определенной строке нет элементов, соответствующему элементу массива B присваивается отрицательное число.

Оба массива показаны на рис. 5, где выделен список y_6 . Очевидно, что массив B содержит «указатель» на односвязный список дискретных элементов для каждого ряда (где $i_y = 0, 1, 2, 3, \dots, n_y - 1$) ячеек. Аналогично массив Y указывает на следующий элемент в списке, заканчивающемся отрицательным числом.

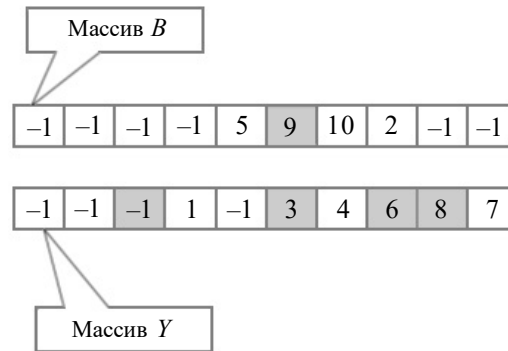


Рис. 5. Односвязные списки для рядов ячеек, представленные двумя массивами

В примере, показанном на рис. 3, в первой строке ячеек нет ни одного дискретного элемента, поэтому односвязный список дискретных элементов, сопоставленных с этой строкой (список y_0), пуст, что достигается с помощью параметра $B[1] = -1$.

Аналогично последним дискретным элементом, отображенным на строку 6 ячеек (список y_6), является дискретный элемент номер 9, поэтому $B[6] = 9$. Следующий дискретный элемент в этом списке номер 8, поэтому $Y[9] = 8$; следующий дискретный элемент номер 6, поэтому $Y[8] = 6$; следующий дискретный элемент номер 3, поэтому $Y[6] = 3$; последний дискретный элемент номер 3, поэтому $Y[3] = -1$ (см. рис. 5).

Этап 2. Для каждого ряда ячеек дискретные элементы теперь готовы к отображению на отдельные ячейки. Для каждого ряда ячеек дискретные элементы, отображенные на этот ряд, помещаются в односвязные списки, как объяснялось на этапе 1. Начало списка представлено соответствующим номером $B[i_y]$ для списка y_{i_y} .

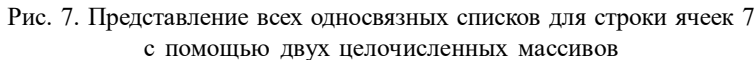
Рассматриваются только непустые списки. Проводится перебор всех дискретных элементов, но не строк ячеек. При обнаружении так называемого «нового» y -списка он помечается как «старый» список, чтобы ни один дискретный элемент больше не рассматривал этот ряд ячеек.

Каждый дискретный элемент из текущего y_{i_y} -списка отображается на ячейки путем выполнения цикла перебора всех дискретных элементов из y_{i_y} -списка:

$$i_x = \text{Int} \left(\frac{x - x_{\min}}{d} \right). \quad (9)$$

Конкретный список x_{i_x, i_y} содержит все дискретные элементы с целочисленными координатами i_x и i_y . Кроме того, все односвязные списки x_{i_x, i_y} (где $i_x = 1, 2, 3, \dots, n_x$) содержат все дискретные элементы из y_{i_y} -списка и представляются двумя массивами целых чисел.

Односвязные списки $x_{ix, iy}$ (где $i_x = 1, 2, 3, \dots, n_x$) с рис. 3 показаны на рис. 6. Численное представление этих списков массивами целых чисел показано на рис. 7, где выделен список $x_{4,7}$.



Список u_7 (см. рис. 3) содержит все дискретные элементы в 7-й строке ячеек, то есть дискретные элементы 10, 7, 4 и 1. Ни один дискретный элемент из списка u_2 не имеет целочисленной координаты i_x , равной 1 или 2, поэтому односвязные списки $x_{1,7}$ и $x_{2,7}$ пусты, то есть $A[1] = -1$ и $A[2] = -1$. Только дискретный элемент 4 имеет целочисленную координату i_x , равную 5, поэтому $A[5] = 4$ и $X[4] = -1$. Только дискретный элемент 7 имеет целочисленную координату i_x , равную 3, поэтому $A[3] = 7$,

$X[7] = -1$. Дискретные элементы 1 и 10 имеют целочисленные координаты i_x , равные 4, поэтому $A[4] = 1$, $X[1] = 10$, $X[10] = -1$.

Важно отметить, как составлены списки y_{iy} и x_{ix} :

- сначала выполняется цикл по всем дискретным элементам, внутри которого конкретный дискретный элемент j добавляется в соответствующий список y_{iy} в зависимости от его целочисленной координаты i_y ;

- затем выполняется цикл по всем дискретным элементам из определенного списка y_{iy} , и внутри этого цикла в соответствующий список $x_{ix,iy}$ добавляется конкретный дискретный элемент в зависимости от его целочисленной координаты i_x .

Для сборки новых списков дискретные элементы из старых списков удаляются аналогично – выполняется цикл по всем дискретным элементам, и для каждого дискретного элемента устанавливается $B[i_y] = -1$; аналогично выполняется цикл по всем дискретным элементам из конкретного списка y_{iy} , и для каждого дискретного элемента устанавливается $A[i_x] = -1$.

Таким образом, при сборке списков не происходит перебора ячеек, из чего можно сделать вывод, что общее время работы процессора, необходимое для выполнения всех операций, описанных ранее в этом разделе, пропорционально общему числу дискретных элементов N и не является функцией общего числа ячеек.

Обнаружение контакта осуществляется путем проверки всех дискретных элементов, сопоставленных с определенной ячейкой, на наличие контакта со всеми дискретными элементами в соседних ячейках.

Например, дискретные элементы, отображенные на ячейку (i_x, i_y) , показанную на рис. 8, проверяются на контакт со всеми дискретными элементами, отображенными на ячейки (i_x, i_y) , $(i_x - 1, i_y)$, $(i_x - 1, i_y - 1)$, $(i_x, i_y - 1)$ и $(i_x + 1, i_y - 1)$. Это эквивалентно проверке всех дискретных элементов из списка $x_{ix,iy}$ на соответствие всем дискретным элементам из списков $x_{ix,iy}$, $x_{ix-1,iy}$, $x_{ix-1,iy-1}$, $x_{ix,iy-1}$ и $x_{ix+1,iy-1}$.

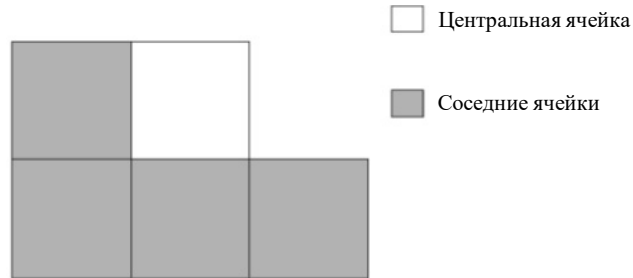


Рис. 8. Шаблон для проверки контактов в двумерной задаче

Таким образом, дискретные элементы, отображенные на любую непустую ячейку, сверяются со всеми дискретными элементами, отображенными на соседние ячейки. В любой момент времени необходимо иметь односвязные списки $x_{ix,iy}$ только для двух соседних строк ячеек i_y и $i_y - 1$, то есть для дискретных элементов из списков y_{iy} и y_{iy-1} . Необходимы два параллельных массива A , и это достигается с помощью двумерного массива $A[2][n_x]$ размером $2n_x$ целых чисел. Массив $A[0]$ указывает на все односвязные списки $x_{ix,iy}$ (где i_y фиксировано, а $i_x = 1, 2, 3, \dots, n_x$). Одномерный массив $A[1]$ указывает на все односвязные списки, состоящие из дискретных элементов, отображенных на строку $i_y - 1$ ячеек, то есть на все списки $x_{ix,iy-1}$ (где $i_y - 1$ фиксировано и $i_x = 1, 2, 3, \dots, n_x$).

Обнаружение контакта выполняется только для ячеек с непустым списком дискретных элементов $x_{ix,iy}$. Для этого используется цикл перебора дискретных элементов из списка y_{iy} для поиска ячейки (i_x, i_y) с одним или более дискретными элементами, назначенными ей, то есть «нового» списка $x_{ix,iy}$. Дискретные элементы, сопоставленные с каждой такой ячейкой, проверяются на контакт со всеми дискретными элементами, сопоставленными с соседними ячейками, как показано на рис. 8. Например, для списка y_7 из рис. 4 цикл перебора всех дискретных элементов из списка будет включать в себя дискретные элементы 10, 7, 4 и 1, в результате чего будут найдены «новые» списки $x_{3,7}$, $x_{4,7}$ и $x_{5,7}$.

В трехмерных задачах алгоритмы поиска дискретных элементов, которые могут вступать в контакт, принципиально не отличаются от изложенных выше. В частности, в скрининг-алгоритме обнаружения контакта массив C в уравнении (1) становится трехмерным, а шаблон для проверки контактов приведен на рис. 9. В алгоритме Munjiza-NBS добавляются два целочисленных массива – аналоги для третьей координаты массивов B , Y и A , X .

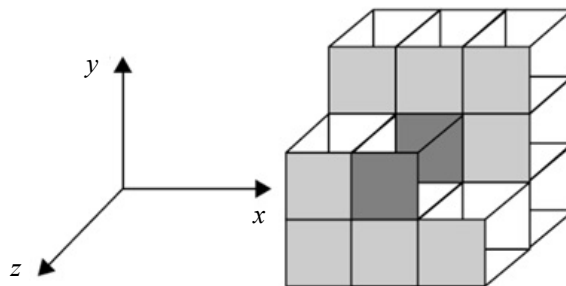


Рис. 9. Шаблон для проверки контактов в трехмерной задаче

3. Программная реализация

Для оценки вычислительной эффективности рассмотренных алгоритмов поиска контактирующих дискретных элементов разработано программное средство (консольное приложение) Demcda. Программное средство Demcda написано на языке программирования Fortran, компилятор Intel Fortran Compiler. Программное средство Demcda состоит из 6 компонентов:

- основная управляющая программа Demcda, в которой считываются входные данные для расчета, производится проверка их корректности, осуществляется вызов подпрограмм, реализующих алгоритмы поиска контактирующих дискретных элементов, производится запись списка найденных контактов в выходной файл;
- модуль *contact_list_processing*, содержащий описание данных и процедур, реализующих работу со списком контактирующих дискретных элементов;
- модуль подпрограммы *screening_contact_detection_2d*, реализующий алгоритм Screening contact detection algorithm в 2D-случае;
- модуль подпрограммы *munjiza_nbs_contact_detection_2d*, реализующий алгоритм Munjiza-NBS Contact Detection Algorithm в 2D-случае;
- модуль подпрограммы *screening_contact_detection_3d*, реализующий алгоритм Screening Contact Detection Algorithm в 3D-случае;
- модуль подпрограммы *munjiza_nbs_contact_detection_3d*, реализующий алгоритм Munjiza-NBS Contact Detection Algorithm в 3D-случае.

Входными данными для программного средства Demcda являются:

- номер алгоритма (1 или 2), по которому будет выполняться поиск контактирующих дискретных элементов;
- файл *bounds.txt*, в котором указываются размерность задачи и координаты границ расчетной области;
- файл *elements.txt*, в котором указываются количество, радиус и координаты центров дискретных элементов.

В результате работы программного средства Demcda формируется файл *contacts.txt*, в котором приводится список контактных пар дискретных элементов.

4. Численное исследование эффективности алгоритмов

Для верификации алгоритмов решена двумерная задача обнаружения контактных пар девяти круговых дискретных элементов диаметром D , равномерно распределенных по пространству и находящихся на расстоянии D друг от друга по направлению x и по направлению y (рис. 10).

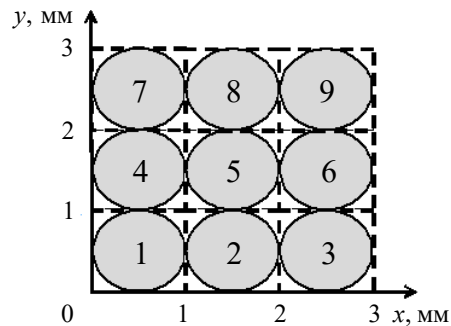


Рис. 10. Тест 1

По результатам расчетов с применением запрограммированных алгоритмов было выделено 12 контактных пар, представленных в таблице 1.

Таблица 1

Номера контактирующих дискретных элементов (двумерная задача)

№	1	2	3	4	5	6	7	8	9	10	11	12
ДЭ 1	9	9	8	8	7	6	6	5	5	4	3	2
ДЭ 2	6	8	5	7	4	3	5	2	4	1	2	1

Анализ таблицы 1 и рис. 10 подтверждает достоверность полученных результатов. Все контактные пары дискретных элементов выделены, и в их списке отсутствуют повторы.

Эта же задача решена в трехмерной постановке. Рассматривалось 27 ($3 \times 3 \times 3$) дискретных элементов сферической формы (рис. 11). Запрограммированные алгоритмы выделили 54 контактные пары дискретных элементов.

В таблице 2 приведены начало и конец выходного файла со списком номеров контактирующих дискретных элементов.

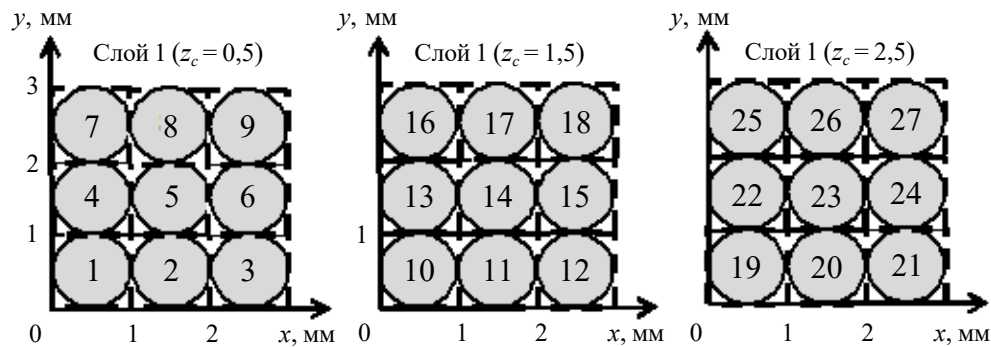


Рис. 11. Тест 2

Таблица 2

Список номеров контактирующих дискретных элементов (трехмерная задача)

№	1	2	3	4	5	6	...	50	51	52	53	54
ДЭ 1	27	27	27	26	26	26	...	5	5	4	3	2
ДЭ 2	26	24	18	25	23	17	...	4	2	1	2	1

Анализ таблицы 2 и рис. 11 подтверждает достоверность полученных результатов.

Для оценки быстродействия алгоритмов решены двумерный и трехмерный варианты этих задач, в которых количество дискретных элементов менялось от $2 \cdot 10^5$ до 10^7 .

Оценивалось время $T = t / \max(t_{3D})$ процессора, затрачиваемое на обнаружение контактных пар, отнесенное к времени $\max(t_{3D})$ решения трехмерной задачи для $N = 6 \cdot 10^6$.

На рис. 12 приведена зависимость T от общего числа дискретных элементов N , пунктирные линии – двумерная задача, сплошные линии – трехмерная задача, красные линии – скрининг-алгоритм обнаружения контакта, черные линии – алгоритм Munji-za-NBS.

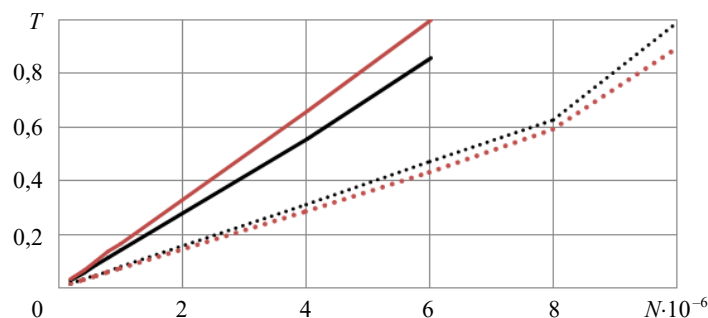


Рис.12. Зависимость времени работы процессора от общего числа дискретных элементов

В диапазоне количества дискретных элементов до $8 \cdot 10^6$ зависимость времени обнаружения контактных пар в двумерной задаче имеет линейный характер. Это означает, что время обнаружения контактных пар, затрачиваемое процессором, пропорционально общему числу дискретных элементов. При дальнейшем увеличении количества дискретных элементов сказывается недостаток оперативной памяти для решения задачи, что приводит к непропорциональному увеличению времени работы

процессора. С точки зрения быстродействия алгоритм обнаружения контактов Munjiza-NBS в двумерных задачах немного уступает скрининг-алгоритму. В трехмерных задачах, наоборот, алгоритм обнаружения контактов Munjiza-NBS превосходит скрининг-алгоритм.

Заключение

При одинаковом количестве дискретных элементов трехмерные задачи требуют большего времени работы процессора по сравнению с двумерными задачами, поскольку шаблон для проверки контактов в трехмерной задаче предполагает большее количество операций.

При малой плотности заполнения дискретными элементами пространства алгоритм обнаружения контактов Munjiza-NBS занимает меньше места в оперативной памяти, чем скрининг-алгоритм. С увеличением плотности заполнения это преимущество уменьшается.

С точки зрения быстродействия алгоритм обнаружения контактов Munjiza-NBS в двумерных задачах уступает скрининг-алгоритму. В трехмерных задачах, наоборот, алгоритм Munjiza-NBS превосходит скрининг-алгоритм, что объясняется особенностями взаимодействия с оперативной памятью.

При достаточном объеме оперативной памяти как скрининг-алгоритм обнаружения контактов, так и алгоритм Munjiza-NBS требуют время работы процессора, пропорциональное общему количеству дискретных элементов. В противном случае время работы процессора увеличивается нелинейно в зависимости от количества дискретных элементов.

Рассмотренные алгоритмы обнаружения контактных пар обеспечивают достоверность результатов и могут быть применены в программной реализации метода дискретных элементов для анализа динамики гранулированных сред.

References

1. Cundall P.A., Strack O.D.L. A discrete numerical model for granular assemblies. *Geotechnique*. 1979. Vol. 29. Iss. 1. P. 47–65. <https://doi.org/10.1680/geot.1979.29.1.47>.
2. Jing L., Stephansson O. Fundamentals of discrete element methods for rock engineering: theory and applications. *Developments in Geotechnical Engineering*. 2007. Vol. 85. P. 1–545. DOI: 10.1016/s0165-1250(07)x8500-6.
3. O'Sullivan C. *Particulate Discrete Element Modelling: A Geomechanics Perspective*. London. CRC Press. 2011. 574 p.
4. Radjai F., Dubois F. *Discrete-Element Modeling of Granular Materials*. Wiley-Iste. 2011. 425 p.
5. Jebahi M. et al. *Discrete Element Method to Model 3D Continuous Materials*. John Wiley & Sons. 2015. 166 p.
6. Thornton C. Granular dynamics, contact mechanics and particle system simulations. In: *Particle Technology Series*. 2015. Vol. 24. 195 p. <https://doi.org/10.1007/978-3-319-18711-2>.
7. Ji S., Liu L. Computational granular mechanics and its engineering applications. In: *Springer Tracts in Mechanical Engineering*. New York. Springer. 2020. 387 p.
8. O'Connor R.M. A distributed discrete element modeling environment: algorithms, implementation and applications. *D. Sci. Dissertation in Civil and Environmental Engineering*. Cambridge, MA, USA. 1996. 203 p.
9. Van Baars S. Discrete element analysis of granular materials. *Doctoral Thesis*. Delft, Netherlands. 1996. 133 p.
10. Ouafdel H. Numerical simulations of granular assemblies with three-dimensional ellipsoid-shaped particles. *Doctoral Thesis*. Ouafdel, Canada. 1998. 204 p.

11. Raji A.O. Discrete element modelling of the deformation of bulk agricultural particulates. *Doctoral Thesis*. Newcastle, UK. 1999. 165 p.
12. Wan Y. Discrete element method in granular material simulations. *Master Thesis*. Kaiserslautern, Deutschland. 2011. 79 p.
13. Wellmann C. A two-scale model of granular materials using a coupled DE-FE approach. *Doctoral thesis*. Hannover, Deutschland. 2011.
14. Chen J. Discrete element method for 3d simulations of mechanical systems of non-spherical granular materials. *Doctoral Thesis*. Tokyo. 2012. 183 p.
15. Samiei K. Assessment of implicit and explicit algorithms in numerical simulation of granular matter. *Doctoral Dissertation*. Luxembourg. 122 p.
16. Vu-Quoc L., Zhang X., Walton O. R. A 3-D discrete-element method for dry granular flows of ellipsoidal particles. *Computer Methods in Applied Mechanics and Engineering*. 2000. Vol. 187. Iss. 3-4. P. 483–528. [https://doi.org/10.1016/S0045-7825\(99\)00337-0](https://doi.org/10.1016/S0045-7825(99)00337-0).
17. Ghaboussi J., Barbosa R. Three-dimensional discrete element method for granular materials. *International Journal for Numerical and Analytical Methods in Geomechanics*. 1990. Vol. 14. Iss. 7. P. 451–472. <https://doi.org/10.1002/nag.1610140702>.
18. Lin X., Ng T.-T. Contact detection algorithms for three-dimensional ellipsoids in discrete element modelling. *International Journal for Numerical and Analytical Methods in Geomechanics*. 1995. Vol. 19. Iss. 9. P. 653–659. <https://doi.org/10.1002/nag.1610190905>.
19. Bićanić N. Discrete element methods. *Encyclopedia of Computational Mechanics. Vol. 1. Fundamentals*. Eds. E. Stein, R. De Borst, T.J.R. Hughes. John Wiley & Sons. 2004. P. 311–337. <https://doi.org/10.1002/0470091355.ecm006.pub2>.
20. Nezami E.G., Hashash Y.M.A., Zhao D, Ghaboussi J. Shortest link method for contact detection in discrete element method. *International Journal for Numerical and Analytical Methods in Geomechanics*. 2006. Vol. 30. Iss. 8. P. 783–801. <https://doi.org/10.1002/nag.500>.
21. Zhao D., Nezami E.G., Hasha Y.M.A., Ghaboussi J. Three-dimensional discrete element simulation for granular materials. *Engineering Computations*. 2006. Vol. 23. Iss. 7. P. 749–770. <https://doi.org/10.1108/02644400610689884>.
22. Wachs A., Girolami L., Vinayet G., Ferrer G. Grains3D, a flexible DEM approach for particles of arbitrary convex shape – Part I: Numerical model and validations. *Powder Technology*. 2012. Vol. 224. P. 374–389. <https://doi.org/10.1016/j.powtec.2012.03.023>.
23. Zheng Z., Zang M., Chenet S., Zhaoal C. An improved 3D DEM-FEM contact detection algorithm for the interaction simulations between particles and structures. *Powder Technology*. 2017. Vol. 305. P. 308–322. <https://doi.org/10.1016/j.powtec.2016.09.076>.
24. Seelen L.J.H., Padding J.T., Kuipers J.A.M. A granular discrete element method for arbitrary convex particle shapes: method and packing generation. *Chemical Engineering Science*. 2018. Vol. 189. P. 84–101. <https://doi.org/10.1016/j.ces.2018.05.034>.
25. Rakotonirina A.D., Delenne J.Y., Radjai F., Wachs A. Grains3D, a flexible DEM approach for particles of arbitrary convex shape – Part III: extension to non-convex particles modelled as glued convex particles. *Computational Particle Mechanics*. 2019. Vol. 6. P. 55–84. <https://doi.org/10.1007/s40571-018-0198-3>.
26. Lai Z., Chen Q., Huang L. Fourier series-based discrete element method for computational mechanics of irregular-shaped particles. *Computer Methods in Applied Mechanics and Engineering*. 2020. Vol. 362. Article No 112873. <https://doi.org/10.1016/j.cma.2020.112873>.
27. Zhang P., Dong Y., Galindo-Torres S.A., Scheuermann A., Li L. Metaball based discrete element method for general shaped particles with round features. *Computational Mechanics*. 2021. Vol. 67. P. 1243–1254. <https://doi.org/10.1007/s00466-021-02001-9>.
28. Alonso-Marroquín F., Wang Y. An efficient algorithm for granular dynamics simulations with complex-shaped objects. *Granular Matter*. 2009. Vol. 11. P. 317–329. <https://doi.org/10.1007/s10035-009-0139-1>.
29. Munjiza A. *The Combined Finite-Discrete Element Method*. 2004. UK. West Sussex. John Wiley & Sons Ltd. 333 p.
30. Pöschel T., Schwager T. *Computational Granular Dynamics: Models and Algorithms*. Berlin. Heidelberg. New York. Springer Science & Business Media. 2005. 322 p. <https://doi.org/10.1007/3-540-27720-X>.

31. Hogue C. Shape representation and contact detection for discrete element simulations of arbitrary geometries. *Engineering Computations*. 1998. Vol. 15. Iss. 3. P. 374–390. <https://doi.org/10.1108/02644409810208525>.

32. Sutmann G. Classical molecular dynamics. In: *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms, Lecture Notes*. Eds. J. Grotendorst, D. Marx, A. Muramatsu. 2002. Vol. 10. P. 211–254.

EFFICIENCY ANALYSIS FOR THE CONTACT DETECTION ALGORITHMS IN THE DISCRETE ELEMENT METHOD*

Gorokhov V.A., Kibets A.I., Nagornyykh E.V.

*National Research Lobachevsky State University of Nizhny Novgorod,
Nizhny Novgorod, Russian Federation*

kibec@mech.unn.ru

Received by the Editor 2024/10/02

The proposed article provides a brief overview of publications devoted to solving problems of granular media mechanics using the discrete element method. These publications discuss the conditions for the successful application of algorithms and software developed for the effective functioning of computing systems implementing the discrete element method. It is noted that the most labor-intensive aspect of modeling deformation processes using the discrete element method are calculations associated with determining contact forces. The key problem is the evolutionary nature of contacts: during deformation of the medium under study, discrete elements can come into contact, which can subsequently be interrupted depending on the loading conditions. In this regard, it is necessary to carefully study the organization and management of memory associated with computer modeling of contacts. The main attention in the article is paid to algorithms for searching for discrete elements coming into contact. These algorithms should be robust (i.e. reliable), easy to implement; effective in terms of speed and use of RAM. Two variants of the algorithm based on the use of auxiliary (screening) grids are compared: the screening contact detection algorithm and the Munjiza-NBS contact detection algorithm. A brief description of these algorithms and their software implementation is given. The results of verification of the algorithms on test problems and a comparative analysis of their performance are presented. It is shown that the selected algorithms provide reliable information about discrete elements contacting during motion and can be successfully used in the software implementation of the discrete element method as applied to problems of granular media dynamics.

Keywords: nonlinear, non-stationary, deformation, discrete element method, contact, search for contacting elements.

*This study was conducted within the scientific program of the National Center for Physics and Mathematics, stage #2 “Mathematical Modeling on Exa-scale and Zetta-scale supercomputers. Stage 2023–2025”.