

РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ ДЛЯ МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ СТРОЕМ С ИНТЕГРАЦИЕЙ В ПЛАТФОРМУ «ОПТИМУС»

Н. А. Баринов, Д. В. Ежов, Д. С. Собанин

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Описание проблем

Имитационное моделирование позволяет решать множество задач, в которых иные методы работают плохо. Особенно это относится к динамике поведения людей, в том числе во время боевых действий. Движение агентов (людей, единиц техники и т. д.) в таких случаях, как правило, происходит не хаотично, а в выбранном строю. Поэтому очень важной задачей в имитационном моделировании является задача движения агентов в строю в пространстве, насыщенном различными препятствиями.

Движение в строю можно разделить на набор следующих подзадач:

- движение строя в целом;
- поддержание места в строю;
- избегание столкновения между агентами;
- избегание проникновения в непроходимые зоны (препятствия);
- обход препятствий;
- выход из тупиков.

Под движением строя в целом понимается движение точки, от координат которой отсчитываются координаты каждого из агентов в этом строю.

Поддержание места в строю означает следование той точке в строю, что задана для агента. При этом строй может меняться по заданной программе и в таком случае агент должен отреагировать на это изменение. Эта проблема порождает проблему гашения колебаний, которые образуются как при внешнем воздействии на агента, так и при изменении строя.

Если по каким-то причинам агент попал в тупик, то он должен найти путь выхода оттуда и присоединиться к строю.

Предлагаемый вариант решения

Для решения выше указанных проблем было предложено разделить навигацию на два «уровня».

На верхнем уровне ищется путь для группы агентов в целом. При этом используется модуль Navmesh [1]. Этот модуль представляет пространство в виде набора выпуклых многогранников (полигонов) и обеспечивает поиск оптимального пути между этими многогранниками по проходимым областям. Таким образом получается путь для группы агентов в целом.

Нижний уровень навигации – это движение собственно агента. Он должен поддерживать заданное ему место в строю (может меняться со временем в соответствии с программой), избегать столкновения друг с другом и не проникать в препятствия, а в случае попадания в тупик выходить из него. На этом уровне управление движением агента происходит с помощью «сил», обеспечивающих изменение ускорения, а через него и скорости агента. Эти «силы» аналогичны физическим по свойствам: могут векторно складываться, вычитаться, умножаться на число и т. д. Действуя на агент, они также порождают ускорение, как и обычная сила.

В случае столкновения с препятствием агент движется, используя алгоритмы модуля Navmesh, позволяющие двигаться вдоль препятствий, не нарушая их границ. В случае захода в тупик вновь задействуется модуль Navmesh для получения индивидуального пути для этого агента к его месту в строю. Во всех иных случаях движение обеспечивается с помощью расчета «сил»

Применяемые алгоритмы

Конфигурация строя в целом, так же, как относительное положение каждого агента в строю в каждый момент, являются заданными.

При движении агенты взаимодействуют с соседними объектами. Ими могут быть как агенты, так и препятствия. Для того, чтобы не тратить время на рассмотрение объектов, заведомо не являющимися соседними, необходим поиск по пространственному индексу.

Поиск по пространственному индексу реализован с помощью разбиения на прямоугольные ячейки всего пространства моделирования. Соседними являются агенты и препятствия, которые попадают в ячейку, в которой находится агент, и непосредственно соседствующие с ней ячейки.

При движении строя, как единого объекта, задается траектория точки, относительно которой каждый из агентов, составляющих строй, имеет определенное местоположение, которое он должен занимать.

Для моделирования движения в строю требуется, чтобы агенты следовали к заданному для текущего момента положению, при этом избегали столкновений друг с другом и с препятствиями. Кроме того, необходимо, чтобы эти движения были реалистич-

ными. В данной модели для решения этих задач за основу взят подход, описанный в [2] с некоторыми добавлениями из [3].

В модель вводятся величины, аналогичные физическим силам. Также, как и физические силы, они являются векторами, но безразмерными. Эти «силы» приложены к агентам и связываются с ускорением соотношением:

$$\vec{a} = \vec{F} / M \quad (1)$$

В зависимости от степени инерционности объекта величина m , соответствующая массе в физическом мире, может быть разной. В данной модели принято $m = 1$, так как агенты считаются одинаковыми и имеют одинаковую инерцию.

Скорость и путь на одном шаге расчета определяются формулами

$$\vec{v} = \vec{a} * \Delta t \quad (2)$$

$$\vec{s} = \vec{v} * \Delta t + \frac{\vec{a} * \Delta t^2}{2} \quad (3)$$

Здесь Δt – шаг по времени.

Агенты движутся в соответствии с вычисленными силами, однако, в случае столкновения с непреодолимой зоной (т. е. препятствием), они перемещаются вдоль границ этих зон, руководимые модулем Navmesh.

1. Расчет силы, действующей на агента

Эти «силы» складываются из нескольких «сил», моделирующих отдельные аспекты движения:

$$\vec{F} = \vec{F}_{\text{строй}} + \vec{F}_{\text{трения}} + \sum_i \vec{F}_{\text{расталк}} + \vec{F}_{\text{преп}} \quad (4)$$

Здесь $\vec{F}_{\text{строй}}$ – «сила», определяющая движение в направлении точки, которую в данный момент должен занимать агент. $\vec{F}_{\text{трения}}$ – «сила», обеспечивающая демпфирование колебаний относительно заданной для агента скорости в движущемся строю. – «сила» отталкивания агента, к которому приложена эта сила, с агентом i . $\vec{F}_{\text{преп}}$ – сила, действующая на агента со стороны препятствия.

2. Расчет $\vec{F}_{\text{строй}}$ – «силы», определяющей движение в направлении точки, которую в данный момент должен занимать агент

Эта «сила» определяется по формуле:

$$\vec{F}_{\text{строй}} = \vec{r} + \vec{v} * \Delta t - \vec{r}_{\text{задан}} \quad (5)$$

Здесь \vec{r} – текущее положение агента, $\vec{r}_{\text{задан}}$ – заданное положение агента, \vec{v} – текущая скорость, Δ – шаг по времени. На рис. 1 показан принцип определения этой силы.

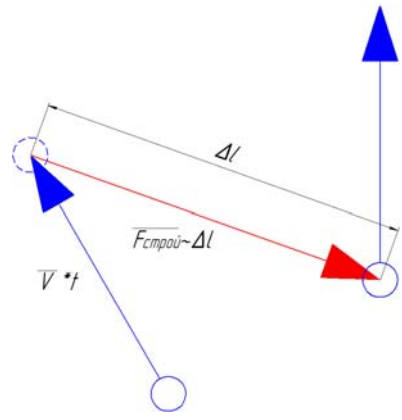


Рис. 1. Определение «силы», ведущей агента к заданному месту в строю

3. Расчет $\vec{F}_{\text{трения}}$ – «силы», обеспечивающей демпфирование колебаний относительно заданной для агента скорости в движущемся строю

$$\vec{F}_{\text{трения}} = k * (\vec{V}_{\text{строй}} - \vec{V}_{\text{агента}}) \quad (6)$$

Заданная скорость агента – относительно неподвижной точки.

4. Расчет $\vec{F}_{\text{расталк}}^i$ – «силы» отталкивания агентов

$\vec{F}_{\text{расталк}}^i$ – «сила» отталкивания (см. рис. 2) текущего агента от агента i .

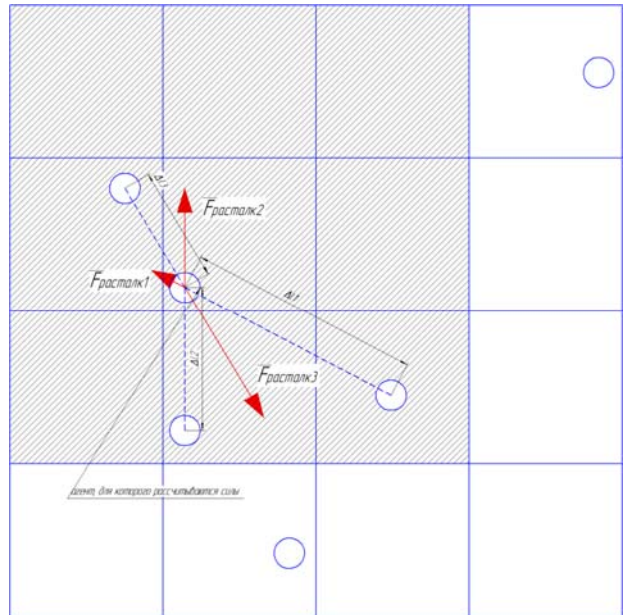


Рис. 2. Определение «силы» отталкивания. Штриховкой показаны соседние с рассматриваемым агентом ячейки. Агенты, находящиеся в не соседних ячейках не учитываются

Она направлена противоположно направлению вектора от центра рассматриваемого агента до агента i и по величине равна

$$|\vec{F}_{\text{расталк}}| = \frac{k}{\Delta l^2} \quad (7)$$

Здесь Δl – расстояние между агентами, м; $k=1 \text{ м}^2$ – коэффициент, позволяющий соблюсти размерность. «Силы» отталкивания не допускают столкновения агентов. Все силы от агентов, ближайших к рассматриваемому агенту, векторно суммируются. Ближайшими агентами считаются те, что попадают в соседние ячейки (показаны на рис. 2 заштрихованными). Остальные агенты игнорируются (их ячейки показаны незаштрихованными).

5. Расчет «силы», позволяющей избежать препятствия

Для избегания столкновения с препятствием вводится «сила», уводящая агент от препятствия. На пути агента возможно нахождение множества препятствий. Наиболее важным считается первое, лежащее на пути.

В случае, если агент не движется, данная сила не вычисляется.

При вычислении силы определяется линия, по которой движется агент. Первой ее точкой является текущее положение агента, второй – предположительное положение агента при движении, вычисляемое по формуле

$$\vec{l}_{\text{motion}} = \vec{p} + \vec{v} * t + \vec{a} * \frac{t^2}{2} \quad (8)$$

Здесь \vec{p} – исходное положение, \vec{v} – скорость, \vec{a} – ускорение.

Далее перебираются все близко расположенные к агенту препятствия. Для поиска расстояния до препятствия используется алгоритм, изложенный в 1.7. Линия, по которой движется агент, аппроксимируется прямой. Тогда ее можно представить в параметрическом виде:

$$\vec{r}1 = \vec{r}0 + \vec{k} * l \quad (9)$$

Здесь \vec{k} – вектор между начальной и предполагаемой точкой \vec{l}_{motion} , l – параметр. Если он попадает в интервал от 0 до 1, то точка $\vec{r}1$ находится на векторе движения.

Аналогичным образом параметризуется и линия препятствия:

$$\vec{r}1_{\text{obs}} = \vec{r}0_{\text{obs}} + \vec{k}1 * l1 \quad (10)$$

Тогда, решая систему уравнений, можно найти значения параметров l и $l1$ для точки пересечения этих линий. Если оба параметра при этом принадлежат интервалу 0...1, то линии пересекаются. Если

$l1$ входит в этот интервал, а $l > 1$, то пересечения нет, но препятствие лежит на пути агента. В общем случае

$$l = \frac{r0.x - r0obs.x - \frac{k1.x}{k1.y} * ro.y + \frac{k1.x}{k1.y} * ro.obs.y}{k1.x * \frac{k.y}{k1.y} - k.x} \quad (11)$$

Когда $k1.x * (k.y/k1.y) - k.x = 0$, данная формула дает деление на 0. В таком случае возможны два варианта. Первый – если $r1.x = r0obs.x$, то агент находится в точке пересечения с препятствием, т. е. $l = 0$. Второй – движение будет параллельно препятствию и, в таком случае, $l = \infty$.

Когда $k1.y = 0$, координата по y равна этой координате препятствия. Предположительная координата пересечения по x вычисляется следующим образом:

$$\text{predpol_crossing_point_X} = ro.x + k.x * l \quad (12)$$

$$\text{Здесь } l = (r0_obstacle_y - r0.y) / k.y \quad (13)$$

Но в этом случае координата должна попадать в интервал между $r0obs.x$ и $r1obs.x$. Иначе пересечения линий тоже не происходит.

Если предполагаемая траектория агента проходит через несколько препятствий, то для дальнейшего рассмотрения выбирается ближайшее из таковых препятствий.

Когда выбрано препятствие, столкновения с которым следует избежать, вычисляется сила (см. рис. 3), позволяющая это сделать. В случае если агент на текущем шаге пересекает препятствие (при неизменных параметрах движения агента), необходимо определить, в какую сторону от препятствия следует уклоняться.

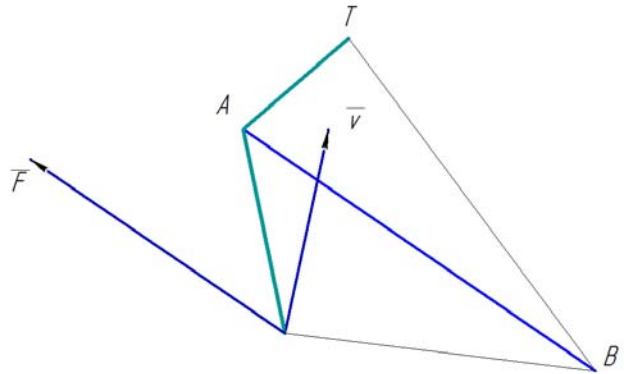


Рис. 3. Определение силы избегания препятствий

Вектор силы направлен параллельно препятствию и равен 100 Н. Поскольку препятствие представляет собой отрезок или последовательность отрезков, то движение параллельно препятствию означает движение параллельно ближайшей его стороне.

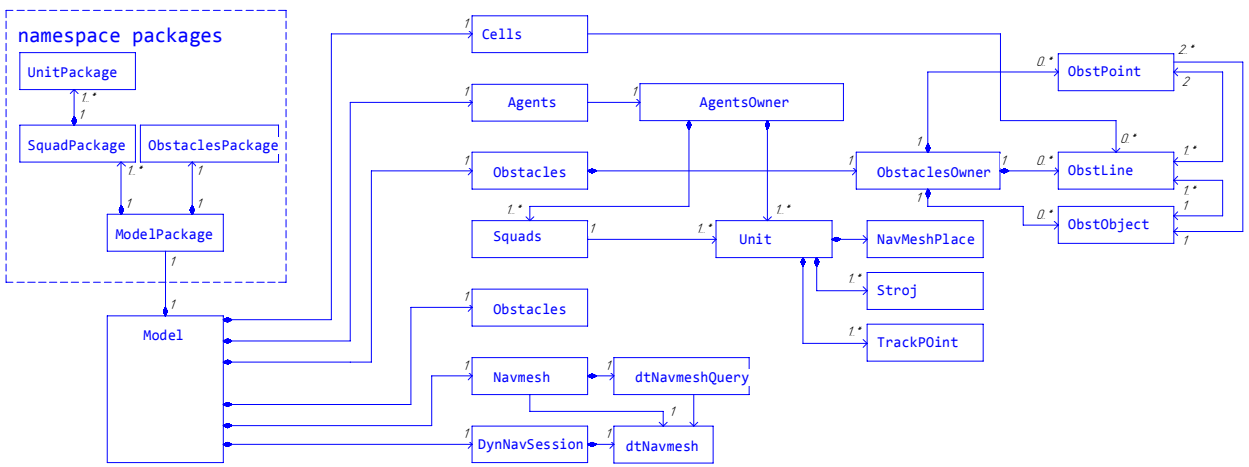


Рис. 4. Диаграмма классов

Так как $m = 1$ кг, а максимальное ускорение заведомо меньше 100 м/с^2 , то при ограничении суммы действующих на агента сил, равнодействующая будет практически совпадать по направлению с данным вектором.

В случае, если препятствие только лежит на пути агента, но на текущем шаге оно не будет достигнуто, сила равна 100 Н и направлена вдоль перпендикуляра от линии препятствия.

6. Расчет движения в случае захода в тупик

Ситуация попадания агента в тупик определяется по разнице между движением агента без учета препятствий и реальным движением агента. На каждом шаге находится изменение координат, которое произошло бы, если бы препятствий не было. Затем рассчитывается аналогичная величина уже с учетом препятствий. После этого изменение координат с учетом препятствий вычитается из изменения координат без учета препятствий и находится модуль итоговой разницы. Эта итоговая разница суммируется на каждом шаге до тех пор, пока не достигнет заданного предела. Если этот предел преодолевается, то считается, что агент застрял. Предел в текущей реализации положен равным 1 м.

Если агент застрял, то начинается поиск выхода из тупика. От агента посылаются «лучи» – отрезки заданной длины, начало которых лежит в точке агента, а конец – на заданном расстоянии в заданном направлении. Цель этого – найти «луч», не пересекающийся с препятствием. Это и будет направлением выхода из тупика для агента. Первый «луч» посылается вдоль направления движения, второй – с углом в 10 градусов по часовой стрелке от первого, третий с углом 10 градусов от первого против часовой стрелки и так далее. Если среди них находится такой «луч», что не встречает препятствий, то перебор «лучей» прекращается и агент получает ускорение вдоль этого луча.

Программная реализация

Программный модуль состоит из расчетной части, написанной на C++ и ответственной за визуализацию части, написанной на Python.

Расчетная часть написана с применением методологии объектно-ориентированного программирования.

В программе разделяется ввод данных и их обработка. Основным классом программы является класс Model, включающий в себя структуры:

- Cell – отвечает за разбиение пространства на ячейки для локальной навигации
- Navmesh – обеспечивает доступ к функционалу Navmesh библиотеки Detour [1]
- DynNavSession – обеспечивает построение навигационной сетки.
- Obstacles – класспрепятствий.
- Squad – классоднойгруппыобъектов
- Agents – классагентов

Структура представлена на рис. 4. Класс Vector 3D условно не показан.

Визуализационная часть разбита на три файла: главный файл visualizer.py, ответственный за считывание данных из файлов с результатами расчетов xml_reader.py и модуль визуализации vis_base.py.

Использование модуля движения строем

В качестве примера работы программы взято движение 200 агентов. Время конца моделирования 50 с, шаг 0,01 с, длина ячейки 0,5 м, масштаб содержимого файла obs.svg по x 0,0009, по y 0,0009. Движение группы агентов происходит по заранее заданному в Input.xml маршруту. В заданный момент изменяется строй. Для демонстрации пути отдельного агента выбран агент под номером 1 с координатой относительно нулевой точки строя $\{1, 0\}$.

В начальный момент времени агенты выстроены в 10 шеренг по 20 агентов в каждом. Координаты

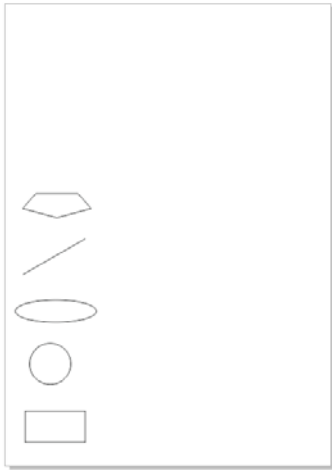


Рис. 5. Препятствия, представленные в файле obs.svg

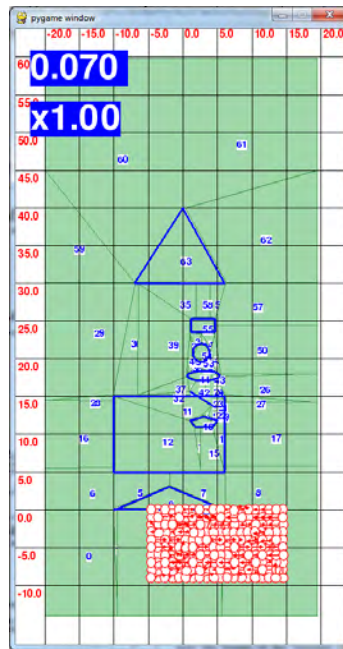


Рис. 6. Начальное положение

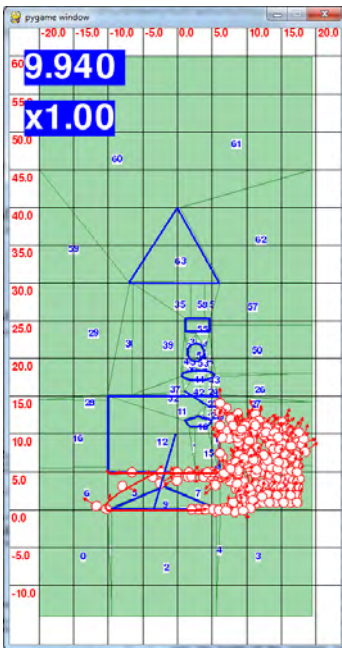


Рис. 7. Строй при обходе препятствия

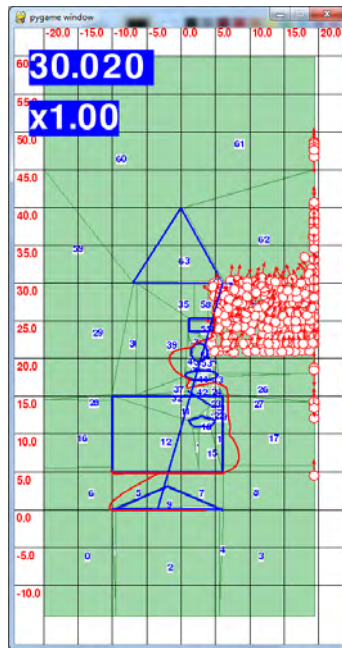


Рис. 8. Группа агентов перед перестроением

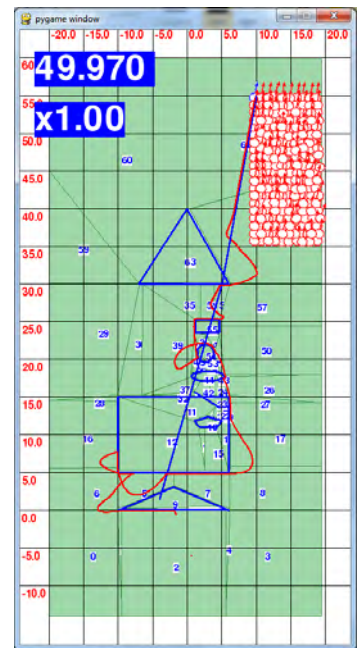


Рис. 9. Группа агентов в конце движения

точки, относительно которой даются координаты в строю $\{-4,5; 0\}$. Расстояние между агентами в шеренге и между шеренгами по 1.

Далее к моменту 30 с с начала моделирования группа агентов должна достичь точки с координатами $\{5;30\}$. При этом первый агент должен занять точку с координатами $\{6;30\}$, второй агент точки с координатами $\{7;30\}$ и так далее. В этот момент агенты начинают менять строй на шеренги по 10 агентов в каждой. К моменту 50 с отделение

достигает точки $\{10;55\}$. При этом агентам встречаются препятствия, описанные в Input.xml и в созданном в CorelDraw файле obs.svg.

В файле Input.xml записаны координаты линий трех препятствий: прямоугольника, равностороннего треугольника и равнобедренного треугольника.

В файле obs.svg представлены: прямоугольник, круг, эллипс, прямая, пятиугольник. На рис. 5 представлен пример препятствий, нарисованных в CorelDraw и использованный в тестовом расчете.

Важно, что в CorelDraw картинка развернута на 180 градусов в силу иного расположения осей координат.

На рис. 6 представлено положение агентов сразу после старта.

На рис. 7 представлен строй во время обхода препятствия, когда точка, от которой отсчитываются координаты внутри строя находится внутри препятствия.

На рис. 8 представлена группа агентов перед перестроением.

На рис. 9 представлена группа агентов в конце движения.

Выводы

Разработан программный модуль, позволяющий моделировать в двумерном режиме движение строя в целом, поддержание места в строю каждым из агентов, избегание столкновения между агентами, избегание проникновения в непроходимые зоны (препятствия), обход препятствий, выход из тупиков. Программный модуль выполнен на C++ (расчетная часть) и Python 3.5 (часть, ответственная за визуализацию результатов. При тестировании показана возможность задания движения 200 объектов в пространстве, наполненном препятствиями.

Литература

1. Коваленко О. В., Крючков И. А., Ерошкина И. В., Собанин Д. С., Хочкин Н. И. Алгоритмы генерации графодинамической модели по планам охраны и обороны производственных площадок в ПО ОЭ СФЗ //ОтчетВНИИЭФ инв.№ 01/6148 дсп. Саров, 2017.

2. Reynolds C. Steering behaviors for autonomous characters // Game developers conference. 1999. P. 763–782.

3. Искусственный интеллект: современный подход, 2-е изд.. : Пер. с англ. М.: Издательский дом «Вильямс», 2007. С. 1408. : ил. Парал. тит. англ. ISBN 978-5-8459-0887-2 (рус.)

4. Корн Г., Корн Т. Справочник по математике (для научных работников и инженеров). М.: Наука, 1973. С. 832.