

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
МАТЕМАТИКО-МЕХАНИЧЕСКИЙ ФАКУЛЬТЕТ  
КАФЕДРА СИСТЕМНОГО ПРОГРАММИРОВАНИЯ

---

**Моделирование взаимодействий систем твердых тел**

*Дипломная работа студента 59-А группы  
Симонова Станислава Вадимовича*

Научный руководитель: ..... Уфнаровский В.В.

Рецензент: ..... к.ф.-м.н. Кучкова И.Н.

“Допустить к защите”,  
заведующий кафедрой,  
д.ф.-м.н., профессор ..... Терехов А.Н.

*Санкт-Петербург  
2003*

## Оглавление

Оглавление .....	2
Введение .....	3
Уравнения движения тела в пространстве .....	4
Обзор существующих алгоритмов моделирования взаимодействий твердых тел .....	5
Collision detection алгоритмы .....	6
Алгоритмы обработки взаимодействия тел .....	8
Алгоритмы, основанные на ограничениях (Constrained-based алгоритмы)....	9
Предлагаемый алгоритм моделирования взаимодействий систем твердых тел ..	11
Особенности постановки задачи .....	11
Общая схема работы предлагаемого алгоритма.....	11
Collision detection алгоритм .....	13
Обработка взаимодействия тел .....	14
Вывод величины импульса, получаемого телом при столкновении .....	14
Обработка столкновений .....	16
Алгоритм верификации положения тела. ....	17
Основные отличия предлагаемого алгоритма от уже существующих решений .....	18
Заключение.....	19
Литература.....	20

## Введение

В данной работе будет рассмотрена проблема моделирования взаимодействий твердых тел. При обсуждении данного вопроса, прежде всего хочется показать в каких областях программирования он актуален, а также какие существуют достижения и разработки в данной области.

В настоящее время существует много программных систем, имеющих дело с моделированием реального мира и процессов, протекающих в нем. К таким системам относятся всевозможные тренажеры для обучения управлению каким-либо транспортным средством, системы проектирования сложных механических объектов, которые помимо функциональности, помогающей непосредственно созданию объекта, должны иметь функциональность, отвечающую за тестирование поведения этого объекта в реальном мире (например, в тестах на прочность, аэродинамику и т.д.). Также, системы VR (virtual reality – виртуальной реальности), несущие как развлекательную, так исследовательскую нагрузки, требуют наличия модуля реалистичной физики. Проект, в процессе работы над которым, появилась необходимость в исследовании вопроса моделирования поведения физических тел – тактический симулятор, подобный тем, которые используются для отработки действий спецслужб в чрезвычайных ситуациях.

Далее рассмотрим проблемы, возникающие при попытке создания системы, с большой степенью точности моделирующей взаимодействие тел в пространстве. Первая, и самая основная, состоит в том, что теоретическая физика не имеет точной математической модели, для описания сил, действующих на тела при их взаимодействии. Поэтому, во всех моделях, разработанных ранее, как и в модели, предлагаемой в данном исследовании, применяются некоторые приближенные математические модели, дающие результаты, похожие на те, что были получены эмпирически. Вторая проблема связана с тем, что все разработанные алгоритмы моделирования взаимодействия твердых тел требуют больших вычислительных ресурсов, в то время как системы моделирования, о которых мы говорим, являются real-time системами, т.е. системами, в которых время работы одного такта, а также время отклика системы на любое изменение, измеряется десятками и сотыми долями секунд. В первую очередь необходимость больших вычислительных затрат связана с частью процесса моделирования, ответственной за расчет точек взаимодействия тел. Как известно, во всех системах, имеющих дело с изображением объемных объектов, эти объекты представлены множеством треугольников, полученных в процессе триангуляции реального объекта. Соответственно, блок алгоритма моделирования, ответственный за просчет точек взаимодействия (точек столкновения), на каждом шаге вычисляет точки пересечения между всевозможными объектами в моделируемой сцене, т.е. между всевозможными треугольниками, принадлежащими разным объектам. Проблема в том, что количество треугольников в моделируемой сцене может достигать нескольких сотен тысяч, что делает метод полного перебора абсолютно неприемлемым в условиях real-time системы.

Дальнейшие трудности, возникающие перед программистом, занимающимся разработкой системы моделирования взаимодействия твердых тел, связаны с несовершенством разработанных моделей и подходов, применяемых для расчета изменений движения тел при их взаимодействии. Как будет показано в следующих главах, на сегодняшний день существуют две основные модели, известные как constrained-based подход (подход, основанный на ограничениях) и impulse-based подход (импульсный подход). Недостаток первого связан с тем, что при во-первых его применение ограничено его

конструктивными особенностям, о которых будет сказано чуть позднее (так, например, данный метод практически неприменим в случае, описанном на рис. 1, т.е. результат он выдавать будет правильный, но при этом будет затрачивать огромные вычислительные мощности)

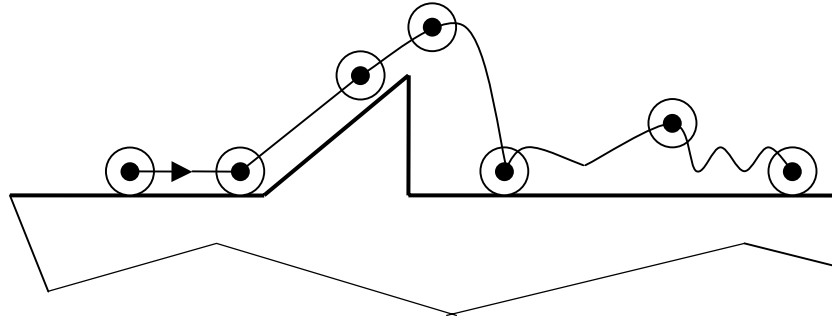


Рис. 1

Недостатками импульсной модели являются высокая степень условности используемой математической модели физических взаимодействий, а также огромное число обрабатываемых столкновений. Если в Constrained-based модели отдельно разбираются случаи вращения и скольжения, то в impulse-based модели они обрабатываются по общему алгоритму, что приводит к обработке на порядок большего числа столкновений, чем в моделях, основанных на ограничениях. В некоторых системах частота возникновения столкновений достигает 1-3 kHz.

В данной работе будет сделан обзор упомянутых двух алгоритмов, а также предложен новый. Будет также показано его преимущество.

## Уравнения движения тела в пространстве

Перед описанием существующих алгоритмов моделирования взаимодействий твердых тел, необходимо сделать отступление в теоретическую механику и ввести математический аппарат, описывающий движение тел в пространстве.

Введем набор параметров, однозначно задающих положение тела (т.е. любой его точки) в пространстве:

- $x(t)$  – положение тела в пространстве
- $R(t)$  – ориентация тела в пространстве

Тогда

$$p(t) = R(t) \cdot p_0 + x(t),$$

где  $p(t)$  положение точки в пространстве в момент времени  $t$ ,  $p_0$  - начальное положение этой точки в нулевой момент времени.

Далее введем параметры, отвечающие за скорость движения тела:

- $v(x)$  – линейная скорость движения.  $v(x) = \dot{x}(t)$
- $\omega(x)$  - угловая скорость вращения:  $\dot{R}(t) = \omega(t) \times R(t)$

Для учета сил, действующих на тело (пользуясь законами Ньютона[1]) (например, сила тяжести и т.д.) введем понятие вращения

$$\tau(t) = \sum (r_i(t) - x(t)) \times F_i(t),$$

где  $F_i$  - силы действующие на тело,  $r_i$  - точки их приложения.

Далее, чтобы упростить уравнения движения тела, введем понятия линейного и углового моментов движения

- $P(x)$  – линейный момент  $P(x) = M \cdot v(x)$
- $L(x)$  – угловой момент  $L(t) = I(t) \cdot \omega(t)$ , где  $I(t)$  - тензор инерции тела,  $I(t) = R(t)I_{body}R(t)^T$

Из теоретической механики известны следующие соотношения:

$$\dot{P}(t) = F(t)$$

$$\dot{L}(t) = \tau(t)$$

Соответственно, эти соотношения позволяют нам, зная силы, действующие на тело, и точки их приложения, выяснить скорости движения тела в пространстве. Таким образом, можно ввести вектор состояния тела, однозначно задающий его положение в пространстве и угловую и линейную скорости движения

$$Y(t) = \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix}.$$

Уравнения движения тела тогда выглядят следующим образом:

$$v(t) = \frac{P(t)}{M}$$

$$\omega(t) = I(t)L(t)$$

$$I(t) = R(t)I_{body}R(t)^T$$

Таким образом, зная силы, действующие на тело и точки их приложения, а также начальное состояние тела, мы можем вычислить вектор состояния тела  $Y(t)$  в любой момент времени  $t$ .

## Обзор существующих алгоритмов моделирования взаимодействий твердых тел.

Рассмотрим процесс моделирования движения твердых тел. При попытке построения корректной модели взаимодействия тел, возникает ряд серьезных ограничений. Первое заключается в том, что все процессы, протекающие в реальном мире, в том числе и взаимодействие твердых тел, непрерывны. В условиях же компьютерного моделирования мы можем иметь дело только с дискретными процессами (строго говоря, моделирование непрерывного течения времени иногда возможно, но только на определенных этапах, таких как вычисление точек и времени взаимодействия тел. На всех же остальных этапах приходится иметь дело с дискретным течением времени) В силу этого, непрерывность течения времени имитируется разбиением процесса на временные такты, длительность которых выбирается настолько малой, насколько позволяют вычислительные ресурсы.

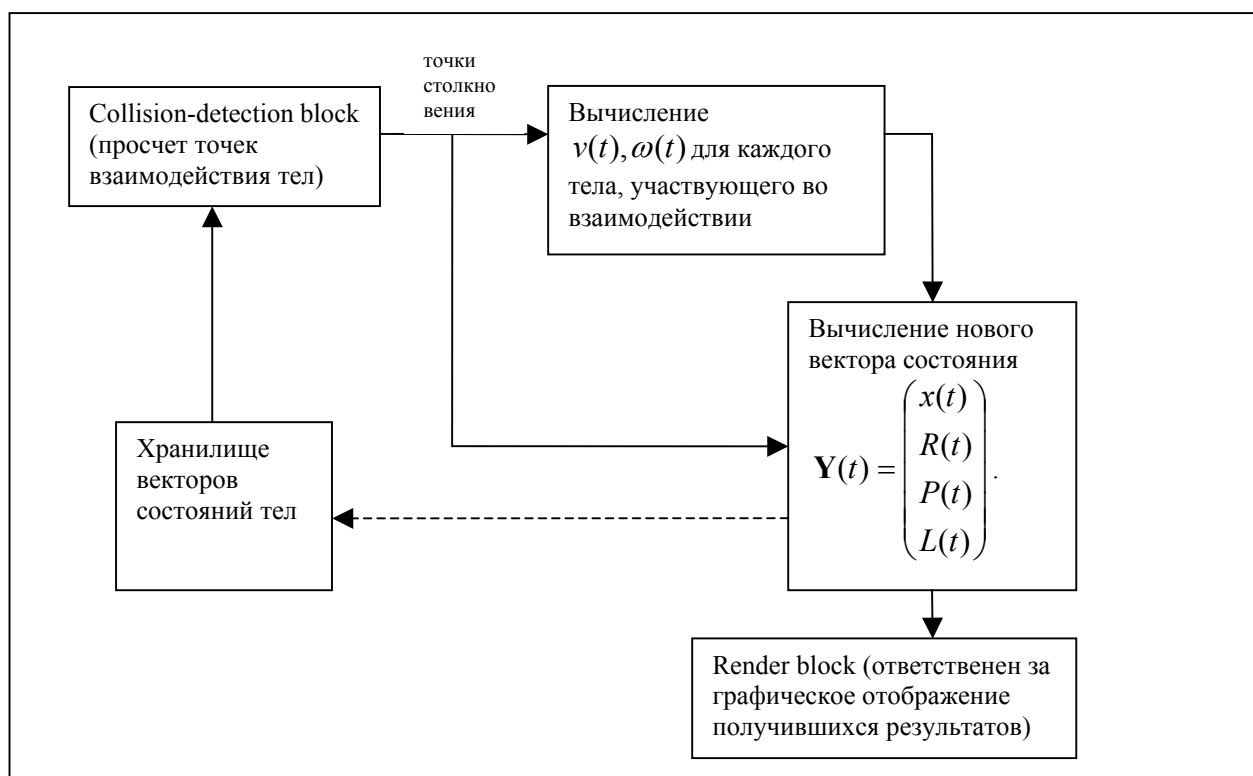


Рис. 1

За годы работы исследователей над проблемой моделирования взаимодействия систем тел сложилась вполне определенная схема работы таких алгоритмов, показанная на рис. 1. На каждом такте процесс состоит из нескольких стадий:

- просчет возможных столкновений тел на текущем временном такте.
- для тех тел, которые на данном такте сталкиваются, проводится пересчет скоростей и импульсов. Далее, в зависимости от точности алгоритма, либо считается, что столкнувшиеся тела на данном такте не совершили перемещения, либо производится расчет перемещения этих тел с учетом изменившихся скоростей и времени, прошедшего от начала такта до столкновения.
- для тел, не вступавших во взаимодействие на данном такте, в соответствии с уравнениями теоретической механики, показанными ранее, производится вычисление перемещения и вращения тел.

Рассмотрим эти стадии более детально.

### Collision detection алгоритмы

На данной стадии производится просчет столкновений между телами, участвующими в моделируемой сцене. Эта часть процесса наиболее просто в понимании и реализации, однако при больших количествах объектов требует больших вычислительных затрат. В силу этого, предыдущими исследователями были разработаны многочисленные алгоритмы, уменьшающие время обсчета столкновений. Рассмотрим основные из них:

- метод "объектно-ориентированных параллелепипедов" (известный в англоязычной литературе как "object-oriented bounding boxes method").

Для каждого тела, участвующего в моделировании, строится описывающий его прямоугольный параллелепипед (называемый далее 'bounding-box'), ограничивающий данное тело и при этом имеющий ориентацию в пространстве, аналогичную ориентации тела (и, соответственно, меняющий ее при вращении тела). Далее в простых вариациях данного алгоритма при проверке тел на пересечение, сначала проверяются их bounding-box'ы, и только в случае, если они пересекаются, проводится полная проверка на пересечение тел.

В усовершенствованных модификациях, все пространство разбивается сначала на 4 больших box'а, далее каждый из них разбивается на четыре вложенных и т.д. до некоторого разумного предела. Таким образом, для пространства строится дерево вложенных ячеек, и проверка на пересечение осуществляется следующим образом – сначала для каждого тела определяются ячейки дерева, с которыми пересекается его (тела) box. Затем на пересечение проверяются только те тела, box'ы которых занимают одинаковые ячейки пространства.

- следующий метод основан на условии, что все моделируемые тела являются выпуклыми многогранниками (но он довольно легко может быть продлен на случай произвольных многогранников). Согласно теории для любых двух непересекающихся или соприкасающихся (т. е. невзаимопроникающих) выпуклых многогранников существует разделяющая их плоскость, т.е. такая плоскость, что многогранники находятся в разных полупространствах относительно нее. Алгоритм состоит в следующем – в начальный момент времени для каждой пары тел вычисляется разделяющая их плоскость (считается, что в начальный момент времени тела не взаимопроникают друг в друга). Если тела не взаимопроникают друг в друга, одна из плоскостей, проходящая через грань тела, или проходящая через вершину одного тела и параллельная грани другого, является плоскостью, разделяющей эти тела. Далее, на каждом такте алгоритма, разделяющая плоскость пересчитывается в соответствии с движением тела. После это производится проверка того, что плоскость осталась разделяющей – т.е. все вершины каждого многогранника лежат в одном полупространстве относительно неё. В случае, если плоскость осталась разделяющей, тела считаются непересекающимися. В противном случае, алгоритм пытается выбрать новую разделяющую плоскость. Если таковая отсутствует, тела считаются взаимопроникающими друг в друга и управление передается блоку, ответственному за обработку столкновений.

- еще один метод основан на алгоритме Лина-Кэнни (Lin-Canny) определения расстояния между двумя выпуклыми многогранниками. Механизм работы метода показан на рис. 2 – для каждой пары тел, участвующих в моделировании, по методу Lin-Canny вычисляется расстояние между ними, далее в соответствии с направлением их движения и их скоростями вычисляется предполагаемое время до их столкновения (BC). Полученные данные заносятся в collision heap (хранилище данных о возможных столкновениях – в нем в упорядоченном по времени до столкновения порядке хранятся данные о всех парах объектов). На каждом такте происходит перевычисление BC. Проверка на столкновение производится только для объектов, BC для которых  $\leq 0$ . В случае, если происходит столкновение, для всех

объектов, участвующих в нем, производится перевычисление их параметров и collision heap сортируется заново.

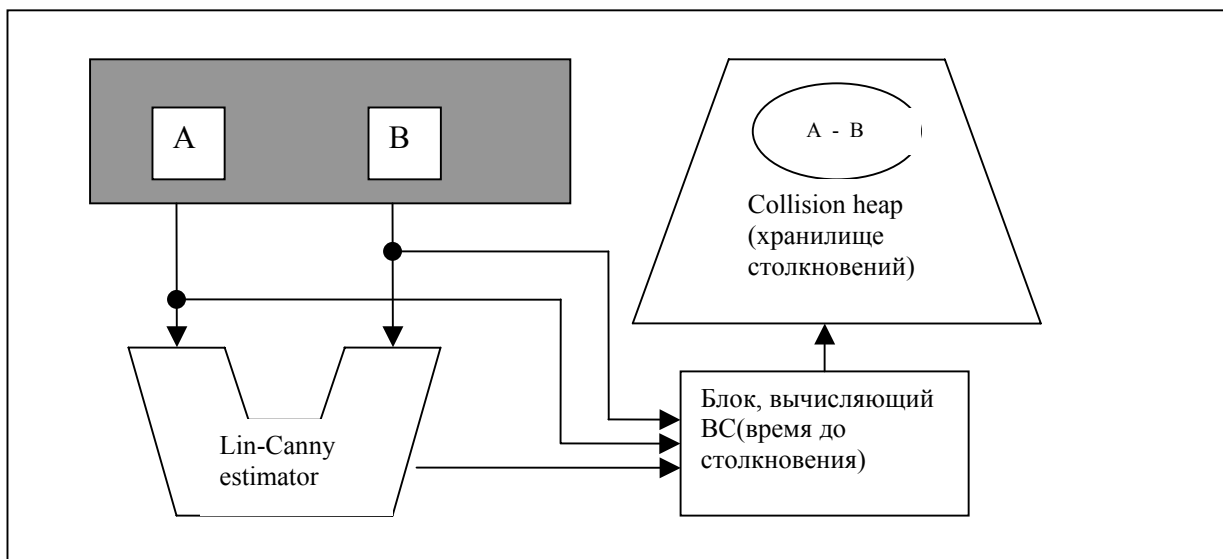


Рис. 2

Фактически, все известные системы моделирования взаимодействия физических тел для вычисления столкновений используют либо описанные выше методы, либо их смешанные вариации.

### Алгоритмы обработки взаимодействия тел

Рассмотрим теперь непосредственно алгоритмы, моделирующие изменение параметров движения тела, при его взаимодействии с другими телами. Почти все существующие алгоритмы моделирования могут быть разделены на два больших класса – алгоритмы, основанные на ограничениях, и импульсные алгоритмы.

Основной принцип работы алгоритмов, основанных на ограничениях, заключается в следующем – вводится определенная математическая интерпретация движение тел при взаимодействиях, и на основе этой интерпретации строятся системы уравнений относительно параметров движения тел. Данные системы называются ограничениями, и результатом работы таких методов является решение полученных систем.

Импульсные алгоритмы подходят к решению проблемы моделирования движения тел с другой стороны – точно так же, как и в алгоритмах, основанных на ограничениях, вводится некоторая математическая модель взаимодействия тел. В процессе моделирования каждый случай, когда тела находятся во взаимодействии, считается столкновением тел, то есть, в то время как в алгоритме, основанном на ограничениях, случай качения цилиндра по поверхности будет рассматриваться отдельно, в импульсном алгоритме в каждый момент времени качения к телу будет прикладываться импульс, соответствующий случаю столкновения тел.

Далее, рассмотрим описанные алгоритмы чуть подробнее.



## Алгоритмы, основанные на ограничениях (Constrained-based алгоритмы)

Как уже было сказано выше, основной принцип алгоритмов, основанных на ограничениях, заключается в решении систем уравнений, получаемых из соответствующей математической интерпретации законов движения систем твердых тел.

Если говорить точнее, решаемые системы уравнений получаются из нескольких условий, принимаемых за ключевые. Первое из них заключается в следующем – силы, возникающие при взаимодействии тел должны по величине быть достаточными для того, чтобы избежать проникновения тел друг в друга. Второе условие гласит, что возникающие силы должны быть отталкивающими, то есть заставлять сталкивающиеся тела двигаться друг от друга. Третье условие накладывает условие равенства нулю на силы взаимодействия в случае, если тела прекратили контактировать.

Еще одно важное условие, накладываемое на взаимодействие тел, выполняется для систем тел с трением, и называется “Закон трения Колумба”, который гласит, что если тело А скользит вдоль тела В в направлении  $\hat{t}$ , и при этом на тело А действует нормальная сила величиной  $f$ , то сила трения равна

$$F_{Coulomb} = -\mu \cdot f \cdot \hat{t},$$

где  $\mu$  - коэффициент трения.

На основе приведенных ограничений в процессе моделирования и строятся решаемые системы уравнений. В случае, если в процессе моделирования участвуют не только отдельные тела, но и системы связанных между собой тел, к полученным системам добавляются уравнения, соответствующие существующим между телами связям.

В статье [3] было показано, что одним из главных недостатков алгоритмов, основанных на ограничениях, является трудоемкость решения получаемых систем. Было также показано, что в ряде случаев задача решения получаемых систем является NP-трудной, также приведены примеры, когда решение либо неоднозначно, либо вообще не существует.

Еще одним недостатком приведенного метода является резкое увеличение затрачиваемых вычислительных мощностей при моделировании ситуаций, схожих с той, что изображена на Рис.1. Это связано с тем, что моделируемые тела в такой ситуации достаточно часто меняют характер взаимодействия с поверхностью (обычное столкновение, качение и т.д.), а данный алгоритм обрабатывает каждый вид взаимодействия отдельно. В результате усложняется вид решаемых систем, что приводит к значительному росту сложности их решения.

## Импульсные алгоритмы

При описании данного вида алгоритмов моделирования следует заметить, что особую важность при вычислениях здесь имеет математическая модель упругости тел и трения. В работах предыдущих исследователей [2], [3], [6] были разработаны уравнения, дающие результаты, хорошо приближенные к результатам, получаемым на опыте, и наиболее удобные для численного интегрирования. На основе этих уравнений рассматриваемые импульсные

алгоритмы опираются на три основных предположения относительно движения тел при взаимодействии, а именно:

- бесконечно малое время столкновения
- Гипотеза Пуассона
- Трение Колумба

Бесконечная малость времени столкновения – условие, общее для всех видов алгоритмов моделирования. Оно необходимо для того, чтобы было истинным предположение о мгновенном изменении скорости движения тела при столкновении.

Гипотеза Пуассона дает приближенное представление о сложных деформациях и потерях энергии в случае, когда сталкиваются два реальных твердых тела. Сама гипотеза является простым эмпирическим правилом, описывающим простейшее поведение тел при столкновении.

Процесс столкновения тел разделяется на две основных фазы – фазу сжатия и фазу растяжения. Точка разделения фаз – точка максимального сжатия тела. Предположим,  $p_{total}$  - величина общего импульса, полученного телом в результате столкновения, а  $p_{м.с}$  - величина импульса, полученного телом после момента максимального сжатия. Тогда гипотеза Пуассона гласит, что

$$p_{total} = (1 + \varepsilon)p_{м.с},$$

где  $\varepsilon$  - константа, зависящая от материала, из которого изготовлено тело,  $0 \leq \varepsilon \leq 1$ .

Закон трения Колумба был описан чуть выше.

Сам процесс моделирования делится на две принципиально различные части – в случае, если тела при столкновении получают достаточный импульс, чтобы направление их движения было направлено друг от друга, алгоритм просто меняет движение тел в соответствии с полученным ими импульсом.

Если же после столкновения тела не прерывают контакта, то есть наблюдается статичное столкновение (то есть столкновение, при котором точка контакта остается точкой контакта и в дальнейшем, а тело совершает только вращательное движение вокруг этой точки под действием инерции и силы тяжести) или случай скольжения одного тела вдоль другого, алгоритм считает, что данные тела испытывают микростолкновения и прикладывает к ним соответствующие микроимпульсы.

Таким образом, импульсные алгоритмы избавлены от недостатков алгоритмов, основанных на ограничениях. Тем не менее, данный вид алгоритмов имеет свои недостатки, связанные с тем, что во-первых, математическая модель взаимодействия тел, используемая данным алгоритмом более условна, чем модель, используемая алгоритмами, основанными на ограничениях. Во-вторых, в силу того, что всякое взаимодействие рассматривается данным алгоритмом как столкновение, количество обрабатываемых столкновений возрастает на порядки, что приводит к увеличению затрачиваемых вычислительных мощностей.

# **Предлагаемый алгоритм моделирования взаимодействий систем твердых тел**

## **Особенности постановки задачи**

Прежде чем описывать предлагаемую схему моделирования взаимодействий твердых тел, необходимо понять в рамках какой задачи она будет использоваться, и, соответственно, выяснить какие условия и ограничения накладывает среда использования на алгоритм моделирования. Ведь, как было показано ранее, все существующие схемы довольно сильно отличаются по множеству параметров, таких как вычислительные мощности, необходимые для нормального функционирования алгоритма, степень соответствия смоделированного поведения тел реальной физической модели, разновидность систем тел, используемых в моделировании, и др.

Необходимость в алгоритме, предложенном в данной работе, появилась при разработке симулятора тактических действий, подобного тем, что используются для отработки действий различных служб в чрезвычайных ситуациях. При подробном рассмотрении специфики данного проекта становятся ясными два основных ограничения, накладываемых на разрабатываемую модель – достаточно небольшие требования к вычислительным ресурсам, а также не столько физическая точность смоделированного поведения тел, сколько его визуальная правдоподобность. Есть еще одна особенность данной предметной области, которая позволяет для моделирования взаимодействий твердых тел использовать упрощенные модели. Дело в том, что в данном проекте результат моделирования именно потому и может полностью не соответствовать реальному физическому поведению тел, что характер их в данном случае носит достаточно случайный характер, что приводит к случайному характеру взаимодействий тел. В таком случае, если заменить высокодетализированную модель цилиндра, являющуюся правильной многогранной пирамидой (число граней может превышать 100) низкополигональным аналогом (с числом граней порядка 20), общий характер взаимодействий с окружающими телами не изменится (не считая случая качения данного цилиндра по какой-либо поверхности - такие случаи обрабатываются отдельно). Таким образом, перед началом процесса моделирования, для каждого тела участвующего в процессе, строится упрощенная модель. В алгоритме, реализованном в рамках данной работы, этот процесс производится вручную. В дальнейшем планируется разработка алгоритмов, автоматически строящих низкополигональные приближения высокодетализированных объектов.

## **Общая схема работы предлагаемого алгоритма**

Теперь рассмотрим алгоритм моделирования взаимодействий твердых тел, предложенный в данной работе.

На каждом такте работы алгоритма хранятся следующие данные:

$$Y(t) = \begin{pmatrix} x(t) \\ R(t) \\ P(t) \\ L(t) \end{pmatrix} - \text{вектор, определяющий скорость и направление движения}$$

тела в пространстве в условиях отсутствия на тело внешних воздействий (уравнения движения тела в таких ограничениях были представлены выше). Вводится также StaticInfoArray – массив точек тела, находящихся в контакте с другими телами, скорость тел в которых относительно друг друга направлена в плоскости касания тел.

На этапе выполнения алгоритма, отвечающем за вычисление нового значения  $Y(t)$  и StaticInfoArray для каждой пары тел, участвующих в моделировании, вводится дополнительная структура данных – CollisionInfoArray. В данный массив будут храниться точки столкновения этих тел на данном такте. Вводится также CollisionCycles – счетчик повторных вызовов этапа алгоритма, ответственного за вычисление нового положения тела в пространстве и обработку столкновений (4-7 шаги).

Сам алгоритм моделирования состоит из семи шагов:

1. Производится обнуление информации в CollisionInfoArray. Обнуляется CollisionCycles
2. Сохраняется значение  $Y(t)$
3. Пересчитываются значения скоростей тела, а соответственно  $Y(t)$  с учетом силы тяжести, действующей на тело, а также StaticInfoArray.
4. Производится пересчет положения тела в пространстве, а соответственно  $Y(t)$
5. Для каждой пары тел, участвующих в моделировании, выполняется проверка на их пересечение. В случае отсутствия точек контакта, просто пересчитываем положение тела в пространстве в соответствии с новым значением. На этом цикл работы алгоритма на данном такте заканчивается. В случае наличия точек контакта, алгоритм переходит к шагу 6.
6. В случае наличия точек контакта между телами вся информация о уже обрабатываемых столкновениях переносится из StaticInfoArray в CollisionInfoArray (т.к. при любом столкновении тел изменяется направление скорости движения тела в данных точках, что может привести к изменению статуса точки контакта). Значение  $Y(t)$  восстанавливается из значения, сохраненного на шаге 2, т.к. в данном варианте алгоритма считается, что за данный такт времени тела совершили нулевое перемещение в пространстве, благодаря своему столкновению. Данное предположение допустимо в условиях заданной степени точности алгоритма моделирования, накладываемой решаемой задачей.
7. В каждой точке контакта тел производится вычисление скорости столкновения тел, а также импульса, полученного каждым телом в данной точке после столкновения. В соответствии с вычисленными значениями импульсов, производится пересчет скоростей тела. Если значение счетчика CollisionCycle не превышает некоей заранее заданной константы (ее значение устанавливается эмпирическим путем в рамках каждой конкретной задачи), алгоритм переходит к шагу 4.

Смысл многократного вызова блока алгоритма, отвечающего за вычисление и обработку столкновений, состоит в следующем – в реальной жизни время, которое тела находятся в контакте, может быть ничтожно мало, и силы, действующие на тела при столкновении, могут приводить к повторному контакту в других точках, что практически мгновенно приводит к возникновению новых сил и т.д. В условиях дискретности времени и, соответственно, достаточно большого размера временных тактов, если не обрабатывать случаи повторного столкновения на одном такте, модель в достаточно большом проценте случаев (что подтверждено эмпирически) будет страдать отсутствием реалистичности. В предлагаемом варианте алгоритма многократные столкновения в один момент времени эмулируются многократным вызовом блока, отвечающего за вычисление и обработку столкновений.

Рассмотрим теперь каждый этап предлагаемого алгоритма подробнее.

### **Collision detection алгоритм**

Схема работы алгоритма, вычисляющего точки столкновения движущихся тел, такова – при запуске процесса моделирования, все тела, участвующие в процессе, разбиваются на два больших класса – динамические и статические объекты. Статические – неподвижные объекты, такие как всевозможные строения и т.д. Динамические – объекты, имеющие свободу для движения.

Для каждого динамического тела, участвующего в моделировании, в начальный момент времени строится описывающий его прямоугольный параллелепипед, имеющий такую же ориентацию в пространстве, что и само тело (OOBB – Object-Oriented Bounding Box). Статические тела в данном алгоритме collision-detection рассматриваются как объединение составляющих их треугольников. Каждый треугольник обрабатывается аналогично динамическому телу, с тем лишь отличием, что импульс, получаемый им при столкновении, не учитывается. В результате разбиения всех тел на статические и динамические за счет того, что для всех статических тел уменьшается количество этапов, ответственных за просчет столкновений с динамическими телами, уменьшается время работы всего алгоритма.

Далее, в процессе движения тела в пространстве, положение и ориентация параллелепипеда меняется в соответствии с положением и ориентацией тела. Также, для каждого динамического тела строится сфера с центром в центре тяжести тела, описывающая его. Сам алгоритм проверки тел на столкновение является модификацией алгоритма основанного на алгоритме Lin-Canny и использующего время до предполагаемого столкновения тел.

На начальном этапе запуска алгоритма для каждой пары тел, участвующих в моделировании, рассчитывается приблизительное время до их столкновения. В случае если оба тела принадлежат динамической части обрабатываемой сцены, значение предсказываемого времени до столкновения вычисляется следующим образом - если линейная скорость движения тел относительно друг друга  $v$ , начальное расстояние между центрами масс  $r$ , а радиусы описывающих их сфер  $r_{on.1}, r_{on.2}$ , то приблизительная скорость до их столкновения

$$\tilde{t} = (r - (r_{on.1} + r_{on.2})) / v$$

очевидно, что реальное время до их столкновения  $t \geq \tilde{t}$ . Полученный результат заносится в соответствующий список, отсортированный по возрастанию. Далее на каждом такте работы алгоритма производится пересчет времени, оставшегося

до столкновения. В случае если для какой-либо пары тел оставшееся до столкновения время  $\leq 0$ , производится проверка ООВВ этих тел на столкновение, и только если ООВВ также пересекаются, производится проверка на пересечение самих тел.

Рассмотрим подробнее, как именно вычисляются точки пересечения между телами. Заметим, что в случае столкновения двух многогранников, точки столкновения могут представлять собой либо конечное множество, либо, если, например, ребро или грань одного из них в момент столкновения лежит в плоскости другого, отрезок или более сложную конструкцию. Ясно, что подобные случаи необходимо обрабатывать отдельно. Еще одну трудность представляет собой следующий факт – в момент проверки на столкновение взаимодействие тел уже может носить взаимопроникающий характер, в этом случае мы получим не точку столкновения, а только ее довольно грубое приближение. Для устранения вышеперечисленных трудностей был разработан следующий алгоритм. В случае если алгоритм collision-detection выдает результат, говорящий о том, что два исследуемых тела сталкиваются, для каждой пары вида грань-ребро, грань-грань, ребро-ребро участвующей в столкновении проводится проверка, не являлись ли они на данном такте моделирования параллельными (с некоторой погрешностью  $\varepsilon$ ). В случае если параллельности не наблюдалось, точкой столкновения считается соответствующая вершина, являющаяся ближайшим к точке действительного пересечения концом ребра, задействованного в данном пересечении. В случае же, если произошло не точечное столкновение, точками столкновения считаются те вершины взаимодействующих ребер (или граней), которые действительно принадлежат области взаимодействия.

Покажем теперь, что предложенный алгоритм в контексте данной конкретной задачи не хуже методов предложенных предыдущими исследователями.

Во-первых, заметим, что алгоритм Lin-Canny дает не точное время до столкновения, а приблизительное, причем, как показывают опыты, средняя дисперсия оценки, даваемой этим алгоритмом, не сильно ниже дисперсии оценки, полученной в результате применения метода, основанного на описанных вокруг тел сферах. В условиях же решаемой задачи, когда все используемые модели являются достаточно низкополигональными (т.е. представляют из себя набор треугольников, число которых порядка нескольких сотен), применение метода Lin-Canny неоправданно. Также при оценке предлагаемого метода необходимо помнить, что в случае проверки тел пересечение, используется техника ООВВ, которая более чем в 90% случаев позволяет не производить дальнейших проверок. Таким образом, сочетание модифицированного алгоритма, основанного на вычислении времени до столкновения, и алгоритма проверки тел на пересечение, основанного на идее ООВВ, дает результаты не хуже алгоритмов, предложенных в работах предыдущих исследователей.

### **Обработка взаимодействия тел**

После определения точек контакта двух тел, необходимо изменить их скорости, в соответствии с импульсами, полученными ими в этих точках. Для начала выведем формулу, определяющую величину импульса при столкновении.

#### **Вывод величины импульса, получаемого телом при столкновении**

Пусть величина импульса  $J$ . Тогда изменение линейной скорости тела

$$v(t) = \frac{J}{M},$$

вращение, придаваемое телу данным импульсом, равно

$$\tau_{impulse} = (p - x(t)) \times J.$$

Пусть в момент времени  $t_0$  сталкиваются тела А и В. Обозначим  $\dot{p}_a^-(t_0), \dot{p}_a^+(t_0)$  - скорости точки контакта тела А до и после столкновения соответственно. Аналогично определим  $\dot{p}_b^-(t_0), \dot{p}_b^+(t_0)$  для тела В. Относительная скорость этих тел относительно друг друга до столкновения

$$v_{rel}^- = \hat{n}(t_0) \cdot (\dot{p}_a^-(t_0) - \dot{p}_b^-(t_0)),$$

после столкновения

$$v_{rel}^+ = \hat{n}(t_0) \cdot (\dot{p}_a^+(t_0) - \dot{p}_b^+(t_0)),$$

где  $\hat{n}(t_0)$  - нормаль к поверхности тела в точке.

Для скоростей тела до и после столкновения эмпирически был получен закон

$$v_{rel}^+ = -\varepsilon v_{rel}^-.$$

$\varepsilon$  – коэффициент затухания, удовлетворяющий неравенствам  $0 \leq \varepsilon \leq 1$ . В нашем случае, неравенства предполагаются строгими.

Введем радиус-вектора точек столкновения

$$r_a = p - x_a(t_0),$$

$$r_b = p - x_b(t_0).$$

Тогда скорости точек столкновения равны

$$\dot{p}_a^+(t_0) = v_a^+(t_0) + \omega_a^+(t_0) \times r_a, \quad \dot{p}_b^+(t_0) = v_b^+(t_0) + \omega_b^+(t_0) \times r_b.$$

Опытным путем было показано, что импульс, получаемый телом при столкновении направлен в направлении нормали к поверхности столкновения, т.е.

$$J = j\hat{n}(t_0).$$

Учитывая уравнения движения тела, описанные в первой части, можно получить:

$$\dot{p}_a^+(t_0) = \dot{p}_a^-(t_0) + j\left(\frac{\hat{n}(t_0)}{M_a} + I_a^{-1}(t_0)(r_a \times \hat{n}(t_0))\right) \times r_a,$$

$$\dot{p}_b^+(t_0) = \dot{p}_b^-(t_0) - j\left(\frac{\hat{n}(t_0)}{M_b} + I_b^{-1}(t_0)(r_b \times \hat{n}(t_0))\right) \times r_b,$$

тогда в предположении, что вектор  $\hat{n}(t_0)$  единичный, т.е.  $\hat{n}(t_0) \cdot \hat{n}(t_0) = 1$ , получаем

$$v_{rel}^+ = v_{rel}^- + j\left(\frac{1}{M_A} + \frac{1}{M_B} +$$

$$\hat{n}(t_0) \cdot (I_a^{-1}(t_0)(r_a \times \hat{n}(t_0))) \times r_a + \hat{n}(t_0) \cdot (I_b^{-1}(t_0)(r_b \times \hat{n}(t_0))) \times r_b\right)$$

но  $v_{rel}^+ = -\varepsilon v_{rel}^-$ . Отсюда получаем конечное выражение для величины импульса, действующего на тела при столкновении:

$$j = \frac{-(1 + \varepsilon)v_{rel}^-}{\frac{1}{M_A} + \frac{1}{M_B} + \hat{n}(t_0) \cdot (I_a^{-1}(t_0)(r_a \times \hat{n}(t_0))) \times r_a + \hat{n}(t_0) \cdot (I_b^{-1}(t_0)(r_b \times \hat{n}(t_0))) \times r_b}$$

Полученным выражением для импульса, получаемого телами при столкновении, используется при вычислении нового значения вектора состояния тела  $Y(t)$ .

Далее более подробно опишем механизм обработки столкновений.

### Обработка столкновений

На начальном этапе алгоритма, отвечающего за обработку столкновений, по формуле, приведенной выше, вычисляется импульс, получаемый телами.

В случае, если тела под действием полученного импульса получают скорости, направленные друг от друга так, что контакт тел прерывается, алгоритм просто пересчитывает параметры движения в соответствии с полученным импульсом и прекращает свою работу на данном этапе.

В случае, если наблюдается так называемое статическое столкновение, то есть, несмотря на полученный телами импульс, они остаются в контакте, управление передается фазе алгоритма, отвечающей за обработку статических столкновений.

### Обработка статических столкновений

Как было описано ранее, для каждого тела, участвующего в процессе моделирования, хранится массив статических точек контакта, который обнуляется на каждом новом такте при обработке нового столкновения, и накапливается в процессе обработки одного такта.

Работа данного этапа начинается с занесения точки статического контакта в массив подобных точек, описанный выше.



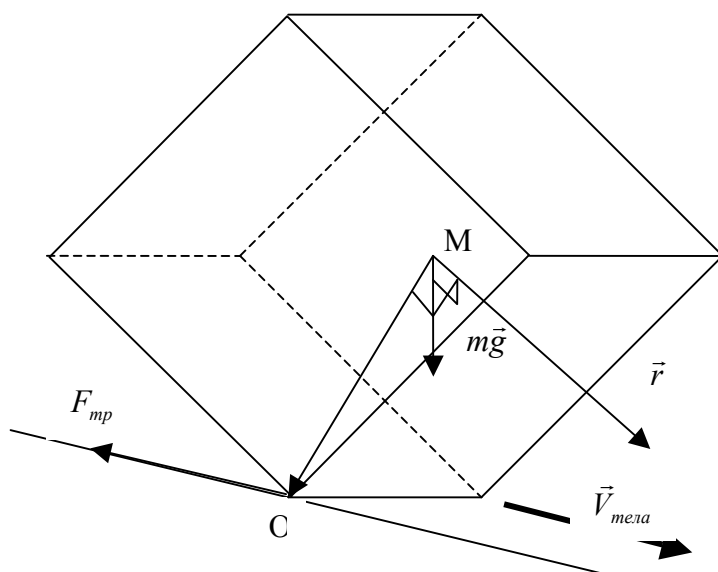


Рис. 4

Для обработки параметров движения, вводится разделение случаев взаимодействия тел в точке статического контакта на несколько основных категорий:

- Покой
- Скольжение
- Качение

На рис. 4 показаны случаи покоя и скольжения. Случай качения обрабатывается аналогичным им образом, только добавляется дополнительное вращение вокруг оси симметрии тела.

Далее при обработке столкновения, для каждой точки в массиве статических контактов определяется ее тип. Далее, в зависимости от их типа, для каждой статической точки определяется направление ее движения, а также ограничение, накладываемое ей на движение тела. А именно, точка покоя накладывает следующее ограничение – движение тела является вращением вокруг оси, проходящей через данную точку, причем, если отсутствуют другие точки контакта, то ( $M$  - центр тяжести тела,  $O$  - точка контакта,  $\vec{r}$  - ось вращения)

$$\vec{r} = \overrightarrow{MO} \times m\vec{g}$$

Точка скольжения накладывает похожее ограничение на движение, отличающееся только тем, что ось вращения меняет свою ориентацию вместе с движением точки контакта.

#### Алгоритм верификации положения тела.

Как известно, в реальном физическом мире предметы могут иметь положения неустойчивого равновесия, то есть положения, в котором любом достаточно маленькое воздействие приведет к нарушению равновесия, в

результате чего тело перейдет в состояние устойчивого равновесия. В предлагаемом алгоритме мы имеем дело с идеально гладкими поверхностями и отсутствием внешних вибраций, характерных для реального случая. Это может привести к тому, что положение, являющееся для тела в реальном мире неустойчивым, при моделировании окажется вполне стационарным, то есть в результате возможен так называемый “эффект застывания”, когда моделируемое тело застывает в положении, совершенно невозможном в реальном случае. Для исправления таких дефектов и разработан предлагаемый алгоритм “верификации положения тела”.

Схема работы алгоритма такова – в случае, если скорость движения моделируемого тела становится нулевой, то есть тело принимает положение равновесия, проверяется на какой именно поверхности тело остановилось. В случае, если моделируемое тело приняло стационарное положение, опираясь на несколько поверхностей, расположенных под углом друг к другу, алгоритм заканчивает свою работу, так как в таком случае, как показали практические наблюдения, вероятность принятия телом положения неустойчивого равновесия близка к нулю. В случае, если тело опирается на одну поверхность, производится проверка на то, действительно ли моделируемое тело приняло положение неустойчивого равновесия. Для этого вычисляется многоугольник, составленный из крайних точек опоры тела. В случае если угол дополнительного наклона тела, необходимый для того, чтобы проекция центра тяжести на плоскость опоры вышла за пределы вычисленного многоугольника, меньше некоторого заранее заданного и достаточно малого  $\varepsilon$ , положения тела считается неустойчивым. В таком случае телу придается импульс в центре тяжести, достаточный для того, чтобы вывести проекцию центра тяжести за пределы опоры.

### **Основные отличия предлагаемого алгоритма от уже существующих решений**

В данном параграфе я сформулирую основные отличия предлагаемого мной алгоритма от уже существующих решений, а также покажу его преимущества, позволяющие использовать его в рассматриваемой предметной области наравне с алгоритмами, разработанными исследователями ранее.

Следует заметить, что предлагаемое решение использует алгоритмы, используемые как в методах, основанных на ограничениях, так и импульсных методах. А именно - в случае, когда тела взаимодействуют таким образом, что после столкновения они получают импульс, прерывающий их контакт, то есть они начинают двигаться друг от друга, используется часть импульсного алгоритма, отвечающего за обработку как раз таких столкновений. В случае же, когда наблюдается так называемое статическое столкновение, то есть моделируемое тело получает в точке контакта в результате столкновения настолько малый импульс, что он не является достаточным для прерывания контакта в данной точке, разработанный алгоритм входит в фазу обработки статического случая. При использовании предлагаемой схемы становится возможным избежать как необходимости решения систем линейных неравенств и уравнений, решение которых приводит к решению задач линейного программирования, так и постоянной обработки столкновений и микростолкновений, как это происходит в импульсных методах.

Естественно, что предлагаемое решение за счет уменьшения вычислительной трудоемкости имеет несколько меньшую физическую точность моделируемых процессов. Для разрешения возникающих в результате этого погрешностей был предложен алгоритм верификации положения тела, который в случае, если моделируемое тело стремится занять положение неустойчивого равновесия, добавляет в определенном направлении импульсные шумы, выводящие тело из текущего состояния.

## Заключение

В работе были проанализированы классические подходы к моделированию взаимодействий систем твердых тел. В результате проведенных исследований, была предложена новая схема моделирования, включающая в себя как уже известные алгоритмы, так и разработанные автором.

Предлагаемая система моделирования основывается на нескольких ключевых идеях.

Во-первых, в рамках предметной области, в которой работает автор, реализация физической модели поведения систем твердых тел не обязательно должна строго соответствовать реальным экспериментам. Это позволяет, во-первых, не требовать от разрабатываемых алгоритмов излишней точности, а, во-вторых, использовать для моделирования упрощенные модели взаимодействующих тел. Эта же специфика разрабатываемого алгоритма позволяет использовать модифицированный вариант алгоритма определения столкновений, основанный на сферической модификации алгоритма Лина-Кэнни.

Вторая идея состоит в том, что в случае обычного столкновения, предлагаемый алгоритм работает как импульсный, а в случае статического принцип работы схож с поведением алгоритмов, основанных на ограничениях, но в отличие от них, вместо построения линейных систем и их решения, были построены некоторые эвристики, основанные на разделении случаев статических столкновений на несколько базовых типов. Для устранения неточностей моделируемого поведения тел, связанных с несовершенством применяемых алгоритмов, был разработан алгоритм верификации положения тела, который в случае, если тело стремится занять положение неустойчивого равновесия, путем

создания импульсных шумов, применяемых к телу в определенном направлении, моделирует реальные условия воздействия на тело окружающей среды.

Разработанный в данной работе алгоритм был апробирован в игровом проекте “E5”, где доказал свою состоятельность и низкую требовательность к вычислительным ресурсам.

## Литература

- [1] **Sir Isaac Newton** “Philosophiae naturalis principia mathematica,” 1687
- [2] **David Barraf** “Fast contact force computation for Nonpenetrating Rigid Body“, *ACM SIGGRAPH’94*
- [3] **Francois Faure** “An energy-based approach for contact force computation”, *iMAGIS/GRAVIR-IMAG*
- [4] **Brian Mirtich, John Canny** “Impulse-based simulation of rigid bodies”, University of California at Berkley, *Symposium on Interactive 3D Graphics, April 1995*
- [5] **Chang, Colgate** “Impulse based simulation”, 1997
- [6] **Matti Larsson, Christian Holmboe** “VR Dynamics – A primer”, *Nya Medier, 5p, VT02, University of Umeå* 2002-03-14
- [7] **Ronen Barzel, Alan H. Barr**: “A modeling system based on dynamic constraints,” *Computer Graphics* 22(4): 179-188, 1988
- [8] **Graham S. Rhodes** “Stable Rigid-Body Physics,” *Game Developer Conference 2001*, [http://www.gdconf.com/archives/proceedings/2001/rhodes\\_paper.pdf](http://www.gdconf.com/archives/proceedings/2001/rhodes_paper.pdf)

- [9] **Andrew Witkin, David Baraff, Michael Kass** “An Introduction to Physically Based Modeling”, *ACM SIGGRAPH*’95
- [10] **Moore, Wilhelms** “Collision detection and response for computer animation”, *ACM SIGGRAPH* ‘88
- [11] **D. Baraff** “Analytical methods for dynamic simulation of non-penetrating rigid bodies”. *Computer Graphics* 23(3): 223-232, 1989.