

АННОТАЦИЯ

Механическая одномерная модель описывающая взаимодействие тела из полидисперсного материала с абсолютно твердой стенкой. Работа посвящена исследованию процесса возникновения диссипации энергии в системе, моделирующей неабсолютно упругое соударение. Диссипативная пара сил образуется при динамическом взаимодействии тела и стенки, в процессе преобразования кинетической энергии тела во внутреннюю энергию, а так же во время преобразования внутренней энергии тела в кинетическую.

Моделирование поведения полидисперсного материала производится с помощью механической модели. Работа нацелена на изучение процессов диссипации в ходе упругого соударения, поэтому исследуемая модель не учитывает потери энергии из-за трения и т.п. То есть в пределах одного столкновения тел система является консервативной, соударения считаются абсолютно упругими а тела абсолютно жесткими. В работе исследуется процесс преобразования кинетической энергии тела во внутреннюю, при столкновении тела и стенки.

В ходе соударения упругого тела и стенки в теле на микроуровне возникают волны сжатия и растяжения, до тех пор пока вся энергия таких колебаний не перейдет во внутреннюю энергию тела. Аналогичной моделью таких растяжений и сжатий является пружина. Так как в данной задаче все возникающие волны являются продольными, то механическую систему можно рассмотреть в виде одномерной задачи. В механической системе массы представляют частицы материала, упругие взаимодействия со стенкой, а так же внутри тела моделируются пружинами. Энергия запасенная в пружинах, а так же энергия движения частиц относительно общего центра масс является аналогом внутренней энергии.

В работе исследуется влияние значений масс и жесткостей пружин на конечное поведение системы, определяется качественный характер поведения модели и оптимальные параметры системы: минимальная и максимальная запасенная телом энергия, возможные состояния, при которых вся энергия

переходит во внутреннюю, либо внутренняя энергия равна нулю после взаимодействия.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 ПОСТАНОВКА ЗАДАЧИ	6
2 МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ.....	11
2.1 Математическая модель.....	11
2.2 Аналитическое решение.....	13
3 ВЫЧИСЛИТЕЛЬНЫЙ МЕТОД	17
3.1 Метод Рунге-Кутта 4-го порядка	17
3.2 Конечно-автоматная модель	19
4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	21
4.1 Используемые программные средства	21
4.2 Параллелизация вычислений	25
4.3 Взаимодействие с кластером	27
5 ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ	34
ЗАКЛЮЧЕНИЕ	44
Список использованных источников.....	46

ВВЕДЕНИЕ

В современном мире принято считать, что при помощи законов Ньютона мы можем не только объяснять наблюдаемые механические явления, но и предсказывать их течение. И это утверждение вполне справедливо, но с развитием технологий все чаще появляются материалы, поведение которых мы предсказать не в силах. Композиты, полимеры и сплавы могут обладать такими свойствами, которые изначально были не очевидны. Это связано с тем, что данные материалы зачастую неоднородны по своей структуре. Эта неоднородность в свою очередь приводит к взаимодействию внутри этих материалов на микроуровне. На микроуровне могут иметь место уже не только механические взаимодействия, но и химические, электромагнитные и тд. Такие взаимодействия могут быть уже так сложны, что предсказывать поведения тел из таких материалов становится нетривиальной задачей.

Но рассмотрим более простой случай, где нет никаких взаимодействий в материале, кроме механических. Рассмотрим полидисперсные материалы. Полидисперсные материалы - материалы, состоящие из зерен различной крупности. Ярким примером полидисперсного материала является щебень в баластном слое железной дороги. Так же полидисперсными являются многие полимерные композиционные материалы. Полидисперсные полимеры представляют собой микроскопические частицы, которые способны взаимодействовать друг с другом на микроуровне, что зачастую приводит к неожиданному поведению данных материалов.

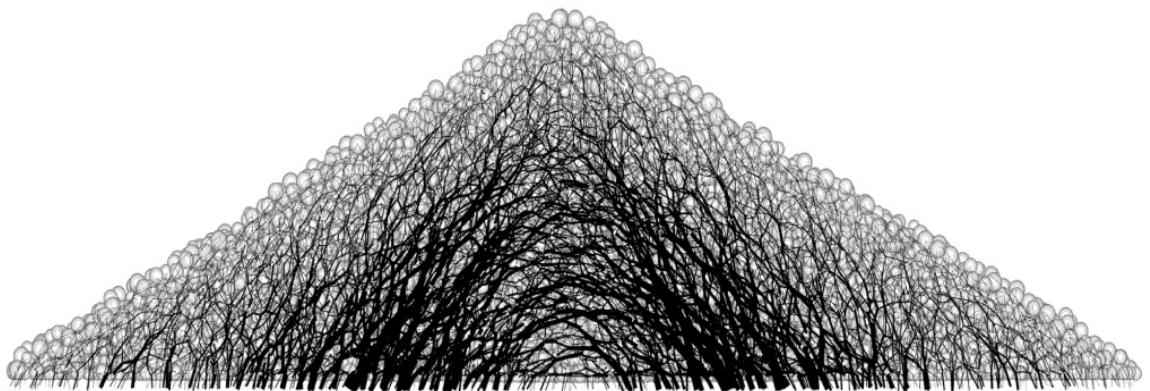


Рис. 1 — Силовая сеть в гранулированной насыпи

Такое поведение объясняется полидисперсностью данного материала, за счет чего образуются силовые сети. Силовые сети состоят из частиц внутри гранулированного материала под давлением, которые удерживаются вместе и заклиниваются сетью взаимных сжимающих сил. Между этими цепями находятся участки с низким напряжением, зерна которых "защищены" от воздействия вышеперечисленных зерен сводами и арками (рисунок 2). Набор взаимосвязанных силовых цепей известен как силовая сеть. Очевидно то, что такие полидисперсные материалы способны поглощать энергию и рассеивать ее в дальнейшем, ведь зачастую такие материалы используются в качестве демпфирующих, но каким образом образующиеся силовые сети влияют на процессы поглощения и рассеивания энергии, а так же каким образом они формируются изучено еще недостаточно подробно. Изучение преобразования кинетической энергии поступательного движения во внутреннюю энергию относительного движения в простейшей силовой цепи представлено в данной работе.

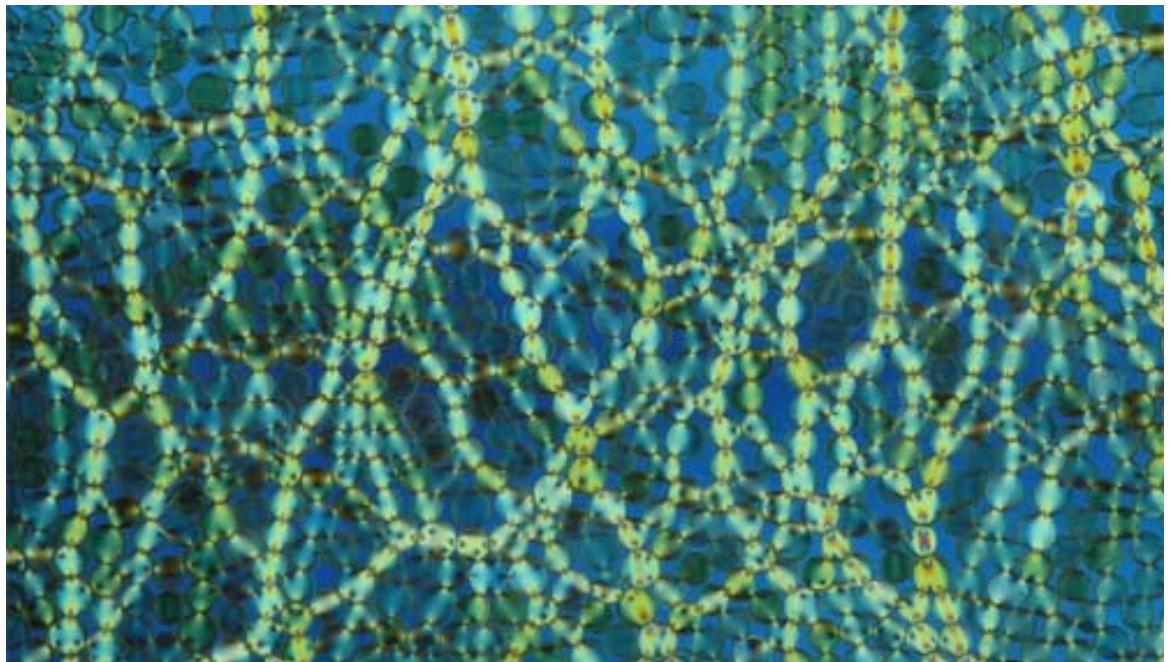


Рис. 2 — Пример силовой сети полученной при нагружении фотоэластичных гранул

1 ПОСТАНОВКА ЗАДАЧИ

Контактные силы в классической механике

Согласно классической механике, когда к твердому телу применяется сила, центр масс тела в соответствии со вторым законом Ньютона, начнет двигаться в направлении вектора единицы силы с ускорением, равным отношению величины силы и массы тела. Поскольку реальные тела не являются абсолютно жесткими, предположение, что все тело будет двигаться с этим ускорением конечно неверно. В действительности приложенная сила будет инициировать волны двигающиеся по телу, которые в конечном итоге приведут к движению всего тела. Следовательно, важным вопросом является понимание перехода от волновых процессов к движению объекта на макроуровне

Другой важный вопрос это разница/взаимосвязь между кинетической энергией волн, распространяющихся внутри тела, и тепловой энергией тела, и переход от одного к другому. Силы можно условно разделить на первичные и вторичные:

— **Первичные силы** - являются фундаментальной характеристикой материи и представляют собой силу соответствующего потенциального энергетического поля, к которому они принадлежат. В классической механике Ньютона мы как правило имеем дело с гравитационными и электромагнитными силами. Действие сил всегда осуществляется посредством полей, создаваемых телами и воспринимаемых рассматриваемым телом.

— **Вторичные силы** - являются результатом взаимодействия вещества, поскольку для возникновения требуют существования контакта между объектами, то есть являются силами второго порядка.

В соответствии с третьим законом Ньютона действие одной силы способствует появлению равной по модулю, но противоположной по направлению силы - пары сил. При взаимодействии двух макроскопических объектов их атомы меняют свое положение под действием соответствующих электромагнитных силовых полей, что в макромасштабе наблюдается как деформации взаимодействующих объектов, что может быть постоянным процессом, периодическим

или их комбинацией. Остающийся вопрос - как такое взаимодействие между двумя объектами может быть обеспечено?

Контактные силы можно разделить на статические и динамические:

- **статические** - обусловленные первичными силами,
- **динамические** - образованные в процессе включающем движение материи (динамический способ генерации контактной силы / пары сил).

Взаимодействие двух тел, как было сказано выше, инициирует волны материи на микроуровне, которые в дальнейшем инициируют движение на макроуровне, то возникает вопрос, какая часть энергии, запасенной телом перейдет в поступательную кинетическую энергию тела? Аналогом такого взаимодействия может быть любое не абсолютно упругое взаимодействие двух тел, при котором часть энергии взаимодействия переходят во внутреннюю энергию тела, то есть в тепловую. Демпфирующие полидисперсные материалы так же являются аналогом такой системы и для демпфера важно эффективное преобразование кинетической энергии во внутреннюю энергию системы[1].

Поведение гранулированных материалов может проходить через различные фазы, иметь хаотичный характер, поэтому создание общей теории, описывающей поведение гранулированных материалов, является затруднительным. При этом такие материалы могут поглощать приложенную энергию колебаний, а ввиду возможности создания демпфирующих элементов из прочных материалов с небольшим весом, они нашли применение в высокотехнологичных отраслях: в работе [2] приводится пример применения демпфирующих элементов, основанных на гранулированных материалах, в аэрокосмической отрасли.

В данной работе также указано на тот факт, что при низких амплитудах вибрации, когда частицы материала постоянно находятся в контакте, эффект демпфирования будет зависеть от способности отдельных частиц к диссипации энергии. То есть можно анализировать поведение материала под нагрузкой на основе анализа взаимодействия его отдельных частиц.

Работа [2] также ссылается на аналитический метод, позволяющий найти эффективный модуль упругости и эффективное отношение Пуассона случайной упаковки для одинаковых упругих сфер, когда они находятся в постоянном контакте (аналог низкоамплитудной вибрации), где постоянные сжимающие силы между любыми контактирующими частицами могут быть вызваны гидростатическим сжатием вещества.

В научной литературе принято связывать процесс диссипации с переносом энергии вещества из различных форм в тепловую. При этом возможны процессы, также потребляющие энергию, при действии которых тепловая энергия незамкнутой системы будет меняться незначительно. Разумеется, в конечном итоге энергия будет переходить в тепловую, но если ограничить рассматриваемую область областью без диссиационных процессов, то можно рассматривать диссиацию или поглощение энергии системой, не обусловленную процессами трения и не связанные с выделением тепла. Данная работа основана в большой степени на гипотезе Игоря Эмри, о том что источниками диссипации энергии в определенных системах могут выступать контактные силы, вследствии чего потребляемая системой энергия уходит не полностью в тепло, а частично на создание силовой сети, засчет перераспределения волновой энергии взаимодействия тел. Данная гипотеза основана на экспериментах [3, 4] с гранулированными материалами.

С помощью экспериментальной установки [3] позволяющей исследовать поведение материалов под давлением, как описано в работах [3, 4] было установлено что в определенных условиях материал (полимерное вещество под давлением), поглощающий энергию, изменяет свою температуру незначительно, т.е. не вся переданная веществу энергия перешла в тепло в этом веществе. Этот эффект, как оказалось, практически не исследован в литературе.

Среди исследований данной тематике можно привести работу [5] в которой с помощью метода конечных элементов исследуются свойства гранулированных материалов находящихся под нагрузкой. Результаты полученные численно для определенных параметров системы не являются универсаль-

ными, следовательно не являются достаточным результатом для анализа процессов поглощения энергии в демпфирующем элементе.

Похожее исследование [6] изучает динамическое поведение механической системы с одной степенью свободы содержащей демпфер частиц. Результаты работы демонстрируют регулярное или хаотическое движение гранулированного слоя при частоте возбуждения в качестве параметра управления. Результаты демонстрируют поведение слоя вещества, но элементы трения в модели не дают изучить требуемый эффект нетепловой диссипации. Взаимосвязь диссипации энергии и образования пары сил или силовой сети в гранулированных системах не изучена, а поскольку процессы диссипации могут иметь практическое применение, было решено исследовать процесс, обуславливающий работу демпфирующего элемента, то есть процесс диссипации энергии в гранулированных материалах.

Так же был произведен эксперимент, в котором по пластине из полидисперсного полимера заданной ширины стреляют пулей со скоростью 45 м/с, что приводит в разрушению пластины. Затем такой же пулей, с той же скоростью стреляют по пластине из того же полимера, но теперь ширина пластины уменьшена в два раза. В результате взаимодействия полимер не только не разрушается, но и останавливает пулю, что говорит о скрытых процессах внутри материала и влиянии процесса создания силовой сети на распространение деформаций и диссипацию энергии в материалах.

Таким образом для понимания процессов, происходящих в гранулированных материалах было решено проанализировать многоуровневый энергетический обмен в простейшей силовой сети и ее образования. Данный процесс рассматривается на двух различных уровнях: на внешнем - макроуровне(рисунок 1.1 (а)), и на внутреннем - микроуровне(рисунок 1.1 (б)).

Простейшим примером взаимодействия на макроуровне является столкновение эластичного сферического тела массой M , которое движется с постоянной скоростью v_0 , с жесткой стенкой. В процессе взаимодействия, согласно третьему закону Ньютона, между телом и жесткой стенкой сформируется пара сил: тело будет давить на стенку с силой F , а стенка будет давить на

тело с силой $-F$. Последняя сила, в свою очередь, в течении времени Δt остановит тело и ускорит его в обратном направлении со скоростью $-v_0$. Во время соударения внешняя кинетическая энергия будет превращена во внутреннюю потенциальную энергию эластичного тела. При этом многоуровневом взаимодействии и преобразовании энергии формируется контактная пара сил. Таким образом контактные силы являются результатом многоуровневого преобразования энергий.

Рассмотрим данный процесс на микроуровне. Представим тело в качестве системы нескольких тел с массами m_1, m_2, m_3 , соединенных пружинами с жесткостями k_2, k_3 , тогда внутренняя энергия системы будет заключена в кинетической энергии относительного движения частиц и энергии пружин. При этом взаимодействие тела с жесткой стенкой происходит посредством пружины жесткостью k_1 . Преобразование энергий заключается в переходе внешней кинетической энергии движения масс, во внутреннюю энергию, аналогом которой в данной постановке задачи являются потенциальные энергии пружин, соединяющих массы, и кинетические энергии относительного движения масс. Таким образом, найдя энергию поступательного движения тела после соударения, мы можем узнать диссиацию энергии в системе, что не позволяет сделать анализ на макроуровне.



Рис. 1.1 — Модель тела из полидисперсного материала на макроуровне (а) и на микроуровне (б)

2 МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

2.1 Математическая модель

Система, показанная на рисунке ?? может быть описана следующей системой уравнений движения:

$$m_1 \ddot{x}_1 = k_1(l - x_1) - k_2 [l - (x_2 - x_1)], \quad (2.1)$$

$$m_2 \ddot{x}_2 = k_2 [l - (x_2 - x_1)] - k_3 [l - (x_3 - x_2)], \quad (2.2)$$

$$m_3 \ddot{x}_3 = k_3 [l - (x_3 - x_2)]. \quad (2.3)$$

Сдвинув координаты, получим:

$$x_1 = \tilde{x}_1 + l, \quad (2.4)$$

$$x_2 = \tilde{x}_2 + 2l, \quad (2.5)$$

$$x_3 = \tilde{x}_3 + 3l, \quad (2.6)$$

и, подставив полученные значения в уравнения движения, получим:

$$m_1 \ddot{\tilde{x}}_1 = -k_1 \tilde{x}_1 + k_2 (\tilde{x}_2 - \tilde{x}_1), \quad (2.7)$$

$$m_2 \ddot{\tilde{x}}_2 = -k_2 (\tilde{x}_2 - \tilde{x}_1) + k_3 (\tilde{x}_3 - \tilde{x}_2), \quad (2.8)$$

$$m_3 \ddot{\tilde{x}}_3 = -k_3 (\tilde{x}_3 - \tilde{x}_2). \quad (2.9)$$

Разделив уравнения на соответствующие массы получим новые переменные: $\omega_i^2 = \frac{k_i}{m_i}$ and $\omega_{ij}^2 = \frac{k_j}{m_j}$. В итоге система уравнений имеет вид:

$$\ddot{\tilde{x}}_1 = -\omega_1^2 \tilde{x}_1 + \omega_{21}^2 (\tilde{x}_2 - \tilde{x}_1), \quad (2.10)$$

$$\ddot{\tilde{x}}_2 = -\omega_2^2 (\tilde{x}_2 - \tilde{x}_1) + \omega_{32}^2 (\tilde{x}_3 - \tilde{x}_2), \quad (2.11)$$

$$\ddot{\tilde{x}}_3 = -\omega_3^2 (\tilde{x}_3 - \tilde{x}_2). \quad (2.12)$$

Для составления системы ОДУ введем вектор q :

$$\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6]^T = [\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dot{\tilde{x}}_4, \dot{\tilde{x}}_5, \dot{\tilde{x}}_6]^T. \quad (2.13)$$

Таким образом система ОДУ $\frac{d}{dt}q$ имеет вид:

$$\frac{d}{dt} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\omega_1^2 - \omega_{21}^2 & \omega_{21}^2 & 0 & 0 & 0 & 0 \\ \omega_2^2 & -\omega_2^2 - \omega_{32}^2 & \omega_{32}^2 & 0 & 0 & 0 \\ 0 & \omega_3^2 & -\omega_3^2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} \quad (2.14)$$

Таким образом нахождение значений вектора q , путем решения ОДУ, позволяет найти значения скоростей и координат масс m_1, m_2, m_3 в любой момент времени и понять какие процессы протекают в системе во время взаимодействия.

2.2 Аналитическое решение

Определитель матрицы A равен $\omega_1^2\omega_2^2\omega_3^2$, что позволяет нам найти точное решение при помощи спектрального разложения.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\omega_1^2 - \omega_{21}^2 & \omega_{21}^2 & 0 & 0 & 0 & 0 \\ \omega_2^2 & -\omega_2^2 - \omega_{32}^2 & \omega_{32}^2 & 0 & 0 & 0 \\ 0 & \omega_3^2 & -\omega_3^2 & 0 & 0 & 0 \end{bmatrix} \quad (2.15)$$

Спектральное разложение имеет вид:

$$A = P^{-1}\Lambda P \quad (2.16)$$

, где A - изначальная матрица, P - матрица собственных векторов, Λ - диагональная матрица, на диагонали которой расположены собственные числа матрицы A . Для данной задачи система ОДУ имеет вид:

$$\frac{d}{dt}q = Aq \quad (2.17)$$

Применив спектральное разложение к матрице A , получим:

$$\frac{d}{dt}q = P^{-1}\Lambda Pq, \quad (2.18)$$

$$\frac{d}{dt}Pq = \Lambda Pq \quad (2.19)$$

Произведем замену переменных $\tilde{q} = Pq$. Теперь уравнение имеет вид:

$$\frac{d}{dt}\tilde{q} = \Lambda\tilde{q} \quad (2.20)$$

Отсюда вектор \tilde{q} :

$$\tilde{q} = \begin{bmatrix} e^{\lambda_1 t} \\ \ddots \\ e^{\lambda_n t} \end{bmatrix} \tilde{q}(0), \quad (2.21)$$

где t - время. В результате обратной замены получим $q = P^{-1}\tilde{q}$.

Таким образом решение системы ОДУ сводится к нахождению собственных чисел матрицы A . Собственные числа можно найти аналитически, для этого запишем характеристический полином матрицы A :

$$\begin{aligned} \lambda^6 + \lambda^4(\omega_1^2 + \omega_2^2 + \omega_{21}^2 + \omega_3^2 + \omega_{32}^2) + \lambda^2(\omega_1^2\omega_2^2 + \omega_{21}^2\omega_3^2 + \omega_{21}^2\omega_{32}^2 + \omega_3^2\omega_2^2 + \omega_3^2\omega_1^2 + \omega_{32}^2\omega_1^2) \\ + \omega_1^2\omega_2^2\omega_3^2 = 0 \end{aligned} \quad (2.22)$$

Можно легко увидеть, что данное выражение легко сводится к кубическому уравнению вида:

$$k^3 + k^2W_1 + kW_2 + W_3 = 0 \quad (2.23)$$

, где

$$k = \lambda^2 \quad (2.24)$$

$$W_1 = \omega_1^2 + \omega_2^2 + \omega_{21}^2 + \omega_3^2 + \omega_{32}^2 \quad (2.25)$$

$$W_2 = \omega_1^2\omega_2^2 + \omega_{21}^2\omega_3^2 + \omega_{21}^2\omega_{32}^2 + \omega_3^2\omega_2^2 + \omega_3^2\omega_1^2 + \omega_{32}^2\omega_1^2 \quad (2.26)$$

$$W_3 = \omega_1^2\omega_2^2\omega_3^2 \quad (2.27)$$

Решим данное уравнение при помощи тригонометрической формулы Виета для кубического уравнения:

$$Q = \frac{W_1^2 - 3W_2}{9} \quad (2.28)$$

$$R = \frac{2W_1^3 - 9W_1W_2 + 27W_3}{54} \quad (2.29)$$

$$S = Q^3 - R^2 \quad (2.30)$$

Если $S > 0$, то имеем три действительных корня, вычисляемых при помощи тригонометрических функций. При $S < 0$, тригонометрические функции заменяются на гиперболические и пара корней является комплексными. При $S = 0$ уравнение вырождено и имеет два корня. Программная реализация решателя на языке Python, использующего тригонометрическую формулу Виета, представлена в листинге 2.1.

Листинг 2.1 — Программная реализация решателя кубического уравнения при помощи теоремы Виета

```

def cbslv(a, b, c):
    Q = (a**2 - 3*b)/9
    R = (2*a**3 - 9*a*b+27*c)/54
    S = Q**3 - R**2
    if S > 0:
        phi = cm.t.acos(R/cm.t.sqrt(Q**3))/3
        x1 = -2*cm.t.sqrt(Q)*cm.t.cos(phi) - a/3
        x2 = -2*cm.t.sqrt(Q+np.pi/3)*cm.t.cos(phi) - a/3
        x3 = -2*cm.t.sqrt(Q-2*np.pi/3)*cm.t.cos(phi) - a/3
    elif S < 0:
        if Q > 0:
            phi = cm.t.acosh(abs(R)/cm.t.sqrt(Q**3))/3
            x1 = -2*np.sign(R)*cm.t.sqrt(Q)*cm.t.cosh(phi) - a/3 # real root
            x2 = np.sign(R)*cm.t.sqrt(Q)*cm.t.cosh(phi) - a/3 +
                1j*cm.t.sqrt(3)*cm.t.sqrt(Q)*cm.t.sinh(phi)
            x3 = np.sign(R)*cm.t.sqrt(Q)*cm.t.cosh(phi) - a/3 -
                1j*cm.t.sqrt(3)*cm.t.sqrt(Q)*cm.t.sinh(phi)
        elif Q < 0:
            phi = cm.t.asinh(abs(R)/cm.t.sqrt(abs(Q)**3))/3
            x1 = -2*np.sign(R)*cm.t.sqrt(abs(Q))*cm.t.sinh(phi) - a/3 # real root
            x2 = np.sign(R)*cm.t.sqrt(abs(Q))*cm.t.sinh(phi) - a/3 +
                1j*cm.t.sqrt(3)*cm.t.sqrt(abs(Q))*cm.t.cosh(phi)
            x3 = np.sign(R)*cm.t.sqrt(abs(Q))*cm.t.sinh(phi) - a/3 -
                1j*cm.t.sqrt(3)*cm.t.sqrt(abs(Q))*cm.t.cosh(phi)
        elif Q == 0:
            x1 = -np.cbrt(c-(a**3)/27) - a/3
            x2 = -(a+x1)/2 + 1j/2*cm.t.sqrt(abs((a - 3*x1)*(a+x1)-4*b))
            x2 = -(a+x1)/2 - 1j/2*cm.t.sqrt(abs((a - 3*x1)*(a+x1)-4*b))
    elif S==0:
        x1 = -2*np.cbrt(R)-a/3
        x2 = np.cbrt(R)-a/3
        x3 = None
    return x1, x2, x3

```

Изначальной идеей нахождения аналитического решения было избавление от переменных и от большого количества арифметических операций. Это могло позволить найти прямые зависимости состояний системы от каких либо из ее параметров. Но в ходе подстановки не удалось сократить какие либо переменные, а вкупе с использованием ресурсозатратных гиперболических функций и вычислением сложных степеней больших чисел аналитическое ре-

шение приводит к ошибкам вычислений и отклонениям. А так же к большему времени расчетов, относительно численных методов.

3 ВЫЧИСЛИТЕЛЬНЫЙ МЕТОД

3.1 Метод Рунге-Кутта 4-го порядка

В связи с тем, что точное решение имеет большую вычислительную сложность, то воспользуемся методом Рунге-Кутта 4-го порядка для решения системы ОДУ[7]. Согласно методу приближенное значение в последующих точках вычисляется по итерационной формуле:

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (3.1)$$

Коэффициенты в общем виде находятся по формулам:

$$k_1 = f(x_i, y_i) \quad (3.2)$$

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \quad (3.3)$$

$$k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) \quad (3.4)$$

$$k_4 = f(x_i + h, y_i + hk_3) \quad (3.5)$$

$$(3.6)$$

, где h - величина шага сетки по x .

Относительно задачи, рассматриваемой в данной работе аналогом x является время t . А аналогом y - вектор q , описывающий состояние системы (2.13). Тогда уравнение $\frac{d}{dt}q = f(t, q)$ принимает вид (2.14). Начальные условия задачи Коши нам известны из постановки задачи. Легко можно заметить, что по сравнению с аналитическим решением, в методе Рунге-Кутта требуется произвести меньшее количество арифметических операций, что критично для данной задачи, потому что требуется проводить анализ большого количества возможных состояний системы, а это значит каждый раз симулировать процесс. А вкупе с тем что данный численный метод имеет порядок точности $O(h^4)$, то было решено использовать данный метод для получения приближенных значений состояний системы для каждого момента времени. На рисунке 3.1 продемонстрированы графики зависимости состояния системы от времени

для параметров $k_1 = k_2 = k_3 = 1.5$ и $m_1 = m_2 = m_3 = 1$ со значением шага $h = 0.01\text{c}$, что дает в итоге точность $O(h^4) = 10^{-8}$. На основе графиков и порядка точности можно сделать вывод, что данной точности интегрирования достаточно для решения поставленной задачи.

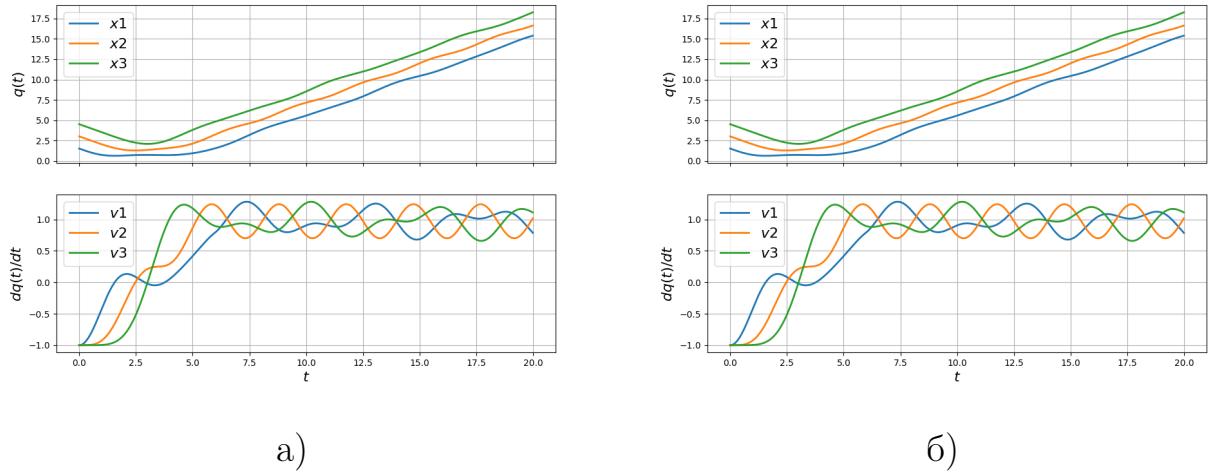


Рис. 3.1—Траектории скоростей и координат системы трех тел, полученные аналитически (а) и численно (б) .

3.2 Конечно-автоматная модель

Так как система тел при движении во времени может принимать несколько состояний, то мы можем рассмотреть данную систему в качестве конечного автомата с тремя возможными состояниями, описывающий динамическое состояние системы. Всего возможно три состояния системы:

- а) тело взаимодействует со стенкой;
- б) тело не взаимодействует со стенкой (свободное движение);
- в) нефизический случай.

При наличии взаимодействия со стенкой в уравнении появляется пара сил: сила упругости первой пружины F и, согласно третьему закону Ньютона действует обратная сила со стороны стенки. При отсутствии взаимодействия, очевидно, данная пара сил не существует, поэтому в состоянии оторванном от стенки считается, что первая пружина отсутствует. Взаимодействие со стенкой происходит, когда значение координаты массы m_1 меньше или равняется длине пружины l в ненапряженном состоянии.

Нефизическими случаем является такое состояние системы, при котором значение координаты массы m_{i+1} становится меньше значения координаты массы m_i , изначально находящейся ближе к стенке. Также нефизическими случаем является, когда значение координаты какой-либо массы становится меньше 0 (масса "прошла" сквозь стенку). Граф, описывающий конечный автомат, представлен на рисунке ???. Программная реализация автомата представлена на блок-схеме ??.

Листинг 3.1 — Программная реализация конечного автомата

```
//UpdateState - checks the conditions of state machine
func (st *StateMachine) UpdateState() error {
    cond := ConditionFromVec(*st.CurCond)
    if cond.X[0] <= -st.Length || cond.X[1] <= -2*st.Length || cond.X[2] <=
        -3*st.Length || cond.X[1] <= cond.X[0]-st.Length || cond.X[2] <=
        cond.X[1]-st.Length {
        st.CurState = NonPhis
    }
    switch st.CurState {
```

```

case Started:
    if !st.Mute {
        fmt.Printf("HitWall: %v sec\n", st.CurTime)
    }
    st.CurState = HitWall
case HitWall:
    if cond.X[0] > 0 {
        if !st.Mute {
            fmt.Printf("BouncedBack: %v sec\n", st.CurTime)
        }
        st.CurState = BouncedBack
    }
case BouncedBack:
    if cond.X[0] <= 0. {
        if !st.Mute {
            fmt.Printf("HitWall: %v sec\n", st.CurTime)
        }
        st.CurState = HitWall
    }
case NonPhis:
    return errors.New("NonPhis conditions reached")
}
if st.CurTime >= st.FinTime {
    return errors.New("Final time reached, simulation stopped")
}
return nil
}

```

4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

4.1 Используемые программные средства

В данной работе было использованы следующие языки программирования:

- Python
- Golang

Изначально все математические модели были реализованы на языке Python с использованием библиотеки numpy, это было сделано для проверки их работоспособности. Несомненным преимуществом работы Python является его простота в использовании, а так же большое количество готовых библиотек, сильно упрощающих разработку. Но так же Python является языком динамической типизации, что значительно влияет на скорость его работы, поэтому для задач, требующих больших вычислительных мощностей, которой как раз и является данная задача, использование Python является нецелесообразным. Поэтому дальнейшая разработка производилась на языке Golang.

Язык Golang

Среди современных языков программирования следует выделить язык программирования Go, реализация которого была выполнена командой разработчиков фирмы Google [8]. Язык Go разрабатывался как язык программирования для создания высокоеффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языкам Си и С++.

Основные возможности Golang:

- Go — язык со строгой статической типизацией. Доступен автоматический вывод типов, для пользовательских типов — "утиная типизация".
- Полноценная поддержка указателей, но без возможности применять к ним арифметические операции, в отличие от C/C++/D.

- Использование динамических массивов, хеш-таблиц, срезов (слайсов), вариант цикла для обхода коллекции.
- Средства функционального программирования: неименованные функции, замыкания, передача функций в параметрах и возврат функциональных значений.
- Автоматическое управление памятью со сборщиком мусора.
- Средства объектно-ориентированного программирования, но без поддержки наследования реализации (наследуются только интерфейсы). По большому счёту, Go является процедурным языком с поддержкой интерфейсов.
- Средства параллельного программирования: встроенные в язык потоки (`go routines`), взаимодействие потоков через каналы и другие средства организации многопоточных программ.
- Достаточно лаконичный и простой синтаксис, основанный на Си.

Go содержит конструкцию, которая позволяет запустить исполнение вызова некоторой функции в отдельном потоке. Такая функция называется в Go горутиной (англ. goroutine), ниже приведен пример оператора вызова горутины:

Листинг 4.1 — Пример вызова горутины

```
go f(args)
```

Каналы используются в качестве средства взаимодействия между горутинами. Помимо каналов в пакете `sync` содержатся такие примитивы синхронизации, как мьютексы или условные переменные, однако их рекомендуется применять только для низкоуровневых библиотек.

Каналы в Go типизированы и являются объектами первого класса. Их можно посыпать в функции, возвращать из функций, присваивать переменным и полям структур, а также пересыпать по каналам. Они могут работать только в одну сторону или в обе. Причем, если обычный двунаправленный канал передать в функцию, где он используется, например, только на прием, его можно промаркировать соответствующим образом в списке параметров

функции и компилятор проверит правильность его использования. Обычные каналы блокируют горутину, которая пытается получить из или послать что-то в канал, пока с другой стороны не будет произведено соответствующее обратное действие. Но канал может иметь буфер, который может накапливать и отдавать накопленное определенное количество значений без блокировки. Канал может быть закрыт встроенной функцией *close*, после этого посылка в него значений приведет к "панике" так в данном языке называется экстренное завершение программы в случае ошибки. Если закрытый канал имеет непустой буфер, эти значения все еще могут быть получены. Но когда буфер уже пуст или его нет, закрытый канал при попытке чтения будет выдавать нулевые значения того типа, для пересылки которых он создан. То есть, закрытый строковый канал будет выдавать пустые строки, целочисленный — нули, канал указателей — nil. Более подробно горутины и в целом язык Golang описывается в книге [9]

Также неоспоримым преимуществом Golang является его скорость работы и компиляции, сопоставимая с такими языками, как Java, C, C++. На рисунке 4.1 сопоставлены популярные языки и Go, на примере часто встречающихся задач.

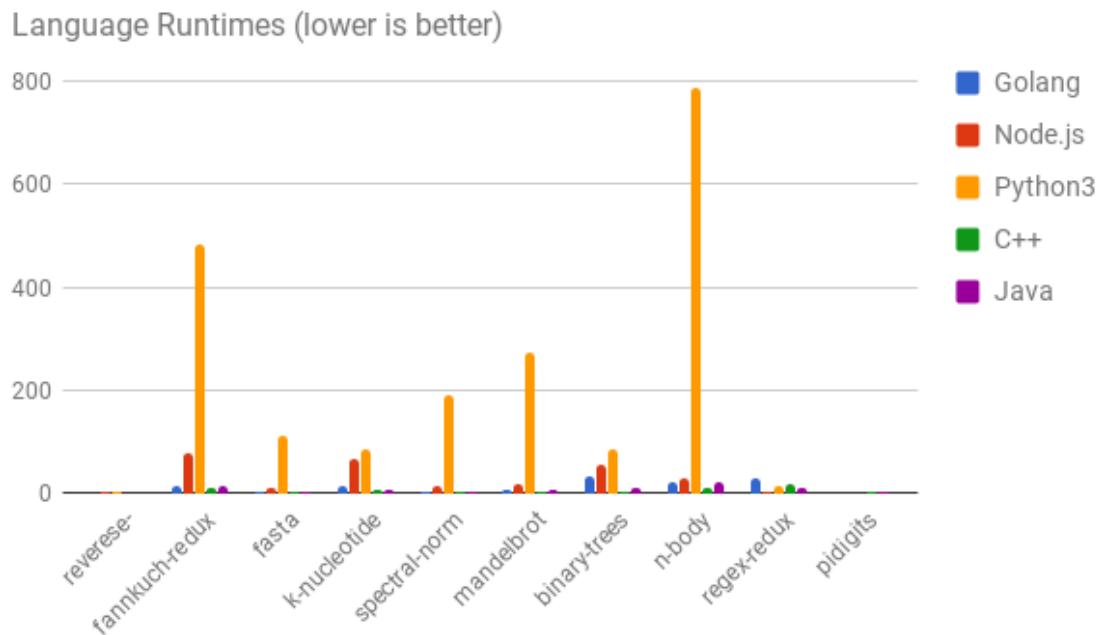


Рис. 4.1 — Сопоставление Golang, Python, C++, Java, Node.js.

Учитывая наличие сборщика мусора, времени компиляции сопоставимым с C++, а так же простотой синтаксиса и читабельностью кода, было решено остановиться на данном языке как на основном языке разработки. Так же в Golang есть возможность исполнения кода, написанного на языке C, что позволяет уменьшить время работы программ, а так же использовать математические библиотеки blas и lapack, которые реализованы в пакете **gonum**[10].

4.2 Параллелизация вычислений

Задача параллелизации представляет собой программное ускорение расчета полей накопленных энергий после соударения системы трех тел со стенкой в зависимости от конфигурации системы. Расчет энергий происходит путем моделирования взаимодействия тел, что представляет собой решение системы ОДУ для каждого момента времени при помощи метода Рунге-Кутта 4 порядка [7]. Для понимания какой из параметров вносит больший вклад в накопление энергии требуется провести большое количество симуляций, которые линейно влияют на время расчетов. Было решено использовать возможности языка Golang, для программного ускорения времени расчетов и более полного использования ресурсов компьютера, а в дальнейшем и вычислительного кластера.

Для ускорения вычисления были реализованы функции считающие двумерные поля(листинг ??) и трехмерные облака точек(листинг ??). Обе функции используют все ядра центрального процессора.

Параллелизация работы в случае функции представленной в листинге ?? достигается за счет того, что на каждую горутину выделяется одна плоскость в трехмерном пространстве, которую необходимо расчитать, после окончания расчета по каналу отправляется строковая переменная содержащая в себе данные расчетов, полученные данной горутиной.

Параллелизация работы в случае функции представленной в листинге ?? достигается, за счет разделения плоскости расчетов между всеми доступными ядрами процессора, что позволяет параллельно производить несколько симуляций. Во время расчетов, аналогично трехмерному случаю формируется строковая переменная, которая отправляется по каналу.

Листинг 4.2 — Функция барьер

```
func barrier(nrouts int, ch chan string, fil *os.File) {
    for r := 0; r < nrouts; r++ {
        x := <-ch
        log.Println("goroutine ", r, "writed", len(x))
        fil.WriteString(x)
    }
}
```

}

Полученные через канал строки обрабатываются в функции представленной в листинге 4.2. Данная функция принимает по каналу определенное количество переменных типа строка, которое равно число запущенных горутин, тем самым реализует работу барьера и блокирует выполнение родительской горутины до завершения работы всех дочерних горутин. После строковая переменная записывается в файл для дальнейшей визуализации.

4.3 Взаимодействие с кластером

В современных задачах зачастую возникают задачи предъявляющие высокие требования к аппаратным мощностям, которые не всегда могут быть доступны. Зачастую в таком случае прибегают к работе с удаленной машиной, обладающей достаточной вычислительной мощностью, чтобы удовлетворить требования. Очевидным решением для облегчения взаимодействия конечного пользователя и ПО является клиент-серверное приложение. Данный подход позволит свести взаимодействие конечного пользователя с исходным кодом программ к минимуму.

В данной работе для работы с серверной частью был использован язык Golang, а в частности пакет **gin-gonic** [11], который является веб-фреймворком, предоставляющий более быструю работу чем стандартный сетевой пакет. Основной частью пакета является роутер, определяющий какой запрос пришел со стороны клиента. Разработчик же описывает каким образом должен реагировать сервер на тот или иной запрос, путем назначения функций обработчиков на различные варианты запросов. Golang не требует установки какого либо дополнительного программного обеспечения для реализации сервера(например Apache), все эти функции уже есть в стандартном сетевом пакете.

Клиентский интерфейс содержит в себе следующие страницы:

- а) Главная страница
- б) Симуляция системы трех тел
- в) Вычисление двумерного поля энергий
- г) Вычисление трехмерного облака точек
- д) Результаты симуляций
- е) Страница ошибки

Данные страницы реализованы на языке html и формируют GET или POST запросы на сервер. Страницы 2,3,4 - содержат в себе поля для ввода

параметров симуляции и интегрирования, которые формируют POST запрос и на сервере, который представляет собой кластер, обрабатываются и запускают симуляцию. В листинге 4.3 реализован парсинг формы запроса, получение данных, требующихся для запуска симуляции и непосредственно запуск симуляции.

Листинг 4.3 — Реализация обработчика POST запроса

```
r.POST("/sim", func(c *gin.Context) {
    var m [3]float64
    m[0], _ = strconv.ParseFloat(c.PostForm("m1"), 64)
    m[1], _ = strconv.ParseFloat(c.PostForm("m2"), 64)
    m[2], _ = strconv.ParseFloat(c.PostForm("m3"), 64)

    var k [3]float64
    k[0], _ = strconv.ParseFloat(c.PostForm("k1"), 64)
    k[1], _ = strconv.ParseFloat(c.PostForm("k2"), 64)
    k[2], _ = strconv.ParseFloat(c.PostForm("k3"), 64)

    v, _ := strconv.ParseFloat(c.PostForm("vel"), 64)
    l, _ := strconv.ParseFloat(c.PostForm("len"), 64)

    step, _ := strconv.ParseFloat(c.PostForm("stpt"), 64)
    fint, _ := strconv.ParseFloat(c.PostForm("fint"), 64)

    c.Redirect(301, "/dat")
    avg := c.Request.FormValue("avg")

    body := slv.NewThreeBodyModel(m, k)
    //y, mon, d := time.Now().Date()
    filename := fmt.Sprintf("sym%v.dat",
        time.Now().Format("15:04:05_02Jan06"))
    file, err := os.Create(fmt.Sprintf("./dat/", filename))
    if err != nil {
        log.Println("Cannot create file! : ", err)
        c.File("./www/error.html")
        return
    }

    machine := slv.StateMachFromModel(body, v, l, step, 0, fint)
    if avg == "on" {
        go func(machine grn.StateMachine, body grn.ThreeBodyModel, file
            *os.File, filename string) {
            filsys[filename] = false
            slv.SimulateAv(machine, body, file)
        }
    }
}
```

```

        filsys [filename] = true
    }(machine, body, file, filename)

} else {
    go func(machine grn.StateMachine, file *os.File, filename string) {
        filsys [filename] = false
        time.Sleep(30 * time.Second)
        slv.Simulate(machine, file)
        filsys [filename] = true
    }(machine, file, filename)
}
})

```

Симуляция запускается отдельной горутиной, что позволяет работать серверу и расчетному ПО параллельно и не останавливать работу сервера. Также для реализации отслеживания процесса расчетов был реализован ассоциативный массив *filsys*, который содержит текущее состояние для каждого файла симуляции, где имя файла является ключом, а значение отображает состояние файла, если значение *false*, то вычисление данного файла еще не закончено, иначе файл может быть загружен с кластера для дальнейшей визуализации.

Страница 5, формируется динамически сервером, в зависимости от состояния файла *filsys*. Для автоматической генерации html страницы используется пакет *template*, который позволяет формировать страницу в зависимости от контекста.

Листинг 4.4 — Html шаблон

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
</head>
<body>
    <div>
        <p><h2>Simulation results</h1></p>
        <hr>
        {{ range $key, $value := . }}
        <li>
            {{ $key }}
            {{ if $value }}

```

```

<form method="POST" action="/dat">
    <input name="file" type="hidden" value="{{ $key
        }}><input type="submit" value="Download">
</form>
{{ else }}
    <strong>File in progress...</strong>
{{ end }}
</li>
{{ end }}
</div>
<a href="/">Back to menu</a>
</body>
</html>

```

Шаблон в листинге 4.4, формирует страницу в зависимости от значений *filesys*, *range \$key, \$value := .* - цикл, который перебирает все ключи и значения, переданные программой, формирующей страницу(листинг 4.5). В зависимости от значений *\$value*, формируется либо сообщение о том что файл еще рассчитывается, либо форма POST запроса со скрытым полем, содержащим имя файла. При нажатии на кнопку выбранный файл расчетов скачивается с кластера на клиентскую машину.

Листинг 4.5 – Формирование html

```

r.GET("/dat", func(c *gin.Context) {
    t, err := template.ParseFiles("./www/dat.html")
    if err != nil {
        panic(err)
    }

    err = t.Execute(c.Writer, filsys)
    if err != nil {
        panic(err)
    }
})

```

Simulate

- **Enable center of mass trajectories**
- **Start parameters:**
Velocity: Spring length:
- **Body parameters:**
Masses: Stiffnesses:
- **Integration parameters:**
Final time: Time step:
-

Рис. 4.2—Зависимость $\Delta\tilde{E}$ от различных значений масс. Каждый график соответствует одной переменной массе.

Calculate 2D slice

- Select what to variate
- Start parameters:
Velocity: Spring length:
- Integration parameters:
Start mass: Final mass: Grid step: Final time:
-

Рис. 4.3—Зависимость $\tilde{\Delta E}$ от различных значений масс. Каждый график соответствует одной переменной массе.

Calculate 3D point cloud

- Select what to variate
 - Masses
 - Stiffnesses
- Start parameters:
Velocity: Spring length:
- Integration parameters:
Start mass: Final mass: Grid step: Final time:
-

Рис. 4.4—Зависимость \tilde{E} от различных значений масс. Каждый график соответствует одной переменной массе.

5 ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Основным вопросом, поставленным в данной работе является нахождение таких состояний системы трех тел, что внутренняя энергия, запасенная в пружинах будет максимальна или минимальна. А так же нахождение таких состояний, при которых внутренняя энергия тела после соударения равна 0 или же наоборот вся вся кинетическая энергия перейдет во внутреннюю.

Для нахождения потерь энергии во время взаимодействия запишем Закон сохранения энергии:

$$\frac{Mv_0^2}{2} = \frac{m_1v_1^2}{2} + \frac{m_2v_2^2}{2} + \frac{m_3v_3^2}{2} + \frac{k_2\Delta l^2}{2} + \frac{k_3\Delta l^2}{2} \quad (5.1)$$

, где $M = m_1 + m_2 + m_3$. Энергия, накопленная первой пружиной отсутствует, так как вся энергия пружины переходит в ускорение тел и растяжение и сжатие остальных пружин.

Так как нас интересует потери энергии для тела на макроуровне, то рассмотрим систему трех тел после соударения как единое тело с постоянной скоростью, тогда разница между кинетическими энергиями до и после соударения и будет показывать накопленную внутреннюю энергию тела. Для этого запишем уравнение для скорости центра масс:

$$v_c = \frac{m_1v_1 + m_2v_2 + m_3v_3}{M} \quad (5.2)$$

, тогда Закон сохранения энергии примет вид:

$$\frac{Mv_0^2}{2} = \frac{Mv_c^2}{2} + \Delta E \quad (5.3)$$

, где ΔE - энергия накопленная в пружинах и кинетическая энергия движения масс относительно центра масс.

На рисунке 5.1 на графике продемонстрирован результат симуляции с нахождением скорости центра масс, а так же его координат, из чего можно сделать вывод что после взаимодействия часть кинетической энергия тела перешла во внутреннюю энергию тела, соответственно скорость тела после соударения уменьшилась. Массы для данной симуляции заданы следующим

образом: $m_1 = m_2 = 1; m_3 = 4$. Начальная скорость $v_0=1\text{м/с}$. При этом можно увидеть, что итоговая скорость тела после взаимодействия меньше первоначальной, засчет преобразования энергий. Энергия, перешедшая во внутреннюю и есть искомая ΔE .

Проведем вычислительный эксперимент с различными входными параметрами системы (длина пружин $l = 1$). Для определения параметра, в большей степени влияющего на состояние системы измерим ΔE для каждого параметра в некоторых пределах. Но очевидно что, например, большая масса способна сильней повлиять на численное значение ΔE . Тогда введем $\Delta \tilde{E} = \frac{\Delta E}{E_0}$, где E_0 - изначальная энергия системы. $\Delta \tilde{E}$ - коэффициент эффективности диссипации энергии. Для случая представленного на рисунке 5.1 $\Delta \tilde{E} = 27.26\%$. На графиках на рисунке 5.2 продемонстрированы зависимость коэффициента эффективности диссипации от различных масс.

Аналогичный эксперимент проведем для переменных жесткостей пружин:

На первом графике на рисунке 5.3 видно, что $\Delta \tilde{E}$ сильнее всего зависит от значения жесткости первой пружины. При этом значение $\Delta \tilde{E}$ линейно

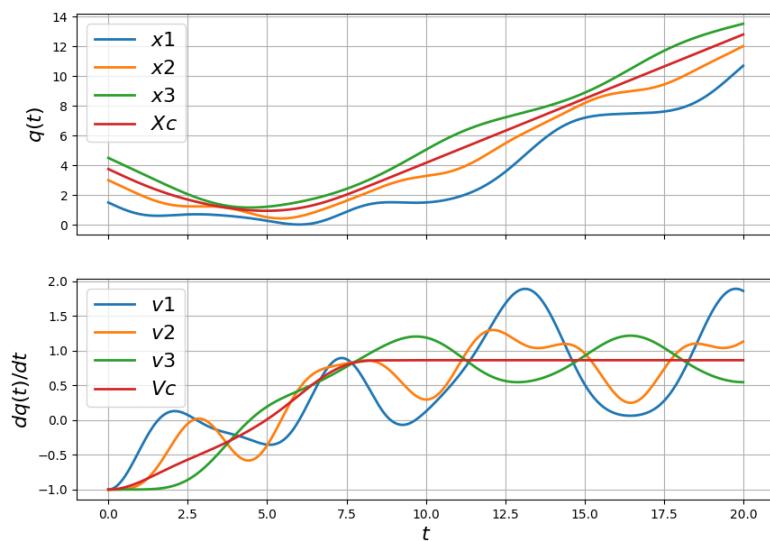


Рис. 5.1 — График координат и скоростей тел на микроуровне и на макроуровне(красным).

достигает максимума, а затем логарифмически убывает. На нижнем графике продемонстрировано количество ударов о стенку в зависимости от жесткости пружины, важно заметить, что количество соударений зависит только от

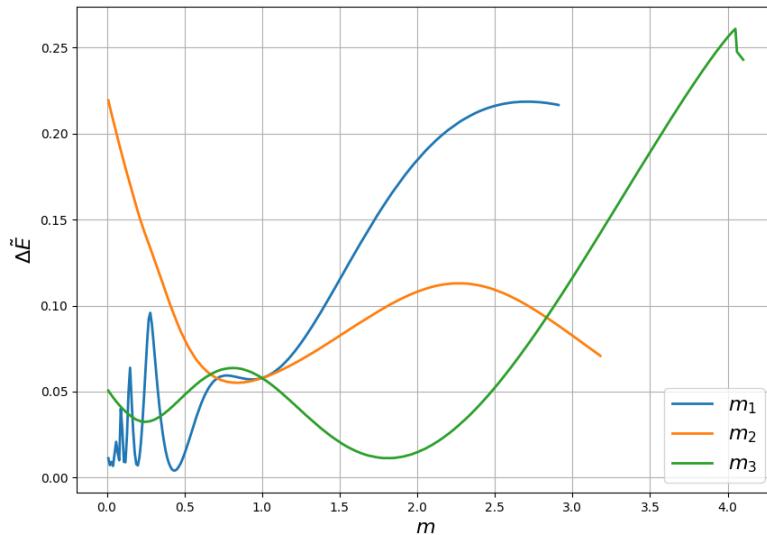


Рис. 5.2—Зависимость $\Delta\tilde{E}$ от различных значений масс. Каждый график соответствует одной переменной массе.

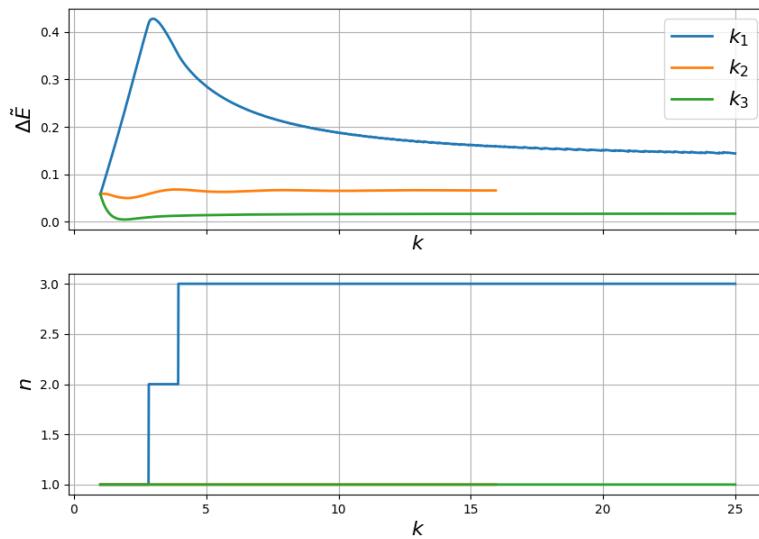


Рис. 5.3—Зависимость $\Delta\tilde{E}$ от различных значений жесткости пружин. Каждый график соответствует одной переменной массе. Нижний график демонстрирует зависимость количества соударений со стенкой от жесткости.

жесткости k_1 . На основе рисунка 5.3 можно сделать вывод, что максимальные значения $\Delta\tilde{E}$ находятся при значении $n = 2$. При этом можно заметить, что значение жесткости k_2 практически не влияет на состояние системы, а увеличение значения жесткости k_3 , наоборот приводят к уменьшению $\Delta\tilde{E}$, а следовательно и накопленной системой энергии.

Рассмотрим поведение системы для двух переменных параметров, для определения значений $\Delta\tilde{E}$. Для этого построим срезы по различным плоскостям шестимерного пространства значений $k_1 k_2 k_3 m_1 m_2 m_3$. Белые области на графиках обозначают нефизические случаи. Для того чтобы избавиться от нефизических случаев было решено увеличить длину пружин и принять ее значение $l = 10$.

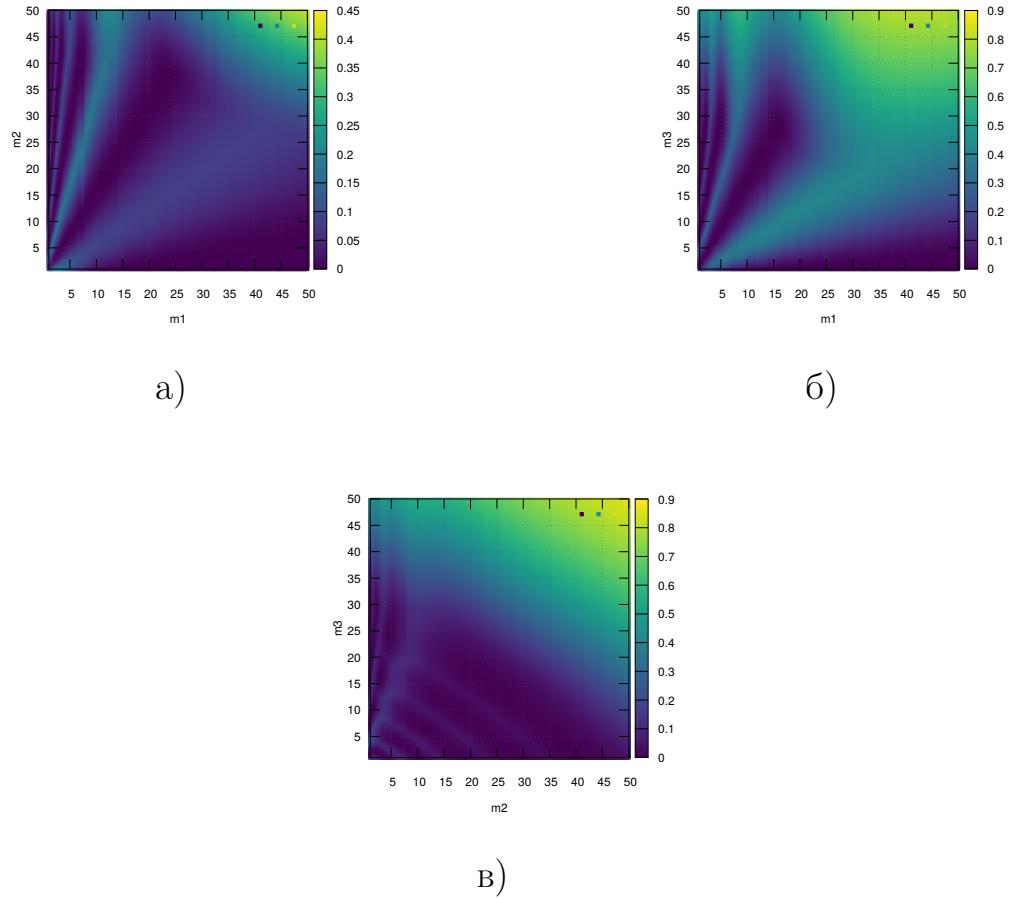


Рис. 5.4—Срезы значений $\Delta\tilde{E}$ для двух переменных масс: $m_1 m_2$ (а), $m_1 m_3$ (б), $m_2 m_3$ (в). $l = 10$ Фиксированные жесткости и массы =1

Полученные срезы позволяют понять какие переменные вносят больший вклад в диссипацию энергии в системе. Но данная выборка не может считаться полностью репрезентативной, так как в большинстве случаев области

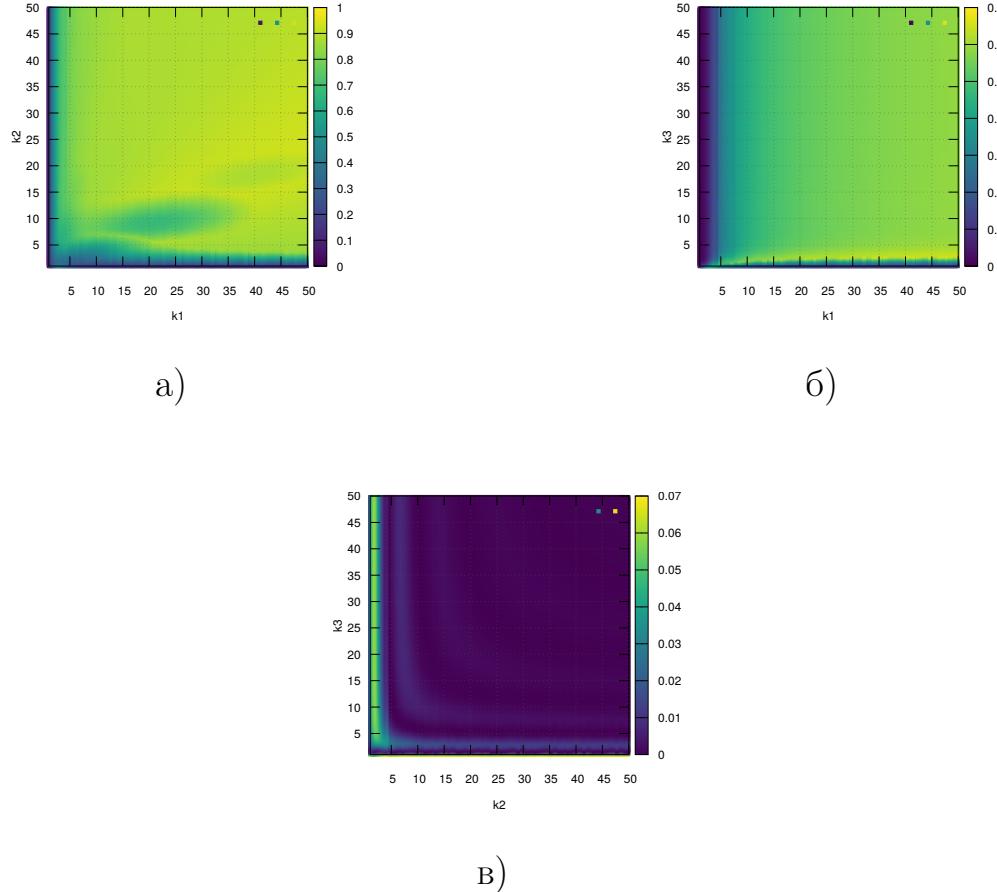


Рис. 5.5—Срезы значений $\Delta \tilde{E}$ для двух переменных жесткостей: k_1 k_2 (а), k_1 k_3 (б), k_2 k_3 (в). $l = 10$ Фиксированные жесткости и массы $= 1$

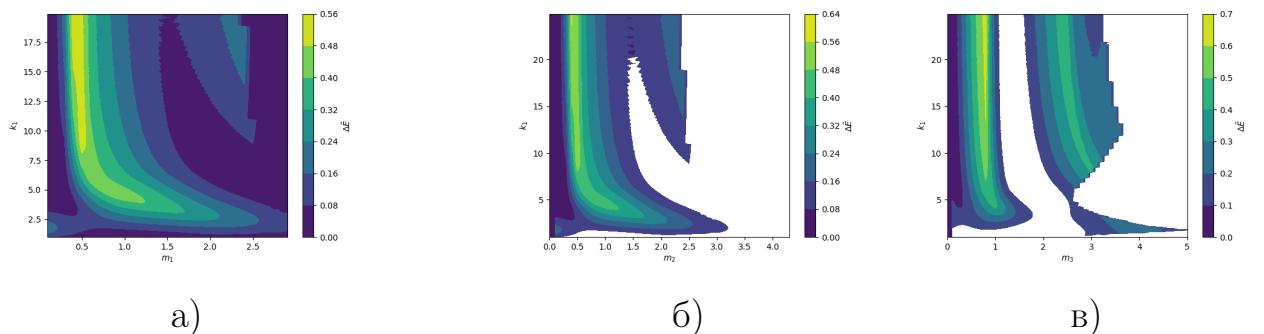


Рис. 5.6—Срезы значений $\Delta \tilde{E}$ для переменной жесткости k_1 и переменной массы: m_1 (а), m_2 (б), m_3 (в)

значений подобраны таким образом чтобы ограничить область нефизических случаев. Но даже такие данные позволяют сделать некоторые выводы насчет поведения системы. В частности на рисунке 5.4 можно увидеть периодичность максимумов, при этом для графиков (а) и (б) она зависит от $\operatorname{tg}\alpha$ равное соотношению масс. Для графика (в), область оптимальных значений находится в области $m_3 \rightarrow \min$, но в то же время максимумы чередуются в зависимости от значения m_2 . Что позволяет сделать вывод, о том что $\Delta\tilde{E}$ зависит в большей степени не от значений масс системы, а от их отношения.

При этом если рассматривать графики, изображенные на рисунке 5.5, то можно легко заметить по графикам (а) и (б), что $\Delta\tilde{E}$ практически не зависит от значений k_2 и k_3 , но при этом линейно зависят от значений k_1 . Основной вклад в диссиацию энергии вносит первая пружина, а на по-

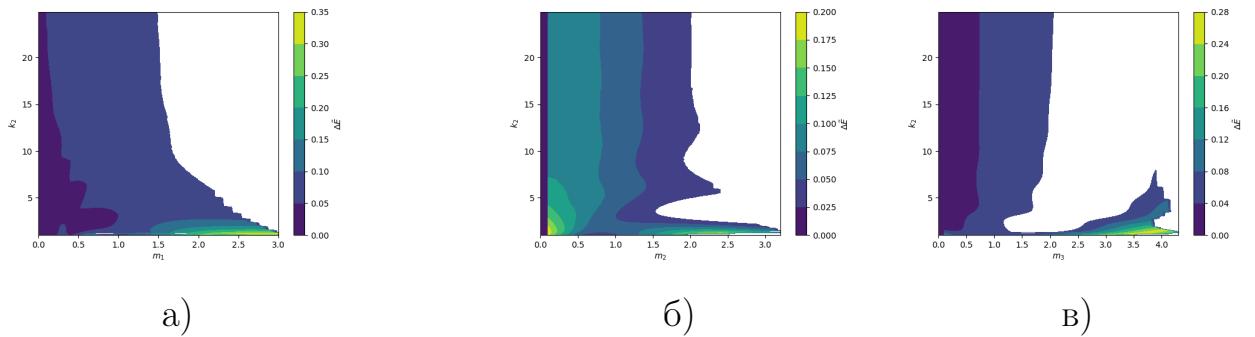


Рис. 5.7—Срезы значений $\Delta\tilde{E}$ для переменной жесткости k_2 и переменной массы: m_1 (а), m_2 (б), m_3 (в). Цветом обозначена величина накопленной энергии

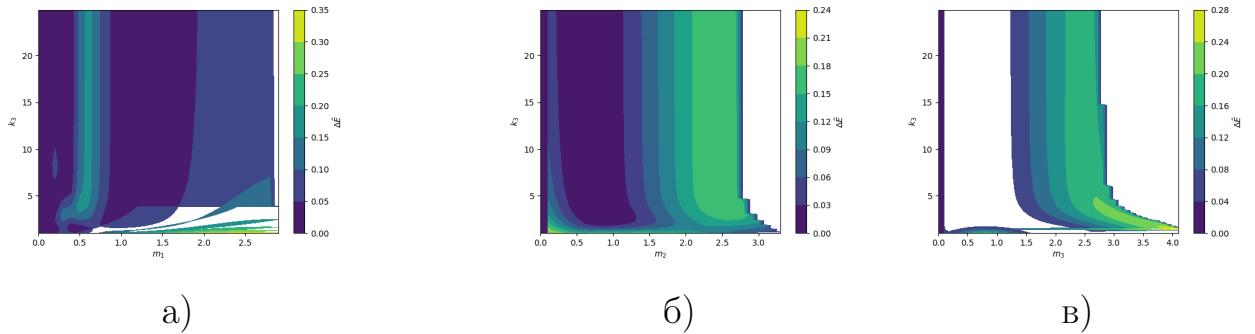


Рис. 5.8—Срезы значений $\Delta\tilde{E}$ для переменной жесткости k_3 и переменной массы: m_1 (а), m_2 (б), m_3 (в).

графику (в) видно что при минимизации жесткости k_2 происходит увеличение $\Delta\tilde{E}$, что соответствует данным, полученным на графике (а).

На рисунке 5.6 можно увидеть, что $\Delta\tilde{E} \rightarrow \max$ при определенных значениях массы и линейно зависят от k_1 в окрестностях оптимальной массы. Так же можно предположить некоторую периодичность процесса, но по причине того, что для данных переменных, система достаточно быстро переходит в нефизическое состояние, то делать сложно делать какие либо однозначные выводы.

Рисунки 5.7 и 5.8 демонстрируют зависимость $\Delta\tilde{E}$ для переменных масс и жесткостей k_2 и k_3 соответственно. Но легко можно заметить, что максимальные значения $\Delta\tilde{E}$ находятся на границе с нефизическими состояниями системы. Скорей всего данные отношения не представляют какой либо практический интерес.

Контуры срезов, полученные на рисунках 5.4 - 5.8 соответствуют данным, полученным на рисунках 5.2 и 5.3. На графиках присутствует периодичность для масс, которую можно заметить и на полученных срезах, а так же видна линейная зависимость $\Delta\tilde{E}$ от k_1 .

Рассмотрим трехмерные пространства изменяемых параметров масс и жесткостей пружин. Для этого провердем численный эксперимент для каждого значения массы с выбранным шагом($h = 0.5$), и получим облака точек, изображенные на рисунке 5.9, значения накопленной энергии в которых, обозначено цветом.

Так как сами по себе графики представленные на рисунке 5.9 не очень репрезентативны, то построим графики содержащие только точки в которых значение $\Delta\tilde{E} < 0.01$. точек, отображающие зоны минимальных и максимальных накопленных энергий.

Посторим траектории системы трех тел в полученных областях максимальных и минимальных энергий. Полученные траектории могут продемонстрировать механизмы перехода энергии с макроуровня на микроуровень.

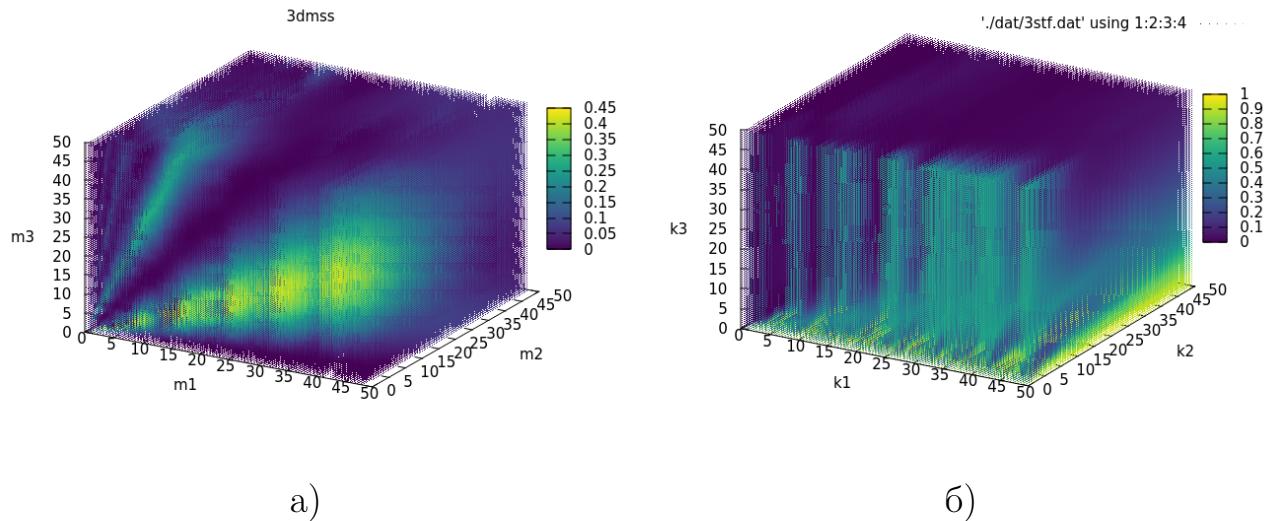


Рис. 5.9—Зависимость $\tilde{\Delta E}$ от различных значений масс (а) и различных значений жесткостей (б). Значение $\tilde{\Delta E}$ указано цветом.

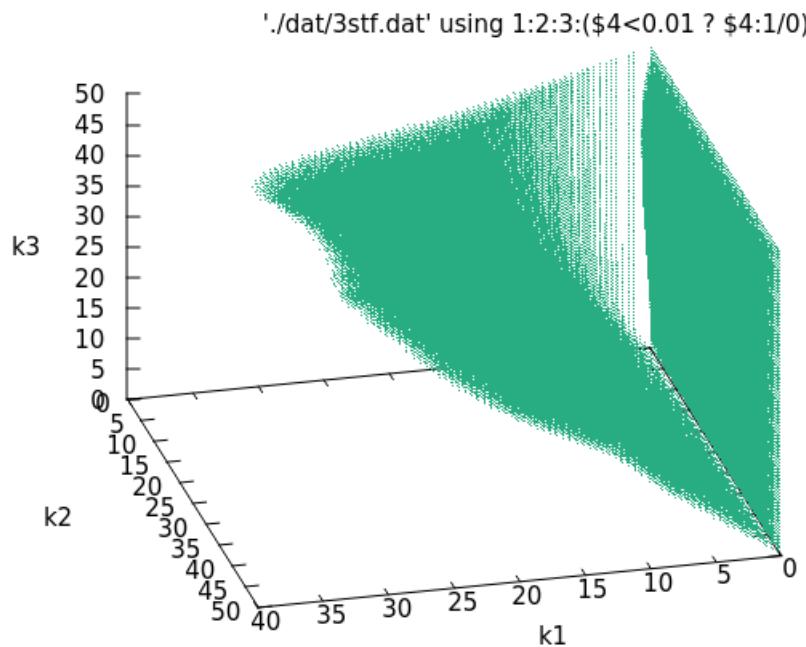


Рис. 5.10—Области $\tilde{\Delta E} < 0.01$ для переменных жесткостей

На основе рисунков ?? и ?? можно сделать вывод, что в предельных случаях система трех тел сводится к системе двух тел. Значение диссипации

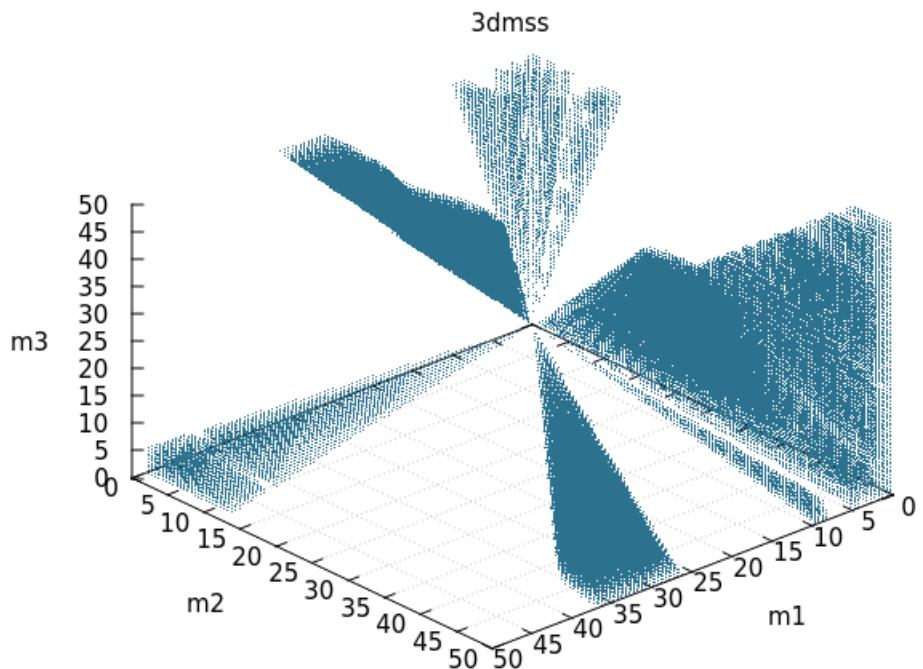


Рис. 5.11 – Области $\Delta \tilde{E} < 0.01$ для переменных масс

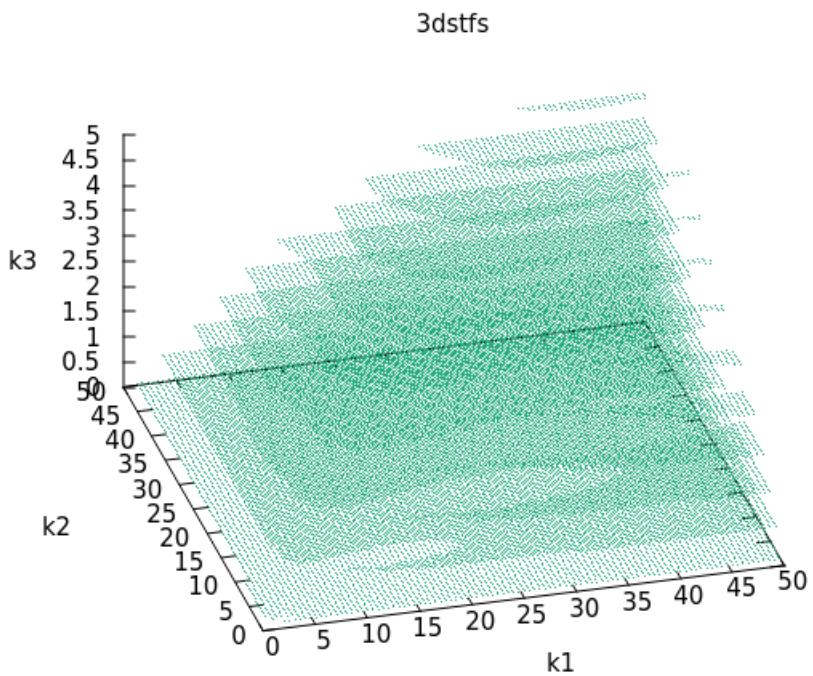


Рис. 5.12 – Области максимальных значений $\Delta \tilde{E}$ для переменных жесткостей

будет зависеть от того каким образом система из трех тел преобразуется в систему двух тел при взаимодействии.

Для максимума накопленной энергии система преобразуется в систему двух тел, в которой одно тело представляет из себя два тела, которые совершают гармонические колебания относительно общего центра масс. Данный центр масс в свою очередь будет колебаться относительно оставшейся массы. Такая конфигурация позволяет преобразовать наибольшее количество кинетической энергии тела на макроуровне во внутреннюю энергию тела.

В случае минимума наколенной энергии система стремится к единому абсолютно упругому телу, такие конфигурации сводятся к случаям, когда одна масса является максимальной и вклад от других масс не является значительным. Так же есть случаи когда две массы совершают гармонические колебания, но третья масса является столь незначительной, либо засчет жесткости пружин не может внести достаточного вклада в данную систему.

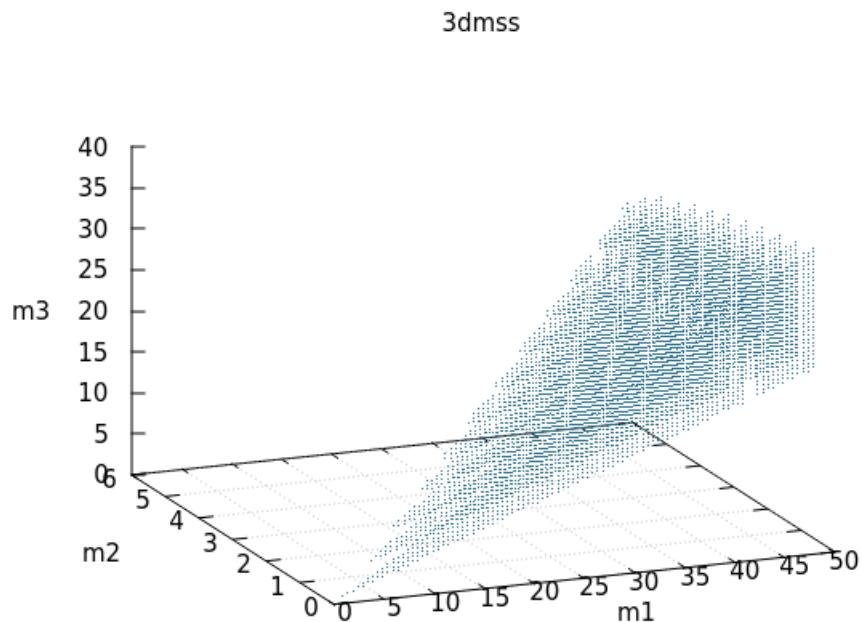


Рис. 5.13 — Области максимальных значений $\Delta\tilde{E}$ для переменных масс

ЗАКЛЮЧЕНИЕ

В ходе работы были выполнены следующие задачи:

- По теме "Силовые сети в гранулированных средах" был проведен обзор литературы и было обнаружено отсутствие подходящих исследований, описывающих проблематику проекта
- Определена актуальность проекта и определена цель исследования в соответствии с актуальными вопросами по теме проекта
- Составлена математическая модель неабсолютно упругого соударения системы трех тел со стенкой.
- Построен и реализован конечный автомат, описывающий состояния системы.
- Проведен анализ методов решения и выбор оптимального.
- Реализовано аналитическое решение системы ОДУ
- Реализован метод Рунге-Кутта для решения ОДУ
- Произведен анализ эффективности методов и скорости расчетов.
- Проведен анализ зависимости коэффициента эффективности диссипации от значений переменных системы.
- Были построены двумерные и трехмерные облака точек, показывающие зависимости коэффициента диссипации от параметров системы
- Определены принципы поведения системы трех тел в областях максимумов и минимумов энергий.

Данные, полученные в ходе работы над курсовым проектом, являются стартовой точкой для понимания процессов, происходящих в неабсолютно упругих телах при соударении. Данная математическая модель и понимание какие из параметров сильней всего влияют на состояние системы, позволяют в дальнейшем найти такие значения параметров системы для которых на-

копленная внутренняя энергия будет максимальна, что позволит определить оптимальные параметры для полидисперсных демпфирующих систем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Lawyer G. Understanding the influence of all nodes in a network // Scientific reports. — 2015. — 03. — Vol. 5. — P. 8665.
2. Darabi B. Viscoelastic behaviour of element of particle damper: Experimental and numerical study // Iranian Journal of Science and Technology, Transactions of Mechanical Engineering. — 2019. — 01. — P. 1–11.
3. Emri I., Prodan T. A measuring system for bulk and shear characterization of polymers // Experimental Mechanics. — 2006. — 08. — Vol. 46. — P. 429–439.
4. Viscoelasticity of new generation thermoplastic polyurethane vibration isolators / Marko Bek, Joris Betjes, Bernd-Steffen von Bernstorff, Igor Emri // Physics of Fluids. — 2017. — Vol. 29, no. 12. — P. 121614. — <https://doi.org/10.1063/1.5000413>.
5. Energy dissipation in granular materials in triaxial tests / Rector Mukwiri, Yousef Motlagh, William Coombs, Charles Augarde. — 2017. — 04.
6. Sanchez M., Carlevaro M. Nonlinear dynamic analysis of an optimal particle damper // Journal of Sound and Vibration. — 2011. — 10. — Vol. 332.
7. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. — Москва : Наука, 1987. — С. 600.
8. Go Tour. — <https://tour.golang.org/welcome/1>.
9. Mark Summerfield. Programming in Go: Creating Applications for the 21st Century. — 2012.
10. GoNum. — <https://www.gonum.org/>.
11. GoDoc gin-gonic. — <https://godoc.org/github.com/gin-gonic/gin>.