**Глава 4. Функциональное тестирование программного комплекса**

**4.1 Введение в цели тестирования**

Функциональное тестирование предназначено для проверки соответствия работы разработанного программного обеспечения заданным требованиям и спецификациям[gb.ru](https://gb.ru/blog/funktsionalnoe-testirovanie-po/#:~:text=%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5%20%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%20%D1%81%D0%BE%D0%B1%D0%BE%D0%B9,%D0%BA%D0%BE%D0%BD%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D1%8B%D1%85%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F%D1%85%20%D1%80%D0%B5%D1%88%D0%B0%D1%82%D1%8C%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B5%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8). В данной главе рассматриваются методы подтверждения корректности вычислителя, обеспечивающие правильную и воспроизводимую работу модели в типичных условиях. Основными целями тестирования являются проверка правильности интерпретации входных данных, адекватности математической постановки задачи и воспроизводимости результатов при повторных запусках.

**4.2 Методика тестирования**

Тестирование проводилось разработчиком с учётом внутренней структуры системы (белый ящик)[gb.ru](https://gb.ru/blog/funktsionalnoe-testirovanie-po/#:~:text=%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%C2%AB%D1%87%D0%B5%D1%80%D0%BD%D1%8B%D0%B9%20%D1%8F%D1%89%D0%B8%D0%BA%C2%BB%20%D0%B1%D0%B5%D1%80%D0%B5%D1%82%20%D0%B7%D0%B0,%D0%BA%D0%B5%D0%B9%D1%81%D1%8B%20%D0%BD%D0%B0%20%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%B5%20%D0%BA%D0%BE%D0%B4%D0%B0%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B). Использовалась следующая методика:

* **Аналитическое сравнение:** результаты численного решения проверялись на корректность посредством сравнения с аналитическими решениями или результатами авторитетных источников и других средств моделирования.
* **Повторные запуски:** моделирование повторялось при одинаковых параметрах, чтобы убедиться в детерминированности и стабильности результатов.
* **Граничные и частные случаи:** исследовались тривиальные сценарии (например, отсутствие потоков в системе или отсутствие начального импульса), проверяющие адекватность работы системы в простейших условиях.
* **Проверка этапов решения:** контролировалась последовательность событий (переходы режимов, столкновения и т. д.) и правильность их обработки согласно логике модели.

Таким образом обеспечивалось тщательное функциональное тестирование ключевых компонентов системы.

**4.3 Задача о двух баках**

Данная модель описывает два соединённых резервуара с перетеканием жидкости между ними и возможным сбросом из первого бака. Гидродинамические процессы в баковой системе задаются системой обыкновенных дифференциальных уравнений с учётом закона Торричелли при расчёте расхода через отверстия. Цель теста – проверить, что вычислитель правильно решает эти уравнения и воспроизводит физическое поведение системы.

Рисунок 4.1 – Схема модели двух баков.

В ходе тестирования задавались начальные уровни жидкости в баках и параметры потоков. После запуска моделирования фиксировались временные зависимости уровней жидкости. Наблюдался процесс перетекания: уровень первого бака постепенно снижался, передавая жидкость во второй, и система стремилась к установившемуся состоянию. Полученные зависимости уровней сравнивались с аналитическими оценками или ожидаемым поведением. Расхождения между численным решением и эталонными данными оставались в пределах допустимой погрешности, что подтверждает корректность вычислителя. При повторных запусках с теми же параметрами наблюдалось полное совпадение результатов, указывающее на детерминированность реализации.

Результаты теста подтвердили, что вычислитель корректно моделирует процесс двух баков: уровни жидкости меняются согласно физическим законам, а итоговые значения соответствуют ожидаемым решениям.

**4.4 Модель прыгающего мячика**

Эта модель объединяет непрерывное движение тела под действием гравитации и дискретные события – столкновение с поверхностью и последующий отскок. Пусть $h(t)$ и $v(t)$ – высота и скорость мячика. В отсутствие столкновения движение описывается уравнениями  
h˙=v,v˙=−g,\dot h = v,\quad \dot v = -g,h˙=v,v˙=−g,  
а при достижении $h = 0$ скорость мгновенно изменяет знак (с учётом коэффициента упругости $c<1$) по правилу $v \to -c,v$[openmodelica.org](https://openmodelica.org/doc/ModelicaLongCourse/Lecture09_Hybrid_Systems.pdf#:~:text=der%28height%29%20%3D%20velocity%3B%20der%28velocity%29%3D,velocity%29%3B%20end%20when%3B%20end%20BouncingBall). Такая гибридная постановка позволяет смоделировать серию последовательных падений и отскоков шарика, постепенно затухающих во времени.

Рисунок 4.2 – Схема модели прыгающего мячика.

Проведённые тесты включали моделирование движения мячика с различными начальными условиями. Как ожидается, амплитуда подъёма между ударами убывает геометрически. Графики положения и скорости демонстрировали последовательное уменьшение максимальных высот и времени полёта, что совпадало с теоретической зависимостью. Время свободного падения между столкновениями соответствовало расчетным значениям, а в момент каждого удара отмечалось мгновенное изменение скорости в соответствии с законом отскока.

Интерпретация результатов показывает полное совпадение динамики системы с ожидаемым поведением при заданном коэффициенте упругости. Ошибок в вычислениях не обнаружено, что подтверждает корректную обработку как непрерывных, так и дискретных этапов в модели.

**4.5 Подведение итогов**

Выполненное функциональное тестирование подтвердило, что разработанный программный комплекс корректно интерпретирует входные данные, осуществляет расчёты в соответствии с математической постановкой и выдаёт воспроизводимые результаты. В качестве тестовых сценариев использованы эталонные модели («двух баков» и «прыгающего мячика»), результаты по которым полностью совпали с ожидаемыми решениями. Таким образом, можно сделать вывод, что программная система в своём текущем виде соответствует заявленным функциональным требованиям[gb.ru](https://gb.ru/blog/funktsionalnoe-testirovanie-po/#:~:text=%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5%20%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%20%D1%81%D0%BE%D0%B1%D0%BE%D0%B9,%D0%BA%D0%BE%D0%BD%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D1%8B%D1%85%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F%D1%85%20%D1%80%D0%B5%D1%88%D0%B0%D1%82%D1%8C%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B5%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8).

**Литература:** Приведённые примеры тестовых моделей основаны на классических постановках задач (например, см. [OpenModelica BouncingBall][8] для модели прыгающего мяча) и на методиках функционального тестирования ПО[gb.ru](https://gb.ru/blog/funktsionalnoe-testirovanie-po/#:~:text=%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5%20%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%20%D1%81%D0%BE%D0%B1%D0%BE%D0%B9,%D0%BA%D0%BE%D0%BD%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D1%8B%D1%85%20%D1%83%D1%81%D0%BB%D0%BE%D0%B2%D0%B8%D1%8F%D1%85%20%D1%80%D0%B5%D1%88%D0%B0%D1%82%D1%8C%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B5%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8)[gb.ru](https://gb.ru/blog/funktsionalnoe-testirovanie-po/#:~:text=%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%C2%AB%D1%87%D0%B5%D1%80%D0%BD%D1%8B%D0%B9%20%D1%8F%D1%89%D0%B8%D0%BA%C2%BB%20%D0%B1%D0%B5%D1%80%D0%B5%D1%82%20%D0%B7%D0%B0,%D0%BA%D0%B5%D0%B9%D1%81%D1%8B%20%D0%BD%D0%B0%20%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%B5%20%D0%BA%D0%BE%D0%B4%D0%B0%20%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B), использованных в качестве отправной точки для верификации разработанного вычислителя.