

Санкт-Петербургский государственный университет

Кафедра системного программирования

Программная инженерия

Привалихин Алексей Александрович

Создание системы тестирования функций пакета MATLAB и ее интеграция в проект RefleXion X1

Отчет по учебной практике

Научный руководитель:
ст. пр. кафедры
системного программирования
Я. А. Кириленко
Консультант:
глав. инж. RefleXion Medical
Е. С. Вороненко, PhD

Санкт-Петербург
2020

Содержание

Введение	2
1 Постановка задачи	3
2 Обзор	4
3 Реализация	6
4 Заключение	10

Введение

Компания RefleXion Medical — одна из немногих на рынке аппаратов лучевой терапии. Ее основным продуктом является RefleXion X1 — аппарат, который кроме традиционных для современной радиотерапии методов, таких как IMRT [1] и SBRT [2], поддерживает также находящийся в стадии клинических испытаний метод VgRT [3].

Линейный акселератор аппарата RefleXion X1 расположен на непрерывно вращающемся вокруг продольной оси кушетки кольце. На протяжении одного оборота линейный акселератор проходит 50 позиций (firing position), в каждой из которых может быть выпущено не более 4 импульсов. Форма луча контролируется многолепестковым коллиматором (MLC), имеющим 64 лепестка. Также кушетка может перемещаться вдоль продольной оси, занимая фиксированные положения с интервалом 2.1мм (beam stations). Элементарным лучом (beamlet) называется часть луча, проходящая через конкретный лепесток MLC в конкретной позиции кольца и кушетки.

Одной из важных частей программного обеспечения аппарата RefleXion X1 является планировщик лечения. В режиме IMRT/SBRT, получив на вход заданный врачом набор предписаний и ограничений по дозе для каждого органа и целевой области, планировщик задает оптимальную дозу для каждого элементарного луча. В режиме VgRT планировщик строит "матрицу стрельбы" (firing matrix). Также планировщик рассчитывает такие параметры как предполагаемое распределение дозы в теле пациента и продолжительность лечения.

Планировщик лечения в проекте RefleXion X1 реализован с использованием вычислительного пакета MATLAB [4] версии R2017b.

Учитывая область применения, критически важно, чтобы программное обеспечение RefleXion X1, в том числе планировщик, функционировало исправно. Исходя из имеющейся в распоряжении RefleXion информации (подтвержденной экспериментальными данными и службой поддержки разработчика MATLAB, компании MathWorks) о существующих в версиях MatLab R2019 и MatLab R2020 неполадках в функции `convn`, а также учитывая планы в будущем перевести проект RefleXion X1 на более новую версию пакета MATLAB, было решено создать систему тестирования, проверяющую корректность работы используемых в проекте функций пакета MATLAB.

Поскольку в проекте RefleXion X1 используется библиотека Armadillo [5] для языка C++, предоставляющая набор функций, аналогичный представляемому пакетом MatLab, в качестве метода тестирования был выбран запуск аналогичных функций пакета MatLab и библиотеки Armadillo и сравнение полученных результатов. Такой подход позволяет также достичь вторичной цели — удостовериться корректную работу функций библиотеки Armadillo.

1 Постановка задачи

В рамках проекта по созданию системы тестирования, проверяющей корректность работы используемых в проекте функций пакета MATLAB были поставлены следующие задачи.

- Необходимо создать интерфейс для вызова нужных функций библиотеки Armadillo из среды MatLab.
- Необходимо создать для каждой из исследуемых функций набор тестовых сценариев, реализовать тестовый метод для проверки каждого сценария.
- Необходимо создать пакет тестов, удостоверяющий корректную работу используемых в проекте Reflexion X1 функций пакета MatLab с помощью библиотеки Armadillo.
- Необходимо интегрировать созданный пакет в существующий пакет тестов планировщика лечения проекта Reflexion X1.

2 Обзор

Пакет MatLab Созданный компанией MathWorks вычислительный пакет MatLab предоставляет широкий набор функций для произведения вычислений, относящихся к различным областям математики, а также для представления результатов вычислений как в текстовом виде, так и в виде графиков.

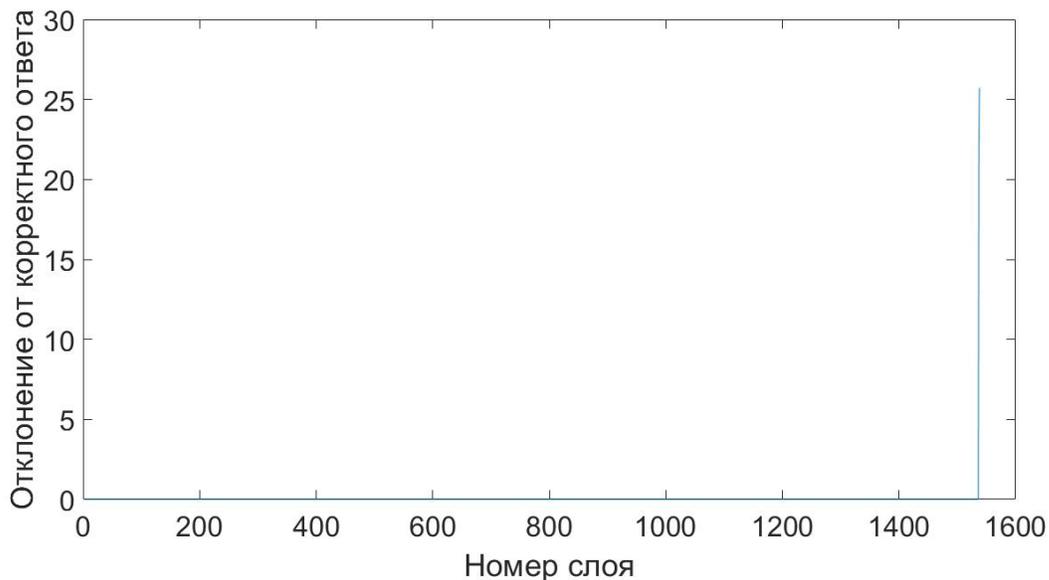
В текущей версии проекта RefleXion X1 используется MATLAB R2017b.

МEX функции Пакет MATLAB предоставляет возможность компилировать и связывать файлы с исходным кодом на языках C и C++ в исполняемый файл (MEX — MATLAB Executable), который можно вызывать как функцию из среды MATLAB.

Известные ошибки в пакете MATLAB Одна из функций, предоставляемых пакетом MATLAB — функция многомерной свертки `convn`. Было замечено, что в версиях R2019 и R2020 при некоторых размерах входных матриц функция работает некорректно, выдавая существенную ошибку на последних слоях трехмерной матрицы ответа.

Был произведен запрос в службу поддержки компании MathWorks, и она подтвердила наличие данной проблемы.

На данном графике продемонстрирован модуль разницы между послойными суммами корректного ответа и реального вывода функции `convn` при размерах входных матриц $7 \times 7 \times 1538$ и 7×7 . Первая матрица заполнена случайными числами из равномерного распределения на интервале $(0, 1)$, вторая заполнена нулями, кроме первого элемента первой строки, которому присвоено значение 1.



Библиотека Armadillo Библиотека Armadillo для языка C++ предоставляет набор вычислительных функций, аналогичных функциям пакета MatLab.

В текущей версии проекта RefleXion X1 используется Armadillo 7.

Пакет MATLAB.unittest MATLAB.unittest [6] — это предоставляемый компанией MathWorks вместе с основным пакетом MATLAB пакет для создания основанных на классах модульных тестов.

Поскольку существующий пакет тестов планировщика лечения проекта RefleXion X1 реализован с использованием пакета MATLAB.unittest, тот же пакет был выбран для создания данной системы тестирования.

Способы вызова функций библиотек C++ из MATLAB Существует несколько способов вызова функций библиотек C++ из MATLAB:

- Пакеты `loadlibrary` [7] и `callib` [8] позволяют загрузить библиотеку на языке C++ и использовать ее функции из среды MATLAB, однако все написанные на C++ функции библиотеки должны придерживаться C calling convention, а Armadillo не удовлетворяет этому условию
- Пакет `slib` [9] позволяет загрузить библиотеку на языке C++ и использовать ее функции из среды MATLAB, однако для этого во-первых необходим специальный интерфейс, а во-вторых `slib` не поддерживает многие особенности C++, в том числе классы, что делает данный пакет непригодным для вызова функций Armadillo
- Использование MEX C API [10] в сочетании с MATLAB C Matrix API [11] для создания исполняемого файла MATLAB из исходного кода на C/C++
- Использование MEX C++ API [12] в сочетании с MATLAB Data API [13] для создания исполняемого файла MATLAB из исходного кода на C/C++ (доступен начиная с MATLAB R2018a)

3 Реализация

Создание интерфейса для вызова функций библиотеки Armadillo из среды MATLAB С учетом того что версии MATLAB и Armadillo, используемые в проекте на данный момент, уже устарели, и в ближайшем будущем предполагается переход на более новую версию обоих продуктов, было создано две версии интерфейса для вызова функций библиотеки Armadillo из среды MATLAB.

Поскольку в используемой на данный момент в проекте Reflexion X1 версии MATLAB (R2017b) отсутствует mex C++ API, в первой версии интерфейса используется mex C API и связанный с ним C Matrix API. Также в данной версии отсутствует проверка корректности функции `interp2`, поскольку этой функции нет в Armadillo 7.

Вторая версия интерфейса использует mex C API и MATLAB Data API, а также Armadillo 9.900, что позволяет включить функцию `interp2` в список исследуемых.

Сборка исполняемого мех-файла из соответствующего исходного файла добавлена в скрипт непрерывной интеграции.

Создание набора тестовых сценариев Для каждой из исследуемых функций создан класс-наследник класса `TestCase`, в котором для каждого из рассматриваемых сценариев реализован тестовый метод. На данный момент в список входят базовые сценарии — работа с плотными/разреженными матрицами, массивы различного размера, различные параметры конфигурации, такие как метод интерполяции и метод вычисления нормы.

Размеры массивов по каждому из измерений выбирались из набора списков: малые (10, 30, 50, 70, 90), средние (500, 1500, 2500, 3500, 4500) и большие (10000). Однако, по соображениям экономии оплачиваемого времени работы тестового сервера, а также учитывая конкретные случаи использования тех или иных функций в проекте Reflexion X1, не для всех функций были использованы все перечисленные размеры. В приведенной ниже таблице исследуемых функций буквой S обозначен список малых размеров, M — средних, L — больших. В случаях, когда были использованы не все значения из списка, приведены индексы использованных элементов.

Таблица 1: Список исследуемых функций

Функция	S	M	L
min	+	+	+
max	+	+	+
abs	+	+	+
clamp	+	+	+
conv	+	+	+
conv2	+	1	-
convn	+	-	-
exp	+	+	+
pow	+	1, 2, 3, 4	-
repmat	+	-	-
transpose	+	+	+
approx_equal	+	+	+
norm	+	+	+
sum (плотные матрицы)	+	1	-
sum (разреженные матрицы)	+	+	+
find	+	+	+
операторы отношений	+	+	+
interp1	+	+	+
interp2	+	+	-
fft	+	+	+
imag	+	+	+
real	+	+	+
умножение матриц	+	1, 2, 3	-
остальные арифметические операторы	+	+	+
операторы создания подмассивов (2D)	+	+	+
операторы создания подмассивов (3D)	+	1	-

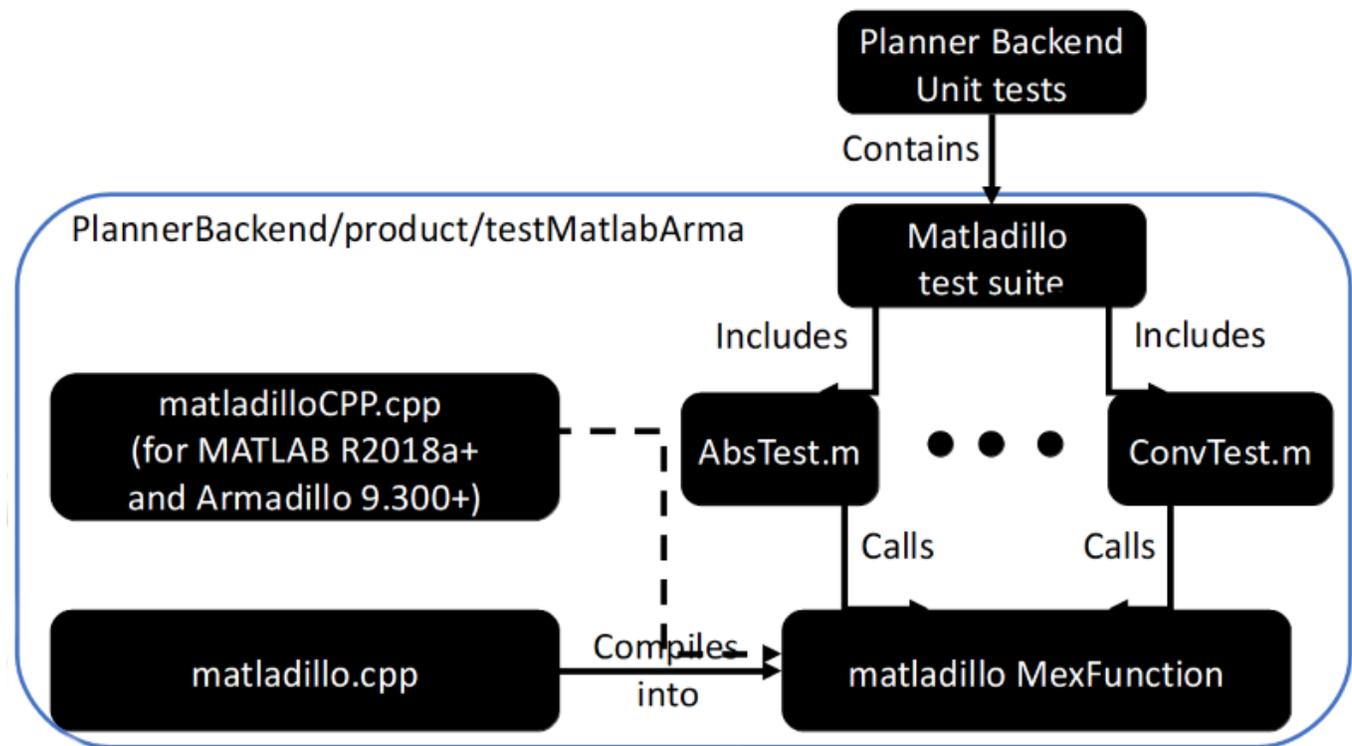
Функция `convn` поддерживает только трехмерные массивы. Поскольку данная функция отсутствует в библиотеке Armadillo, она была реализована согласно определению трехмерной свертки с помощью функции `conv2`. Также для функции `convn` включены сценарии, на которых эта функция некорректно работает в MATLAB R2019+: свертка трехмерного массива с малыми первым и вторым измерениями, но большим третьим с двумерной матрицей с обоими малыми измерениями.

Функция `interp2` доступна только в версии интерфейса для MATLAB R2018a+ и Armadillo 9.300+.

Создание пакета тестов и его интеграция в проект Создан экземпляр класса `TestSuite` (массив `TestCase`), содержащий в себе по одному экземпляру каждого из созданных в прошлом пункте классов. Данный массив включен в экземпляр `TestSuite`, отвечающий за все модульные тесты планировщика лечения.

Таким образом все созданные тесты автоматически исполняются при выполнении модульных тестов планировщика лечения в ходе непрерывной интеграции.

Архитектура системы



Интеграция созданной системы в проект Reflexion X1 В рамках политики branch-per-feature в репозитории проекта была создана отдельная ветка, в которой были произведены необходимые изменения, после чего был произведен pull request в базовую ветку. Pull request прошел все необходимые проверки, включая статический анализ, тестирование и рецензию уполномоченными сотрудниками, после чего изменения были внесены в базовую ветку проекта. Основная задача считается успешно завершенной, однако в дальнейшем возможно расширение списка тестируемых функций и/или списка тестовых сценариев в рамках усилий по максимально полному покрытию проекта тестами.

4 Заключение

В ходе данной работы достигнуты следующие результаты.

- Были созданы две версии интерфейса для вызова функций библиотеки Armadillo из среды MATLAB.
- Был создан набор тестовых сценариев для каждой из исследуемых функций.
- Тестовые сценарии были собраны в пакет, который в свою очередь был включен в пакет тестов планировщика лечения проекта RefleXion X1.
- Был создан pull request в базовую ветку репозитория проекта.
- Pull request прошел все необходимые проверки, как ручные, так и автоматические, и был принят.
- Была подтверждена корректность работы исследуемых функций в используемых в проекте версиях пакета MATLAB и библиотеки Armadillo.

В рамках усилий по максимально полному покрытию проекта тестами, набор тестовых сценариев, а также сам набор исследуемых функций может в дальнейшем расширяться.

Список литературы

- [1] Описание технологии IMRT:
<https://www.radiologyinfo.org/en/info.cfm?pg=imrt>
- [2] Описание технологии SBRT:
<https://www.mskcc.org/cancer-care/diagnosis-treatment/cancer-treatments/radiation-therapy>
- [3] Описание технологии VgRT:
<https://reflexion.com/our-technology/>
- [4] Веб-страница пакета MATLAB:
<https://www.mathworks.com/products/matlab.html>
- [5] Документация библиотеки Armadillo:
<http://arma.sourceforge.net/docs.html>
- [6] Документация пакета Matlab.unittest:
<https://www.mathworks.com/help/matlab/class-based-unit-tests.html>
- [7] Документация пакета loadlibrary:
<https://www.mathworks.com/help/matlab/ref/loadlibrary.html>
- [8] Документация пакета calllib:
<https://www.mathworks.com/help/matlab/ref/calllib.html>
- [9] Документация пакета clib:
<https://www.mathworks.com/help/matlab/use-prebuilt-matlab-interface-to-c-library.html>
- [10] Документация MEX C API:
<https://www.mathworks.com/help/matlab/call-mex-files-1.html>
- [11] Документация C Matrix API:
<https://www.mathworks.com/help/matlab/cc-mx-matrix-library.html>
- [12] Документация MEX C++ API:
https://www.mathworks.com/help/matlab/matlab_external/cpp-mex-api.html
- [13] документация MATLAB Data API:
<https://www.mathworks.com/help/matlab/matlab-data-array.html>