

ВОПРОС РАСПАРАЛЛЕЛИВАНИЯ В РАЗРАБОТКЕ ЯДРА ГЕОМЕТРИЧЕСКОГО МОДЕЛИРОВАНИЯ

О.В. Шаповалов, Е.С. Сергеев, В.С. Чалышев, А.В. Катаев, Д.И. Крыжановский, А.Е. Андреев

Введение

Настоящая разработка является частью крупномасштабного проекта по созданию отечественного геометрического ядра. Проект выполняется под эгидой Московского государственного технологического университета «СТАНКИН» совместно с ведущими САПР компаниями и ВУЗами России. В проекте участвуют профессиональные команды программистов из различных городов России и стран СНГ, координация проекта возложена на компанию Топ Системы. Наша команда, представленная сотрудниками Волгоградского государственного технического университета и компании Сингулярис Лаб, отвечает за одно из направлений разработки.

В перспективе разработанное в рамках проекта 3D-ядро (геометрическое ядро) должно стать основой для целого ряда современных и конкурентоспособных приложений, обеспечивающих решение задач конструкторско-технологической подготовки производства в станкостроении и промышленности в целом. Данная разработка будет выступать конкурентом лидерам рынка: Parasolid (компания Siemens PLM Software) и ACIS (выпускаемый Spatial Corp. — дочерней компании Dassault Systèmes).

Геометрическое ядро (пакет геометрического моделирования) – это набор библиотек с программным интерфейсом (API), с помощью которого можно пользоваться функциями геометрического (например, твердотельного) моделирования. Типичной функциональностью пакета геометрического моделирования является предоставление набора программных интерфейсов (структур данных, функций и классов) для создания приложения каркасного, поверхностного, твердотельного или немногообразного моделирования.

В ядре RGK для описания модели использован подход к описанию модели методами граничного представления (Boundary Representation, B-Rep). Под телом в статье будет подразумеваться граф, вершинами которого являются грани, ребра и вершины разбиения границы тела, при этом с топологическими сущностями связано их геометрическое представление (грань – поверхность, ребро – кривая, вершина – точка). В целях оптимизации скорости работы функций ядра, а также для обеспечения точности хранения и расчётов модели используются как общие типы кривых и поверхностей (плоскости, прямые, кривые и поверхности второго порядка и т.д.), так и NURBS-кривые и NURBS-поверхности. NURBS – Non-uniform rational B-spline – неоднородный рациональный B-сплайн, широко распространенный вид сплайнов, используемый для генерации и представления кривых и поверхностей.

Расчет сложных моделей может занимать существенное время, и в статье будут рассмотрены вопросы оптимизации и распараллеливания вычислений применительно к нескольким задачам, которые решала наша команда. Оптимизации подверглись методы и алгоритмы, предназначенные для построения тел по сечениям, по сетке направляющих кривых, а также методы деформации модели. Эти методы и алгоритмы применяются в подсистеме управления моделью и топологической подсистеме ядра 3-мерного моделирования.

По причине того, что работа над проектом (и над его оптимизацией) продолжается, оценка скорости работы алгоритмов носит приблизительный характер, но уже сейчас можно видеть конкурентоспособность скорости работы наших алгоритмов по сравнению с аналогичными в ядре Parasolid.

1. Построение тела по траектории

Операция построения тела по траектории позволяет получать поверхности путем перемещения контура вдоль пространственной траектории. Движение контура осуществляется за счет его многократного копирования вдоль выбранной траектории.

Опишем более детально алгоритм работы генератора:

1. Анализ контура и пространственной траектории.
2. Вычисление множества систем координат минимального вращения, движущихся из начала в конец траектории, если траектории периодическая или ломаная, требуются дополнительные вычисления.
3. Трансформация множества систем координат минимального вращения с учетом законов масштабирования и вращения.
4. Построение контуров в начальном положении для каждого ребра траектории.
5. Построение поверхностей для граней, если результат движения контура не распознается как аналитическая поверхность, тогда поверхность строится путем интерполяции контуров, полученных в результате движения по текущему ребру траектории.
6. Обработка изломов в случае необходимости.
7. Сборка топологии результирующего тела.

Этап 1 не содержит трудоемких вычислений и сводится к анализу G1 гладкости траектории и расширению кривых в ребрах в случае излома [4].

Этапы 2-3. На основе анализа длины вектора второй производной на разных участках кривых траектории с учетом законов вращения и масштабирования вычисляется достаточное количество точек для построения движущейся системы координат минимального вращения [2,5]. Построение систем координат – последовательная процедура, которая может быть оптимизирована, но не может быть распараллелена.

Этап 4. На данном этапе кривые из начального положения перемещаются в начало каждого ребра траектории, для того, чтобы построение поверхностей можно было выполнить параллельно.

Этап 5. В общем случае построения тела по траектории значительную часть времени может занимать построение поверхностей. Вычислительная сложность этапа напрямую зависит от количества точек контрольного полигона кривых в основе рёбер контура и кривизны кривых в основе рёбер траектории. Значительное влияние на размерность контрольного полигона результирующих поверхностей оказывают законы вращения и масштабирования. Следовательно, для поиска “узких мест” в работе генератора нужно выполнить профилирование на тяжелых кривых с большими контрольными полигонами с заданными законами масштабирования и вращения.

Этап 6. Построение геометрии на изломах траектории. Выполняется большое количество таких сложных операций, как пересечение плоскостей, пересечение кривых, построении поверхности по сечениям с G1 условием на границе.

Этап 7. Построение топологии - последовательная процедура сборки результирующего тела.

После проведения оптимизаций основную часть метода построения тела занимает построение поверхностей для граней. Поверхности для граней строятся путем интерполяции через копии контура. Для каждого ребра контура и ребра траектории строится отдельная поверхность.

Для оптимизации процедуры построения поверхностей можно оптимизировать процедуру интерполяции и распараллелить построение поверхностей для граней. Для параллельного построения поверхностей следует подготовить все необходимые для построения поверхности данные: множество систем координат вдоль пространственной траектории, задающих закон движения и трансформацию контура, также требуется первый контур в начале каждого ребра траектории. В результате была выбрана схема распараллеливания по ребрам траектории, и построение боковых поверхностей полученных в результате движения контура по ребрам траектории осуществляется в параллельных потоках.

Для оценки эффективности распараллеливания мы создали стресс-тесты, с использованием сложных контуров и траекторий. Сложность контура определяется, прежде всего, количеством рёбер и сложностью кривых в ребрах, сложность кривых в свою очередь зависит от количества точек контрольного полигона и степени кривой. Результаты тестирования приведены в таблицах 2 и 3.

Таблица 1. Характеристики тестовых моделей

Номер модели	Количество рёбер в контуре	Количество точек контрольного полигона в кривой ребра	Количество рёбер в траектории
1	36	200	8
2	72	200	8
3	108	200	8
4	36	200	80
5	72	200	80
6	108	200	80

Таблица 2. Время построения тел на AMD FX8120, 8 ядер

Модель	Последовательная версия		Параллельная версия		Ускорение	Ускорение построения поверхностей
	Время, с	Время построения поверхностей, с	Время, с	Время построения поверхностей, с		
1	0,87	0,82	0,32	0,27	2,71	2,992
2	1,75	1,65	0,577	0,5	3,03	3,3
3	3,42	3,3	1,1	0,951	3,10	3,47
4	3	3,1	0,9	0,7	3,33	4,42
5	6,46	6,1	1,63	1,26	3,96	4,84

6	12,75	12,02	3,17	2,45	4,02	4,9
---	-------	-------	------	------	------	-----

Таблица 3. Время построения тел на Intel Core i5 2310

Модель	Последовательная версия		Параллельная версия		Ускорение	Ускорение построения поверхностей
	Время, с	Время построения поверхностей, с	Время, с	Время построения поверхностей, с		
1	0,62	0,6	0,22	0,19	2,81	3,15
2	1,21	1,165	0,42	0,37	2,88	3,14
3	2,4	2,31	0,84	0,76	2,85	3,03
4	2,26	2,14	0,71	0,57	3,18	3,75
5	4,52	4,28	1,35	1,12	3,34	3,8
6	9,23	8,78	2,84	2,39	3,25	3,67

Из таблиц 2-3 можно сделать следующие выводы:

1. Чем больше трудоёмкость построения боковых граней, тем большего ускорения от распараллеливания можно ожидать.
2. На небольших моделях эффект от распараллеливания незначительный. Например, для модели №1 время построения менее одной секунды, если использовать модели меньшего размера, время построения уменьшится до десятых долей секунды.



Рис. 1. Движение проволочного контура по замкнутой траектории с заданным направлением выравнивания секционных кривых

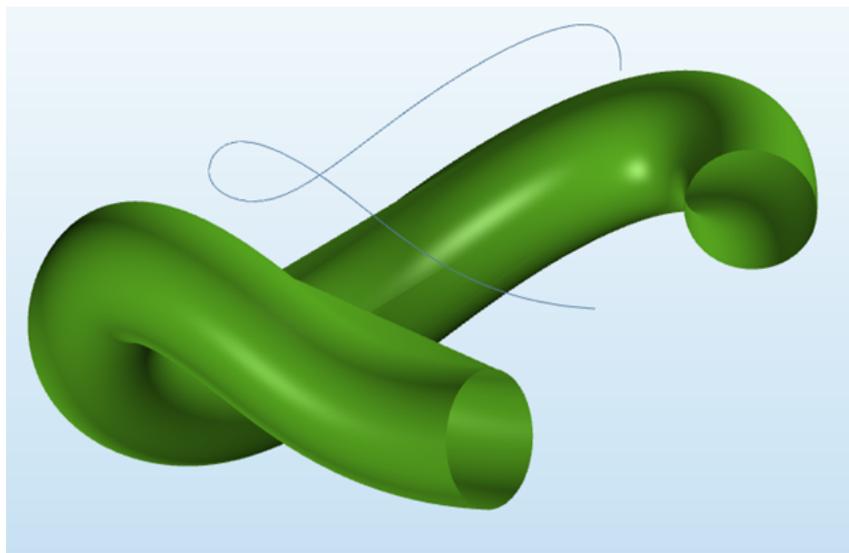


Рис. 2. Движение окружности по гладкой траектории, заданной nurbs-кривой

1. ОПЕРАЦИЯ ДЕФОРМАЦИИ ТЕЛ

В ядре реализуются различные операции деформации, осуществляемые над телами: сгибание, растяжение и скручивание, скульптурная деформация, деформация по кривой и по поверхности.

Деформация тела заключается в деформации вершин, кривых и поверхностей, задающих геометрию тела. Деформация осуществляется путем выбора большого количества точек с кривых и поверхностей, применения деформации к отдельным точкам и последующей интерполяции деформированных кривых и поверхностей.

Деформацию всех кривых для рёбер и поверхностей для граней можно осуществлять независимо. Следовательно, можно распараллелить процедуру деформации рёбер и процедуру деформации граней. Тестирование осуществлялось на трех моделях, количественные характеристики которых приведены в таблице 4, с использованием тестового стенда на базе процессора Intel Core i5 2310.

Таблица 4. Описание тестовых моделей

	Количество граней	Количество рёбер	Количество вершин
Модель №1	484	1121	644
Модель №2	22	44	22
Модель №3	2194	5282	3168

В качестве тестовой операции была выбрана деформация сгибанием, где основное время вычисления состоит из интерполяции и проецирования деформированных кривых рёбер на поверхности граней.

После оптимизации последовательного кода и распараллеливания процедур деформации граней и рёбер, получаются результаты, приведенные в таблице 5.

Таблица 5. Ускорение процедуры деформации тела (сгибание)

	Время выполнения последовательной версии, с	Время выполнения параллельной версии, с	Ускорение
Модель №1	4,06	3,01	1,34
Модель №2	0,017	0,013	1,3
Модель №3	1,07	0,36	3

В ходе профилирования параллельного кода было обнаружено, что вычислительная нагрузка распределяется между потоками неравномерно. Пока часть потоков выполняет вычисления, другая часть потоков простаивает и ожидает завершения загруженных потоков. Устранить дисбаланс нагрузки можно с помощью директив OpenMP [3]. Мы воспользовались оператором `schedule(dynamic)` для динамического распределения нагрузки между потоками. Оператор `schedule` для параллельной конструкции `for` указывает, как поступать с потоками, ожидающими завершения других потоков. В данном случае часть нагрузки будет распределена между потоками, ожидающими на выходе из цикла. Результаты после применения оператора приведены в таблице 6.

Таблица 6. Ускорение процедуры деформации тела (сгибание) после динамического распределения нагрузки на стенде №1

	Время выполнения последовательной версии, с	Время выполнения параллельной версии, с	Ускорение
Модель №1	4,06	1,66	2,45
Модель №2	0,017	0,01	1,7
Модель №3	1,07	0,33	3,24

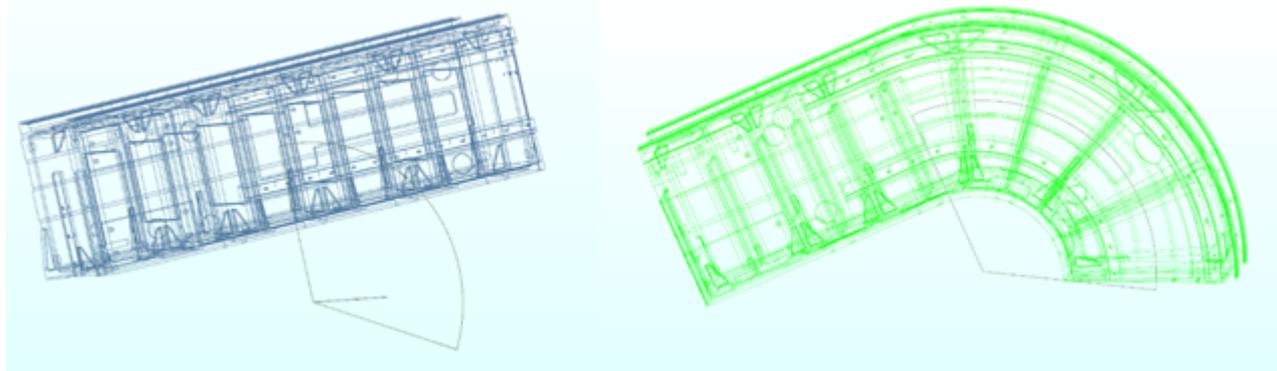


Рис. 3. Деформация гибки стандартной модели Cast из Tflex

1. ОПЕРАЦИЯ ПОСТРОЕНИЯ ТЕЛА ПО НАБОРУ СЕЧЕНИЙ

Операция позволяет строить тело по набору сечений. Для построения тела сечения разбиваются на сектора, для которых независимо строятся поверхности, после чего на основе каждой построенной поверхности в тело добавляются грани.

Общий алгоритм построения тела включает следующие шаги:

1. Автоматическое определение направлений обхода сечений (если направления не заданы явно).
2. Синхронизация сечений в вершинах, для которых явно заданы соответствия.
3. Синхронизация сечений по направляющим.
4. Синхронизация сечений в вершинах, в которых ребра соединяются не гладко.
5. Построение боковых поверхностей тела.
6. Сборка топологии тела.

На 1 шаге, если направления обхода сечений не заданы явно либо через соответствия точек сечений, применяется алгоритм, требующий вычисления длин сечений и отображения длин в параметры кривых. Эти операции являются трудоемкими и могут выполняться независимо для каждого сечения.

На 2 шаге строится первичное разбиение сечений на сектора, которое уточняется на 3 и 4 шагах, тяжелые операции на этом шаге отсутствуют.

3 шаг выполняется, только если заданы направляющие для построения тела. Для каждой направляющей проверяется ее G1 гладкость [4] и находятся точки пересечения со всеми сечениями. Эти операции также являются трудоемкими и могут быть выполнены независимо.

На 4 шаге в каждом сечении находятся точки изломов, в которых ребра сечений не образуют гладкого соединения, после чего происходит разбиение сечений на сектора по точкам излома. Если количество изломов в сечениях различается, то разбиение на сектора в сечениях с меньшим числом изломов будет происходить пропорционально длине секторов с изломами в соседних сечениях. На этом шаге выполняются трудоемкие операции поиска точек изломов в кривых, вычисления длин кривых и отображения длин в параметры.

На 5 шаге для каждого сектора строятся поверхности, интерполирующие кривые в сечениях и направляющих, после чего строится топология боковых граней тела. Если заданы ограничения на производные, то по ним строятся кривые, описывающие законы изменения производных. Построение поверхностей является трудоемкой операцией и может быть выполнено независимо для каждого сектора. Построение законов изменения производных, задаваемых гранями, также является трудоемкой операцией. При включении опции построения аналитических поверхностей также запускаются методы распознавания, которые в некоторых случаях могут требовать значительных вычислительных затрат.

На 6 шаге выполняется завершение топологии тела: для твердых тел добавляются замыкающие торцевые грани и проверяется корректность ориентации нормалей тела. Для проверки ориентации применяется метод, основанный на пересечении луча с каждой гранью тела, который требует значительных вычислений и может быть оптимизирован.

В генераторе по сечениям были распараллелены следующие методы:

На шаге 1: Набор равномерно распределенных точек сечений (параллельно для каждого сечения).

На шаге 3: Проверка непрерывности направляющих (параллельно для каждой направляющей) и нахождение точек пересечения с сечениями (параллельно для каждой пары сечение-направляющая).

На шаге 4: Проверка гладкости сечений и нахождение точек изломов (параллельно для каждого сечения), синхронизация точек изломов по соседним сечениям (параллельно для каждой пары сечение-точка излома).

На шаге 5: Подготовка исходных данных для построения поверхности (параллельно для каждой пары сечение-сектор), построение поверхностей (параллельно для каждого сектора).

На шаге 6: Проверка пересечения луча с гранями тела (параллельно для каждой грани).

Были рассмотрены следующие группы тестов.

Группа 1: выборка из 82 тестов основной функциональности генератора, покрывающих все ветки выполнения.

Группа 2: тесты с направляющими (повышенная нагрузка на 3 и 5 шагах).

Группа 3: тесты с изломами без точек соответствия (повышенная нагрузка на 1 и 4 шагах).

Группа 4: тесты, в которых строятся сплошные тела (повышенная нагрузка на 6 шаге).

Из таблицы 7 видно, что распараллеливание привело к увеличению производительности генератора практически на всех тестах. В то же время, эффективность далека от теоретического предела, что объясняется относительной простотой тестов во всех группах. Малое количество сечений и небольшое число изломов в сечениях приводят к тому, что вычислительные ресурсы оказываются загруженными на параллельных участках не полностью, а общее время работы последовательных участков составляет существенную часть времени работы генератора. В то же время, практически ни на каких тестах не наблюдалось падение производительности, что говорит о том, что даже для простейших тестов накладные расходы на организацию параллельной обработки данных оказываются небольшими по сравнению со временем вычислений. Стоит отметить, что столь незначительное ускорение на 5 шаге в общей группе тестов по сравнению с группами 3 и 4 объясняется большим временем работы на данном шаге в тестах с направляющими. Используемый при этом алгоритм не содержит внутреннего параллелизма, а отдельные задачи построения поверхностей обладают существенно различной вычислительной сложностью.

Таблица 7. Результаты тестирования производительности генератора тел

Группа тестов	Количество тестов в группе X количество запусков	Шаги	Общее время на составление и решение СЛАУ, с		Ускорение
			до модификаций	после модификаций	
Группа 1	82 X 5	1	0,1630 (9,3%)	0,0832 (6,0%)	1,96 (49,0%)
		3	0,0304 (1,7%)	0,0146 (1,1%)	2,08 (52,0%)
		4	0,1378 (7,7%)	0,0777 (5,6%)	1,77 (43,6%)
		5	0,8745 (49,9%)	0,7353 (53,2%)	1,19 (15,9%)
		6	0,4752 (27,1%)	0,3881 (28,1%)	1,22 (18,3%)
		прочие	0,0757 (4,3%)	0,0827 (6,0%)	0,9 (-9,24%)
		общее	1,7537 (100%)	1,3817 (100%)	1,27 (21,2%)
Группа 2	5 X 5	1	0,0035 (1,6%)	0,0017 (0,8%)	2,06 (51,4%)
		3	0,0360 (16,0%)	0,0159 (7,7%)	2,26 (55,8%)
		4	0,0002 (0,1%)	0,0002 (0,1%)	1,0 (0%)
		5	0,1835 (81,3%)	0,1872 (90,3%)	0,98 (-2%)
		6	0,0001 (0,1%)	0,0001 (0,1%)	1,0 (0%)
		прочие	0,0020 (1,3%)	0,0020 (1,3%)	1,0 (0%)
		общее	0,2256 (100%)	0,2073 (100%)	1,08 (8,1%)
Группа 3	10 X 5	1	0,0227 (8,0%)	0,0107 (6,1%)	2,12 (52,9%)
		3	0,0000 (0,0%)	0,0000 (0,0%)	1,0 (0,0%)
		4	0,0330 (11,6%)	0,0200 (11,4%)	1,65 (39,4%)
		5	0,1468 (51,6%)	0,0752 (42,8%)	1,95 (48,8%)

		6	0,0710 (25,0%)	0,0589 (33,5%)	1,20 (17,0%)
		прочие	0,0108 (3,8%)	0,0111 (6,3%)	0,97 (-2,7%)
		общее	0,2843 (100%)	0,1757 (100%)	1,64 (39,0%)
Группа 4	15 X 5	1	0,0164 (4,0%)	0,0100 (3,4%)	1,64 (39,0%)
		3	0,0000 (0,0%)	0,0000 (0,0%)	1,0 (0%)
		4	0,0316 (7,7%)	0,0228 (7,8%)	1,38 (27,8%)
		5	0,1207 (29,2%)	0,0740 (25,4%)	1,63 (38,7%)
		6	0,2244 (54,3%)	0,1643 (56,3%)	1,36 (26,8%)
		прочие	0,0200 (4,9%)	0,0206 (7,1%)	0,97 (-3%)
		общее	0,4131 (100%)	0,2918 (100%)	1,42 (29,4%)

Были также проведены стресс-тесты.

Стресс-тест 1: тестовые данные состоят из 4700 сечений с различным (1-3) числом изломов в сечениях и явным указанием направлений обхода сечений. Основное время работы генератора составляют шаг синхронизация изломов (4 шаг - 43,6%) и построение поверхности (5 шаг 48%). В результате тестирования видно, что на 4 шаге эффективность распараллеливания достаточно близко к теоретическому пределу (в тесте учитывалось общее время 4 шага, включая операции, которые не могут выполняться параллельно). Для 5 шага выделяется 7 последовательных задач, различного объема, что не позволяет судить об эффективности распараллеливания данного шага.

Стресс-тест 2: тестовые данные состоят из 990 сечений без изломов с включенным автоматическим определением направлений обхода сечений. Основное время работы генератора приходится на определение направления сечений (1 шаг — 99,5%). По результатам тестирования видно, что ускорение на данном шаге составило 3,86, что близко к теоретическому пределу.

Стресс-тест 3: тестовые данные состоят из 50 сечений без изломов и 50 направляющих. Основное время работы генератора составляет обработка направляющих (3 шаг — 44,5%) и построение поверхности (5 шаг — 52,2%). Тестирование показывает, что на 3 шаге достигается распараллеливание в 3,9 раза, что достаточно близко к теоретическому пределу. Время работы на 5 шаге осталось практически без изменений, формируется лишь одна задача построения поверхности, и основная работа на данном шаге происходит в методе построения NURBS поверхности по двум семействам кривых, который не обладает внутренним параллелизмом. Так как задачи, выделяемые для параллельной обработки на данном шаге, различаются по вычислительной сложности, показанное увеличение производительности достигнуто при использовании динамического планирования в параллельной реализации.

Таблица 8. Результаты тестирования производительности генератора тел

Тест	Шаги	Время работы последовательной версии (%)	Время работы параллельной версии (%)	Ускорение
Стресс-тест 1: 4700 сечений с изломами	4	0,7315 (43,6%)	0,1940 (29,0%)	3,77 (73,5%)
	5	0,8054 (48,0%)	0,3337 (49,8%)	2,42 (58,7%)
	прочие	0,1410 (8,4%)	0,1420 (21,2%)	1,10 (9,3%)
	общее	1,679 (100%)	0,6696 (100%)	2,50 (60,1%)
Стресс-тест 2: 990 сечений без указания соответствий	1	28,3811 (99,5%)	7,3448 (98,2%)	3,86 (74,1%)
	5	0,1234 (0,4%)	0,1192 (1,6%)	1,04(3,4%)
	прочие	0,0175 (0,1%)	0,0169 (0,2%)	1,0 (0,3%)
	общее	28,5221 (100%)	7,4809 (100%)	3,81 (73,8%)
Стресс-тест 3: 50 сечений и 50 направляющих	3	2,4088 (44,5%)	0,6122 (17,4%)	3,9 (74,6%)
	5	2,8793 (52,2%)	2,8664 (81,3%)	1,00 (0,45%)
	прочие	0,1209(2,2%)	0,0480 (1,4%)	2,52 (60,3%)
	общее	5,4089 (100%)	3,5266 (100%)	1,53 (34,8%)

На рисунках 4 и 5 представлены результаты работы данного генератора.

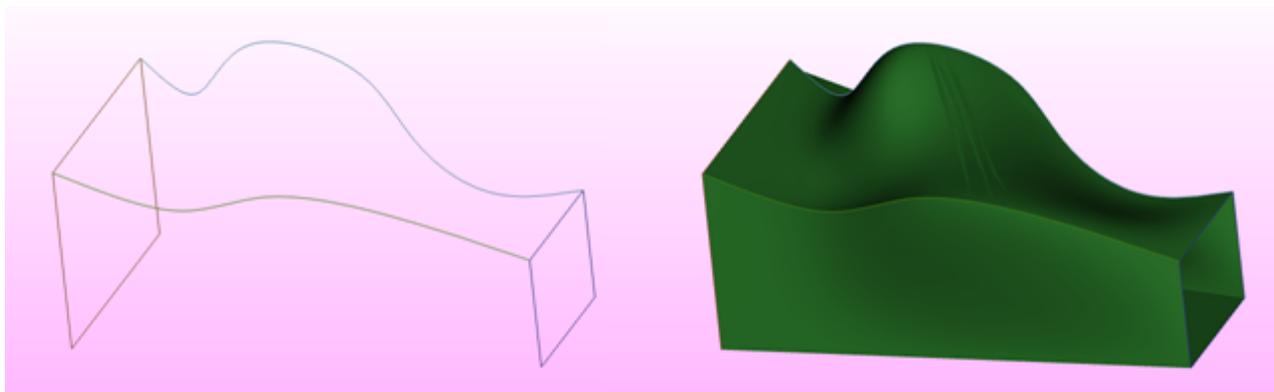


Рис. 4. Деформация гибки стандартной модели Cast из Tflex

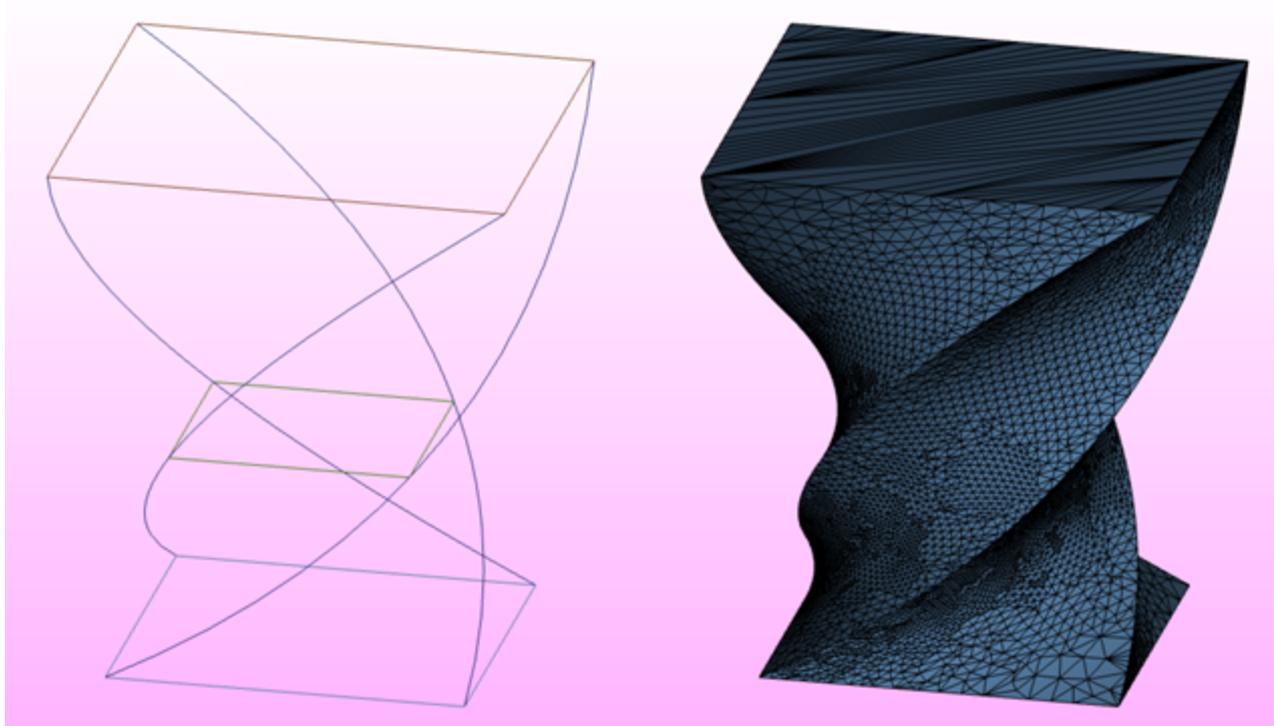


Рис. 5. Деформация гибки стандартной модели Cast из Tflex

Заключение

В процессе выполнения работ были оптимизированы методы построения тел по сечениям, по траектории, а также генератора деформации тел. Целью доработки являлось сокращение времени выполнения вычислений на многопроцессорных системах с общей памятью. В результате по каждому из рассмотренных методов удалось добиться ускорения обработки тестовых моделей. Так, для операции построения тела по траектории ускорение достигает 4,9 раз для 8-ядерной системы, для генератора деформаций – 3,5 раз для 4-ядерной системы, для построения тела по набору сечений – до 3,8 раз для 4-ядерной системы.

Геометрическое ядро на данный момент представляет собой высокоэффективную распараллеленную библиотеку с внушительной функциональностью. В качестве технологий распараллеливания используются открытые стандарты OpenMP и OpenCL. В перспективе предполагается оптимизация пакета под разные архитектуры, в том числе Xeon Phi.

Работа выполняется в рамках НИОКР «Разработка библиотеки алгоритмов генерации базовых тел, построения тел по сечениям, по сетке направляющих кривых и проверки корректности модели для ядра 3-мерного моделирования», договор № 152М-2- ВЛГ-12/1 от 02.04.2012 г.

ЛИТЕРАТУРА:

1. Hoffman С.М. Geometry and solid modeling. Cornell University, 1989. 344 p.
2. Piegl L., Tiller W. The NURBS Book. – 2nd Edition, 646 p.

3. OpenMP API Specification http://www.openmp.org/mp-documents/OpenMP_4.0_RC2.pdf
4. Dinesh Manocha, John F. Canny. Detecting cusps and inflection points in curves // Computer Aided Geometric Design, Computer Aided Geometric Design Volume 9, Issue 1, May 1992, Pages 1–24
5. Piegl L., Tiller W. Computing offsets of NURBS curves and surfaces // Computer-Aided Design, No. 31 (1999), p. 147 – 156.
6. Ian Stroud, Boundary Representation Modeling Techniques, 787 p.
7. http://isicad.ru/ru/articles.php?article_num=15235