

УДК

Пересторонин Аким Максимович

Трушин Владислав Алексеевич

Архипов Кирилл Александрович

студент

«Московский Политехнический Университет», 107023, г. Москва, ул.

Большая Семёновская, 38

**СЕМАНТИЧЕСКИ УПРАВЛЯЕМАЯ ПЛОТНОСТНАЯ
ТОПОЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ КОННЕКТОРОВ В
ПАРАМЕТРИЧЕСКИХ OPENSCAD-СЦЕНАХ**

Аннотация: Предложен метод и программный комплекс для синтеза соединительных элементов непосредственно из семантически размеченных OpenSCAD-сцен. В отличие от Workflow, где геометрия, проектная область и граничные условия подготавливаются отдельно, входной файл содержит и геометрию, и инженерные роли объектов: anchor, load, preserve и obstacle. Проектная область строится автоматически, далее выполняются вокселизация, трёхмерный линейно-упругий расчёт и плотностная топологическая оптимизация по схеме SIMP с обновлением по критерию оптимальности. Для стабилизации решения используются локальный фильтр чувствительности/плотности и гиперболическая проекция, а перевод серого поля плотностей в экспортируемую бинарную геометрию выполняется связностно-ориентированным постпроцессором: кратчайший путь по полю обратных плотностей, наращивание связной области, морфологическое сглаживание и удаление разреженных вокселей.

Ключевые слова: генеративный дизайн, топологическая оптимизация, SIMP (Solid Isotropic Material with Penalization), OpenSCAD, воксельный метод конечных элементов, коннектор, матрично-свободный решатель.

Perestoronin Akim Maksimovich

Trushin Vladislav Alekseevich

Arkhipov Kirill Alexandrovich

student

«Moscow Polytechnic University», 107023, Russian Federation, Moscow, st.

Bolshaya Semyonovskaya, 38

E-mail: mospolytech@mospolytech.ru

SEMANTIC DENSITY-BASED TOPOLOGY OPTIMIZATION OF CONNECTORS IN PARAMETRIC OPENS CAD SCENES

Abstract: A method and software workflow for connector synthesis directly from semantically annotated OpenSCAD scenes is presented. Unlike conventional workflows where geometry, design domain and boundary conditions are prepared separately, the input scene stores both geometry and engineering roles such as anchor, load, preserve and obstacle. The design domain is generated automatically, followed by voxelization, three-dimensional linear-elastic analysis and density-based topology optimization using SIMP and an optimality-criteria update. A neighborhood filter and a hyperbolic projection are used to regularize the density field, while the conversion of gray densities into an exportable binary connector is performed by a connectivity-aware postprocessor combining a shortest path on inverse densities, connected-region growth, morphological smoothing and sparse-voxel pruning

Key words: generative design, topology optimization, SIMP (Solid Isotropic Material with Penalization), OpenSCAD, voxel finite element method, connector, matrix-free solver.

1. Введение

Топологическая оптимизация остаётся одним из наиболее результативных инструментов вычислительного проектирования [1], поскольку позволяет

распределять материал внутри заданной области не по интуитивной схеме, а по критерию механической эффективности. Для задач генеративного дизайна особенно важны плотностные методы семейства SIMP, которые допускают градиентное обновление распределения материала и естественно сочетаются с воксельными и конечно-элементными представлениями [2, 3]. В то же время в реальных инженерных Workflow трудность часто смещается от собственно оптимизационного алгоритма к этапу подготовки расчётной постановки: необходимо синхронизировать геометрию, проектную область, поверхности закрепления, области приложения нагрузки и зоны запрета материала.

Сценарное моделирование в OpenSCAD удобно тем, что геометрия задаётся параметрически и воспроизводимо, однако типовой OpenSCAD-файл не несёт прямой информации о механической роли объекта в расчёте [4]. Поэтому между параметрической CAD-моделью и постановкой задачи топологической оптимизации возникает разрыв, который обычно закрывается ручным переносом данных в отдельные CAE-препроцессоры. Для задач синтеза соединительных элементов это особенно неудобно: коннектор должен связывать уже существующие тела, обходить препятствия, не разрушать сохраняемые детали и одновременно быть экспортируемым обратно в CAD-экосистему.

В представленной работе этот разрыв устраняется за счёт семантической надстройки над OpenSCAD-сценой и связки её с плотностной топологической оптимизацией на регулярной гексаэдральной решётке. Научный интерес работы состоит не только в объединении параметрического CAD-описания и расчёта, но и в процедуре перевода серого поля плотностей в бинарную геометрию коннектора без потери связности между опорой и нагрузкой. В отличие от ранее использовавшейся эвристики поштучного удаления низкоэнергетических элементов, в текущей версии применяются SIMP, фильтрация, гиперболическая проекция и постпроцессор, ориентированный на кратчайший плотностный путь и наращивание связной области.

Подходы к оптимизации информационных моделей САПР рассматриваются также в работе [5], что подтверждает актуальность переноса части инженерной логики на уровень параметрического описания модели.

2. Постановка задачи

Пусть параметрическая сцена задаётся множеством объектов $S = \{E_i\}$. Каждый объект E_i характеризуется геометрией G_i , ролью r_i , материалом m_i , а также атрибутами для формирования расчётной постановки. В рассматриваемой реализации используются роли `anchor`, `load`, `preserve`, `obstacle` и `part`. Семантические блоки `GD_SCENE` и `GD_ENTITY` позволяют хранить в одном входном файле как собственно геометрию, так и инженерные параметры: вектор закрепления, вектор внешней силы, флаги участия в построении проектной области и запрета материала [6, 7].

После автоматического построения ограничивающего параллелепипеда вокруг объектов с флагом `connect` формируются множества Ω_a , Ω_l , Ω_p и Ω_o , соответствующие опоре, нагрузке, сохраняемым телам и препятствиям. Проектная область Ω_d строится как разность допустимого объёма и уже занятых либо запрещённых областей. Оптимизационные переменные ρ_e определяются только на вокселях из Ω_d . В данной версии пакета строгая оптимизационная постановка сохраняет классическую задачу минимума комплаенса при ограничении на объём; напряжения и перемещения рассчитываются на каждой итерации как инженерные диагностические показатели и используются для итоговой проверки конструкции, но не вводятся как активные ограничения в шаге ОС-обновления.

$$\min_{\{\rho\}} C(\rho) = F^T u \quad (1)$$

$$\begin{aligned} K(\rho)u = F, \quad (1/n_d)\sum_e \rho_{phys,e} \leq \bar{V}, \quad \rho_{min} \\ \leq \rho_e \leq 1 \end{aligned} \quad (2)$$

Таким образом, требуется найти такое распределение плотностей $\rho = \{\rho_e\}$, которое минимизирует податливость конструкции в связке «опора —

коннектор — нагружаемая область» при фиксированном ресурсе материала и допускает последующую бинаризацию без разрыва несущего пути.

3. Математическая модель

Линейно-упругая часть постановки использует стандартное уравнение равновесия $K(\rho)u = F$, где $K(\rho)$ — глобальная матрица жёсткости, u — вектор перемещений, F — вектор внешних сил. Локальные матрицы жёсткости строятся на первом порядке гексаэдрального элемента, а интерполяция модуля Юнга в проектной области задаётся схемой SIMP. Для коннектора применяются свойства PETG в системе единиц мм–Н–МПа; сохранённые детали рассматриваются как сплошной ABS-материал с собственной жёсткостью.

$$E_e(\rho_{\text{phys},e}) = E_{\text{min}} + \rho_{\text{phys},e}^p (E_0 - E_{\text{min}}) \quad (3)$$

Чтобы подавить сеточную зависимость и «шахматные» структуры, на поле проектных переменных накладывается локальный взвешенный фильтр [8]. После фильтрации используется гиперболическая проекция, повышающая контраст между пустотой и материалом. Параметры ρ_{enl} и β изменяются по continuation-схеме: по мере роста номера итерации возрастает жёсткостная пенализация промежуточных плотностей и усиливается бинаризирующий эффект проекции. Обновление осуществляется по критерию оптимальности и ограничивается параметром `move_limit`, который стабилизирует шаг по плотности.

$$\tilde{\rho}_e = (\sum_{\{i \in N_e\}} H_{ei} \rho_i) / (\sum_{\{i \in N_e\}} H_{ei}), \quad (4)$$

$$H_{ei} = \max(0, r_f - \|x_e - x_i\|_2)$$

$$\rho_{\text{phys},e} = [\tanh(\beta\eta) + \tanh(\beta(\tilde{\rho}_e - \eta))] / [\tanh(\beta\eta) + \tanh(\beta(1 - \eta))] \quad (5)$$

$$\rho_e^{k+1} = \text{clip}(\rho_e^k \sqrt{-(\partial C / \partial \rho_e) / (\lambda \partial V / \partial \rho_e)}, \rho_e^k - m, \rho_e^k + m) \quad (6)$$

Важной особенностью является постпроцессор бинаризации. Из поля физических плотностей формируется связная маска: сначала вычисляется

кратчайший путь между фронтами опоры и нагрузки по стоимости, обратной локальной плотности; затем область наращивается по максимуму ρ_{phys} до целевого числа вокселей; после этого выполняются морфологическое замыкание/размыкание и отсечение разреженных фрагментов. Итоговая бинарная структура принудительно проверяется на связность в 6-соседстве. Такой шаг позволяет экспортировать коннектор обратно в OpenSCAD не как абстрактное поле плотностей, а как геометрически определённый объект.

$$\sigma_{\text{vm}} = \{0.5[(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2] + 3(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2)\}^{\{1/2\}} \quad (7)$$

4. Алгоритмическая схема и вычислительная реализация

Программный комплекс реализован на Python [9] и состоит из пяти основных уровней. Парсер извлекает параметры сцены, примитивы, трансформации, булевы операции и пользовательские модули OpenSCAD. Геометрический модуль строит воксельную решётку, маски ролей и обязательную окрестность anchor/load, которая не должна исчезать при бинаризации. FEM-модуль формирует локальные матрицы, собирает оператор жёсткости для активных элементов и выбирает режим решения: sparse-сборка для умеренных размеров или матрично-свободный итерационный режим для крупных задач. Оптимизационный модуль реализует фильтр, проекцию, ОС-обновление и связностно-ориентированную бинаризацию [10, 11]. Наконец, слой визуализации и экспорта сохраняет итерационные кадры, карты напряжений, метрики, итоговый SCAD-файл коннектора и обзорную сцену.

Интерфейс системы оптимизации коннектора на базе OpenSCAD организован как одностраничное рабочее пространство с тремя основными зонами. Слева расположена панель быстрых настроек (рисунок 1), содержащая параметры расчета и дискретизации: пресет качества, материал, размер вокселя, целевой объем, максимальное перемещение, коэффициент запаса и расширенные параметры. В центральной части (рисунок 2) размещена область загрузки исходного .scad-файла и сводка по расчетной модели, где отображаются число

объектов, закреплений, нагрузок, препятствий, активных вокселей, узлов, степеней свободы и оценка размера матрицы. Верхняя часть также содержит элементы управления сценой.

Нижняя часть интерфейса разделена на вкладки, соответствующие основным этапам работы: просмотр сцены (рисунок 3), редактирование объектов (рисунок 4), запуск и анализ результатов, а также генерация структурированного.scad-описания (рисунок 5).

Во вкладке сцены доступна 3D-визуализация модели с цветовым разделением ролей объектов и краткой сводкой ограничений расчета.

Во вкладке объектов реализована форма поэлементного редактирования: имя, роль, материал, закрепления по осям, компоненты нагрузки и логические флаги участия объекта в оптимизации.

Такая структура обеспечивает последовательный переход от задания исходных данных к визуальному контролю, настройке модели и получению итогового кода.

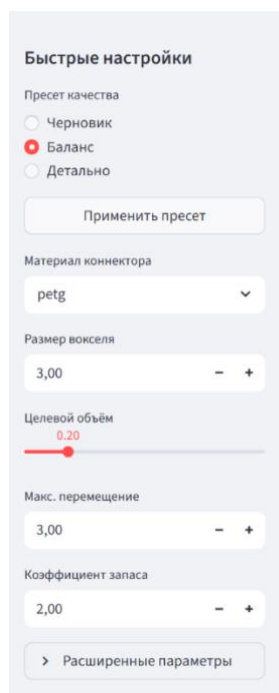


Рисунок 1 – Боковая панель

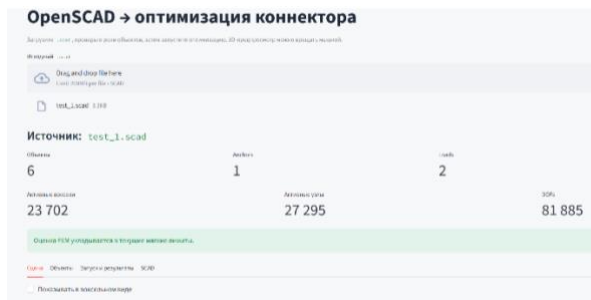


Рисунок 2 – Центральная панель

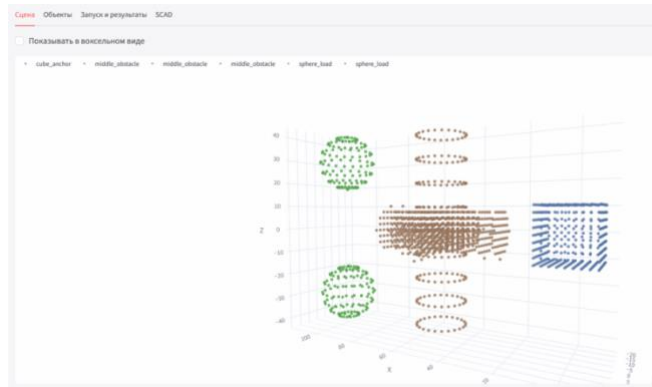


Рисунок 3 – Просмотр сцены

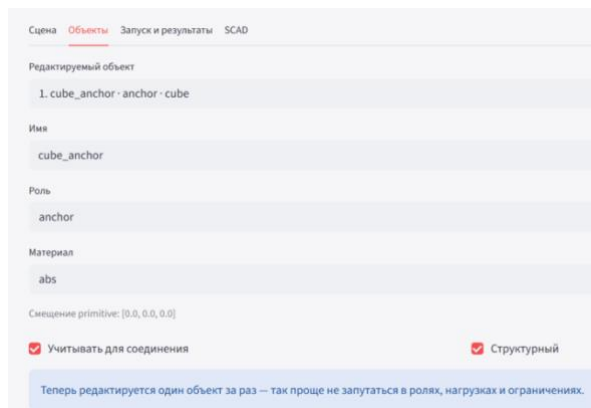


Рисунок 4 – Редактирование объекта

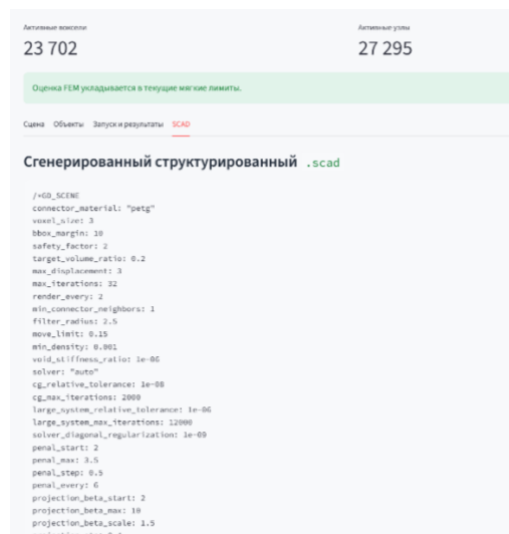


Рисунок 5 – SCAD-объект

Для высокоразрешённых постановок ключевым является переход к matrix-free-режиму [12, 13]. В архивном сценарии 0,2 мм оценка активного контекста составляет около 3,21 млн степеней свободы и 8,72 ГБ как ориентировочный объём собранной разреженной матрицы. Поэтому оптимизатор выбирает матрично-свободную схему применения оператора жёсткости и решает систему методом сопряжённых градиентов; это снижает риск исчерпания памяти и позволяет довести сцену до последней сохранённой итерации без полного LU-факторизационного шага.

5. Вычислительный эксперимент

В качестве тестовой задачи рассмотрен узел из двух моторно-энкодерных сборок, расположенных одна над другой. Верхний вал играет роль области нагрузки (load) и нагружается силой $F_z = -1$ Н; нижний корпус двигателя — anchor-область с полным закреплением по трём координатам; отдельный корпус нижнего энкодера сохранён в составе конструкции; части верхнего двигателя, верхнего энкодера и нижнего вала используются как препятствия для размещения материала. Такая постановка воспроизводит типичную задачу генеративного синтеза переходного элемента между уже заданными телами.

Первый этап – составление таблицы, которая отражает изменение результатов в зависимости от размера вокселя.

Здесь V_x – размер вокселей, использованный при расчётах, V_c – их итоговое количество, V_m – рассчитанный объём объекта.

Таблица 1. Параметры среды в зависимости от выбранного размера вокселя

V_x	Память, ГБ	V_c	V_m	Нагрузка, МПа	Смещение, мм
3.0	0.2	3684	99468.0	0.003	8.401e-05
2.5	0.3	4719	73734.375	0.003	0.0001142
2.0	0.7	6734	53872.0	0.02	0.0002175
1.5	1.6	10319	34826.625	0.01	0.000231
1.0	2.4	20079	20079.0	0.08	0.0005582

В рамках исследования была выполнена серия вычислительных экспериментов для пяти значений размера вокселя: 1,0; 1,5; 2,0; 2,5 и 3,0 мм. Для каждого запуска анализировались показатели вычислительной нагрузки, размер конечно-элементной задачи, динамика уменьшения объёма материала на контрольных шагах 4, 14 и 32, а также итоговые механические характеристики после постобработки.

Результаты показывают выраженную зависимость вычислительной стоимости от размера вокселя. При уменьшении размера вокселя до 1,0 мм время расчёта возрастает до 119,8 мин, число степеней свободы достигает 1 919 730, расчётная ёмкость матрицы — 5,22 GB (рисунок 7). Для вариантов 2,0–3,0 мм время расчёта снижается до 6,5–11,3 мин, а размер задачи уменьшается до диапазона 81 885–251 403 степеней свободы. Таким образом, уменьшение размера вокселя обеспечивает более детализированное описание структуры, однако сопровождается резким ростом вычислительных затрат (рисунок 6).

Установлено, что с уменьшением размера вокселя снижается и относительный финальный объём коннектора (рисунок 8). На 32-й итерации доля объёма составляет 16,1 % для размера 3,0 мм, 12,1 % для 2,5 мм, 9,2 % для 2,0 мм, 5,7 % для 1,5 мм и 3,4 % для 1,0 мм. Это указывает на то, что более мелкая дискретизация позволяет получить более компактную итоговую топологию, но требует существенно большей вычислительной мощности.

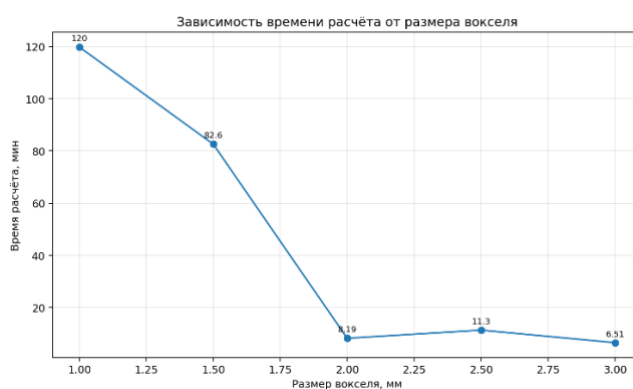


Рисунок 6 – Зависимость времени расчёта от размера вокселя

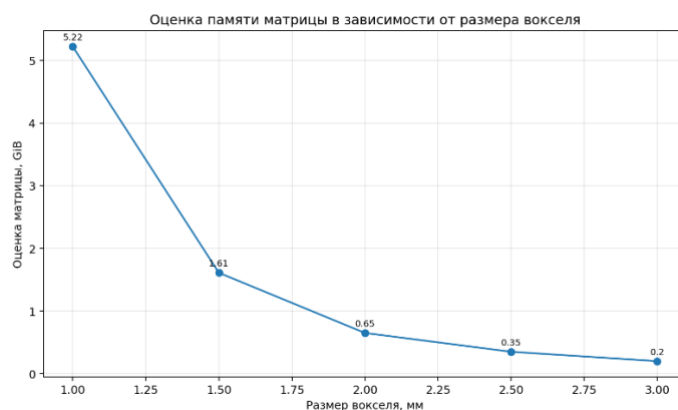


Рисунок 7 – Оценка памяти матрицы в зависимости от размера вокселя

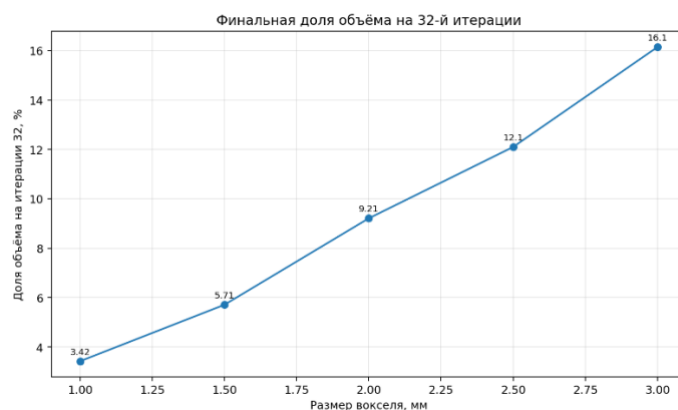


Рисунок 8 – Финальная доля объема на 32-й итерации

Для высокоразрешённого архивного запуска использовались параметры:

- размер вокселя 0.2 мм
- шаг по объёму 0.18
- renal от 1.5 до 3.5
- β от 2.0 до 12.0
- $\eta = 0.45$

Автоматически построенная сетка имела размер $96 \times 55 \times 248$ вокселей. Число проектных вокселей составило 660 911 а обязательная окрестность anchor/load включала 27 103 ячеек.

Итоговые данные занесены в таблицу 2.

Таблица 2. Параметры высокоразрешённого архивного сценария и итоговые показатели

Параметр	Значение
Материал коннектора	PETG
Шаг вокселя, мм	0.2
Размер сетки	96×55×248
Проектные воксели	660 911
Последняя сохранённая итерация	37
Коннектор, количество вокселей	50 792
Объём коннектора, мм ³	406.336
$\sigma_{vm,max}$, МПа	0.165
u_{max} , мм	0.001332
Режим решателя	matrix-free-cg

matrix-free-cg – метод сопряжённых градиентов.

Для удобства сопоставления с экспериментальными данными в статье приведены результаты при масштабированной нагрузке 1 Н; пересчёт на 1000 Н даёт пропорциональное увеличение напряжений и перемещений (до 165 МПа и 1.33 мм соответственно), что превышает предел текучести PETG, поэтому реальная конструкция потребовала бы увеличения объёма материала. На рисунках 9-12 представлена эволюция высокоразрешённого сценария – на рисунке 9 представлена начальная вокселизованная модель, на рисунке 10 – процесс оптимизации – одна из стадий, промежуточная сохранённая бинарная топология, на рисунке 11 – финальная версия, видна карта напряжений.

На рисунке 12 показана автоматически построенная проектная модель в этой системе.

На последней сохранённой 37-й итерации получен коннектор из 50 792 вокселей объёмом 406.336 мм³. Это соответствует бинарному отношению

объёма 0.0769 и уменьшению материала на 92.3% относительно полностью заполненной проектной области объёмом 5287.288 мм³.

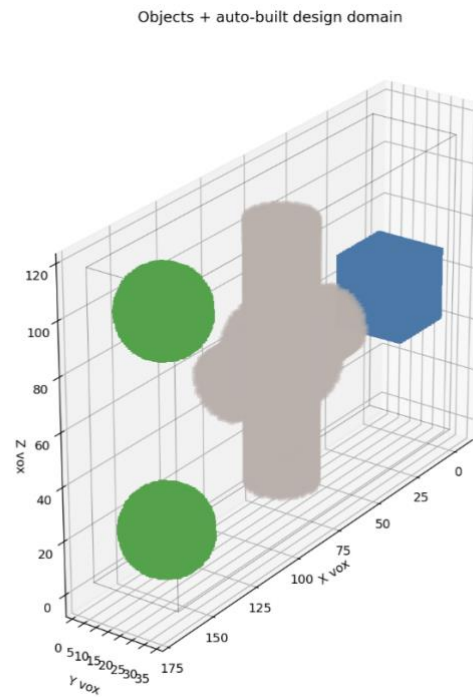


Рисунок 9 – Начальная вокселизированная модель

SIMP+OC topology step — stress
iter=4 | change=0.1500 | vox=68625 | rho_vol=0.129 | vm=0.02 MPa

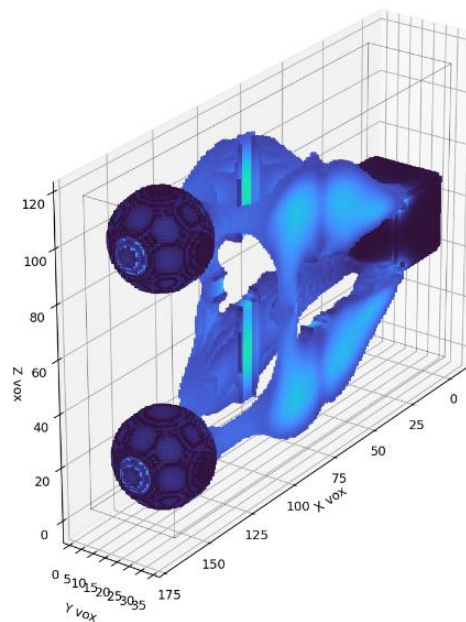


Рисунок 10 – Процесс оптимизации (15-я стадия)

SIMP+OC topology step — stress
iter=20 | change=0.1500 | vox=37484 | rho_vol=0.071 | vm=0.02 MPa

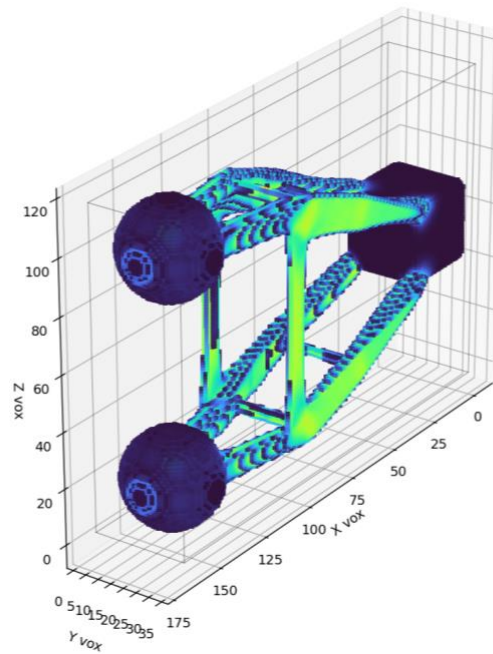


Рисунок 11 – Карта напряжений

SIMP+OC topology step
iter=20 | change=0.1500 | vox=37484 | rho_vol=0.071 | vm=0.02 MPa

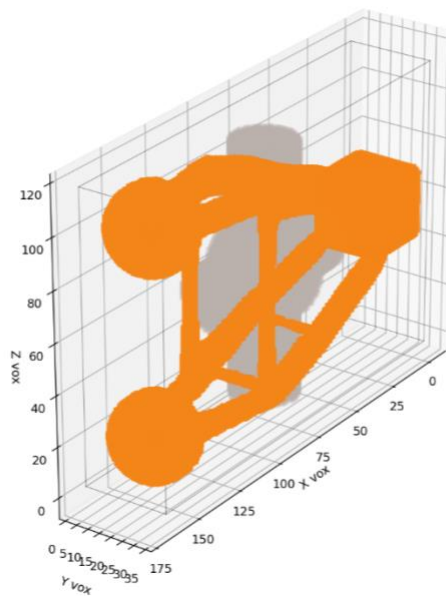


Рисунок 12 – Проектная модель

На рисунке 13 показана модель в САД-системе.

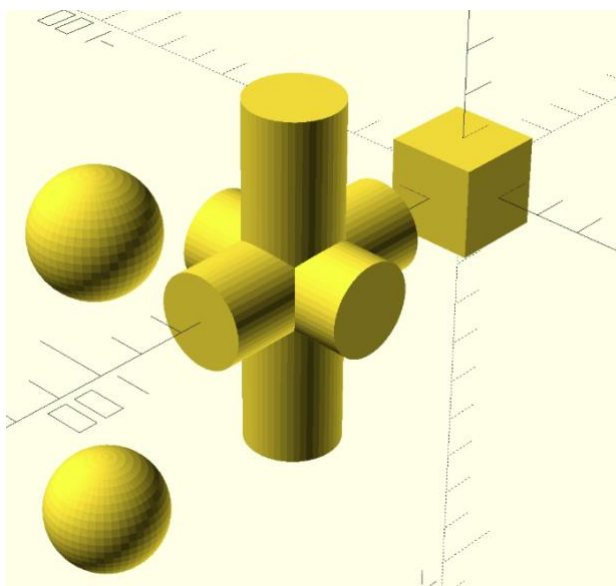


Рисунок 13 – Модель коннектора в OpenSCAD

Диагностические показатели (рисунок 14) на этой итерации составили: $\sigma_{vm,max} = 0.165$ МПа, $\sigma_{abs,max} = 0.091$ МПа, $u_{max} = 0.001332$ мм и $C = 1.302$ в согласованной системе единиц мм–Н–МПа. При выбранном коэффициенте запаса допустимое напряжение для PETG равно $\sigma_{allow} = 29.23$ МПа, так что итоговая диагностическая напряжённость остаётся значительно ниже этого порога. Визуальный анализ показывает формирование двух выраженных вертикальных ветвей силового потока и нижнего охватывающего участка, замыкающего траекторию передачи усилия между сборками.

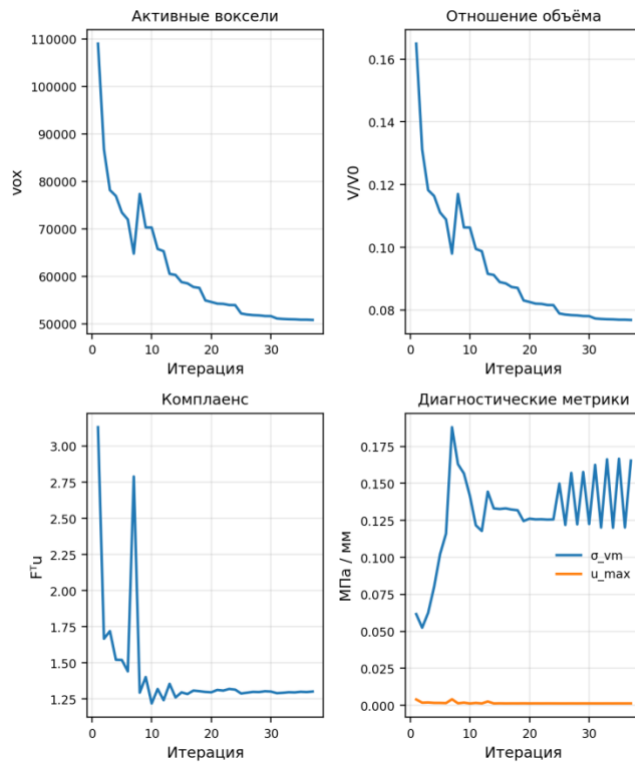


Рисунок 14 – Динамика числа вокселей, отношения объёма, комплаенса и диагностических механических метрик по итерациям высокоразрешённого запуска.

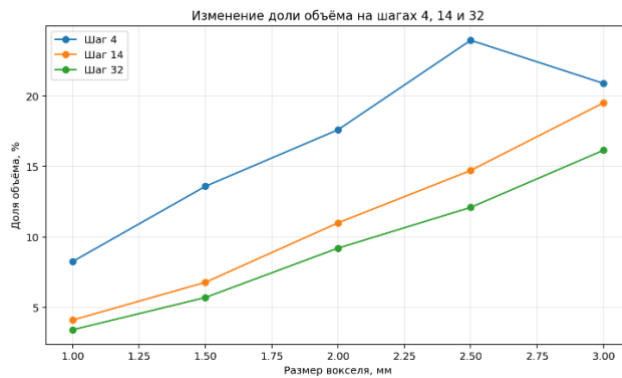
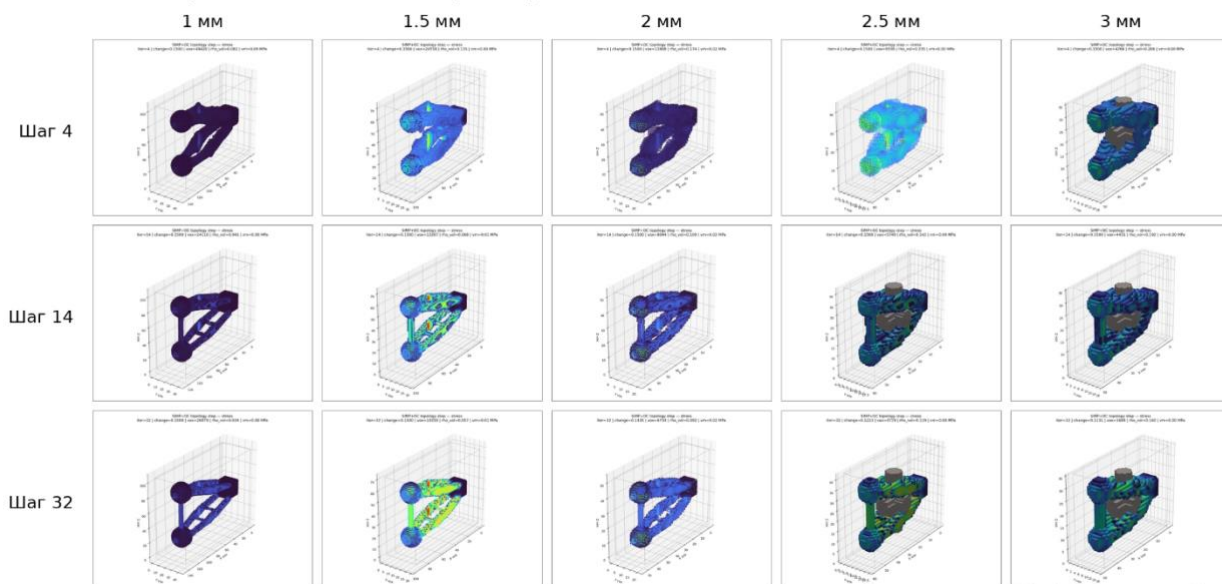


Рисунок 15 – Изменение доли объёма на шагах 4, 14, 32

Анализ контрольных шагов 4, 14 и 32 (рисунок 15) показывает, что основное сокращение объёма происходит на ранних стадиях итерационного процесса.

Сетка stress-кадров: шаги 4, 14, 32 × размер вокселя 1.0–3.0 мм



Например, для размера 1,0 мм доля объёма уменьшается с 8,3 % на шаге 4 до 4,1 % на шаге 14 и до 3,4 % на шаге 32. Аналогичная закономерность сохраняется и для остальных размеров вокселя, что подтверждает устойчивость процесса уменьшения объёма независимо от выбранного пространственного разрешения.

С точки зрения механического отклика после постобработки минимальные значения compliance получены для более грубых сеток: 0,160 при размере 3,0 мм и 0,206 при размере 2,5 мм. Для более мелких сеток значения выше: 0,430 для 2,0 мм, 0,438 для 1,5 мм и 1,104 для 1,0 мм. Перемещения после постобработки также изменяются закономерно: от 0,000084 для 3,0 мм до 0,000558 для 1,0 мм. Это свидетельствует о том, что повышение детализации сетки не всегда приводит к улучшению интегральных жёсткостных характеристик и должно рассматриваться совместно с вычислительной ценой расчёта.

Следует отметить и изменение используемой вычислительной схемы. Для вариантов 1,0 и 1,5 мм применялся режим *matrix-free-cg*, тогда как для 2,0–3,0 мм использовался *cg*. Для расчёта с размером вокселя 1,0 мм зафиксировано 3 предупреждения, для 1,5 мм — 2 предупреждения, что дополнительно указывает на рост вычислительной сложности при чрезмерном уменьшении шага дискретизации.

6. Обсуждение

Полученный результат подтверждает, что даже при прямоугольной регулярной дискретизации и сравнительно простой механике можно получить практически интерпретируемую форму коннектора, если связать CAD-сцену с оптимизацией на уровне семантики. Существенное преимущество подхода заключается в воспроизводимости: одно и то же текстовое описание OpenSCAD сцены одновременно определяет геометрию, роли объектов и конфигурацию оптимизатора. Не менее важен и переход от поля плотностей к бинарной форме. Без этого шага результат оставался бы сугубо численным и требовал бы ручной доработки перед экспортом в CAD-пакет или перед 3D-печатью.

Следует, однако, отметить ряд ограничений. Во-первых, текущий ОС-контур не включает активных ограничений на напряжения, собственные частоты или технологические требования; эти показатели контролируются после расчёта как диагностические величины. Во-вторых, в очень крупных задачах итоговый бинарный re-solve может быть пропущен ради приемлемого расхода памяти, и тогда финальные механические показатели относятся к последнему плотностному состоянию. В-третьих, пока не реализованы многослучайное нагружение, anisotropic-параметры печати и производственные ограничения, характерные для additively manufactured деталей.

Для оценки места предложенного подхода среди существующих решений проведём сравнение с наиболее распространёнными workflow топологической оптимизации. Классические инструменты, такие как Abaqus/OptiStruct или Ansys Mechanical, требуют отдельного определения геометрии, проектной области, граничных условий и конечно-элементной сетки в разных модулях, что создаёт риск потери согласованности при параметрических изменениях. Открытые реализации, например `topopt.py` или 88-строчный код на MATLAB [10, 11], напротив, интегрируют расчёт и оптимизацию, но не поддерживают

прямую связь с параметрической CAD-моделью, а их постпроцессинг часто ограничивается пороговой бинаризацией без контроля связности.

Предложенный метод устраняет оба этих недостатка за счёт семантической разметки внутри OpenSCAD-сцены: проектная область, роли объектов и ограничения извлекаются автоматически, что обеспечивает полную воспроизводимость расчёта при любом изменении геометрических параметров. Кроме того, связностно-ориентированный постпроцессор гарантирует, что итоговая бинарная геометрия коннектора содержит непрерывный путь между опорой и нагрузкой, в отличие от традиционного отсечения по порогу плотности, которое часто приводит к разрывам. Таким образом, представленная работа сокращает разрыв между генеративным CAD-проектированием и топологической оптимизацией, сохраняя при этом возможность экспорта результата обратно в параметрическую сцену.

7. Заключение

Разработан и исследован метод семантически управляемой плотностной топологической оптимизации коннекторов в параметрических OpenSCAD-сценах. Метод объединяет сценовое описание, автоматическое построение проектной области, воксельный МКЭ, SIMP-оптимизацию с фильтрацией и проекцией, а также связностно-ориентированную бинаризацию плотностей. На высокоразрешённом архивном сценарии показано, что подход позволяет получать экспортируемую топологию коннектора при масштабах порядка нескольких миллионов степеней свободы и при этом сохранять инженерно низкие диагностические напряжения и перемещения.

Дальнейшее развитие целесообразно направить на постановки с активными ограничениями по напряжению и перемещениям, многократные случаи нагружения, согласование с технологическими правилами 3D-печати и построение полноценных benchmark-сцен для систематического сравнения с классическими density-based и level-set решателями [14].

Список литературы

1. Tang T., Wang L., Zhu M., Zhang H., Dong J., Yue W., Xia H. Topology Optimization: A Review for Structural Designs Under Statics Problems // *Materials*. 2024. Vol. 17, no. 23. Art. 5970.
2. Birosz M. T., Bátorfi J. G., Andó M. Extending the usability of the force-flow based topology optimization to the process of generative design // *Meccanica*. 2023. Vol. 58. P. 607–618.
3. Wang F., Lazarov B. S., Sigmund O. On projection methods, convergence and robust formulations in topology optimization // *Structural and Multidisciplinary Optimization*. 2011. Vol. 43, no. 6. P. 767–784.
4. OpenSCAD Developers. OpenSCAD Documentation. URL: <https://openscad.org/documentation> (дата обращения: 10.04.2026)
5. Полигональный CAD. URL: <https://github.com/Mika-dot/Cad/tree/V2-Experiment> (дата обращения: 18.04.2026)
6. Bendsoe M. P., Sigmund O. *Topology Optimization: Theory, Methods, and Applications*. Berlin: Springer, 2003. 370 p.
7. Sigmund O., Maute K. Topology optimization approaches: A comparative review // *Structural and Multidisciplinary Optimization*. 2013. Vol. 48, no. 6. P. 1031–1055.
8. Bourdin B. Filters in topology optimization // *International Journal for Numerical Methods in Engineering*. 2001. Vol. 50, no. 9. P. 2143–2158.
9. Gustafsson T., McBain G. D. scikit-fem: A Python package for finite element assembly // *Journal of Open Source Software*. 2020. Vol. 5, no. 52. Art. 2369.
10. Sigmund O. A 99 line topology optimization code written in Matlab // *Structural and Multidisciplinary Optimization*. 2001. Vol. 21, no. 2. P. 120–127.
11. Andreassen E., Clausen A., Schevenels M., Lazarov B. S., Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code // *Structural and Multidisciplinary Optimization*. 2011. Vol. 43, no. 1. P. 1–16.
12. Lazarov B. S., Sigmund O. Filters in topology optimization based on Helmholtz-type differential equations // *International Journal for Numerical Methods in Engineering*. 2011. Vol. 86, no. 6. P. 765–781.
13. El Khadiri I., Zemzami M., Nguyen N.-Q., Abouelmajd M., Hmina N., Belhouideg S. Topology optimization methods for additive manufacturing: a review // *International Journal for Simulation and Multidisciplinary Design Optimization*. 2023. Vol. 14. Art. 12.
14. Bendsoe M. P., Kikuchi N. Generating optimal topologies in structural design using a homogenization method // *Computer Methods in Applied Mechanics and Engineering*. 1988. Vol. 71, no. 2. P. 197–224.

References

15. Tang T., Wang L., Zhu M., Zhang H., Dong J., Yue W., Xia H. Topology Optimization: A Review for Structural Designs Under Statics Problems // *Materials*. 2024. Vol. 17, no. 23. Art. 5970.
16. Birosz M. T., Bátorfi J. G., Andó M. Extending the usability of the force-flow based topology optimization to the process of generative design // *Meccanica*. 2023. Vol. 58. P. 607–618.
17. Wang F., Lazarov B. S., Sigmund O. On projection methods, convergence and robust formulations in topology optimization // *Structural and Multidisciplinary Optimization*. 2011. Vol. 43, no. 6. P. 767–784.
18. OpenSCAD Developers. OpenSCAD Documentation. URL: <https://openscad.org/documentation> (дата обращения: 10.04.2026)
19. Полигональный CAD. URL: <https://github.com/Mika-dot/Cad/tree/V2-Experiment> (дата обращения: 18.04.2026)
20. Bendsoe M. P., Sigmund O. *Topology Optimization: Theory, Methods, and Applications*. Berlin: Springer, 2003. 370 p.
21. Sigmund O., Maute K. Topology optimization approaches: A comparative review // *Structural and Multidisciplinary Optimization*. 2013. Vol. 48, no. 6. P. 1031–1055.
22. Bourdin B. Filters in topology optimization // *International Journal for Numerical Methods in Engineering*. 2001. Vol. 50, no. 9. P. 2143–2158.
23. Gustafsson T., McBain G. D. scikit-fem: A Python package for finite element assembly // *Journal of Open Source Software*. 2020. Vol. 5, no. 52. Art. 2369.
24. Sigmund O. A 99 line topology optimization code written in Matlab // *Structural and Multidisciplinary Optimization*. 2001. Vol. 21, no. 2. P. 120–127.
25. Andreassen E., Clausen A., Schevenels M., Lazarov B. S., Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code // *Structural and Multidisciplinary Optimization*. 2011. Vol. 43, no. 1. P. 1–16.
26. Lazarov B. S., Sigmund O. Filters in topology optimization based on Helmholtz-type differential equations // *International Journal for Numerical Methods in Engineering*. 2011. Vol. 86, no. 6. P. 765–781.
27. El Khadiri I., Zemzami M., Nguyen N.-Q., Abouelmajd M., Hmina N., Belhouideg S. Topology optimization methods for additive manufacturing: a review // *International Journal for Simulation and Multidisciplinary Design Optimization*. 2023. Vol. 14. Art. 12.
28. Bendsoe M. P., Kikuchi N. Generating optimal topologies in structural design using a homogenization method // *Computer Methods in Applied Mechanics and Engineering*. 1988. Vol. 71, no. 2. P. 197–224.

