ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.94

DOI: 10.21822/2073-6185-2025-52-4-165-180 Оригинальная статья/ Original article

Автоматизация процесса армирования железобетонных конструкций в программе Autodesk Revit с использованием Dynamo Studio

Т.А. Юрошева[1], Н.С. Исупов[2], Ш.Р. Кудзиев[1], Е.Н. Акоева[1]
[1]Северо – Кавказский горно-металлургический институт
(Государственный технологический университет),
[1]362021, г. Владикавказ, ул. Николаева, 44, Россия,
[2] Уральский федеральный университет имени первого Президента России Б.Н. Ельцина,
[2]620002, г. Екатеринбург, ул. Мира 19, Россия

Резюме. Цель. Целью работы является исследование возможностей программы Dynamo Studio как инструмента для автоматизации проектирования систем армирования в Revit и разработка алгоритмов армирования различных конструктивных элементов: балок, колонн и плит перекрытий. **Метод.** Исследования основано на методах автоматизация процесса армирования железобетонных конструкций в программе Autodesk Revit с использованием Dynamo Studio. В работе детально рассмотрены принципы работы с основными нодами и геометрическими параметрами элементов. **Результат.** Разработаны методики создания арматурных каркасов для балок, колонн и плит перекрытий с учетом их конструктивных особенностей, и геометрических параметров. Представлено подробное описание алгоритмов армирования балок, колонн и плит перекрытий. Описаны основные ноды и последовательность действий при создании арматурных каркасов. **Вывод.** Результаты исследования демонстрируют возможность эффективного автоматизированного проектирования систем армирования железобетонных конструкций. Применение Dynamo Studio для автоматизации процесса армирования существенно повышает производительность труда и обеспечивает высокое качество проектной документации. Результаты исследования могут служить основой для создания более сложных автоматизированных систем проектирования арматурных конструкций. Выводы подтверждают практическую значимость полученных результатов, которые могут быть использованы проектировщиками при разработке строительных конструкций.

Ключевые слова: автоматизация проектирования, армирование, железобетонные конструкции, Dynamo, Revit, арматурные каркасы

Для цитирования: Т.А. Юрошева, Н.С. Исупов, Ш.Р. Кудзиев, Е.Н. Акоева. Автоматизация процесса армирования железобетонных конструкций в программе Autodesk Revit с использованием Dynamo Studio. Вестник Дагестанского государственного технического университета. Технические науки. 2025;52(4):165-180. DOI:10.21822/2073-6185-2025-52-4-165-180

Automation of the reinforcement of reinforced concrete structures in the Autodesk Revit program using Dynamo Studio

T.A. Yurosheva[1], N.S. Isupov[2], Sh.R. Kudziev[1], E.N. Akoeva[1]
[1]North Caucasian Institute of Mining and Metallurgy
(State Technological University),
44 Nikolaeva Str., Vladikavkaz 362011, Russia,
[2] B.N. Yeltsin first President of Russia Ural Federal University,
19 Mira Str., Ekaterinbur 620002, Russia

Abstract. Objective. The aim of the work is to study the capabilities of the Dynamo Studio program as a tool for automating the design of reinforcement systems in Revit and to develop

algorithms for reinforcing various structural elements: beams, columns, and floor slabs. **Method.** The study is based on methods for automating the reinforcement process of reinforced concrete structures in Autodesk Revit using Dynamo Studio. The paper considers the principles of working with the main nodes and geometric parameters of elements. **Result.** Methods for creating reinforcement cages for beams, columns, and floor slabs have been developed, taking into account their design features and geometric parameters. Algorithms for reinforcing beams, columns, and floor slabs are described. The main nodes and the sequence of actions for creating reinforcement cages are described. **Conclusion**. The results of the study demonstrate the possibility of effective automated design of reinforcement systems for reinforced concrete structures. The use of Dynamo Studio to automate the reinforcement process significantly increases labor productivity and ensures high-quality design documentation. The results of the study can serve as the basis for creating complex automated systems for the design of reinforcement structures. The findings confirm the practical significance of the results obtained and can be used by designers when developing building structures.

Keywords: design automation, reinforcement, reinforced concrete structures, Dynamo, Revit, reinforcement frames

For citation: T.A. Yurosheva, N.S. Isupov, Sh.R. Kudziev, E.N. Akoeva. Automation of the reinforcement of reinforced concrete structures in the Autodesk Revit program using Dynamo Studio. Herald of Daghestan State Technical University. Technical Sciences. 2025;52(4):165-180. (In Russ) DOI:10.21822/2073-6185-2025-52-4-165-180

Введение. В современных условиях проектирования строительных конструкций особую актуальность приобретает автоматизация процессов армирования железобетонных элементов. Использование программного обеспечения для создания арматурных каркасов позволяет существенно сократить время проектирования и минимизировать вероятность ошибок [1].

Постановка задачи. Целью данной работы является исследование возможностей программы Dynamo как инструмента для автоматизации проектирования систем армирования в Revit. В ходе исследования были поставлены задачи по разработке алгоритмов армирования различных конструктивных элементов: балок, колонн и плит перекрытий.

Для достижения поставленной цели решались следующие задачи:

1. Исследовать и проанализировать существующие методы и алгоритмы армирования железобетонных конструкций;
2. Разработать алгоритм автоматизированного проектирования арматурных каркасов для: балок с возможностью определения геометрических параметров и создания арматурного каркаса с заданными характеристиками; колонн с построением местной системы координат и автоматизированным размещением продольных стержней; плиты перекрытия с автоматическим определением верхней грани и формированием арматурного каркаса;
3. Создать комплексную методику автоматизированного проектирования, включающую: определение геометрических параметров элементов и расчет необходимых параметров армирования; формирование арматурного каркаса с учетом конструктивных особенностей; выбор типа арматуры и способа загибки;
4. Разработать систему нодов для: получения параметров элементов; построения геометрии арматурного каркаса; создания арматурных стержней и хомутов; формирования арматурного каркаса в целом;
5. Провести тестирование разработанной системы на различных типах железобетонных конструкций, оценить эффективность предложенного решения и его практическую применимость;
6. Разработать рекомендации по использованию созданной системы в проектной деятельности.

Решение поставленных задач позволило создать эффективный инструмент

для автоматизации процесса проектирования армирования железобетонных конструкций, что существенно повысит производительность труда проектировщиков и обеспечит высокое качество проектной документации.

Научная новизна результатов исследования заключается в следующем:

1. Впервые разработана комплексная методика автоматизированного проектирования систем армирования железобетонных конструкций, объединяющая возможности программ Revit и Dynamo в единый технологический процесс.
2. Предложен инновационный алгоритм создания арматурных каркасов, включающий автоматизированное определение геометрических параметров элементов и формирование арматурного каркаса с использованием семейства нодов Geometry, Element и CodeBlock.
3. Разработана оригинальная система построения местной системы координат для армирования колонн с применением нода Vector.ByTwoPoints, позволяющая оптимизировать процесс размещения продольных арматурных стержней.
4. Создан новый подход к проектированию армирования перекрытий с использованием нодов Curve и Geometry для определения лучей армирования и формирования арматурных стержней.
5. Впервые внедрен метод автоматизированного расчета параметров армирования (длина, ширина, высота элементов) с помощью нода Element.Get ParameterValue By Name, что позволяет существенно повысить точность проектирования.
6. Предложена усовершенствованная технология создания арматурных каркасов с возможностью выбора стиля арматирования, типа арматуры и типа загибки через пакет DynamoForRebar.

Практическая значимость полученных результатов заключается в создании эффективного инструмента для автоматизации процесса проектирования армирования железобетонных конструкций, что позволяет значительно сократить время проектирования и повысить качество проектной документации.

Методы исследования. Алгоритмы процесса армирования каркасов для железобетнных конструкций

Армирование балок. Инициализация алгоритма осуществляется посредством селекции идентификатора требуемой балки. Последующий этап предусматривает детерминирование геометрических характеристик элемента посредством интеграции семейства нодов Geometry, Element и CodeBlock, в рамках которого осуществляется квантификация линейных размеров балки - протяженности и ширины [2].

Параллельно производится определение высотного параметра конструкции путем экстракции отметок верхней и нижней граней с последующим вычислением их разности, что позволяет установить искомое значение высоты H.

По завершении этапа геометрического анализа осуществляется переход к формированию арматурного каркаса. Вектор распространения массива хомутов задается посредством нода Vector.ByTwoPoints. Конфигурация массива хомутов реализуется путем спецификации межцентрового расстояния и необходимых геометрических параметров, что обеспечивает генерацию требуемых линий хомутов.

Последующее формирование геометрии продольных стержней и хомутов осуществляется с учетом селекции соответствующего стиля и типа арматурного каркаса, при этом предусматривается возможность анкеровки арматурных элементов посредством использования дополнения DynamoForRebar.

Финальная стадия алгоритма реализуется посредством Python.Script, обеспечивающего формирование армированной конструкции балки. Функциональную схему скрипта можно видеть на рис. 1.

Армирование колонн. Процесс инициализации алгоритма начинается с экстракции граней колонны и последующей калькуляцией их центроидов.

Рис.1 – Функциональная схема скрипта армирования балки
Fig. 1 – Functional diagram of the beam reinforcement script

На последующем этапе осуществляется определение точки нижней торцевой грани посредством использования координат нижней торцевой грани Z и центроида соответствующей грани, при этом финальная стадия данного этапа реализуется через нод Code Block, выполняющий функцию извлечения первого элемента из сформированного списка [3-5]. Последующий этап алгоритма предусматривает определение центроидов боковых граней с последующим их проецированием на плоскость нижнего торца конструкции. Формирование локальной системы координат осуществляется посредством применения нода Vector.ByTwoPoints. В рамках установленной системы координат производится позиционирование точек размещения продольных арматурных стержней. Формирование массива стержней реализуется через создание списка элементов с последующей селекцией требуемого типа арматуры из кодового перечня. На следующем этапе осуществляется спецификация размеров выпусков и интервалов между стержнями для обеспечения корректной установки продольных арматурных элементов. Заключительная стадия алгоритма предусматривает интеграцию всех предварительно определенных параметров в блок создания арматуры, что обеспечивает генерацию арматурных стержней. Функциональная схема алгоритма представлена на рис. 2.

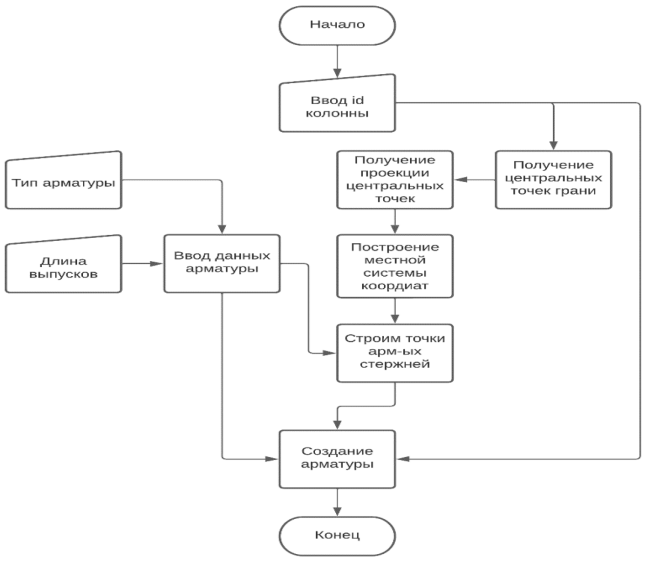

Рис. 2 – Функциональная схема скрипта армирования колонн
Fig. 2 – Functional diagram of the column reinforcement script

Армирование перекрытия. В рамках реализации алгоритма проектирования арматурного каркаса плиты перекрытия, представленного на рис. 3, осуществляется последовательная обработка идентификатора конструктивного элемента. На начальном этапе производится экстракция геометрических характеристик плиты посредством получения массива её граней. В процессе обработки данных особое внимание уделяется определению верхней поверхности перекрытия, что достигается путем обращения к последнему элементу сформированного списка посредством нода List.LastItem.

Последующий этап алгоритма предусматривает генерацию системы направляющих векторов, реализуемую посредством интеграции нодов Curve и Geometry. Данная операция обеспечивает формирование массива лучей, необходимых для корректного позиционирования арматурных элементов в пространстве.

Финальная стадия проектирования характеризуется комплексной обработкой параметров армирования с использованием специализированного пакета DynamoForRebar. В рамках данного этапа осуществляется многоаспектная настройка характеристик арматурного каркаса, включающая: селекцию типоразмера арматурных стержней; определение методики анкеровки арматурных элементов; спецификацию геометрической конфигурации арматурных стержней.

Результатом последовательной реализации описанного алгоритма является формирование полноценного арматурного каркаса плиты перекрытия, соответствующего проектным требованиям и нормативным документам.

Рис. 3 – Функциональная схема скрипта армирования перекрытия
Fig. 3 – Functional diagram of the floor reinforcement script

Обсуждение результатов. Армирование балок. В процессе инициализации алгоритма проектирования, показанного на рис. 4, осуществляется первичная селекция конструктивного элемента посредством нода Revit.Selection, что обеспечивает идентификацию требуемой балки в рамках программного модуля.

Последующий этап характеризуется комплексной обработкой геометрических параметров элемента, реализуемой посредством интеграции семейства нодов Geometry, Element и CodeBlock [6]. В результате данной операции производится квантификация линейных размеров балки, включающая определение её протяженности и ширины.

Параллельно с геометрическим анализом осуществляется детерминирование высотного параметра конструкции. Данный процесс реализуется посредством нода Element.GetParameterValueByName, обеспечивающего получение абсолютных отметок верхней и нижней поверхностей балки. Финальная стадия определения высотного

параметра предусматривает вычисление разности между полученными значениями, что позволяет установить искомое значение высоты H.

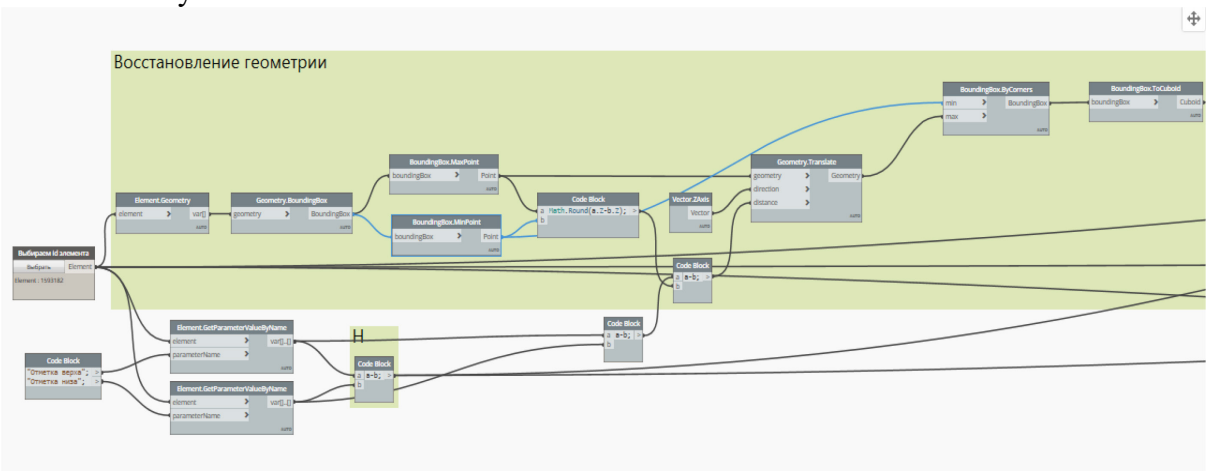

Рис. 4 – Восстановление геометрии балки
Fig. 4 – Restoration of beam geometry

По завершении этапа определения базовых геометрических характеристик осуществляется экстракция конфигурации нижней поверхности балки, что представляет собой финальный этап геометрического анализа конструкции. После успешной реализации данного этапа происходит переход к формированию арматурного каркаса, который включает в себя комплекс операций по проектированию и размещению арматурных элементов в соответствии с расчетными требованиями и нормативными документами (рис. 5).

Данный переход от геометрического анализа к проектированию армирования является логическим продолжением алгоритма, где на основе полученных геометрических параметров формируется арматурный каркас, обеспечивающий необходимую прочность и надежность конструкции [7].

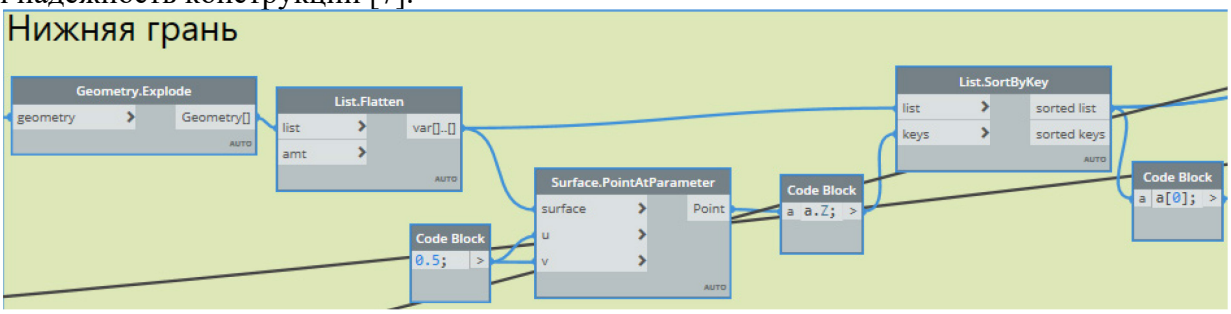

Рис. 5 – Нахождение нижней грани балки
Fig. 5 – Finding the lower edge of the beam

На начальном этапе проектирования поперечного армирования осуществляется определение направления распространения массива хомутов. Данная операция реализуется посредством нода Vector.ByTwoPoints, который обеспечивает генерацию вектора, определяющего ориентацию арматурного каркаса в пространстве (детализация процесса на рис. 6). Последующий этап характеризуется формированием массива хомутов, при котором осуществляется спецификация следующих параметров:

1. Межцентровое расстояние между элементами поперечного армирования.
2. Геометрические характеристики арматурных элементов.
3. Конфигурация размещения хомутов в пространстве.

В результате комплексной обработки указанных параметров формируется объект Geometry.Translate, содержащий полный набор геометрических примитивов, описывающих конфигурацию поперечного армирования конструкции. Данный объект представляет собой совокупность линий, определяющих пространственное расположение хомутов в арматурном каркасе.

Таким образом, представленная методология обеспечивает точное позиционирование элементов поперечного армирования с учетом проектных требований и нормативных предписаний.

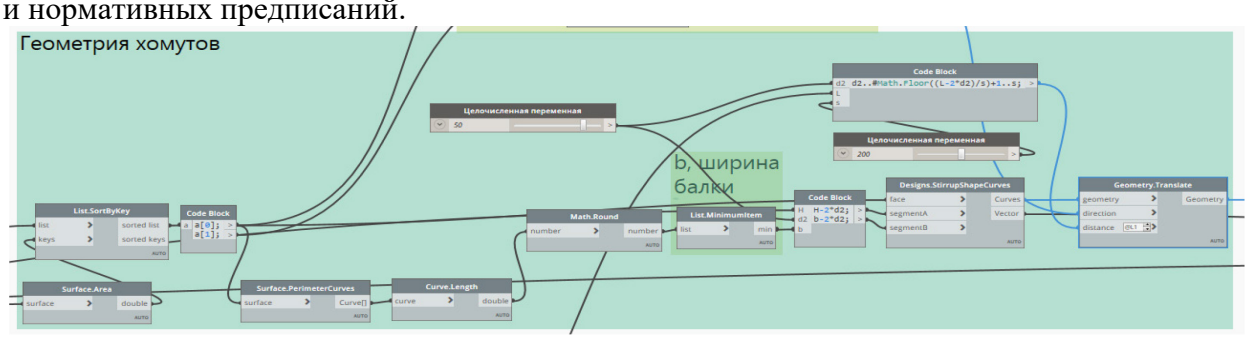

Рис. 6 – Создание геометрии хомутов
Fig. 6 – Creating clamp geometry

На следующем этапе проектирования осуществляется генерация геометрической модели продольного армирования, детальное представление которой представлено на рис. 7. В качестве входных данных для данного процесса используются следующие параметры:

1. Геометрическая модель конструктивного элемента (балки).
2. Расчетная длина анкеровки арматурных стержней.

В рамках данного этапа выполняется комплексная обработка указанных параметров с целью формирования пространственной конфигурации продольного армирования. Результатом данного процесса является создание объекта Geometry.Translate, который представляет собой совокупность геометрических примитивов, описывающих расположение продольных арматурных стержней в конструкции.

В сформированном объекте осуществляется точное позиционирование всех элементов продольного армирования, что обеспечивает:

1. Корректное размещение арматурных стержней в поперечном сечении балки.
2. Соблюдение расчетных длин анкеровки.
3. Соответствие проектной документации.

Таким образом, представленная методология обеспечивает точное моделирование продольного армирования с учетом всех конструктивных особенностей и расчетных требований.

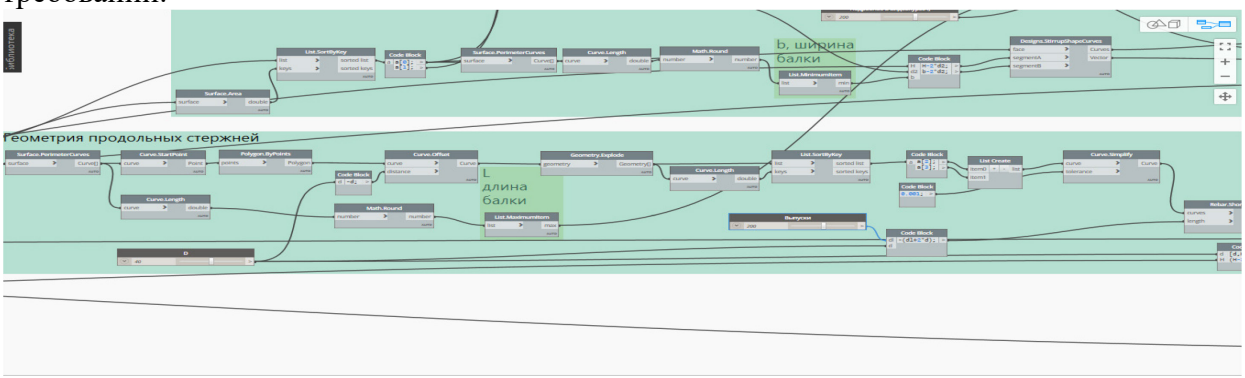

Рис.7 – Создание продольных стержней
Fig. 7 – Creation of longitudinal rods

В процессе синтеза арматурного каркаса осуществляется интеграция геометрических моделей продольных стержней и поперечных хомутов, формирующих комплексную пространственную структуру армирования. На данном этапе происходит консолидация ранее созданных геометрических примитивов в единую систему арматурного каркаса.

После формирования базовой геометрии арматурного каркаса производится настройка его конструктивных характеристик посредством специализированного дополнения DynamoForRebar и семейства Rebar. В рамках данного этапа осуществляется:

1. Селекция стиля армирования в соответствии с проектными требованиями.
2. Определение типоразмера арматурных элементов.

3. Спецификация методики анкеровки и конфигурации загиба арматурных стержней.

Конечная стадия алгоритма реализуется посредством Python.Script, который обеспечивает генерацию итогового арматурного каркаса с учетом всех заданных параметров и характеристик. Практическая реализация описанного алгоритма демонстрирует следующий порядок действий:

1. Создание базовой конструкции балки, показанной на рис.8, с заданными габаритными размерами (400x800 мм)
2. Идентификация конструктивного элемента посредством передачи его уникального идентификатора в скрипт
3. Запуск алгоритма обработки
4. Получение итогового результата (рис. 9).

Таким образом, представленная методология обеспечивает комплексную автоматизацию процесса проектирования арматурного каркаса балки с возможностью гибкой настройки конструктивных параметров и характеристик армирования.

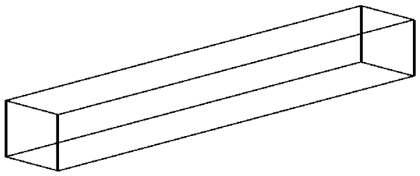

Рис. 8 – Начальный вид балки
Fig. 8 – Initial view of the beam

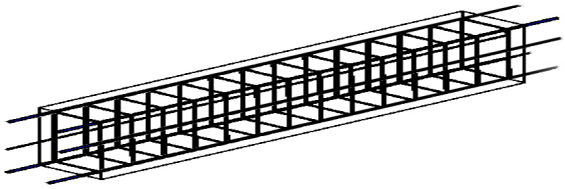

Рис. 9 – Вид балки после работы скрипта
Fig. 9 – View of the beam after running the script

В рамках настоящего исследования представляется систематизированный перечень базовых нодальных элементов, задействованных при реализации алгоритмического скрипта, направленного на проектирование арматурного каркаса балки. Далее следует детальная классификация и функциональное описание ключевых нодов, которые были применены в процессе разработки программного обеспечения для автоматизации процесса армирования конструктивного элемента [8].

Представленные в табл. 1 ноды демонстрируют комплексный подход к решению задачи проектирования арматурного каркаса и отражают специфику взаимодействия различных компонентов программного обеспечения в рамках единой алгоритмической структуры.

Таблица 1. Основные ноды использованные в скрипте армирования балки
Table 1. Main nodes used in the beam reinforcement script

Element.GetParameterValueByName	Получение параметров экземпляра по названию
BoundingBox.Minpoint	Получение нижней точки элемента
Code Block	Нод позволяющий писать свой код на Python.
List.FilterByBoolMask	Он по очереди перебирает все элементы списка и каждый из них помещает либо в слот in, либо в слот out
Categories	С его помощью можно взять ту или иную категорию.

Армирование колонн. На рис. 10 демонстрируется целостная структура алгоритмического скрипта, предназначенного для автоматизации процесса проектирования арматурного каркаса колонны.

Инициализация данного программного модуля осуществляется посредством загрузки уникального идентификатора конструктивного элемента (колонны), что является первостепенным этапом в реализации комплексного алгоритма армирования. Данная методология позволяет обеспечить точное позиционирование арматурных элементов в пространстве с учетом индивидуальных характеристик конкретного конструктивного элемента, что является критически важным для обеспечения требуемой прочности и надежности проектируемой конструкции.

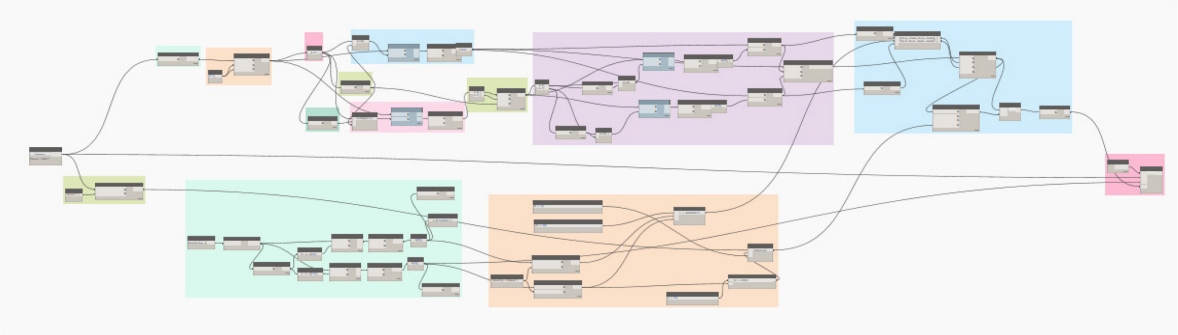

Рис. 10 – Общий вид скрипта армирования колонн
Fig. 10 – General view of the column reinforcement script

На начальном этапе проектирования осуществляется последовательное определение геометрических характеристик колонны посредством следующих операций:

1. Идентификация поверхностного массива колонны с использованием нода Element.Faces.
2. Определение центральных точек каждой грани через функционал Surfaces.PointAtParameter.
3. Селекция минимальных значений посредством List.MinimumItem для определения нижней торцевой грани.
4. Экстракция координат Z и определение центральной точки нижней торцевой грани.
5. Финальная обработка данных через Code Block, обеспечивающая вывод первого элемента сформированного списка.

Последующий этап характеризуется выполнением следующих процедур:

1. Определение центральных точек боковых граней колонны.
2. Выполнение операции проецирования указанных точек на плоскость нижнего торца конструкции.

Данная методология обеспечивает точное позиционирование ключевых точек арматурного каркаса с учетом пространственной конфигурации колонны, что является необходимым условием для корректного проектирования системы армирования. Представленный алгоритм, показанный на рис. 11 позволяет создать надежную основу для дальнейшего формирования арматурного каркаса, обеспечивая точное определение геометрических параметров и пространственного расположения арматурных элементов в конструкции.

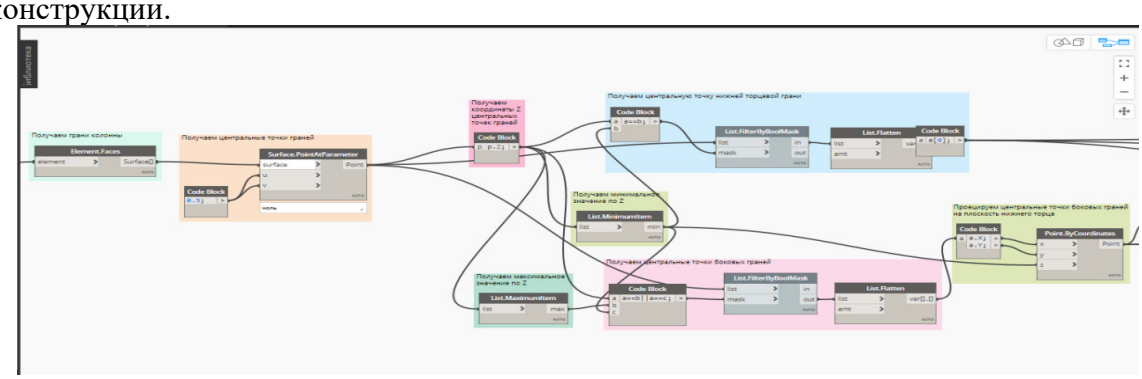

Рис. 11 – Начало скрипта. Получение опорных точек
Fig. 11 – Start of the script. Obtaining reference points

В целях оптимизации функционирования алгоритмического скрипта и обеспечения точности позиционирования арматурных элементов осуществляется построение локальной системы координат. Данный процесс реализуется посредством следующих последовательных операций: формирование двух базовых векторов с использованием нода Vector.ByTwoPoints на основе предварительно определенных точек; конструирование системы координат на основе полученных векторных элементов.

Представленный метод построения локальной системы координат позволяет:

1. Обеспечить корректное позиционирование арматурных элементов в пространстве.
2. Упростить алгоритмическую обработку геометрических параметров.
3. Повысить точность вычислений при проектировании арматурного каркаса.

Детальное описание процесса создания локальной системы координат представлено на рис. 12, где наглядно демонстрируется последовательность операций и их взаимосвязь в рамках общего алгоритма проектирования.

Таким образом, построение локальной системы координат является критически важным этапом в алгоритме армирования, обеспечивающим необходимую точность и эффективность последующих операций проектирования арматурного каркаса.

Рис. 12 – Построение местной системы координат
Fig. 12 – Construction of a local coordinate system

В рамках установленной локальной системы координат осуществляется процесс позиционирования точек размещения продольных арматурных стержней. Данная операция реализуется посредством комплексной интеграции следующих нодальных групп:

1. Vector - обеспечивает векторное моделирование и определение пространственного положения элементов.
2. Code Block - предоставляет возможности для программируемой обработки данных и реализации пользовательских алгоритмов.
3. List - осуществляет управление списками и массивами данных, необходимых для корректного позиционирования арматурных элементов.

На завершающем этапе данного процесса происходит формирование структурированного списка арматурных стержней посредством нода List.Create, который обеспечивает:

1. Систематизацию данных о пространственном расположении арматурных элементов.
2. Создание упорядоченной последовательности для последующей обработки.
3. Формирование базы данных для конструирования арматурного каркаса.

Представленный подход, продемонстрированный на рис.13 к позиционированию арматурных стержней позволяет обеспечить точное и систематическое размещение элементов армирования в соответствии с проектными требованиями и конструктивными особенностями колонны.

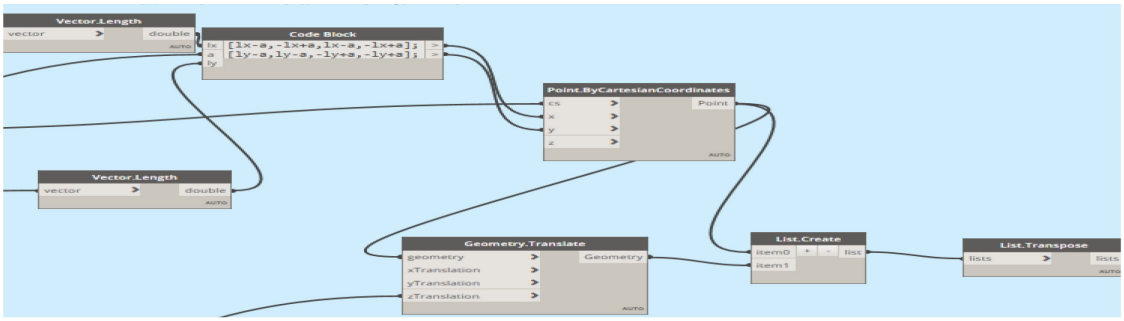

Рис. 13 – Построение точек стержней
Fig. 13 – Construction of rod points

В рамках реализации алгоритма проектирования арматурного каркаса осуществляется следующая последовательность операций:

1. Идентификация типов арматурных стержней производится посредством нода Element.Name (рис. 14), который обеспечивает: категоризацию арматурных элементов по их характеристикам; формирование структурированного перечня доступных типов арматуры.
2. Селекция необходимого типа арматурного стержня осуществляется через специализированное окно кода (рис. 15), что позволяет: обеспечить точный выбор требуемого типа арматуры; учесть специфические параметры конкретного конструктивного элемента.
3. На этапе параметризации арматурного каркаса производится: определение размеров выпусков арматурных стержней; расчет оптимальных расстояний между продольными стержнями; формирование геометрических параметров установки арматурных элементов.
4. Финальная стадия проектирования (рис. 16) характеризуется: интеграцией всех предварительно полученных данных в блок создания арматуры; автоматизированной генерацией арматурных стержней; формированием окончательного арматурного каркаса в соответствии с проектными требованиями.

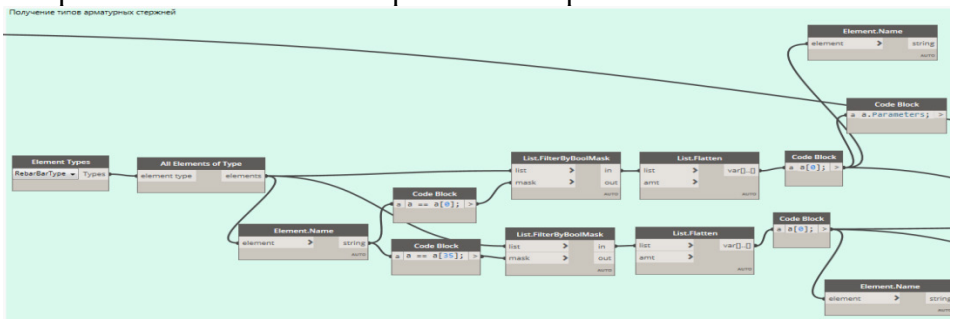

Рис. 14 – Получение типа арматуры
Fig. 14 – Obtaining the reinforcement type

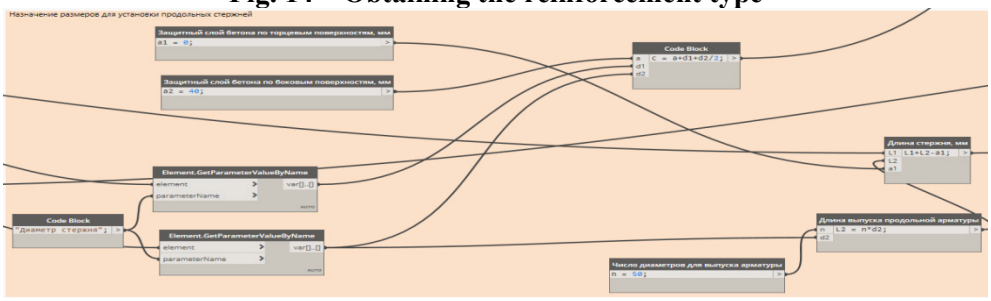

Рис.15 – Назначение размеров для установки продольных стержней
Fig. 15 – Assigning dimensions for installing longitudinal rods

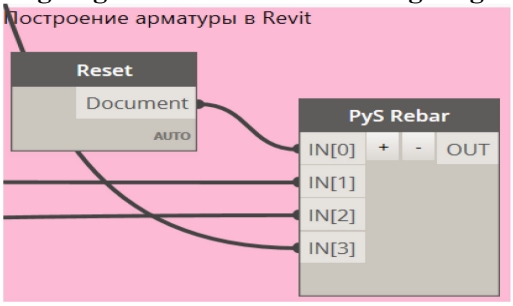

Рис.16 – Построение арматуры
Fig. 16 – Construction of reinforcement

Представленный алгоритм обеспечивает комплексную автоматизацию процесса проектирования арматурного каркаса, начиная от идентификации типов арматуры и заканчивая формированием окончательного конструктивного решения, показанного на рис. 17, 18.

В рамках практической верификации разработанного алгоритмического решения осуществляется следующая последовательность действий:

1. Формирование базовой конструктивной модели (рис. 17): создание колонны с заданными габаритными размерами 400х400 миллиметров; определение пространственного положения элемента в проектной модели.
2. Инициализация алгоритмического модуля (рис. 18): запуск программного скрипта для автоматизации процесса проектирования арматурного каркаса; активация последовательности вычислительных операций в соответствии с заданным алгоритмом.
3. Представленный метод практической реализации позволяет: проверить корректность функционирования разработанного алгоритма; оценить эффективность автоматизированного подхода к проектированию арматурного каркаса; подтвердить соответствие получаемых результатов проектным требованиям.

Таким образом, данный этап работы является критически важным для верификации работоспособности алгоритмического решения и его практической применимости в процессе проектирования железобетонных конструкций.

Рис. 17 – Начальный вид
Fig. 17 – Initial view

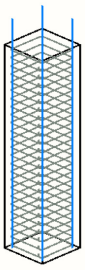

Рис. 18 – Армированная колонна
Fig. 18 – Reinforced column

Систематизированный перечень ключевых нодов, задействованных в алгоритмическом скрипте проектирования арматурного каркаса колонны представлен в табл. 2.

Таблица 2. Основные ноды использованные в скрипте армирования колонны
Table 2. Main nodes used in the column reinforcement script

Element.Name	Получение названия элемента из списка.
List.Create	Создание нового списка. На вход получаем элементы для заполнения списка.
Vector.ByTwoPoints	Создание вектора по двум точкам
List.MinimumItem	Нахождение минимального элемента в списке.
Element.GetJoinedElements	Получение объединенных элементов, для их выбора.
List.Flatten	Удаляет уровни данных из структуры данных. Используется, когда иерархии данных не нужны для работы.
Element.GetParameterNameByValue	Получение имени параметра по значению.

Армирование плит перекрытий. На рис. 19 представлено систематизированное графическое отображение комплексной алгоритмической структуры, обеспечивающее всестороннее понимание принципов функционирования программного скрипта.

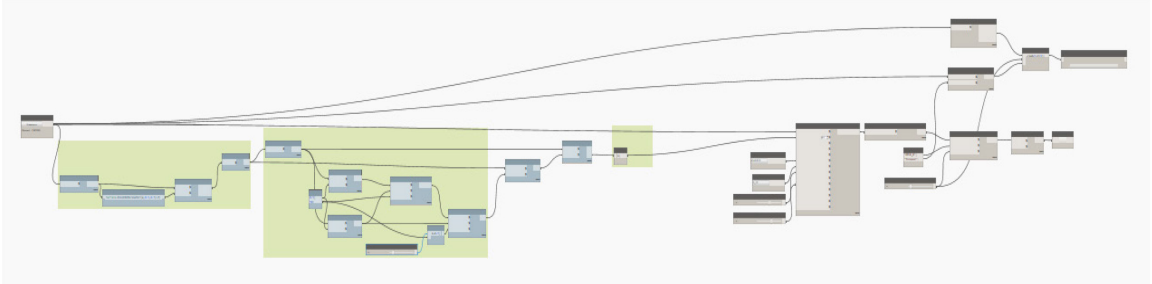

Рис. 19 – Общий вид скрипта
Fig. 19 – General view of the script

В процессе алгоритмической обработки конструктивного элемента осуществляется последовательная реализация ряда операций. Инициализация процесса проектирования начинается с импортирования идентификатора конструктивного элемента (плиты

перекрытия) в рабочую среду, после чего элемент передается в нод Element.Faces для получения его геометрических характеристик. Далее происходит определение пространственных параметров, включающее идентификацию всех граней элемента и локализацию верхней грани перекрытия как ключевого конструктивного элемента.

После этого выполняется сортировка и фильтрация данных посредством применения нода Code Block для программируемой обработки информации и использования List.SortByKey для структурирования данных по определенным критериям. В результате происходит выделение верхней грани перекрытия посредством List.LastItem как конечного элемента отсортированного списка.

Представленный алгоритм обеспечивает точную идентификацию конструктивного элемента, корректное определение его пространственного положения, систематизацию геометрических характеристик и формирование структурированной базы данных для последующего проектирования арматурного каркаса.

Таким образом, описанная последовательность операций позволяет создать надежную основу, показанную на рис. 20, для дальнейшего моделирования арматурного каркаса перекрытия с учетом его конструктивных особенностей и пространственной ориентации.

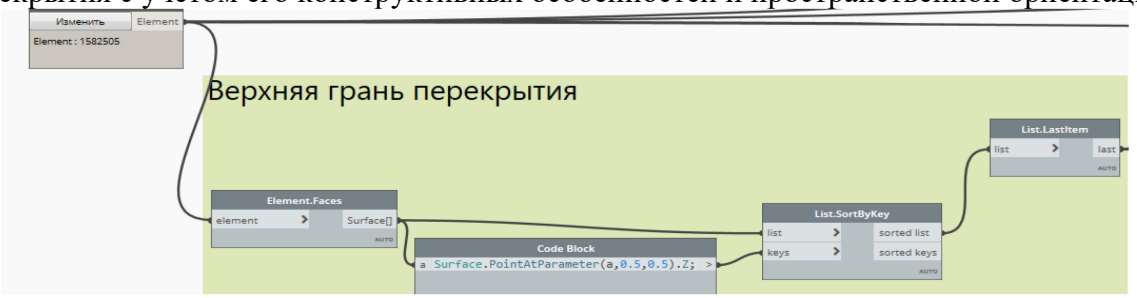

Рис. 20 – Верхняя грань перекрытия
Fig. 20 – Upper edge of the ceiling

В процессе алгоритмической обработки осуществляется генерация лучевых элементов посредством интеграции нодальных групп Curve и Geometry. Результатом данной операции является формирование нода Line.ByStartPointDirectionLength, который обеспечивает создание линейных элементов с заданными параметрами начальной точки, направления и длины. Визуализация данной алгоритмической последовательности представлена на иллюстративном материале под номером 21, где наглядно демонстрируется процесс генерации лучевых элементов и их параметрическая настройка.

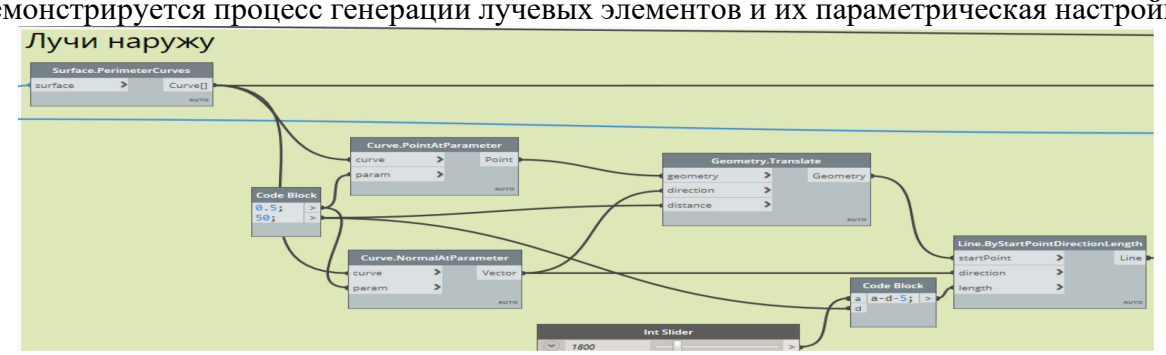

Рис. 21 – Нахождение лучей
Fig. 21 – Finding the rays

В рамках реализации алгоритма проектирования осуществляется генерация геометрической модели арматурного каркаса. Данный процесс предусматривает последовательное выполнение следующих операций:

1. Формирование геометрического описания арматурного каркаса с учетом конструктивных особенностей проектируемого элемента.
2. Имплементация параметров арматурных элементов посредством специализированного программного пакета DynamoForRebar, включающая: определение

типологических характеристик арматурных стержней; спецификацию методов их гибки и формовки; установление геометрических параметров арматурного каркаса.

3. Генерация трехмерной модели арматурных стержней в соответствии с заданными параметрами и конструктивными требованиями.

Визуализация процесса создания арматурного каркаса представлена на рис. 22, где демонстрируется последовательность операций по формированию геометрической модели арматурного каркаса и генерации арматурных элементов.

Таким образом, описанная алгоритмическая последовательность обеспечивает комплексную реализацию процесса проектирования арматурного каркаса с учетом всех необходимых конструктивных и технологических параметров.

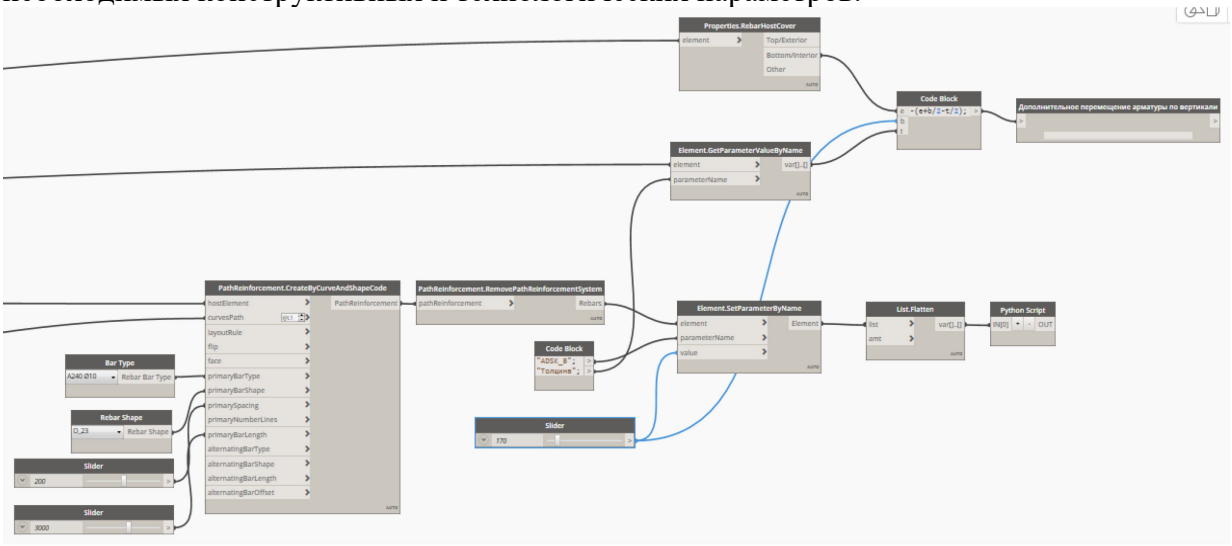

Рис. 22 – Создание стержней
Fig. 22 – Creating rods

Функционирование программного алгоритма осуществляется посредством последовательности операций.

В первую очередь производится инициализация конструктивного элемента - создание базового объекта перекрытия в рабочей среде проектирования. После этого активируется программный скрипт, отвечающий за процесс армирования. На этапе параметрической настройки осуществляется спецификация характеристик армирования и селекция типологических параметров арматурных стержней в соответствии с проектными требованиями.

По завершении конфигурации параметров производится повторный запуск скрипта, инициирующий процесс генерации арматурного каркаса. Результатом успешной реализации алгоритма является формирование комплексного конструктивного элемента - армированного перекрытия, соответствующего заданным проектным характеристикам и нормативным требованиям.

Визуализация конечного результата представлена на рис. 23-24, где демонстрируется итоговый конструктивный элемент с интегрированным арматурным каркасом.

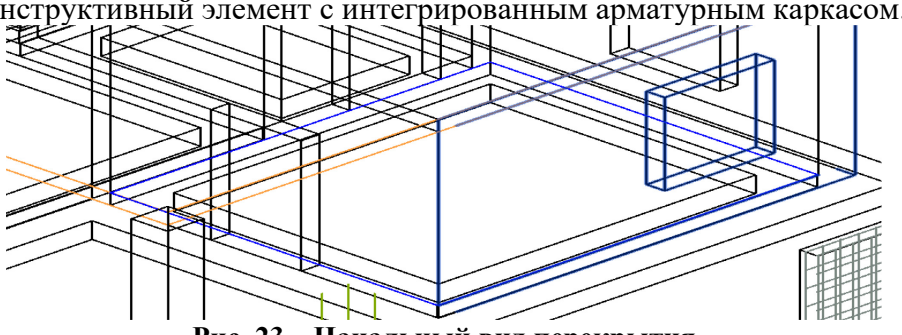

Рис. 23 – Начальный вид перекрытия
Fig. 23 – Initial view of the ceiling

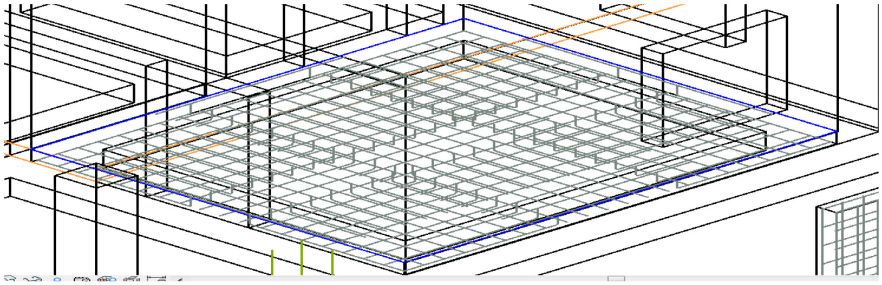

Рис. 24 – Армированное перекрытие
Fig. 24 – Reinforced floor

Таким образом, описанная последовательность операций обеспечивает полную реализацию алгоритма проектирования армированного перекрытия с возможностью адаптации под различные проектные требования и конструктивные решения.

Систематизированный перечень ключевых нодов, задействованных в алгоритмическом скрипте проектирования арматурного каркаса перекрытия представлен в табл. 3.

Таблица 3. Основные ноды, использованные в скрипте армирования перекрытия
Table 3. Main nodes used in the floor reinforcement script

Element.Faces	Получение граней элемента
List.SortByKey	Сортировка списка по ключу. На выходе полуаем отсортированный список
List.LastItem	Возвращает последний элемент списка
Integer Slider	Ползунок для ввода целочисленной переменной
Curve.NormalAtParameter	Получение внешнего контура грани

Вывод. В результате проведенного исследования были детально изучены и описаны алгоритмы автоматизированного проектирования систем армирования железобетонных конструкций в программном комплексе Revit-Dynamo. Разработанные методики позволяют эффективно создавать арматурные каркасы для балок, колонн и плит перекрытий с учетом их геометрических параметров, и конструктивных особенностей.

Практическая ценность работы подтверждается возможностью непосредственного применения описанных алгоритмов в проектной деятельности. Использование Dynamo для автоматизации процесса армирования существенно повышает производительность труда проектировщиков и обеспечивает высокое качество проектной документации.

Дальнейшее развитие темы может быть направлено на совершенствование существующих алгоритмов, расширение их функциональности и адаптацию под различные типы железобетонных конструкций. Полученные результаты могут служить основой для создания более сложных автоматизированных систем проектирования арматурных конструкций.

Библиографический список:

1. Т.А. Юрошева, А.В. Калиниченко, В.Г. Макиев. Алгоритм проектирования несущих конструкций многоэтажного здания с использованием среды визуального программирования DYNAMO STUDIO. Вестник Дагестанского государственного технического университета. Технические науки. 2022; 49(4):126-133. DOI:10.21822/2073-6185-2022-49-4-126-133
2. Смакаев Р.М., Низина Т.А. Автоматизация задач проектирования с помощью среды визуального программирования DYNAMO STUDIO [Электронный ресурс] // Огарев-online. - 2020. - № 3.
3. Поддорогина Е.А., Шумилов К.А., Мазинг А.А. Разработка строительных объектов в DYNAMO REVIT // BIM-моделирование в задачах строительства и архитектуры: материалы Всерос. науч.-практ. конф., 29-30 марта 2018 г. – СПб: СПбГАСУ, 2018. – С. 177–182.
4. Калиниченко А.В., Аликов А.Ю., Юрошева Т.А. Разработка программного модуля автоматической коррекции чертежей, полученных в результате обмена графическими данными между CAD-системами, на примере мебельного производства. Вестник Дагестанского государственного технического университета. Технические науки. 2024;51(4):80-86. https://doi.org/10.21822/2073-6185-2024-51-4-80-86
5. Официальный сайт поддержки продукта Dynamo Studio [Электронный ресурс]. – Режим доступа: https://www.dynamoprimer.com/index.html

6. СП 63.13330.2018. Бетонные и железобетонные конструкции. Основные положения. Актуализированная редакция. СНиП 52-01-2003 / Минрегион России. М., 2012.
7. Юрошева Т.А., Калиниченко А.В. Разработка программного модуля для автоматизации процесса расчёта водопотребления и водоотведения с использованием среды разработки DYNAMO STUDIO. Вестник Дагестанского государственного технического университета. Технические науки. 2025;52(1):173-182. https://doi.org/10.21822/2073-6185-2025-52-1-173-182
8. Агаханов Э.К., Агаханов М.К., Труфанова Е.В., Джабраилов А.З. Моделирование каркаса уникального здания параметрической архитектуры на основе применения поверхности «цилиндр-цилиндр». Вестник Дагестанского государственного технического университета. Технические науки. 2025;52(1):193-201. https://doi.org/10.21822/2073-6185-2025-52-1-193-201

References:

1. T.A. Yurosheva, A.V. Kalinichenko, V.G. Makiev. An algorithm for designing load-bearing structures of a multi-storey building using the DYNAMO STUDIO visual programming environment. Bulletin of the Dagestan State Technical University. Technical sciences.
2. Smakaev R.M., Nizina T.A. Automation of design tasks using DYNAMO STUDIO visual programming [Electronic resource] // Ogarev-online. - 2020. - No. 3.
3. Poddorogina E.A., Shumilov K.A., Mazing A.A. Development of construction objects in DYNAMO REVIT // BIM-modeling in construction tasks and architecture: materials of All-Russian Scientific and Practical Conference, March 29-30, 2018 - St. Petersburg: SPbGA-SU, 2018. – P. 177-182.
4. Kalinichenko A.V., Alikov A.Yu., Yurosheva T.A. Development of a software module for automatic correction of drawings obtained as a result of graphic data exchange between CAD systems, using the example of machine manufacturing. Herald of Dagestan State Technical University. Technical Sciences. 2024;51(4):80-86. https://doi.org/10.21822/2073-6185-2024-51-4-80-86
5. Official website of Dynamo Studio product support [Electronic resource]. – Access mode: https://www.dynamoprimer.com/index.html
6. SP 63.13330.2018. Concrete and reinforced concrete structures. The main provisions. Updated version. SNiP 52-01-2003 / Ministry of Regional Development of Russia. Moscow, 2012.
7. Yurosheva T.A., Kalinichenko A.V. Development of a software module for automating the process of calculating water consumption and water disposal using the DYNAMO STUDIO development environment. Herald of Dagestan State Technical University. Technical Sciences. 2025;52(1):173-182. (In Russ) https://doi.org/10.21822/2073-6185-2025-52-1-173-182
8. Agakhanov E.K., Agakhanov M.K., Trufanova E.V., Dzhabrailov A.Z. Modeling the frame of a unique building of parametric architecture based on the application of the "cylinder-cylinder" surface. Herald of Dagestan State Technical University. Technical Sciences. 2025;52(1):193-201. (In Russ) https://doi.org/10.21822/2073-6185-2025-52-1-193-201

Сведения об авторах:

Татьяна Александровна Юрошева, кандидат технических наук, доцент; кафедра «Информационных технологий и систем»;trini-83@yandex.ru; ORCID: 0009-0002-1365-8787

Исупов Никита Сергеевич, аспирант, Институт строительства и архитектуры; isupovn98@gmail.com; ORCID: 0000-0002-4301-3202

Кудзиев Шота Русланович, магистр, kudziev.mega.ru@mail.ru;

Акоева Евгения Николаевна, старший преподаватель, кафедра «Информационных технологий и систем»; evgeniya-akoeva@mail.ru

Information about the authors:

Tatyana A. Yurosheva, Cand.Sci.(Eng.), Assoc. Prof.; Department of Information Technologies and Systems; trini-83@yandex.ru; ORCID: 0009-0002-1365-8787

Isupov Nikita Sergeevich, Postgraduate student, Institute of Construction and Architecture; isupovn98@gmail.com; ORCID: 0000-0002-4301-3202

Kudziev Shota Ruslanovich, Master,

Akoeva Evgeniya Nikolaevna, Senior lecturer, Department of Information Technologies and Systems; evgeniya-akoeva@mail.ru

Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/Received 02.05.2025.

Одобрена после рецензирования/Reviced 04.07.2025.

Принята в печать/ Accepted for publication 15.07.2025.