

Методика анализа и интерпретации результатов расчета SCAD++ с помощью внешнего постпроцессора

Станислав Федорович Дьяков, Сергей Александрович Агафонов

Санкт-Петербургский политехнический университет Петра Великого (СПбПУ);
г. Санкт-Петербург, Россия

АННОТАЦИЯ

Введение. Одним из ключевых аспектов решения любых задач численными методами является интерпретация полученных результатов с целью их дальнейшего использования. Современные расчетные комплексы, реализующие метод конечных элементов, имеют в своем составе так называемый постпроцессор — модуль, предназначенный для облегчения визуализации и интерпретации результатов выполненных вычислений. Однако, каким бы продвинутым не был встроенный в программный комплекс постпроцессор, его функционал может быть недостаточен для решения конкретной инженерной задачи. Один из способов оптимизации соответствующих этапов работы — применение сторонних программных продуктов и разработка самописных пользовательских расширений.

Материалы и методы. Определение требуемых параметров армирования монолитных железобетонных конструкций сложной геометрической формы выполнено в вычислительном комплексе SCAD++, но дальнейшая их обработка встроенными средствами затруднительна. В рамках предложенной методики интерпретации и анализа полученных в SCAD++ результатов армирования монолитной железобетонной плиты с широкими балками применялись свободно распространяемый программный продукт Gmsh и пользовательский скрипт на языке программирования Python.

Результаты. Реализация предложенного алгоритма обработки данных, полученных из SCAD++, позволила преодолеть ограничения стандартных средств проектно-вычислительного комплекса в части визуализации результатов выполненных расчетов, а также сформировать исходные сведения для их дальнейшего использования в процессе инженерно-строительного проектирования.

Выводы. Предложенная методика удобна для применения в инженерной практике и предоставляет широкие возможности для эффективного анализа результатов расчетов, выполненных методом конечных элементов, и их дальнейшей обработки. Ее использование позволит повысить эффективность принимаемых решений, оптимизировать процесс проектирования геометрически сложных конструкций и их элементов, а также оптимизировать материальные затраты на приобретение программного обеспечения.

КЛЮЧЕВЫЕ СЛОВА: SCAD, МКЭ, Gmsh, Python, скрипт, изополя, постпроцессор

ДЛЯ ЦИТИРОВАНИЯ: Дьяков С.Ф., Агафонов С.А. Методика анализа и интерпретации результатов расчета SCAD++ с помощью внешнего постпроцессора // Вестник МГСУ. 2025. Т. 20. Вып. 12. С. 1853–1866. DOI: 10.22227/1997-0935.2025.12.1853-1866

Автор, ответственный за переписку: Сергей Александрович Агафонов, agafonov_sa@spbstu.ru.

Method of analysis and interpretation of SCAD++ calculation results using external postprocessor

Stanislav F. Diakov, Sergei A. Agafonov

Peter the Great St. Petersburg Polytechnic University (SPbPU); St. Petersburg, Russian Federation

ABSTRACT

Introduction. One of the crucial aspects of solving various mechanical problems using numerical methods is the interpretation of the obtained results for their further use. Modern computational complexes implementing the finite element method have a so-called postprocessor — a module designed to ease visualization and interpretation of the results of performed calculations. However, no matter how advanced the postprocessor built into the CAE programme is, its functionality may be insufficient for solving a particular engineering task. One of the ways to optimize the relevant stages of work is to use third-party software products and develop custom extensions.

Materials and methods. Determination of the required parameters of reinforcement of monolithic reinforced concrete structures of complex geometric shape is performed in the SCAD++ computer complex, but their further processing by built-in tools is challenging. Within the framework of the proposed methodology, interpretation, and analysis of the results of reinforcement of a monolithic reinforced concrete slab with wide beams obtained in SCAD++ were performed using the freeware software product Gmsh and a user script in the Python programming language.

Results. The implementation of the proposed algorithm of data processing obtained from SCAD++ allowed to overcome the limitations of standard means of CAE system in terms of visualization of the results of the performed calculations. As well as to create initial data for their further use in engineering and construction design.

Conclusions. The proposed methodology is convenient for application in engineering practice and provides wide opportunities for effective analysis of the results of calculations performed by the finite element method and their further processing. Its use will increase the efficiency of decisions taken, improve the design process of geometrically complex structures and their elements, as well as optimize the financial costs of software acquisition.

KEYWORDS: SCAD, FEM, Gmsh, Python, script, postprocessor

FOR CITATION: Diakov S.F., Agafonov S.A. Method of analysis and interpretation of SCAD++ calculation results using external postprocessor. *Vestnik MGSU* [Monthly Journal on Construction and Architecture]. 2025; 20(12):1853-1866. DOI: 10.22227/1997-0935.2025.12.1853-1866 (rus.).

Corresponding author: Sergei A. Agafonov, agafonov_sa@spbstu.ru.

ВВЕДЕНИЕ

К современному строительному производству предъявляются требования по реализации решений, способных обеспечить эффективность и технологичность на всех этапах жизненного цикла конечного продукта — объекта капитального строительства. При этом безусловной является необходимость обеспечения надежности и долговечности конструкций и их элементов.

Технология монолитного железобетонного строительства прочно занимает свою нишу в строительной индустрии, позволяя реализовывать сложные объемно-планировочные решения и формировать актуальный архитектурный облик зданий и сооружений. Ребристые монолитные железобетонные плиты широко применяются в практике строительства благодаря уменьшению собственного веса конструкции при сохранении несущей способности в сравнении со сплошными плитами перекрытий и покрытий [1–3]. Однако актуальным остается вопрос не только выбора подходящей расчетной схемы для определения напряженно-деформированного состояния (НДС) таких конструкций [4–6], но зачастую и интерпретации результатов автоматизированного расчета, от которой зависит эффективность принимаемых решений.

Основа таких решений закладывается на этапе архитектурно-строительного проектирования, в процессе которого используются различные типы программных комплексов: CAE (Computer-aided engineering), CAD (Computer-aided design), BIM (Building Information Modeling). BIM-системы дают возможность сформировать цифровую информационную модель, предполагающую увязку геометрических, физических и функциональных (технологических) характеристик строительного объекта и отдельных его элементов [7–9]. CAD-программы реализуют автоматизацию проектирования, но в меньшей степени по сравнению с BIM. CAE программные пакеты позволяют проводить инженерный анализ: расчеты и симуляцию различных физических процессов. Интеграция таких программных продуктов помогает решать задачи высокой сложности в рамках нового строительства или реконструкции существующих объектов [10–13].

Одна из широко распространенных в инженерной практике систем автоматизации инженерных строительных расчетов (CAE) — вычислительный

комплекс SCAD++¹, реализованный в виде интегрированной системы прочностного анализа и проектирования конструкций на основе метода конечных элементов (МКЭ). Идеология метода предполагает представление искомых перемещений внутри плоского или объемного конечного элемента (КЭ) через различные упрощенные зависимости. При этом погрешность вычислений напрямую зависит от шага сетки элементов, т.е. их размера и количества.

Эффективность расчета строительных конструкций с использованием МКЭ базируется на нескольких факторах: выборе расчетной схемы, корректном задании выбранной расчетной схемы в вычислительном комплексе, в правильной интерпретации полученных результатов. Наглядность представления результатов расчета значительно упрощает работу инженера-проектировщика, позволяя в том числе «отловить» ошибки, которые могли быть допущены на начальных стадиях работы. Кроме того, расширяются возможности для принятия более оптимальных и обоснованных решений с учетом требований по материалоемкости и экологичности проектируемых конструкций [14–17].

Для визуального представления результатов определения НДС конструкции в большинстве вычислительных комплексов используется постпроцессор МКЭ, который реализован и в SCAD++. Однако в реальной проектной деятельности нередко возникают ситуации, когда инструментарий, предлагаемый разработчиком программного обеспечения (ПО), недостаточен для решения поставленной задачи. В таком случае существует возможность использования программного интерфейса приложения API (Application Programming Interface) для обмена данными. В SCAD++ реализована возможность расширения недостающего функционала посредством написания пользовательских приложений на языке JScript, взаимодействующих с ядром программного комплекса.

В настоящей работе предложена методика повышения эффективности анализа и интерпретации результатов расчета при недостатке стандартного функционала в постпроцессоре SCAD++.

МАТЕРИАЛЫ И МЕТОДЫ

При расчете и конструировании — подборе арматуры в плитах простого в плане (по форме близко-

¹ URL: <https://scadoffice.ru>

го к прямоугольнику) здания сложностей с анализом результатов не возникает: можно легко определить интенсивность армирования по основному AS1 (армирование по X) и ортогональному ему AS3 (армирование по Y) направлениям.

Ситуация несколько усложняется, если в конструкции плиты присутствуют широкие ребра, необходимость организации которых часто возникает в практике проектирования. Расчетная схема таких плит может быть построена с применением объемных КЭ, однако такая постановка задачи требует значительных вычислительных мощностей. В инженерной практике такие плиты удобно моделировать, используя плоские конечные элементы — оболочка (тип 44), задавая различные жесткостные характеристики соответствующим участкам конструкции. Фрагмент такого перекрытия представлен на рис. 1. Очевидно, что в отличие от рядового (тонкого) участка плиты в ребре (толстый участок плиты) основная нижняя продольная рабочая арматура должна располагаться вдоль оси ребра. В простых плитах при ортогональном расположении ребер требуемая интенсивность арматуры «собирается» в том направлении (AS1 или AS3), которое параллельно оси рассматриваемого ребра.

Однако, если ось широкого ребра не совпадает ни с одной из основных осей, наглядно визуализировать необходимую интенсивность армирования затруднительно: интенсивность, к примеру, требуе-

мого нижнего армирования будет разделена между направлениями X и Y (рис. 2). В SCAD++ присутствует инструментарий, позволяющий менять направление выдачи усилий, направленный как раз на решение возникающей проблемы (рис. 2, b). При этом для радиального направления балок задача смены направления выдачи усилий, хотя и трудоемкая (можно представить, что таких перекрытий не одно, а несколько, например, в нетиповых секциях одного комплекса), но выполнимая. Для кольцевого направления такая задача превращается в сверхтрудную, требующую больших временных затрат.

Альтернативным смене направления выдачи усилий вариантом определения требуемой площади арматуры в элементах, не сонаправленных основным осям, является суммирование подобранных программным комплексом значений армирования AS1 и AS3. Но необходимый для этого функционал в SCAD++ отсутствует, поэтому для решения задачи предлагается связка: пользовательское расширение собственной разработки для SCAD++ и программа Gmsh² [18–20]. Методика такой обработки данных представлена далее.

² Gmsh: A Three-Dimensional Finite Element Mesh Generator with Built-in Pre- and Post-Processing Facilities. URL: <https://gmsh.info>

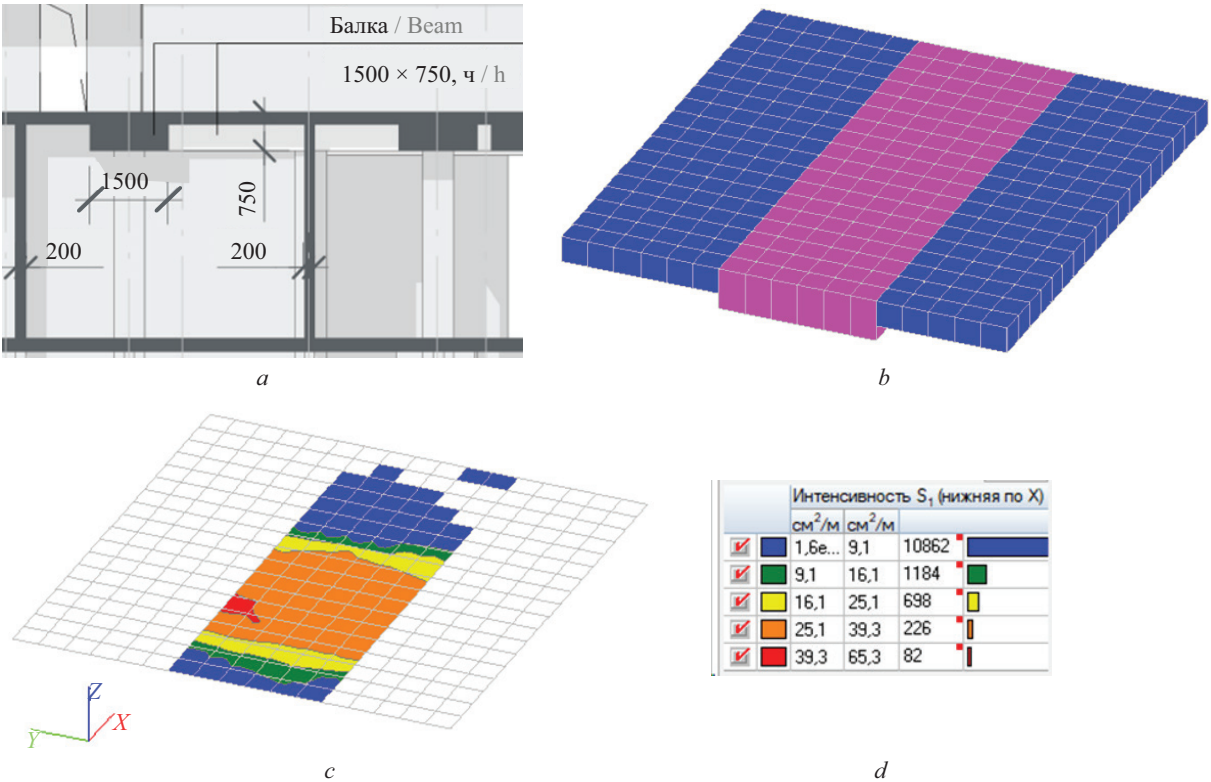


Рис. 1. Перекрытие, выполненное по широким балкам: a — сечение; b — общий вид сетки конечных элементов; c — изополя нижнего продольного армирования по X ; d — интенсивность нижнего продольного армирования по X

Fig. 1. Plate with wide beams: a — section; b — general view of the finite element mesh; c — isofields of bottom longitudinal reinforcement along X ; d — intensity of bottom longitudinal reinforcement along X

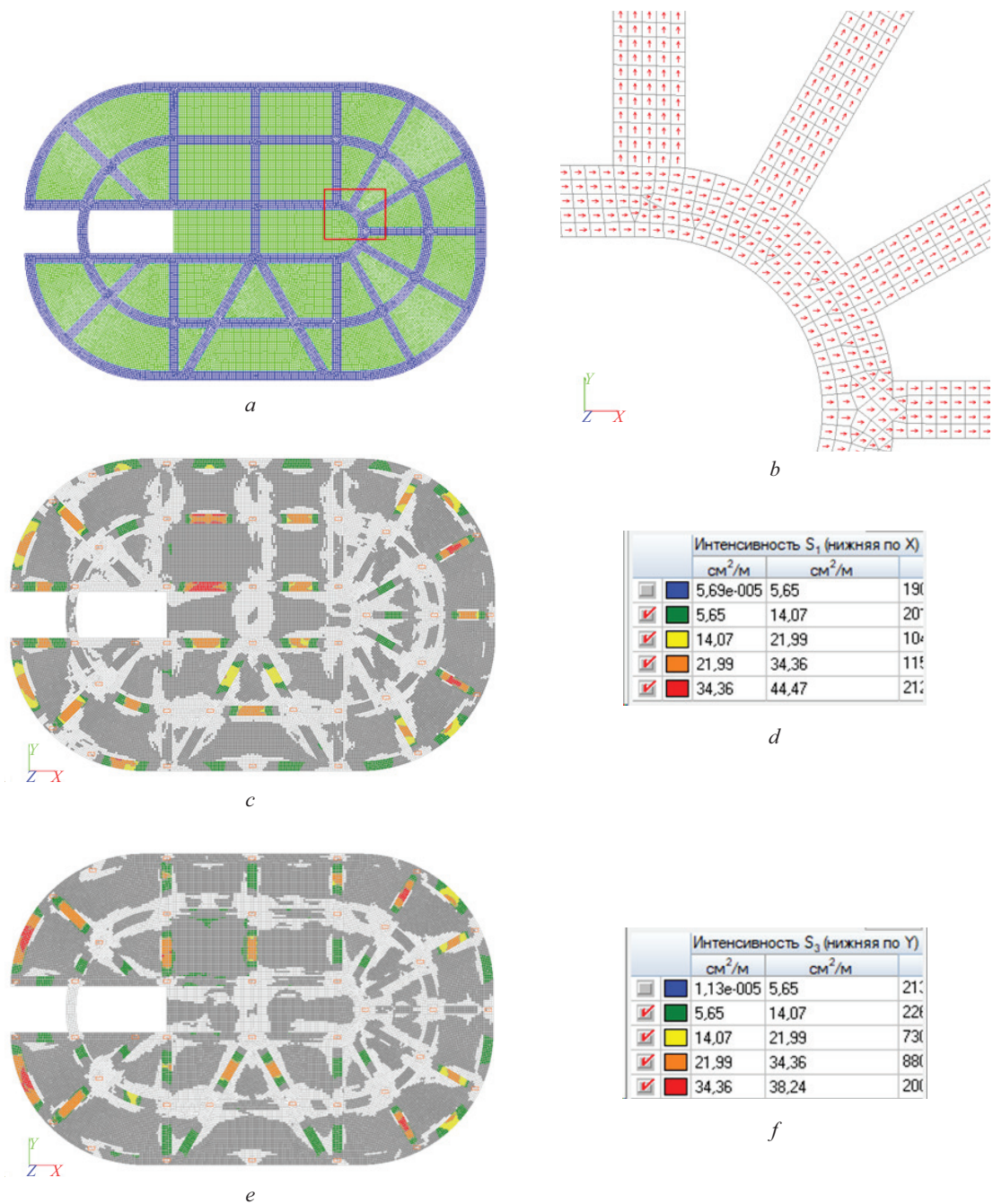


Рис. 2. Перекрытие, выполненное по широким балкам, расположенным не параллельно основным осям: *a* — общий вид; *b* — укрупненный фрагмент перекрытия с измененным направлением выдачи усилий; *c* — изополя нижнего продольного армирования по *X*; *d* — интенсивность нижнего продольного армирования по *X*; *e* — изополя нижнего продольного армирования по *Y*; *f* — интенсивность нижнего продольного армирования по *Y*

Fig. 2. Plate with wide beams that are not parallel to the main axes: *a* — general view; *b* — scaled fragment of the plate with the specified directions of the stress output; *c* — isofields of bottom longitudinal reinforcement along *X*; *d* — intensity of bottom longitudinal reinforcement along *X*; *e* — isofields of bottom longitudinal reinforcement along *Y*; *f* — intensity of bottom longitudinal reinforcement along *Y*

Свободно распространяемый программный продукт Gmsh предназначен для автоматической генерации трехмерных сеток КЭ, имеющий в своем составе собственное CAD-ядро и удобный постпроцессор, который дает возможность визуализировать данные. Приложение широко применяется для подготовки ис-

ходных сведений для решения задач в конечно-элементной постановке, зачастую позволяя выполнить более качественное разбиение задаваемых контуров на КЭ в сравнении со стандартными инструментами, включенными в вычислительные комплексы, реализующие МКЭ [21, 22]. Взаимодействие с програм-

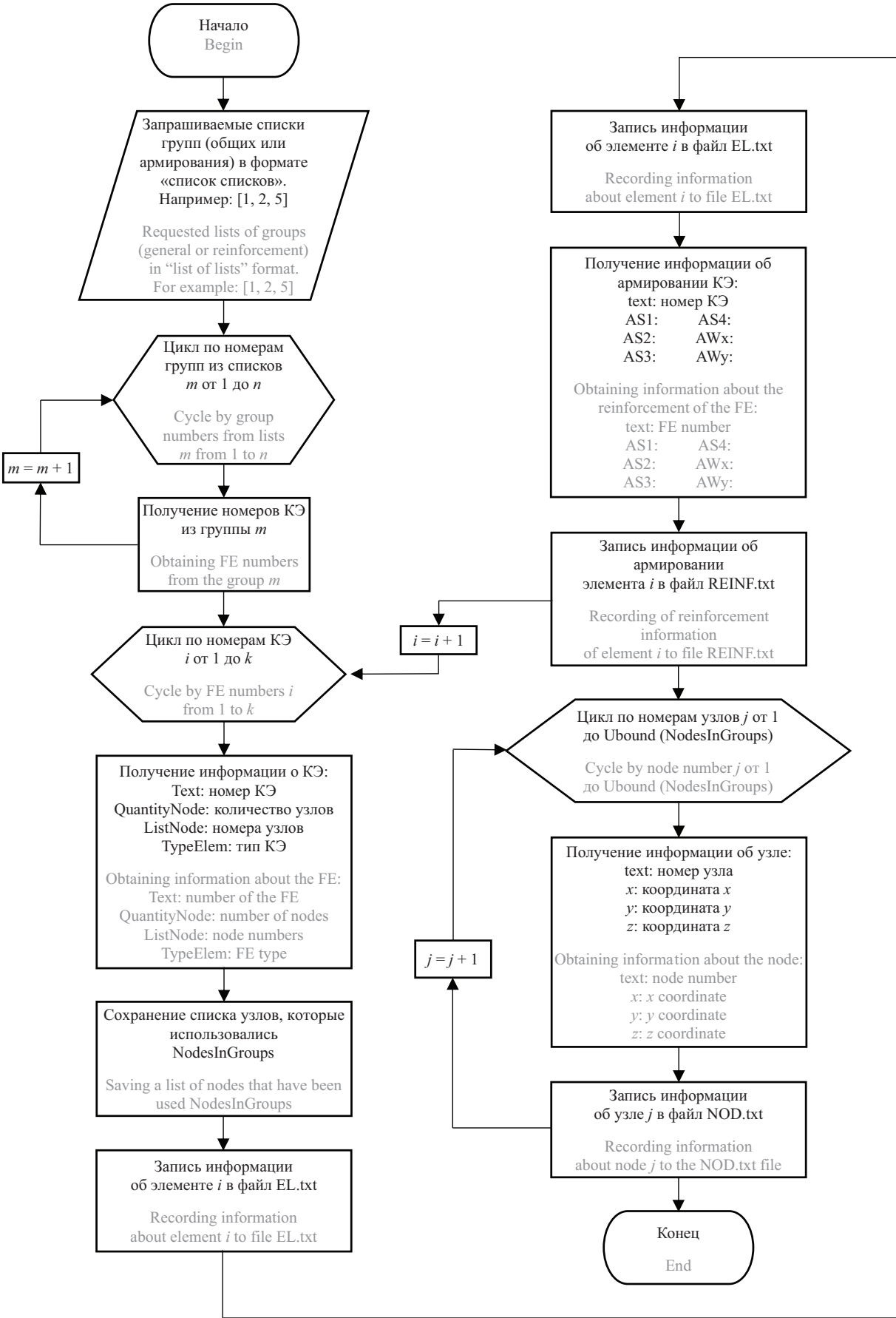


Рис. 3. Блок-схема алгоритма работы пользовательского расширения для SCAD++

Fig. 3. Block diagram of the algorithm of user extension for SCAD++

{«Text»:22347,«x»:-827.6886875,«y»:-574.978625,«z»:0,«Flag»:0}	<i>a</i>
{«Text»:22348,«x»:-827.5070000000001,«y»:-575.1644375,«z»:0,«Flag»:0}	
{«Text»:34964,«QuantityNode»:4,«ListNode»:[34830,34803,34829,34798],«TypeElem»:44,... »NumBed»:0,«Group»:1}	<i>b</i>
{«Text»:34965,«QuantityNode»:4,«ListNode»:[34799,34831,34794,34825],«TypeElem»:44,... »NumBed»:0,«Group»:1}	
{«Text»:34803,«AS1»:0,«AS2»:1.6589594174260138,«AS3»:0,«AS4»:1.9699072431291857,«AwX»:null,«AwY»:null}	<i>c</i>
{«Text»:34804,«AS1»:0,«AS2»:0,«AS3»:1.60825147507554,«AS4»:0,«AwX»:null,«AwY»:null}	

Рис. 4. Фрагменты строк с данными, экспортируемыми из SCAD++: *a* — из файла NOD.txt; *b* — из файла EL.txt; *c* — из файла REINF.txt

Fig. 4. Fragments of lines with data exported from SCAD++: *a* — from file NOD.txt; *b* — from file EL.txt; *c* — from file REINF.txt

мой Gmsh возможно как посредством графического интерфейса GUI (Graphical User Interface), так и с помощью встроенного языка путем формирования файла .geo. Кроме того, функционал Gmsh предоставляет пользователю инструменты для дополнительной обработки и визуального представления результатов

\$MeshFormat 2.2 0 8 \$EndMeshFormat \$Nodes 6 1 0.0 0.0 0.0 2 1.0 0.0 0.0 3 1.0 1.0 0.0 4 0.0 1.0 0.0 5 2.0 0.0 0.0 6 2.0 1.0 0.0 \$EndNodes \$Elements 2 1 3 2 99 2 1 2 3 4 1 2 3 4 2 3 2 99 2 2 5 6 3 2 5 6 3 \$EndElements \$NodeData 1 «A scalar view» 1 0.0 3 0 1 6 1 0.0 2 0.1 3 0.2 4 0.0 5 0.2 6 0.4 \$EndNodeData \$ElementData 1 «Layer Name» 1 0.0 3 0 1 34986 1 90.58 2 72.04 3 70.52 \$EndElementData	six mesh nodes: node #1: coordinates (0.0, 0.0, 0.0) node #2: coordinates (1.0, 0.0, 0.0) etc. two elements: quad #1: type 3, physical 99, elementary 2, nodes quad #2: type 3, physical 99, elementary 2, nodes one string tag: the name of the view («A scalar view») one real tag: the time value (0.0) three integer tags: the time step (0; time steps always start at 0) 1-component (scalar) field six associated nodal values value associated with node #1 (0.0) value associated with node #2 (0.1) etc. one string tag: the name of the view («Layer Name») number of associated element values value associated with element #1 (90.58) value associated with element #2 (72.04) etc.
--	--

Рис. 5. Структура файла .msh

Fig. 5. File structure .msh

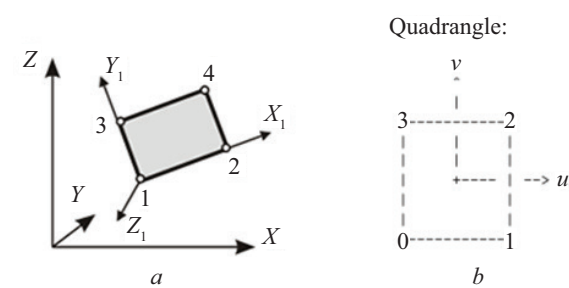


Рис. 6. Порядок нумерации в четырехузловых конечных элементах: *a* — в SCAD++; *b* — в Gmsh

Fig. 6. Numbering order in four-node finite elements: *a* — in SCAD++; *b* — in Gmsh

расчетов в виде скалярных, векторных или тензорных карт, полученных в сторонних вычислительных комплексах. Операции постобработки могут быть автоматизированы пользовательскими приложениями.

Для решения задачи по определению требуемой площади арматуры в сечениях широких ребер монолитной железобетонной плиты, не параллельных основным осям, рассмотрим следующую методику: с применением разработанного пользовательского расширения для SCAD++ (рис. 3) в текстовые файлы выведена информация о геометрии конструк-

ции (файлы NOD.txt и EL.txt), а также о подобранной в каждом КЭ интенсивности арматуры (файл REINF.txt).

Для рационального структурирования большого количества информации в выходных файлах выбран формат обмена данными JSON (JavaScript Object Notation), достаточно широко распространенный и применяемый для передачи информации между различными приложениями [18, 23, 24]. На рис. 4 представлены фрагменты данных, экспортируемых из SCAD++ через пользовательское расширение.

Информация о КЭ, записываемая в файл EL.txt, содержит следующие поля: Text — номер конечного элемента в расчетной схеме; QuantityNode — количество узлов КЭ; ListNode — номера узлов КЭ; TypeElem — тип КЭ. Файл NOD.txt включает в себя данные о позиционировании — координатах узлов, принадлежащих конечному элементу. Совместно файлы EL.txt и NOD.txt позволяют передать в Gmsh информацию о расчетной схеме. Непосредственные результаты выполненного в SCAD++ расчета армирования КЭ группируются в файл REINF.txt, который содержит следующие сведения: Text — номер КЭ; AS1, AS2, AS3, AS4, AWx, AWy — информация о результатах армирования конечного элемента в соответствующем направлении.

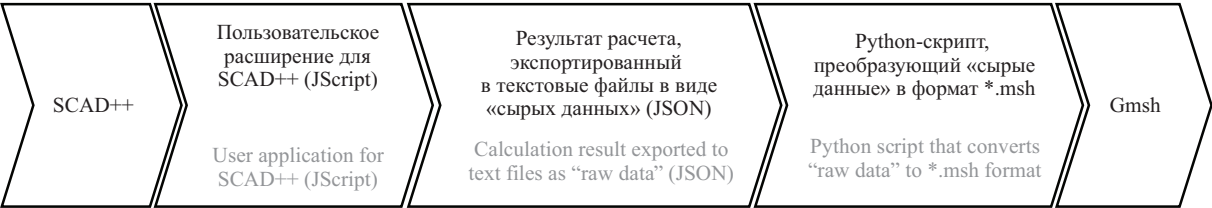


Рис. 7. Порядок обработки данных, полученных в SCAD++

Fig. 7. Procedure of data processing obtained in SCAD++

Примеры обработки данных в программе Gmsh

Examples of data processing in the Gmsh programme

Строка-запрос Request string	Результат обработки Processing result
«AS1» + «AS3»	Вывод суммы «Нижняя по X» (AS1) и «Нижняя по Y» (AS3) Output the sum of “Bottom of X” (AS1) and “Bottom of Y” (AS3)
«AS2» + «AS4»	Вывод суммы «Верхняя по X» (AS2) и «Верхняя по Y» (AS4) Output the sum of “Top of X” (AS2) and “Top of Y” (AS4)
«max(AS1, AS2)»	Вывод слоя (вида) с максимальным значением из AS1 и AS2 Output the layer with the maximum value from AS1 and AS2
«max(AS1, AS2) + max(AS3, AS4) – 20.1»	Вывод результата сложения трех чисел: максимальное значение из AS1 и AS2 прибавить максимальное значение из AS3 и AS4 и вычесть 20,1 см² Output the result of summing three values: the maximum value from AS1 and AS2 add the maximum value from AS3 and AS4 and subtract 20.1 cm²

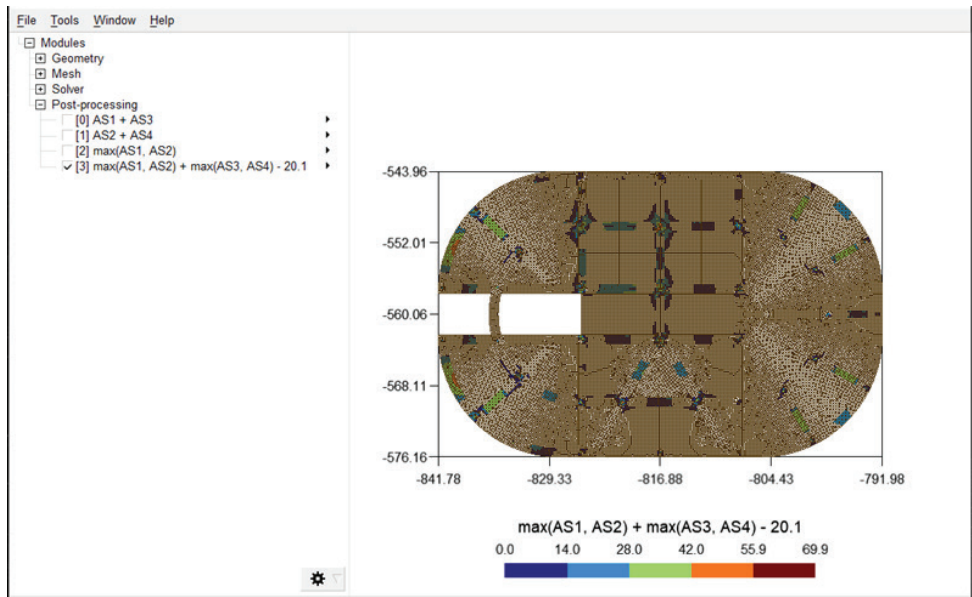
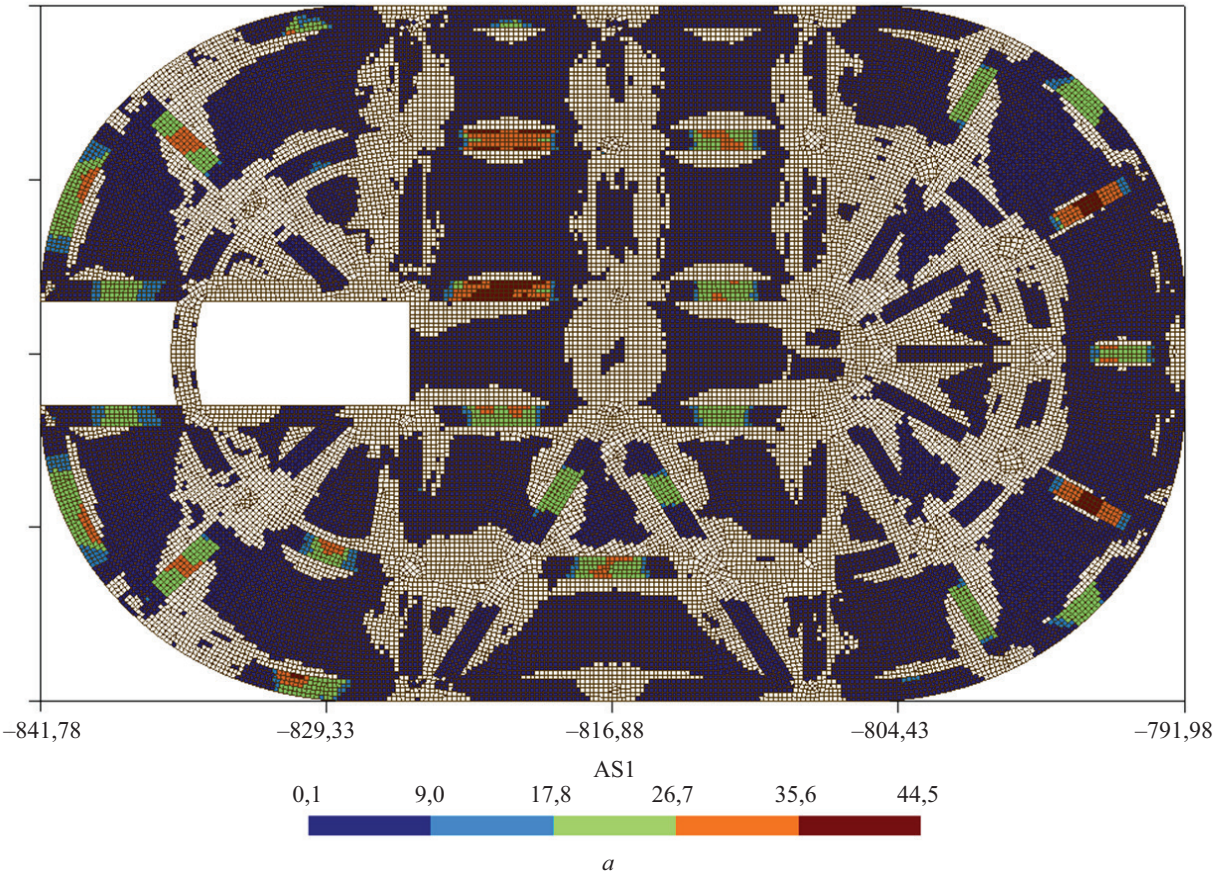
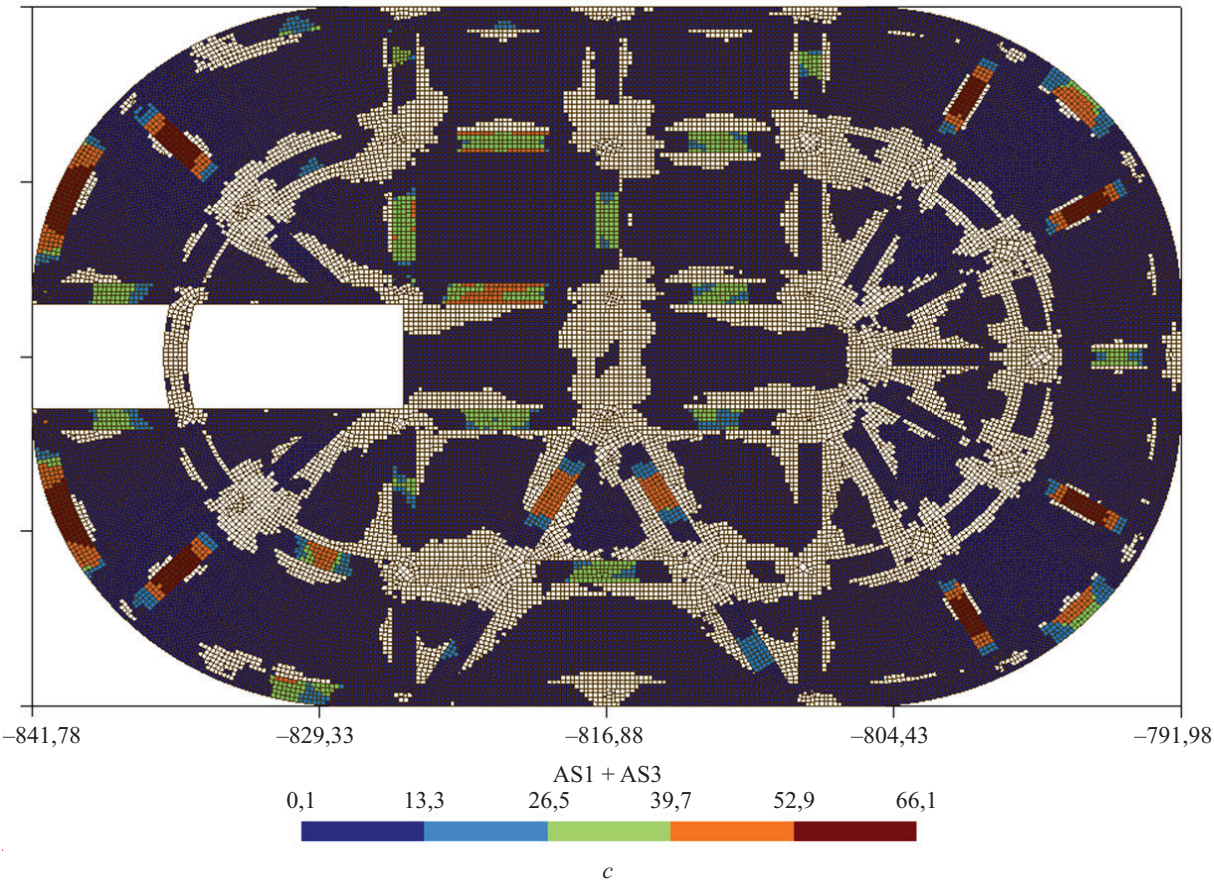
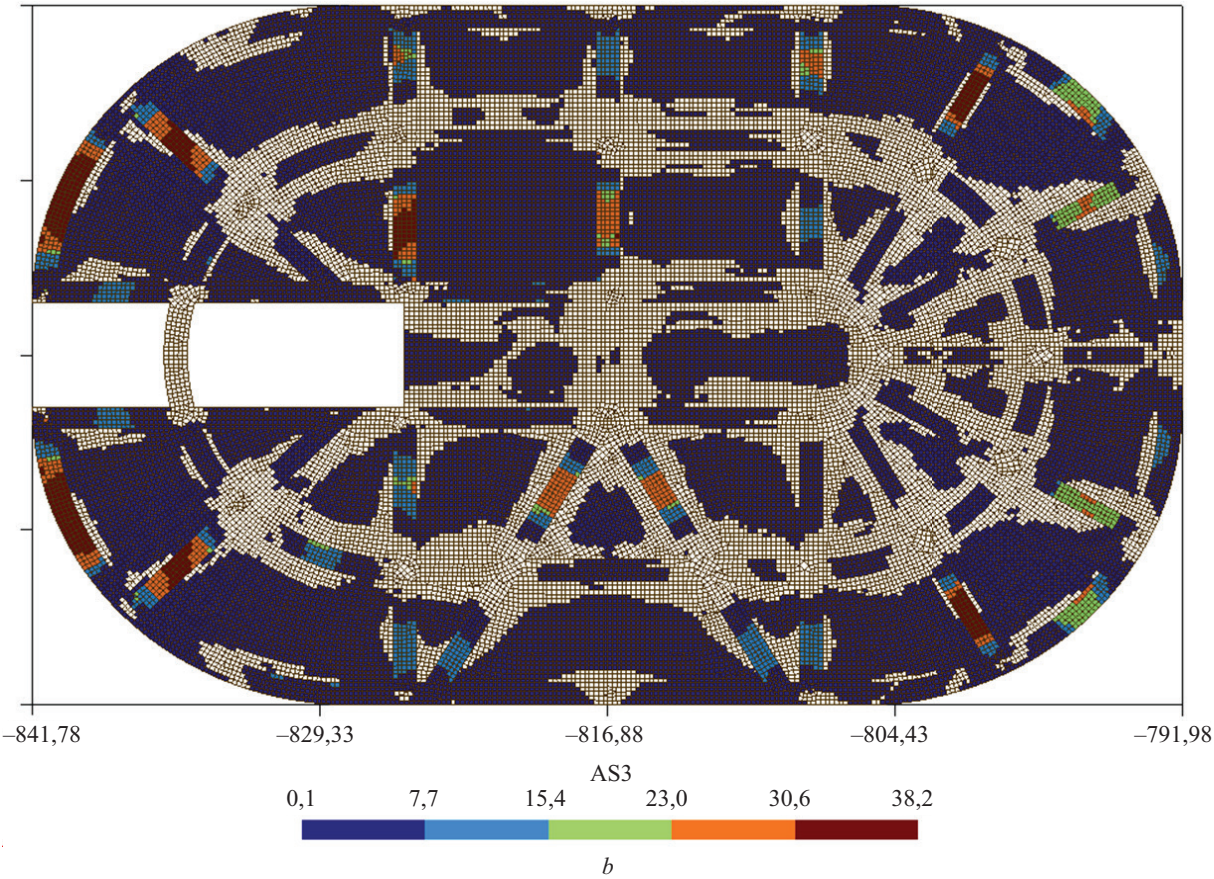


Рис. 8. Общий вид рабочего поля программы Gmsh с результатами обработки пользовательских запросов
Fig. 8. General view of the working field of the Gmsh programme with the results of processing user requests

Далее для переноса экспортированных данных из SCAD++ в Gmsh их требуется преобразовать в файл типа .msh. Структура файла .msh, приведенная на рис. 5, очень проста [20] и имеет блочную структуру. Кратко опишем структуру .msh файла. Первый блок (\$MeshFormat) содержит информацию

о версии используемого языка. В представленном случае используется версия 2. Второй блок (\$Nodes) включает в себя список узлов: в первой строке блока указано количество узлов, во второй и последующих — номер узла и его координаты. Третий блок (\$Elements) содержит список элементов: в первой





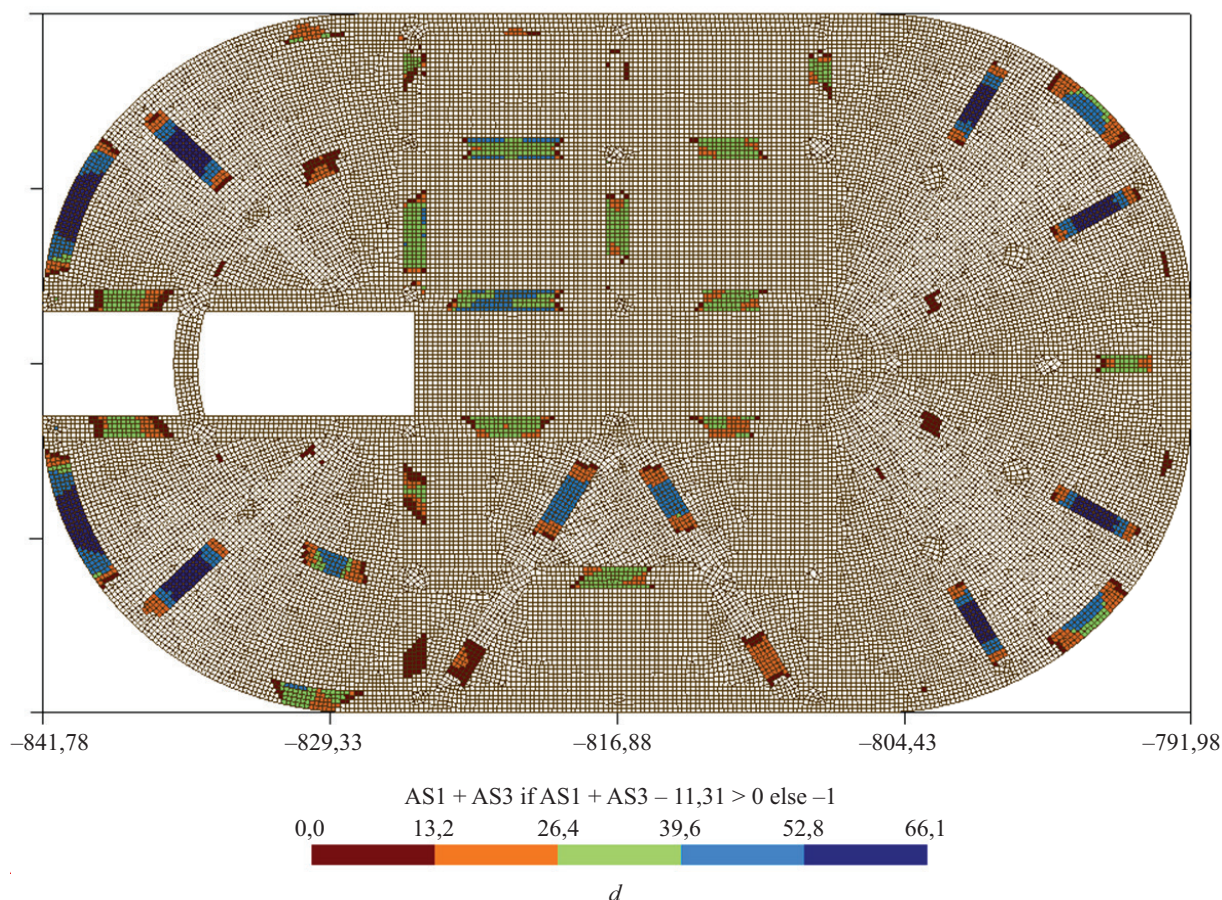


Рис. 9. Отображение в программе Gmsh: *a* — армирование нижнее по *X*; *b* — армирование нижнее по *Y*; *c* — сумма требуемой нижней арматуры по *X* и *Y*; *d* — сумма требуемой нижней арматуры по *X* и *Y*, отображенная для элементов, в которых эта сумма превосходит фоновое армирование ($d12s200 + d12s200$)

Fig. 9. Results, displayed in Gmsh programme: *a* — bottom reinforcement in *X*; *b* — bottom reinforcement in *Y*; *c* — sum of required bottom reinforcement in *X* and *Y*; *d* — sum of required bottom reinforcement in *X* and *Y* displayed for elements in which this sum exceeds the background reinforcement ($d12s200 + d12s200$)

строке указано количество элементов, во второй и последующих — номера элементов, служебная информация и номера узлов, из которых элемент состоит. Служебная информация включает в себя тип КЭ (2-узловой/3-узловой/4-узловой и т.д.) и дополнительные сведения. Четвертый блок (\$NodeData) содержит данные, которые возможно отобразить в узлах. В начале блока представлены различная служебная информация (в том числе название вида) и количество узлов, для которых выводятся данные. После служебной информации построчно выводятся пары чисел: номер узла — значение в узле. Пятый блок (\$ElementData) во многом аналогичен предыдущему блоку, с той лишь разницей, что используется для отображения сведений не в узлах, а в элементах.

Стоит отметить, что количество блоков \$NodeData и \$ElementData неограниченно. Они могут следовать друг за другом, образуя слои (или виды), отображение которых можно при дальнейшей работе отключать и включать. Структура файла .msh оказалась удобной по двум причинам: она полностью повторяет логику

метода конечных элементов, в которой расчетная схема представляет собой набор узлов (каждый из которых имеет уникальный номер и координату) и набор элементов (каждый из которых имеет уникальный номер и список узлов, из которых он состоит). Однако необходимо оговориться, что для четырехузловых КЭ порядок нумерации узлов в SCAD++ и в Gmsh отличается (рис. 6). Все данные, необходимые для формирования .msh файла, были экспортированы в «сыром виде» из SCAD++ в файлы NOD.txt, EL.txt, REINF.txt.

Благодаря простоте своей структуры файл типа .msh может быть сформирован «вручную» через незначительную доработку экспортируемых из SCAD++ данных, например, в программе для работы с электронными таблицами Microsoft Excel или аналогичной (для этого в SCAD++ есть достаточно удобный модуль «Документирование»). Однако более рациональным решением для формирования файла .msh является создание скрипта (программного кода) на языке программирования Python, который автоматически считывает данные JSON и формирует из них .msh файл. При этом проблема с несоответствием

нумерации узлов решается так же просто. Таким образом, алгоритм переноса результатов расчетов, выполненных в ядре SCAD++, имеет вид, показанный на рис. 7.

Такое решение дало возможность не только значительно ускорить перенос сведений, но и обеспечить их обработку для последующего анализа в Gmsh «на лету». Поскольку формат .msh позволяет создавать неограниченное количество блоков с данными (ограничимся примером с блоком \$ElementData), то имеется возможность вывести в один файл разную информацию, которая будет отображаться в различных слоях (видах). Скрипт на языке Python написан таким образом, что может обрабатывать простые формулы и проводить необходимые вычисления. Пример формируемых запросов приведен в таблице.

Результаты обработки пользовательских запросов могут быть выведены для просмотра в отдельном слое (виде), кроме того, реализована возможность быстрого перехода между ними. На рис. 8 представлен общий вид рабочего поля программы Gmsh с результатами обработки запросов, приведенных в таблице.

РЕЗУЛЬТАТЫ

По итогам реализации предложенного алгоритма по обработке результатов подбора арматуры рассматриваемой плиты перекрытия в программном комплексе SCAD++ получена необходимая интенсивность армирования широких ребер, расположенных не параллельно основным осям. Визуальное отображение результатов обработки данных показано на рис. 9.

Применение Gmsh позволяет легко и наглядно отображать обработанную информацию, предоставляя пользователю весь необходимый инструментарий: цветовая настраиваемая шкала; пользовательское задание интервалов; работа со слоями (видами), отображающими различные данные для одних и тех же

конструкций; экспорт настроенных изображений в pdf или png для дальнейшего использования в CAD-приложениях в качестве подложек. Кроме того, возникает возможность передавать смежным специалистам непосредственно сформированные Gmsh-файлы, которые могут быть обработаны ими в удобном формате.

ЗАКЛЮЧЕНИЕ

Использование пользовательских расширений для SCAD++ позволяет выгружать практически любые результаты расчета в виде «сырых данных». Такими данными могут быть не только результаты подбора армирования, но и перемещения, а также значения внутренних усилий и напряжений.

Применение разработанного скрипта на языке программирования Python, преобразующего результаты выполненных расчетов из SCAD++ в исходные данные для Gmsh, дает возможность выполнять эту операцию «на лету», формируя удобные для дальнейшего анализа значения и их комбинации.

Использование Gmsh позволяет осуществлять эффективную обработку результатов, выполненных в SCAD++ расчетов, посредством входящих в него инструментов.

Таким образом, предложенная методика предоставляет широкие возможности для эффективного анализа результатов расчетов, проведенных непосредственно в сертифицированном проектно-вычислительном комплексе SCAD++, и их дальнейшей обработки. Кроме того, реализация предлагаемого алгоритма позволяет организовать эффективное взаимодействие специалистов, работающих над проектом объекта капитального строительства, а также обеспечить значительный экономический эффект через отказ от приобретения дополнительных лицензий дорогостоящего ПО для рабочих мест специалистов, незанятых напрямую расчетом НДС проектируемых конструкций и их элементов, благодаря свободному распространению Gmsh.

СПИСОК ИСТОЧНИКОВ

1. Ma J., He Yu., Zhao Zi.L., Xie Yi.M. Topology Optimization of Ribbed Slabs and Shells // Engineering Structures. 2023. Vol. 277. P. 115454. DOI: 10.1016/J.ENGSTRUCT.2022.115454. EDN PXNBJR.
2. Цветкова А.А., Агафонов С.А., Горячева А.О., Маслак Т.В. Работа монолитного железобетонного ребристого перекрытия при наличии приопорных усадочных трещин в ребрах // Инженерные исследования. 2023. № 1 (11). С. 11–19. EDN OCVMVQ.
3. Xu J., Shu X., Qiao P., Li S., Xu J. Developing a Digital Twin Model for Monitoring Building Structural Health by Combining a Building Information Model and a Real-Scene 3D Model // Measurement. 2023. Vol. 217. P. 112955. DOI: 10.1016/j.measurement.2023.112955

4. Никитин К.Е., Курсанов О.А. Сравнительное исследование конечно-элементных методик расчета ребристых железобетонных перекрытий // Строительная механика инженерных конструкций и сооружений. 2022. Т. 18. № 3. С. 242–254. DOI: 10.22363/1815-5235-2022-18-3-242-254. EDN VXEZDR.
5. Мозголов М.В., Козлова Е.В. К вопросу создания верификационной модели для расчета кессонного железобетонного перекрытия в вычислительном комплексе SCAD // Вестник НИЦ Строительство. 2022. № 1 (32). С. 128–140. DOI: 10.37538/2224-9494-2022-1(32)-128-140. EDN BKAJGQ.
6. Czumaj P., Dudziak S., Kacprzyk Z. Computational Models of Reinforced Concrete Ribbed Floor //

MATEC Web of Conferences. 2018. Vol. 196. P. 01051. DOI: 10.1051/MATECONF/201819601051

7. Tan Y., Xu W., Chen P., Zhang S. Building Defect Inspection and Data Management Using Computer Vision, Augmented Reality, and BIM Technology // Automation in Construction. 2024. Vol. 160. P. 105318. DOI: 10.1016/j.autcon.2024.105318

8. Jin R., Zhong B., Ma L., Hashemi A., Ding L. Integrating BIM with Building Performance Analysis in Project Life-Cycle // Automation in Construction. 2019. Vol. 106. P. 102861. DOI: 10.1016/J.AUTCON.2019.102861

9. Kunieda Y., Codinhoto R., Emmitt S. Increasing the Efficiency and Efficacy of Demolition through Computerised 4D Simulation // Engineering, Construction and Architectural Management. 2019. Vol. 26. Issue 10. Pp. 2186–2205. DOI: 10.1108/ECAM-11-2018-0492

10. Leonardi M.L., Granja J., Oliveira D.V., Azenha M. Scalable BIM Based Open Workflow for Structural Analysis of Masonry Building Aggregates // Computers & Structures. 2024. Vol. 297. P. 107321. DOI: 10.1016/J.COMPSTRUC.2024.107321

11. Ghosh S., Dashti F., Nagae T., Uta H. Numerical Simulation of the 2010 4-Story Reinforced Concrete Structure Tested on the E-Defense Shake Table // Engineering Structures. 2024. Vol. 306. P. 117769. DOI: 10.1016/J.ENGSTRUCT.2024.117769

12. Vishtak O., Golova T., Vishtak N., Shtyrova I., Mikheyev I. Software Module for Automating the Calculation of Building Structures // Procedia Computer Science. 2022. Vol. 213. Pp. 301–306. DOI: 10.1016/J.PROCS.2022.11.071

13. Rasoulzadeh S., Senk V., Königsberger M., Reisinger J., Kovacic I., Füssl J. et al. A Novel Integrative Design Framework Combining 4D Sketching, Geometry Reconstruction, Micromechanics Material Modeling, and Structural Analysis // Advanced Engineering Informatics. 2023. Vol. 57. P. 102074. DOI: 10.1016/J.AEI.2023.102074

14. Andersen M.E.M., Poulsen P.N., Olesen J.F., Hoang L.C. Strength-Based Material Layout Optimization of Solid Reinforced Concrete // Computers & Structures. 2023. Vol. 276. P. 106941. DOI: 10.1016/J.COMPSTRUC.2022.106941

15. Lee S.H., Oh B.K., Choi J., Hong D.H., Hong T., Lee D.E. et al. Eco-Friendly and Economically Optimal Design Model (EEODM) to Reduce the CO₂ Emissions and the Cost of Long-Span Waffle Slabs // Journal of

Cleaner Production. 2021. Vol. 296. P. 126367. DOI: 10.1016/J.JCLEPRO.2021.126367

16. Teodosio B., Wasantha P.L.P., Yaghoubi E., Guerrieri M., Fragomeni S., van Staden R. Environmental, Economic, and Serviceability Attributes of Residential Foundation Slabs: A Comparison between Waffle and Stiffened Rafts Using Multi-Output Deep Learning // Journal of Building Engineering. 2023. Vol. 80. P. 107983. DOI: 10.1016/J.JOBE.2023.107983

17. Ženíšek M., Pešta J., Tipka M., Kočí V., Hájek P. Optimization of RC Structures in Terms of Cost and Environmental Impact — Case Study // Sustainability. 2020. Vol. 12. Issue 20. P. 8532. DOI: 10.3390/SU12208532

18. Guzmán C.F., Mouton T., Habraken A. Using Gmsh as a Mesh Generator and Post-Processor for LAGAMINE // 1st International Workshop on the Finite Element Code LAGAMINE. 2013. DOI: 10.13140/RG.2.2.19824.00002

19. Geuzaine C., Remacle J.F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities // International Journal for Numerical Methods in Engineering. 2009. Vol. 79. Issue 11. Pp. 1309–1331. DOI: 10.1002/NME.2579

20. Damyanova M., Sabchevski S., Zhelyazkov I. Pre- and Post-Processing of Data for Simulation of Gyrotrons by the GYREOSS Software Package // Journal of Physics: Conference Series. 2010. Vol. 207. P. 012032. DOI: 10.1088/1742-6596/207/1/012032

21. López J., Anitescu C., Rabczuk T. CAD-Compatible Structural Shape Optimization with a Movable Bézier Tetrahedral Mesh // Computer Methods in Applied Mechanics and Engineering. 2020. Vol. 367. P. 113066. DOI: 10.1016/J.CMA.2020.113066

22. Lima P., Shauer N., Villegas J.B., Devloo P.R.B. DFNMesh: Finite Element Meshing for Discrete Fracture Matrix Models // Advances in Engineering Software. 2023. Vol. 186. P. 103545. DOI: 10.1016/J.ADVENGSOFT.2023.103545

23. Alavi H., Bortolini R., Forcada N. BIM-Based Decision Support for Building Condition Assessment // Automation in Construction. 2022. Vol. 135. P. 104117. DOI: 10.1016/J.AUTCON.2021.104117

24. Seidenschnur M., Kücükavcı A., Fjerbæk E.V., Smith K.M., Pauwels P., Hviid C.A. A Common Data Environment for HVAC Design and Engineering // Automation in Construction. 2022. Vol. 142. P. 104500. DOI: 10.1016/J.AUTCON.2022.104500

Поступила в редакцию 2 июня 2024 г.

Принята в доработанном виде 13 ноября 2025 г.

Одобрена для публикации 21 ноября 2025 г.

ОБ АВТОРАХ: **Станислав Федорович Дьяков** — кандидат технических наук, доцент Высшей школы промышленно-гражданского и дорожного строительства; **Санкт-Петербургский политехнический университет Петра Великого (СПбПУ)**; 195251, г. Санкт-Петербург, ул. Политехническая, д. 29; SPIN-код: 3766-8749, РИНЦ ID: 636376, Scopus: 57210792974, ResearcherID: AAK-4182-2020, ORCID: 0000-0003-3703-0222, Scholar ID: Yhs1xUEAAAAJ; dyakov_sf@spbstu.ru;

Сергей Александрович Агафонов — ассистент Высшей школы промышленно-гражданского и дорожного строительства; **Санкт-Петербургский политехнический университет Петра Великого (СПбПУ)**; 195251, г. Санкт-Петербург, ул. Политехническая, д. 29; РИНЦ ID: 1784, SPIN-код: 7944-9351, ResearcherID: HHM-5486-2022, ORCID: 0000-0001-5742-8532, Scholar ID: sBe_I2AAAAAJ; agafonov_sa@spbstu.ru.

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.

Авторы заявляют об отсутствии конфликта интересов.

REFERENCES

1. Ma J., He Yu., Zhao Zi.L., Xie Yi.M. Topology Optimization of Ribbed Slabs and Shells. *Engineering Structures*. 2023; 277:115454. DOI: 10.1016/J.ENGSTRUCT.2022.115454. EDN PXNBJR.
2. Tsvetkova A., Agafonov S., Goryacheva A., Maslak T. The performance of a monolithic reinforced concrete ribbed floor with shrinkage cracks in the ribs. *Engineering Research*. 2023; 1(11):11-19. EDN OCVMQV. (rus.).
3. Xu J., Shu X., Qiao P., Li S., Xu J. Developing a Digital Twin Model for Monitoring Building Structural Health by Combining a Building Information Model and a Real-Scene 3D Model. *Measurement*. 2023; 217: 112955. DOI: 10.1016/j.measurement.2023.112955
4. Nikitin K.E., Kirsanov O.A. Comparative study of finite element methods of calculation of ribbed reinforced concrete floors. *Structural Mechanics of Engineering Constructions and Buildings*. 2022; 18(3):242-254. DOI: 10.22363/1815-5235-2022-18-3-242-254. EDN VXEZDR. (rus.).
5. Mozgolov M.V., Kozlova E.V. Creation of a SCAD verification model for the design calculations of a reinforced-concrete waffle slab floor system. *Bulletin of Science and Research Center of Construction*. 2022; 1(32):128-140. DOI: 10.37538/2224-9494-2022-1(32)-128-140. EDN BKAJGQ. (rus.).
6. Czumaj P., Dudziak S., Kacprzyk Z. Computational Models of Reinforced Concrete Ribbed Floor. *MATEC Web of Conferences*. 2018; 196:01051. DOI: 10.1051/MATECONF/201819601051
7. Tan Y., Xu W., Chen P., Zhang S. Building Defect Inspection and Data Management Using Computer Vision, Augmented Reality, and BIM Technology. *Automation in Construction*. 2024; 160:105318. DOI: 10.1016/j.autcon.2024.105318
8. Jin R., Zhong B., Ma L., Hashemi A., Ding L. Integrating BIM with Building Performance Analysis in Project Life-Cycle. *Automation in Construction*. 2019; 106:102861. DOI: 10.1016/J.AUTCON.2019.102861
9. Kunieda Y., Codinhoto R., Emmitt S. Increasing the Efficiency and Efficacy of Demolition through Computerised 4D Simulation. *Engineering, Construction and Architectural Management*. 2019; 26(10):2186-2205. DOI: 10.1108/ECAM-11-2018-0492
10. Leonardi M.L., Granja J., Oliveira D.V., Azenha M. Scalable BIM Based Open Workflow for Structural Analysis of Masonry Building Aggregates. *Computers & Structures*. 2024; 297:107321. DOI: 10.1016/J.COMPSTRUC.2024.107321
11. Ghosh S., Dashti F., Nagae T., Uta H. Numerical Simulation of the 2010 4-Story Reinforced Concrete Structure Tested on the E-Defense Shake Table. *Engineering Structures*. 2024; 306:117769. DOI: 10.1016/J.ENGSTRUCT.2024.117769
12. Vishtak O., Golova T., Vishtak N., Shtyrova I., Mikheyev I. Software Module for Automating the Calculation of Building Structures. *Procedia Computer Science*. 2022; 213:301-306. DOI: 10.1016/J.PROCS.2022.11.071
13. Rasoulzadeh S., Senk V., Königsberger M., Reisinger J., Kovacic I., Füssl J. et al. A Novel Integrative Design Framework Combining 4D Sketching, Geometry Reconstruction, Micromechanics Material Modelling, and Structural Analysis. *Advanced Engineering Informatics*. 2023; 57:102074. DOI: 10.1016/J.AEI.2023.102074
14. Andersen M.E.M., Poulsen P.N., Olesen J.F., Hoang L.C. Strength-Based Material Layout Optimization of Solid Reinforced Concrete. *Computers & Structures*. 2023; 276:106941. DOI: 10.1016/J.COMPSTRUC.2022.106941
15. Lee S.H., Oh B.K., Choi J., Hong D.H., Hong T., Lee D.E. et al. Eco-Friendly and Economically Optimal Design Model (EEODM) to Reduce the CO₂ Emissions and the Cost of Long-Span Waffle Slabs. *Journal of Cleaner Production*. 2021; 296:126367. DOI: 10.1016/J.JCLEPRO.2021.126367
16. Teodosio B., Wasantha P.L.P., Yaghoubi E., Guerrieri M., Fragomeni S., van Staden R. Environmental, Economic, and Serviceability Attributes of Residential Foundation Slabs: A Comparison between Waffle and Stiffened Rafts Using Multi-Output Deep Learning. *Journal of Building Engineering*. 2023; 80:107983. DOI: 10.1016/J.JOBE.2023.107983
17. Ženíšek M., Pešta J., Tipka M., Kočí V., Hájek P. Optimization of RC Structures in Terms of Cost and Environmental Impact — Case Study. *Sustainability*. 2020; 12(20):8532. DOI: 10.3390/SU12208532
18. Guzmán C.F., Mouton T., Habraken A. Using Gmsh as a Mesh Generator and Post-Processor for

LAGAMINE. *1st International Workshop on the Finite Element Code LAGAMINE*. 2013. DOI: 10.13140/RG.2.2.19824.00002

19. Geuzaine C., Remacle J.F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*. 2009; 79(11):1309-1331. DOI: 10.1002/NME.2579

20. Damyanova M., Sabchevski S., Zhelyazkov I. Pre- and Post-Processing of Data for Simulation of Gyrotrons by the GYROSS Software Package. *Journal of Physics: Conference Series*. 2010; 207:012032. DOI: 10.1088/1742-6596/207/1/012032

21. López J., Anitescu C., Rabczuk T. CAD-Compatible Structural Shape Optimization with a Movable Bézier Tetrahedral Mesh. *Computer Methods in Applied*

Mechanics and Engineering. 2020; 367:113066. DOI: 10.1016/J.CMA.2020.113066

22. Lima P., Shauer N., Villegas J.B., Devloo P.R.B. DFNMesh: Finite Element Meshing for Discrete Fracture Matrix Models. *Advances in Engineering Software*. 2023; 186:103545. DOI: 10.1016/J.ADVENGSOFT.2023.103545

23. Alavi H., Bortolini R., Forcada N. BIM-Based Decision Support for Building Condition Assessment. *Automation in Construction*. 2022; 135:104117. DOI: 10.1016/J.AUTCON.2021.104117

24. Seidenschnur M., Küçükavcı A., Fjerbæk E.V., Smith K.M., Pauwels P., Hviid C.A. A Common Data Environment for HVAC Design and Engineering. *Automation in Construction*. 2022; 142:104500. DOI: 10.1016/J.AUTCON.2022.104500

Received June 2, 2024.

Adopted in revised form on November 13, 2025.

Approved for publication on November 21, 2025.

B I O N O T E S: **Stanislav F. Diakov** — Candidate of Technical Sciences, Associate Professor of the Higher School of Industrial, Civil and Road Construction; **Peter the Great St. Petersburg Polytechnic University (SPbPU)**; 29 Politechnicheskaya st., St. Petersburg, 195251, Russian Federation; ID RSCI: 1784, SPIN-code: 3766-8749, ID RSCI: 636376, Scopus: 57210792974, ResearcherID: AAK-4182-2020, ORCID: 0000-0003-3703-0222, Scholar ID: Yhs1xUEAAAAJ; dyakov_sf@spbstu.ru;

Sergei A. Agafonov — assistant of the Higher School of Industrial, Civil and Road Construction; **Peter the Great St. Petersburg Polytechnic University (SPbPU)**; 29 Politechnicheskaya st., St. Petersburg, 195251, Russian Federation; SPIN-code: 7944-9351, ResearcherID: HHM-5486-2022, ORCID: 0000-0001-5742-8532, ScholarID: sBe_I2AAAAAJ; agafonov_sa@spbstu.ru.

Contribution of the authors: all authors made an equivalent contribution to the publication.

The authors declare no conflict of interest.