Теоретические основы информатики

Theoretical foundations of computer science

УДК 539.3+004.02+004.942

МНОГОПОТОЧНОЕ ПРОГРАММИРОВАНИЕ И КЕШИРОВАНИЕ В РАМКАХ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ ДЛЯ ИССЛЕДОВАНИЯ ОБОЛОЧЕЧНЫХ КОНСТРУКЦИЙ

Е. А. БУЙВОЛОВ[1)]***, А. А. СЕМЕНОВ***[1)]

[1)]*Санкт-Петербургский государственный архитектурно-строительный университет, ул. 2-я Красноармейская, 4, 190005, г. Санкт-Петербург, Россия*

Статья посвящена вопросу разработки высокопроизводительного программного обеспечения для расчета тонкостенных оболочечных конструкций, процесс деформирования которых носит существенно нелинейный характер и требует больших вычислительных ресурсов. Использована математическая модель типа Тимошенко (Миндлина – Рейснера), учитывающая геометрическую нелинейность, ортотропию материала, поперечные сдвиги и наличие ребер жесткости. Модель записана в виде функционала полной потенциальной энергии деформации и может быть применена для исследования конструкций различной геометрической формы. Для осуществления расчета использованы метод Ритца и метод Ньютона. При программной реализации показано, каким образом от сервисной архитектуры получилось перейти к эффективной микросервисной архитектуре, заменив один из недостаточно производительных Java-модулей на Python-модуль. Проведена оптимизация вычислительного алгоритма для реализации многопоточного расчета всех стадий вычисления, включая метод Ньютона. Выполнены замеры производительности расчета

Образец цитирования:
Буйволов ЕА, Семенов АА. Многопоточное программирование и кеширование в рамках микросервисной архитектуры для исследования оболочечных конструкций. *Журнал Белорусского государственного университета. Математика. Информатика.* 2023;2:63–79.
https://doi.org/10.33581/2520-6508-2023-2-63-79
EDN: DZSBBW

For citation:
Buyvolov EA, Semenov AA. Multithreaded programming and caching within the framework of microservice architecture for the research of shell structures. *Journal of the Belarusian State University. Mathematics and Informatics.* 2023;2:63–79. Russian.
https://doi.org/10.33581/2520-6508-2023-2-63-79
EDN: DZSBBW

Авторы:
Евгений Алексеевич Буйволов – аспирант кафедры информационных технологий строительного факультета. Научный руководитель – А. А. Семенов.
Алексей Александрович Семенов – кандидат технических наук, доцент; заведующий кафедрой информационных технологий строительного факультета.

Authors:
Evgeniy A. Buyvolov, postgraduate student at the department of computer science, faculty of civil engineering.
Alexey A. Semenov, PhD (engineering), docent; head of the department of computer science, faculty of of civil engineering.
sw.semenov@gmail.com

при различных подходах к реализации многопоточного расчета, а именно `parallelStream` и `ForkJoinPool`. Затронуто использование концепции `MapReduce` в рамках фреймворка Java Stream API. Таким образом, разработано эффективное микросервисное приложение, позволяющее моделировать процесс деформирования оболочечных конструкций, в том числе усиленных ребрами жесткости. Полученный в клиентской части приложения графический результат зависимости прогиба от нагрузки позволяет судить о корректности численного решения. Показана эффективность предложенного алгоритма по сравнению с подходом, реализованным в математическом пакете *Maple* (на основе анализа устойчивости пологой оболочки двоякой кривизны).

Ключевые слова: оболочки; устойчивость; микросервисная архитектура; многопоточное программирование; Python; Vue.js; *Maple*.

MULTITHREADED PROGRAMMING AND CACHING WITHIN THE FRAMEWORK OF MICROSERVICE ARCHITECTURE FOR THE RESEARCH OF SHELL STRUCTURES

E. A. BUYVOLOV [a], ***A. A. SEMENOV*** [a]

[a] *Saint Petersburg State University of Architecture and Civil Engineering, 4 2nd Krasnoarmeiskaya Street, Saint Petersburg 190005, Russia*

Corresponding author: A. A. Semenov (sw.semenov@gmail.com)

This work is devoted to the development of high-performance software for the calculation of thin-walled shell structures. The process of their deformation is essentially non-linear and requires large computing resources. The study is based on a mathematical model of the Timoshenko (Mindlin – Reissner) type, which takes into account geometric non-linearity, material orthotropy, transverse shears and the presence of stiffeners. The model is written in the form of a functional of the total potential energy of deformation, and can be used to study the structures of various geometric shapes. To carry out the calculation, we use the Ritz method and Newton's method. In software implementation, it is shown how it is possible to move from a service architecture to an efficient microservice architecture, replacing one of the insufficiently performing Java modules with a Python module. Optimisation is carried out for the implementation of multithreaded calculation of all stages of calculation, including Newton's method. Measurements of the performance of the calculation are obtained for various approaches to the implementation of multithreaded calculation, namely, `parallelStream` and `ForkJoinPool`. The use of the `MapReduce` concept within the framework of Java Stream API is considered. Thus, an effective microservice application has been developed that allows simulating the process of deformation of shell structures, including those reinforced with stiffeners. The graphical result of the dependence of the deflection on the load obtained in the client part of the application makes it possible to judge the correctness of the numerical solution. The efficiency of the proposed algorithm is shown by comparing it with the implementation in the *Maple* mathematical package. The comparison is made on the basis of an analysis of the buckling of a shallow shell of double curvature.

Keywords: shells; buckling; microservice architecture; multithreaded programming; Python; Vue.js; *Maple*.

Введение

Тонкостенные конструкции активно применяются во многих отраслях промышленности, в частности в авиа-, судо- и ракетостроении, строительстве и других сферах деятельности [1–4].

Тонкостенность таких конструкций позволяет существенно снизить их вес и материалоемкость, при этом они обладают достаточно высокой жесткостью, особенно при усилении ребрами [5–7]. Однако процесс деформирования тонкостенных оболочек носит существенно нелинейный характер, что может приводить к потере устойчивости, когда при малом изменении нагрузки происходит резкое увеличение прогиба. В результате образуются вмятины, конструкция «проваливается», может произойти разрушение материала.

Важное значение для обеспечения безопасной работы конструкций имеет компьютерное моделирование процесса их деформирования в целях выявления опасных режимов работы. При его проведении необходимо учитывать геометрическую нелинейность, наличие ребер жесткости, возможность поперечных сдвигов, а в ряде случаев – особенности деформирования материала (ортотропию, физическую нелинейность, ползучесть и др.) [8–11].

Актуальные исследования, связанные с анализом устойчивости оболочечных конструкций, представлены, например, в публикациях [12–14]. Однако таких работ сравнительно мало. Большинство исследований в последнее десятилетие посвящены расчетам замкнутых цилиндрических оболочек, осуществляемым в конечно-элементных программных комплексах общего назначения, таких как *Ansys*, *Abaqus*, *Nastran* и др. Без сомнения, эти комплексы эффективны для быстрого решения сложных инженерных задач, но в меньшей степени пригодны для детальных научных исследований.

При проведении вычислительного эксперимента с учетом множества факторов и соблюдением требуемой точности формируются нелинейные системы уравнений, решение которых вызывает существенные трудности. Использование современных технологий разработки программного обеспечения и поиск новых алгоритмов решения данной задачи являются актуальной проблемой.

В настоящем исследовании для таких расчетов разработано микросервисное приложение с многопоточным программированием и кешированием, показаны его реализация и эффективность по сравнению с более «классическим» вариантом расчета.

Многопоточное программирование находит применение не только в задачах алгоритмизации, прикладной математики и программирования, но и при автоматизации промышленности [15], а также в физических приборах, например приборах регистрации космических лучей [16]. Идея разделения задачи была совмещена с сетевыми технологиями: в данный момент существует огромное количество облачных решений, использующих вычислительные узлы и балансировку нагрузки между ними [17].

Эффективность многопоточных приложений изучена и подтверждена в тех случаях, когда время на разделение задачи значительно меньше времени выполнения подзадачи [18; 19].

Материалы и методы исследования

Математическая модель. Будем рассматривать математическую модель деформирования оболочечной конструкции, основанную на гипотезах Тимошенко (Миндлина – Рейснера). Модель учитывает геометрическую нелинейность, поперечные сдвиги, ортотропию материала и наличие ребер жесткости. Она состоит из геометрических и физических соотношений, а также функционала полной потенциальной энергии деформации.

Общий вид одного из вариантов оболочечных конструкций (пологая оболочка двоякой кривизны, квадратная в плане) приведен на рис. 1.

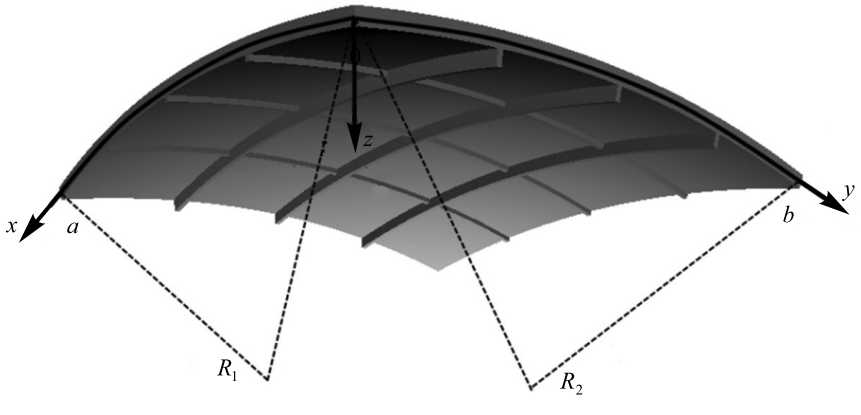

Рис. 1. Общий вид пологой оболочки двоякой кривизны, усиленной ребрами жесткости
Fig. 1. General view of a shallow shell of double curvature, stiffened by ribs

Геометрические соотношения. Данный вид соотношений связывает деформации ε_x, ε_y, γ_{xy} и перемещения U, V, W, Ψ_x, Ψ_y. Для срединной поверхности конструкции имеем [20]

$$\varepsilon_x = \frac{1}{A}\frac{\partial U}{\partial x} + \frac{1}{AB}\frac{\partial A}{\partial y}V - k_x W + \frac{1}{2}\theta_1^2,\ \varepsilon_y = \frac{1}{B}\frac{\partial V}{\partial y} + \frac{1}{AB}\frac{\partial B}{\partial x}U - k_y W + \frac{1}{2}\theta_2^2,$$

$$\gamma_{xy} = \frac{1}{A}\frac{\partial V}{\partial x} + \frac{1}{B}\frac{\partial U}{\partial y}V - \frac{1}{AB}\frac{\partial A}{\partial y}U - \frac{1}{AB}\frac{\partial B}{\partial x}V + \theta_1\theta_2,$$

$$k_x = \frac{1}{R_1},\ k_y = \frac{1}{R_2},\ \theta_1 = -\left(\frac{1}{A}\frac{\partial W}{\partial x} + k_x U\right),\ \theta_2 = -\left(\frac{1}{B}\frac{\partial W}{\partial y} + k_y V\right).$$

Геометрия конструкции задается через параметры Ламе и главные радиусы кривизны, которые для пологой оболочки двоякой кривизны, квадратной в плане, равны следующим величинам:

$$A = 1,\ B = 1,\ R_1 = \text{const},\ R_2 = \text{const}.$$

Деформации слоя, отстоящего от срединной поверхности на расстояние z, выражаются соотношениями

$$\varepsilon_x^z = \varepsilon_x + z\chi_1,\ \varepsilon_y^z = \varepsilon_y + z\chi_2,\ \gamma_{xy}^z = \gamma_{xy} + 2z\chi_{12},$$

где функции изменения кривизн χ_1, χ_2 и кручения χ_{12} принимают вид [20]

$$\chi_1 = \frac{1}{A}\frac{\partial \Psi_x}{\partial x} + \frac{1}{AB}\frac{\partial A}{\partial y}\Psi_y,\ \chi_2 = \frac{1}{B}\frac{\partial \Psi_y}{\partial y} + \frac{1}{AB}\frac{\partial B}{\partial x}\Psi_x,$$

$$\chi_{12} = \frac{1}{2}\left(\frac{1}{A}\frac{\partial \Psi_y}{\partial x} + \frac{1}{B}\frac{\partial \Psi_y}{\partial x} - \frac{1}{AB}\left(\frac{\partial A}{\partial y}\Psi_y + \frac{\partial B}{\partial x}\Psi_y\right)\right).$$

Физические соотношения. Данный вид соотношений связывает напряжения σ_x, σ_y, τ_{xy}, τ_{xz}, τ_{yz} и деформации ε_x, ε_y, γ_{xy}. Для упругого ортотропного материала оболочки имеем [20]

$$\sigma_x = \frac{E_1}{1-\mu_{12}\mu_{21}}\left(\varepsilon_x + \mu_{21}\varepsilon_y + z\left(\chi_1 + \mu_{21}\chi_2\right)\right),\ \sigma_y = \frac{E_2}{1-\mu_{12}\mu_{21}}\left(\varepsilon_y + \mu_{12}\varepsilon_x + z\left(\chi_2 + \mu_{12}\chi_1\right)\right),$$

$$\tau_{xy} = G_{12}\left(\gamma_{xy} + 2z\chi_{12}\right),\ \tau_{xz} = G_{13}kf\left(z\right)\left(\Psi_x - \theta_1\right),\ \tau_{yz} = G_{23}kf\left(z\right)\left(\Psi_y - \theta_2\right).$$

Усилия и моменты. Интегрируя напряжения по z в пределах от $-\frac{h}{2}$ до $\frac{h}{2}$ (т. е. в пределах обшивки), получаем усилия, моменты и перерезывающие силы, приходящиеся на единицу длины сечения и приведенные к координатной поверхности [20]:

$$N_x^0 = \int\limits_{-h/2}^{h/2}\sigma_x dz = \frac{E_1 h}{1-\mu_{12}\mu_{21}}\left(\varepsilon_x + \mu_{21}\varepsilon_y\right),\ N_y^0 = \int\limits_{-h/2}^{h/2}\sigma_y dz = \frac{E_2 h}{1-\mu_{12}\mu_{21}}\left(\varepsilon_y + \mu_{12}\varepsilon_x\right),\ N_{xy}^0 = N_{yx}^0 = G_{12}h\gamma_{xy},$$

$$M_x^0 = \int\limits_{-h/2}^{h/2}\sigma_x z dz = \frac{E_1}{1-\mu_{12}\mu_{21}}\frac{h^3}{12}\left(\chi_1 + \mu_{21}\chi_2\right),\ M_y^0 = \int\limits_{-h/2}^{h/2}\sigma_y z dz = \frac{E_2}{1-\mu_{12}\mu_{21}}\frac{h^3}{12}\left(\chi_2 + \mu_{12}\chi_1\right),$$

$$M_{xy}^0 = M_{yx}^0 = \int\limits_{-h/2}^{h/2}\tau_{xy} z dz = G_{12}\frac{h^3}{6}\chi_{12},$$

$$Q_x^0 = \int\limits_{-h/2}^{h/2}\tau_{xz} dz = G_{13}kh\left(\Psi_x - \theta_1\right),\ Q_y^0 = \int\limits_{-h/2}^{h/2}\tau_{yz} dz = G_{23}kh\left(\Psi_y - \theta_2\right),$$

где N_x^0, N_y^0 – нормальные усилия; N_{xy}^0, N_{yx}^0 – сдвиговые усилия; M_x^0, M_y^0 – изгибающие моменты; M_{xy}^0, M_{yx}^0 – крутящие моменты; Q_x^0, Q_y^0 – перерезывающие силы.

Для получения усилий, моментов и перерезывающих сил, действующих в ребрах, необходимо произвести аналогичное интегрирование, но уже в пределах от $\frac{h}{2}$ до $\frac{h}{2} + H$.

Функция H задает распределение ребер по оболочке с помощью единичных столбчатых функций. Расчет жесткостных характеристик ребер подробно рассматривается в работе [20] и здесь в силу громоздкости не приводится. Отметим лишь, что наличие ребер жесткости на порядок повышает сложность вычислений.

Функционал полной потенциальной энергии деформации. Функционал полной потенциальной энергии деформации оболочки, представляющий собой сумму работ внутренних и внешних сил, имеет вид

$$E_s = E_p^0 + E_p^R - \text{A}, \tag{1}$$

$$E_p^0 = \frac{1}{2}\int_{a_1}^{a}\int_{0}^{b}\Big[N_x^0\varepsilon_x + N_y^0\varepsilon_y + \frac{1}{2}\left(N_{xy}^0 + N_{yx}^0\right)\gamma_{xy} + M_x^0\chi_1 + M_y^0\chi_2 + \left(M_{xy}^0 + M_{yx}^0\right)\chi_{12} +$$

$$+ Q_x^0\left(\Psi_x - \theta_1\right) + Q_y^0\left(\Psi_y - \theta_2\right)\Big] AB\,dx\,dy,$$

$$E_p^R = \frac{1}{2}\int_{a_1}^{a}\int_{0}^{b}\Big[N_x^R\varepsilon_x + N_y^R\varepsilon_y + \frac{1}{2}\left(N_{xy}^R + N_{yx}^R\right)\gamma_{xy} + M_x^R\chi_1 + M_y^R\chi_2 + \left(M_{xy}^R + M_{yx}^R\right)\chi_{12} +$$

$$+ Q_x^R\left(\Psi_x - \theta_1\right) + Q_y^R\left(\Psi_y - \theta_2\right)\Big] AB\,dx\,dy,$$

$$\mathrm{A} = \int_{a_1}^{a}\int_{0}^{b}\left(P_x U + P_y V + qW\right) AB\,dx\,dy,$$

где E_p^0 – потенциальная энергия обшивки; E_p^R – потенциальная энергия ребер жесткости; A – работа внешних сил.

Алгоритм. Для исследования оболочечных конструкций предлагается использовать алгоритм, основанный на методе Ритца и методе Ньютона.

Метод Ритца. Данный метод применяется к функционалу для сведения вариационной задачи к системе нелинейных алгебраических уравнений. Для этого искомые функции перемещений представляются в виде

$$U = U(x, y) = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} U_{kl} X_1^k Y_1^l,\ V = V(x, y) = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} V_{kl} X_2^k Y_2^l,\ W = W(x, y) = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} W_{kl} X_3^k Y_3^l,$$

$$\Psi_x = \Psi_x(x, y) = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} \Psi_{x,\,kl} X_4^k Y_4^l,\ \Psi_y = \Psi_y(x, y) = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} \Psi_{y,\,kl} X_5^k Y_5^l, \tag{2}$$

где U_{kl}, V_{kl}, W_{kl}, $\Psi_{x,\,kl}$, $\Psi_{y,\,kl}$ – неизвестные числовые параметры; $X_1^k - X_5^k$, $Y_1^l - Y_5^l$ – известные аппроксимирующие функции аргументов x и y, удовлетворяющие заданным краевым условиям на контуре оболочки; N – количество членов разложения. Подставляя функции (2) в функционал (1), находим производные по неизвестным числовым параметрам U_{kl}, V_{kl}, W_{kl}, $\Psi_{x,\,kl}$, $\Psi_{y,\,kl}$. Таким образом, получаем систему нелинейных алгебраических уравнений. Для ее решения воспользуемся многомерным методом Ньютона.

Метод Ньютона. Данный метод считается классическим методом решения задач оптимизации. В многомерном случае коэффициенты U_{kl}, V_{kl}, W_{kl}, $\Psi_{x,\,kl}$, $\Psi_{y,\,kl}$ являются корнями, которые определяются при заданной точности для выбранного значения нагрузки. Рассчитанные на предыдущей стадии значения коэффициентов будут использованы для следующего значения нагрузки.

Таким образом, вектор неизвестных числовых параметров вычисляется на каждом шаге нагружения. Используя формулы (2) и найденные значения числовых параметров, можно получить значения перемещений в каждой точке конструкции.

Алгоритм вычисления искомого вектора в соответствии с методом Ньютона может быть записан в виде [21]

$$X_i = X_{i-1} - H^{-1}\left(X_{i-1}\right)\nabla E_s\left(X_{i-1}\right),\ X = \left(U_{kl}, V_{kl}, W_{kl}, \Psi_{x,\,kl}, \Psi_{y,\,kl}\right)^T,\ k, l = 1, \ldots, \sqrt{N},$$

где X_i – значение искомого вектора на текущей итерации; X_{i-1} – значение искомого вектора на предыдущей итерации; H^{-1} – обратная матрица Гессе для E_s; ∇E_s – вектор частных производных (градиент) E_s.

Технологии разработки программного обеспечения. Далее рассмотрим технологии, которые будут использованы при реализации программного обеспечения.

Микросервисная архитектура. Данный вариант сервисно ориентированной архитектуры, подразумевает отказ от единой структуры [22]. Сервисно ориентированная архитектура состоит в формировании модулей приложения для решения бизнес-задач, при этом модули масштабируемы и независимы друг от друга.

Микросервисная архитектура обеспечивает меньшие потери при неисправности одного из модулей, а также простоту масштабирования. С другой стороны, возрастают сетевые издержки и сложность тестирования. Использование докеризации и балансировщика нагрузки позволяет распределить нагрузку между компьютерами вычислительного кластера.

Кеширование. При решении задач аппроксимации и математического моделирования часто базисные функции можно выбрать таким образом, чтобы при упрощении итогового функционала или другого уравнения, которое включает в себя функциональные зависимости, состоящие в том числе из аппроксимирующих функций, бóльшая часть итераций сводилась к использованию уже рассчитанных значений.

Одним из примеров кеширования является использование хеш-таблиц типа «ключ – значение». В языке программирования Java подобные таблицы реализуются в том числе через класс потокобезопасной коллекции `ConcurrentHashMap`.

Многопоточное программирование. Одним из способов повышения скорости расчета является выполнение вычислений параллельно на нескольких компьютерах. В рамках одного компьютера повышение скорости расчета достигается за счет параллельных вычислений на разных ядрах процессора. Существует несколько стратегий параллельных вычислений. В рамках данной статьи рассматривается подход *work stealing*. Он заключается в разделении задачи на минимальные подзадачи, которые разбиваются между возможным в рамках процессора количеством потоков. При освобождении поток забирает следующую подзадачу.

В языке программирования Java данный подход реализуется за счет использования фреймворка Fork/Join. Задача разбивается (`fork`), пока ее сложность (объем) не опустится ниже определенного порога, после чего происходит выполнение подзадач и объединение результатов (`join`). Альтернативным способом реализации (в функциональном стиле) многопоточного расчета является использование метода `parallelStream` из интерфейса Java Stream API, который также по умолчанию создает потоки `ForkJoinPool`. Для обеспечения максимального быстродействия имеет смысл разбивать входные данные таким образом, чтобы издержки на создание объектов не были слишком высокими. В крайнем случае можно разбить входные данные на количество ядер процессора, обеспечив таким образом минимальные издержки на создание объектов и дальнейшую работу сборщика мусора. Однако подобный подход может привести к простою ядер процессора.

Алгоритмизация и программирование

Реализация микросервисной архитектуры. Разработанное приложение (ему присвоено название *JPV-math*) состоит из трех микросервисов:

- клиентского приложения, реализованного с использованием программной платформы Node.js и фреймворков Vue.js и Vuetify;
- сервиса прокси, логики и вычисления основных математических задач, реализованного с помощью фреймворка Java Spring Cloud;
- вспомогательного сервиса, осуществляющего раскрытие скобок и реализованного с помощью фреймворка Python Flask.

Сервисы обмениваются данными через HTTP-запросы. Каждый сервис запускается на отдельном порте. Приложение может быть развернуто с использованием подхода CI/CD (*continuous integration* / *continuous delivery*) на различных устройствах. Работа с Python-сервисом и Java-сервисом ведется в рамках одного проекта в среде IntelliJ IDEA. Код приложения размещен на веб-сервисе *GitHub*[1]. Архитектура приложения представлена на рис. 2.

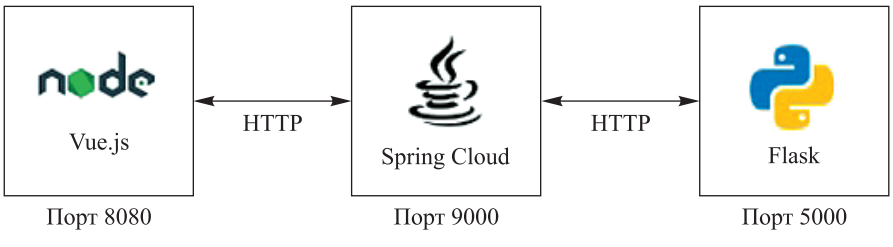

Рис. 2. Архитектура приложения *JPV-math*
Fig. 2. JPV-math application architecture

Java-сервис. Данный сервис использует подход REST (*representational state transfer*). Таким образом, для моделирования процесса деформирования оболочечной конструкции необходимо передать данные (входные параметры) из слоя представления, который в этом случае заменяет сервис веб-клиента, на слой DTO, далее по предоставленному API вызывается метод моделирования, задействующий соответствующий сервис. При разработке приложения применялись блочное тестирование (*unit testing*) и методология *test driven development*.

[1] *JPV-math* (backend) [Electronic resource]. URL: https://github.com/ereborDeveloper/math-modeling (date of access: 11.02.2023) ; *JPV-math* (frontend) [Electronic resource]. URL: https://github.com/ereborDeveloper/math-modeling-gui (date of access: 11.02.2023).

При взятии производной и численном интегрировании используется библиотека символьной алгебры `Symja`[2]. Алгоритм моделирования представлен на рис. 3.

В основе высокоэффективного расчета лежит представление значений функций, результатов интегрирования и дифференцирования в виде пар «ключ – значение». Таким образом, исходная строка на рис. 4, *а*, приводится к объекту на рис. 4, *б*. Основным условием правильного преобразования являются раскрытые скобки (аргументы игнорируются).

Реализованный алгоритм преобразования в виде псевдокода представлен на рис. 5. Алгоритм получения объекта слагаемых с коэффициентами использует алгоритм разбиения строки с игнорированием знаков суммирования и вычитания внутри скобок, который также реализован на основе последовательного перебора символов в строке. Таким образом, после описанного преобразования в объекте не остается повторений по ключу.

При преобразовании исходного функционала изначально выполнялись раскрытие скобок и компоновка слагаемых для дальнейшей подстановки математических формул. Однако Java-сервис не удалось оптимизировать достаточным образом, чтобы раскрытие скобок функционала происходило в допустимое время. Было принято решение использовать Python-сервис с библиотекой `SymEngine`[3], который обеспечил необходимое быстродействие.

Рис. 3. Алгоритм моделирования

Fig. 3. Simulation algorithm

[2]Symja library – Java symbolic math system for Android NCalc calculator [Electronic resource]. URL: https://github.com/axkr/symja_android_library (date of access: 11.02.2023).

[3]SymEngine [Electronic resource]. URL: https://github.com/symengine/symengine (date of access: 11.02.2023).

а/*a*

```
–2.0*u11 + 3.0*v11 –
– 4.0*psiy11 + 3.0*u11*v12 –
– u11*0.7*v12 + 0.5
```

б/*b*

```
{
    “u11”: –2.0,
    “v11”: 3.0,
    “psiy11”: –4.0,
    “u11*v12”: 2.3,
    “number”: 0.5
}
```

Рис. 4. Строка до преобразования (*а*) и объект после преобразования (*б*)
Fig. 4. String before conversion (*a*) and object after conversion (*b*)

```
Возвращает Объект<Ключ, Значение> методПолученияСлагаемыхИзСтроки(строка)
{
    вывод = новый Объект<Ключ, Значение>;
    сомножители = новый Список<Строка>;
    буфер = "";
    начальныйИндекс = 0;
    конечныйИндекс = строка.длина;
    символКаретки = строка(0);
    ...
    // Обработка краевых случаев
    ...
    Для(целого i = 1; i < строка.длина; i++)
    {
        Если(символКаретки == символ разбиения){
            буфер = строка.подстрока(начальныйИндекс, i);
            сомножители = разделитьСтрокуПоСимволуИПропуститьАргументы(буфер, '*');
            произведение = Число(символКаретки.конкатенация("1.0"));
            ДляКаждого(множитель в сомножители)
            {
                Если(множитель является числом)
                {
                    произведение *= множитель;
                }
            }
            сомножители.удалитьВсеЧисла();
            сомножители.отсортироватьПоАлфавиту();
            ключ = сомножители.записатьЧерезСимвол('*');
            Если(ключ пустой)
            {
                ключ = "number";
            }
            Если(вывод.содержитКлюч(ключ))
            {
                вывод.перезаписатьЗначение(ключ, вывод.значениеПоКлючу(ключ) * произведение);
            }
            Иначе
            {
                вывод.добавить(ключ, произведение);
            }
        }
    }
    вернуть вывод;
}
```

Рис. 5. Алгоритм парсинга строки к объекту типа «ключ – значение»
Fig. 5. Algorithm for parsing a string to a key – value object

Python-сервис. Данный сервис представляет собой небольшое Flask-приложение, использующее библиотеку `SymEngine` для раскрытия скобок. Помимо метода раскрытия скобок, в Python-сервисе также был реализован алгоритм, представленный на рис. 5, но скорость его выполнения оказалась ниже, чем в Java-сервисе. Однако Python-сервис на порядок быстрее Java-сервиса при раскрытии скобок, что подтверждает необходимость выбора инструмента под задачу и преимущества использования микросервисного подхода.

Реализация кеширования. Применяя способ хранения математических строк в виде, представленном на рис. 4, *б*, можно реализовать кеширование результатов интегрирования по такому же принципу: сохранять значения посчитанных интегралов и повторно использовать их вместо нового расчета, получая значение по ключу. Алгоритм кеширования при взятии интеграла заключается в применении потокобезопасной неблокируемой коллекции пар «ключ – значение». Подобный подход при сложности обращения к значению по ключу O(1) позволяет существенно ускорить вычисления даже без использования внешнего хранилища за счет интегрирования только сомножителей, зависящих от переменной интегрирования.

Например, при вычислении интеграла для объекта, приведенного на рис. 4, *б*, по переменной u11 в пределах от 0 до 2 получим объект, представленный на рис. 6.

```
{
    “v11”: 6.0,
    “psiy11”: –8.0,
    “v12”: 4.6,
    “number”: –3.0
}
```

Рис. 6. Объект после интегрирования
Fig. 6. Object after integration

При реальном расчете в качестве базисных функций в аппроксимации применяются синусы и косинусы с подобными аргументами. Стоит отметить, что при $N = 2$ время выполнения интегрирования с использованием кеширования составило 1 с, а без использования кеширования – 30 с. Характеристики компьютера, на котором производился расчет, будут представлены далее. При увеличении числа N интегрирование без использования кеширования становится крайне неэффективным.

Реализация многопоточного расчета. В случае применения многопоточного расчета необходимо тестировать скорость выполнения задач. Для различных стратегий в зависимости от задачи и способа ее решения эффективность будет различной.

Основная вычислительная сложность в методе Ньютона заключается в символьной подстановке получаемых на каждой итерации значений вместо переменных в градиент и матрицу Гессе для получения численных значений. В целях повышения скорости вычислений используется параллельная подстановка значений.

Оптимизация проводилась последовательно. В методе Ньютона первым оптимизационным решением является многопоточная подстановка значений интеграла в точке: последовательно берется значение градиента, и уже это значение в несколько потоков суммируется по переменным и значению этих переменных в хеш-таблице значения градиента.

Были рассмотрены следующие варианты расчета градиента:

1) в цикле (без многопоточности);
2) с использованием метода `parallelStream` из интерфейса Java Stream API;
3) с использованием подхода `ForkJoinPool`;
4) с разбиением исходного объекта на количество подобъектов, равное количеству ядер процессора.

Параметры оболочки фиксированные (подробнее см. в разделе «Результаты и их обсуждение»). Расчет выполнен для $N = 2$. Максимальное число итераций в методе Ньютона составляет 300, шаг нагрузки – 0,1 МПа, точность – 10^{-4}. Результаты замеров приведены в табл. 1.

Таблица 1

Время выполнения цикла в методе Ньютона

Table 1

Cycle execution time in Newton’s method

Вариант расчета	Время выполнения, с
Обычный цикл	28
`parallelStream`	22
`ForkJoinPool`	36
Разбиение	18

Использование метода `parallelStream` и конструкции `forEach` позволило дополнительно увеличить скорость перебора при вычислении градиента (15 с) по сравнению с последовательным пере-

бором градиента и разбиением (18 с). Применение аналогичного алгоритма при расчете матрицы Гессе также показало свою эффективность (12 с при использовании схемы «`parallelStream.forEach` + `parallelStream.forEach` + разбиение» против 15 с при использовании обычных циклов).

После данных преобразований нахождение прогиба конструкции W уже не является ресурсозатратной задачей. Способ разбиения по количеству ядер процессора был выбран и для многопоточного интегрирования.

Результаты и их обсуждение

Эффективность предложенного подхода продемонстрируем на примере расчета пологой оболочки двоякой кривизны, квадратной в плане.

Расчет напряженно-деформированного состояния оболочечных конструкций является вычислительно сложной задачей. В итоговый функционал для ребристой пологой оболочки двоякой кривизны в зависимости от точности аппроксимации функций (2) входит минимум 51 слагаемое. При увеличении точности аппроксимации количество слагаемых в функционале возрастает экспоненциально, и при наличии 5 неизвестных функций, состоящих из 25 слагаемых каждая (т. е. при $N = 25$), общее количество слагаемых в функционале после упрощения составляет более 1,5 млн.

Будем рассматривать стальные конструкции, находящиеся под действием внешней равномерно распределенной поперечной нагрузки q (т. е. компоненты нагружения $P_x = P_y = 0$) и шарнирно-неподвижно закрепленные по контуру (граничные условия $U = V = W = M_x = \Psi_y = 0$ при $x = 0, x = a$ и $U = V = W = \Psi_x = M_y = 0$ при $y = 0, y = b$). Для данного вида закрепления с учетом симметрии конструкции в формулах (2) можно взять следующие аппроксимирующие функции:

$$U = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} U_{kl} \sin\left(2k\pi\frac{x}{a}\right)\sin\left((2l-1)\pi\frac{y}{b}\right),\ V = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} V_{kl} \sin\left((2k-1)\pi\frac{x}{a}\right)\sin\left(2l\pi\frac{y}{b}\right),$$

$$W = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} W_{kl} \sin\left((2k-1)\pi\frac{x}{a}\right)\sin\left((2l-1)\pi\frac{y}{b}\right),$$

$$\Psi_x = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} \Psi_{x,kl} \cos\left((2k-1)\pi\frac{x}{a}\right)\sin\left((2l-1)\pi\frac{y}{b}\right),$$

$$\Psi_y = \sum_{k=1}^{\sqrt{N}}\sum_{l=1}^{\sqrt{N}} \Psi_{y,kl} \sin\left((2k-1)\pi\frac{x}{a}\right)\cos\left((2l-1)\pi\frac{y}{b}\right).$$

При наличии ребер жесткости используется ортогональная сетка ребер, равномерно распределенных по конструкции. Ширина ребер $r^j = r^i = 2h$, высота ребер $h^j = h^i = 3h$. Расстояние между ребрами обозначим через x_r и будем полагать, что крайние ребра находятся на расстоянии $0{,}5x_r$ от края конструкции. Входные параметры представлены в табл. 2.

Таблица 2

Входные параметры

Table 2

Input parameters

Параметры	Значение	Расшифровка
$E_1 = E_2$, МПа	210 000	Модули упругости материала
$\mu_{12} = \mu_{21}$	0,3	Коэффициенты Пуассона
G, МПа	80 769,23	Модуль сдвига материала
a, м	5,4	Линейный размер вдоль оси x
b, м	5,4	Линейный размер вдоль оси y
$R_1 = R_2$, м	20,25	Радиусы кривизны
h, м	0,09	Толщина
N	1–25	Количество слагаемых в методе Ритца
Δq, МПа	0,01	Шаг нагрузки

Окончание табл. 2
Ending table 2

Параметры	Значение	Расшифровка
q_{max}, МПа	3,3 (без ребер) 4,3 (с ребрами)	Значение нагрузки, до которой выполняется расчет
Steps	50	Максимальное количество итераций в цикле метода Ньютона
ε	10^{-6}	Точность, при достижении которой в методе Ньютона происходит переход к следующей точке

Устойчивость пологой оболочки двоякой кривизны. В качестве критерия потери устойчивости оболочек будем использовать критерий Ляпунова: нагрузка, при которой малому изменению нагрузки соответствует существенное изменение прогиба, считается критической нагрузкой.

В табл. 3 представлены значения критических нагрузок потери устойчивости q_{cr} для рассматриваемой конструкции без ребер и с сеткой ребер размером 1 × 1, а также значения прогиба конструкции W_c в центральной точке $\left(x=\frac{a}{2},\ y=\frac{b}{2}\right)$ и значения прогиба конструкции W_4 в четвертой части $\left(x=\frac{a}{4},\ y=\frac{b}{4}\right)$ в момент достижения критической нагрузки, полученные с использованием разработанного микросервисного приложения *JPV-math* и математического пакета *Maple*.

Для пакета *Maple* значения при $\sqrt{N} \geq 4$ не приводятся в связи со слишком долгим временем выполнения расчета.

Таблица 3

Результаты расчета для оболочки без ребер и с сеткой ребер размером 1 × 1

Table 3

Calculation results for a shell without ribs and with stiffeners grid 1 × 1

$\sqrt{N}$	*Maple*			*JPV-math*		
	q_{cr}, МПа	W_c, м	W_4, м	q_{cr}, МПа	W_c, м	W_4, м
Расчет для оболочки без ребер						
1	3,31	0,106	0,054	3,30	0,104	0,052
2	3,10	0,079	0,040	3,10	0,078	0,039
3	3,10	0,076	0,040	3,10	0,075	0,039
4	–	–	–	3,10	0,073	0,040
5	–	–	–	3,10	0,072	0,040
Расчет для оболочки с сеткой ребер размером 1 × 1						
1	–	–	–	–	–	–
2	4,80	0,139 6	0,093 2	4,79	0,138 7	0,090 5
3	4,47	0,132 5	0,083 1	4,45	0,131 7	0,082 7
4	–	–	–	4,36	0,125 2	0,075 8
5	–	–	–	4,33	0,126 4	0,075 7

Графики зависимости прогиба W_c и W_4 от нагрузки q представлены на рис. 7 и 8. Для удобства анализа данных также показан укрупненный фрагмент, соответствующий потере устойчивости.

Как показывают результаты расчетов, уже при $\sqrt{N}=3$ и $\sqrt{N}=4$ вид кривых и значения критических нагрузок становятся близки, что говорит о сходимости метода Ритца. Тем не менее для других видов конструкций может понадобиться большее, чем здесь, количество неизвестных коэффициентов в формулах (2), что требует отдельного исследования.

Для рассмотренных конструкций наблюдается общая потеря устойчивости без образования локальных вмятин. Кроме того, добавление двух ортогонально расположенных ребер жесткости увеличивает значение критической нагрузки на 39 %. В данном случае такой существенный эффект объясняется достаточно высокой жесткостью обшивки и массивностью подкрепляющих элементов.

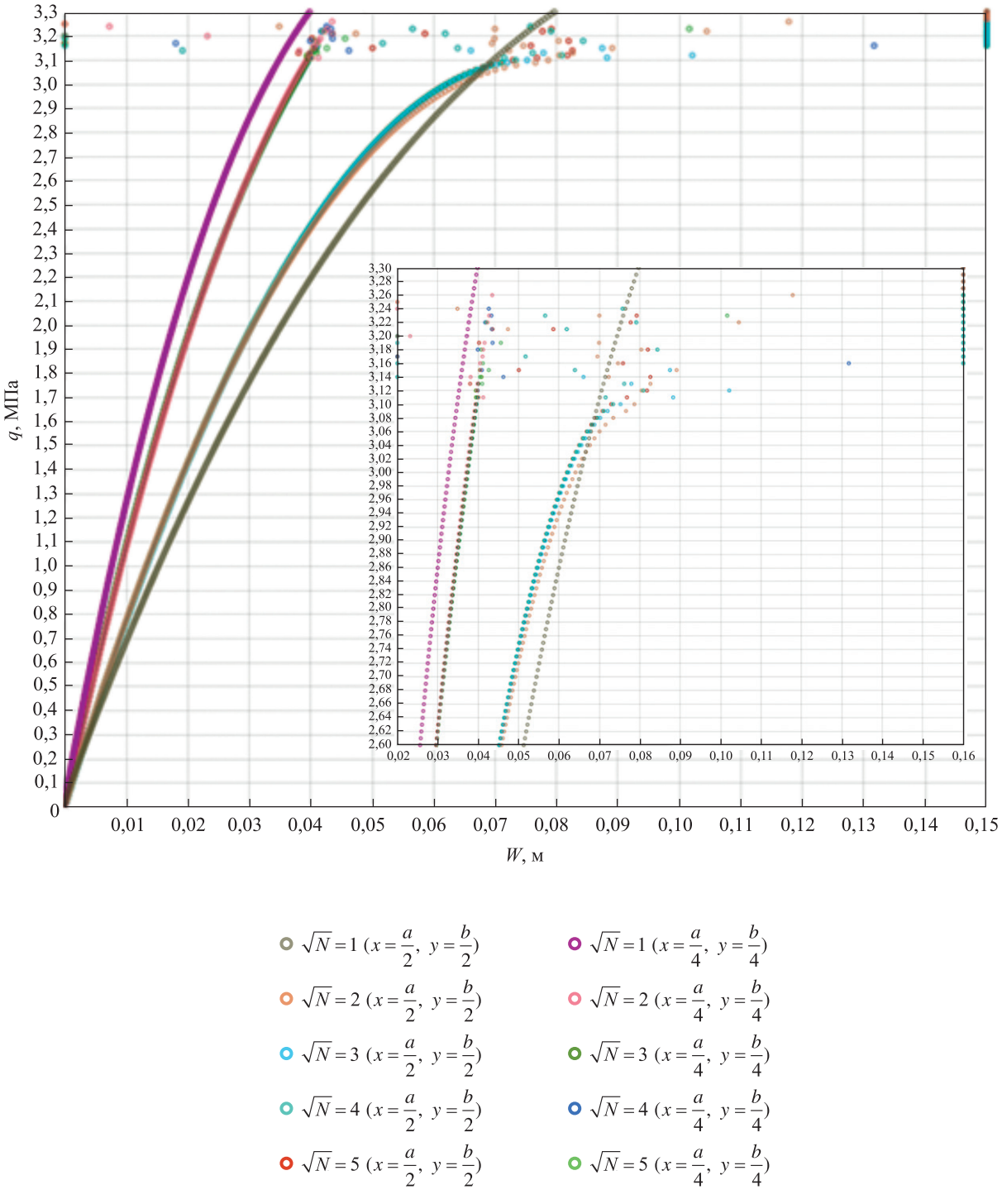

Рис. 7. График зависимости прогиба от нагрузки для конструкции без ребер
Fig. 7. Graph of deflection versus load for a structure without ribs

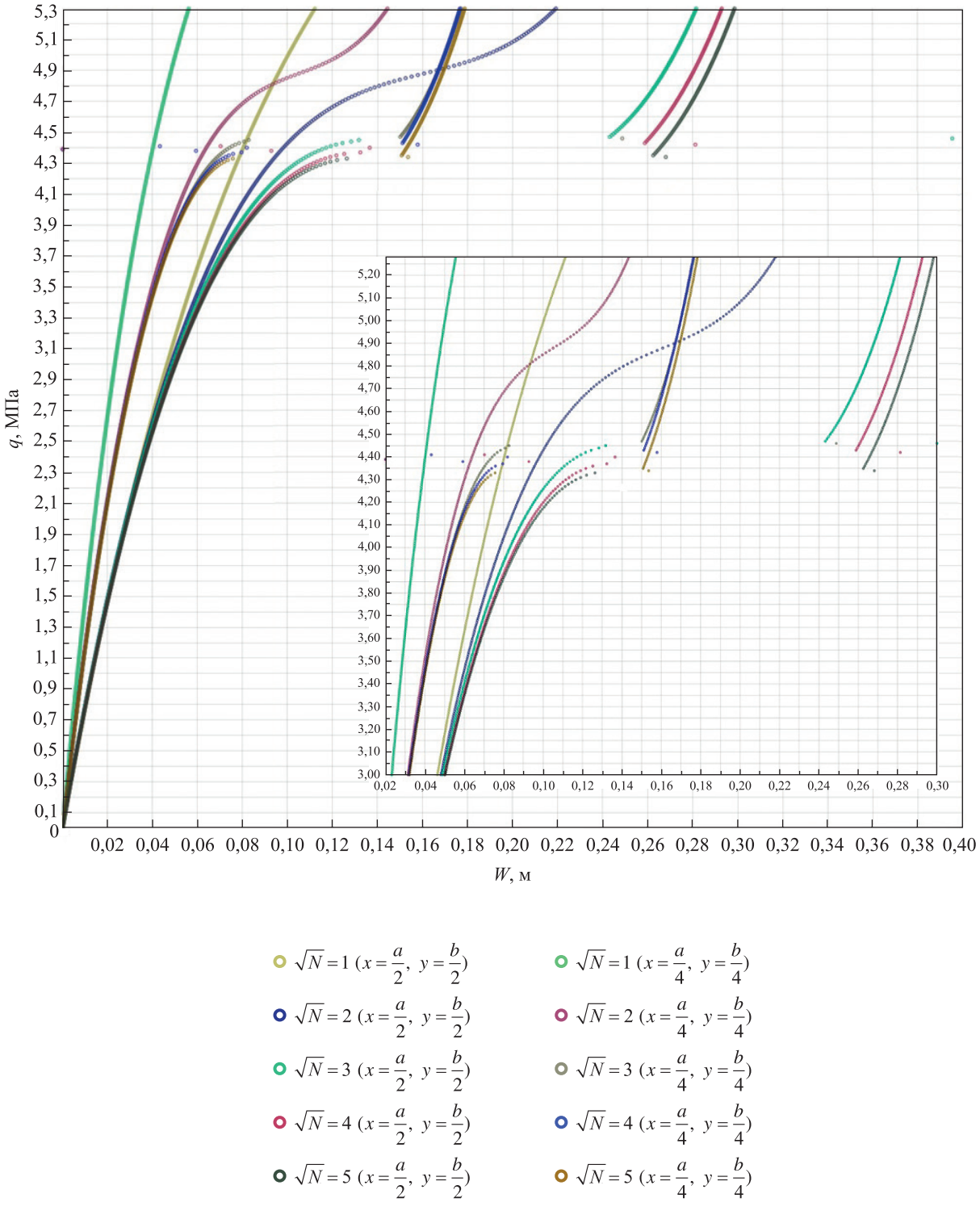

Рис. 8. График зависимости прогиба от нагрузки для конструкции с ребрами
Fig. 8. Graph of deflection versus load for a structure with ribs

Сравнение скорости вычислений. В качестве тестового стенда использовался компьютер, характеристики и результаты бенчмарка которого (на основе сайта *UserBenchmark*) представлены в табл. 4.

Таблица 4

Характеристики и результаты бенчмарка тестового стенда

Table 4

Characteristics and results of the benchmark test bench

Устройство	Торговая марка и основные характеристики	Бенчмарк, %
CPU	AMD Ryzen 5, модель 2600	79,5
GPU	AMD Radeon, модель RX550	13,9
SSD	Intel 660p, интерфейс NVMe PCIe, формфактор M.2, объем 512 Гб	132,4
	HP EX900, интерфейс NVMe PCIe, формфактор M.2, объем 500 Гб	123,1
HDD	WD Blue, объем 1 Тб	83,2
RAM	G.SKILL Ripjaws V, тип DDR4, частота 3200 ГГц, тайминг C16, объем 16 Гб (2×8 Гб)	100,6

Примечание. CPU – процессор; GPU – графический процессор; SSD – твердотельный накопитель; HDD – накопитель на жестких магнитных дисках; RAM – оперативная память.

Первым вычислительным экспериментом является сравнение результатов и скорости вычислений при использовании приложения *JPV-math* и пакета *Maple*. Для этого проводятся замеры времени выполнения расчета для одной и той же оболочки с теми же параметрами в пакете *Maple* и приложении *JPV-math*. Для повышения информативности полученных результатов используется цветовая карта (табл. 5).

Таблица 5

Замеры производительности этапов вычисления в пакете *Maple* и приложении *JPV-math*

Table 5

Performance measurements of calculation steps in the *Maple* package and *JPV-math* application

$\sqrt{N}$	Время выполнения, с				Цветовая карта
	Расчет без ребер		Расчет с ребрами		
	Maple	*JPV-math*	*Maple*	*JPV-math*	
1	4	<1	5	2	1 с и менее
2	34	18	36	25	До 10 с
3	655	102	748	186	До 100 с
4	–	384	–	979	До 250 с
5	–	1799	–	5195	До 500 с
					500 с и более

Результаты замеров времени выполнения различных этапов расчета на основе входных данных из табл. 2 представлены в табл. 6.

Разработанная программа использует 100 % процессорного времени, однако наибольшая производительность может быть достигнута при разделении сервисов по вычислительным мощностям. Тогда процессор не будет распределен между вычислениями и визуализацией. Результаты замеров времени вычислений без использования визуализации также приведены в табл. 6.

Таблица 6

Замеры производительности этапов моделирования в приложении *JPV-math*

Table 6

Performance measurements of simulation steps in the *JPV-math* application

$\sqrt{N}$	Время выполнения, с					
	Et + Rt	It	E	Dt	NWt	At
Расчет без ребер						
1	<1	<1	–	<1	<1	<1
2	<1	<1	–	<1	18	18
3	4	<1	–	3	95	102
4	32	1	–	24	324	384
5	199	6	–	131	1463	1799
Расчет с ребрами						
1	<1	<1	<1	<1	2	2
2	<1	<1	<1	1	24	25
3	4	<1	5	5	172	186
4	33	1	33	48	864	979
5	199	6	269	300	4421	5195
Расчет без ребер и без визуализации						
1	<1	<1	–	<1	2	2
2	<1	1	–	<1	5	6
3	1	3	–	<1	30	34
4	10	8	–	2	136	156
5	54	18	–	11	530	615

Примечания: 1. Et + Rt – время преобразования данных в рамках оптимизации и раскрытия всех скобок под знаком интеграла, замены посчитанных производных и аппроксимирующих функций в строке; It – время взятия двойного интеграла; E – время расчета ребра (раскрытие скобок и преобразование); Dt – время нахождения градиента и матрицы Гессе; NWt – время выполнения оптимизированного метода Ньютона и визуализации данных; At – общее время вычисления. 2. Цветовую карту см. в табл. 5.

Python-модуль для раскрытия скобок при $\sqrt{N} = 5$ потребовал около 12 Гб оперативной памяти, а Java-модуль – всего 2,56 Гб.

Заключение

В ходе исследования разработано микросервисное приложение, позволяющее моделировать деформирование оболочечных конструкций. Оно оказалось в несколько раз более эффективным, чем подход, реализованный в математическом пакете *Maple*. Разработанное приложение обеспечивает взятие двойного интеграла по заданному числу слагаемых на процессоре AMD Ryzen 5 (модель 2600) без использования режима Turbo Boost, внешних файлов или баз данных с уже посчитанными значениями за 18 с.

Взяв за основу идею сервисно ориентированной архитектуры и вдохновившись фреймворком Java Spring Cloud, авторы показали, каким образом от сервисной архитектуры можно перейти к эффективной микросервисной архитектуре, заменив один из недостаточно производительных Java-модулей на Python-модуль.

Проведена оптимизация вычислительного алгоритма для реализации многопоточного расчета всех стадий вычисления, включая метод Ньютона. Выполнены замеры производительности расчета при различных подходах к реализации многопоточного расчета, а именно `parallelStream` и `ForkJoinPool`. Затронуто использование концепции `MapReduce` в рамках фреймворка Java Stream API.

Получены графические результаты вычисления, а также выполнены замеры времени выполнения всех этапов расчета и сравнение предложенного программного решения с математическим пакетом *Maple*. Проект представляет собой готовое к развертыванию микросервисное приложение с клиентской и серверной частями.

Приложение может быть как развернуто в локальной сети, так и доступно извне в зависимости от действий `DevOps`.

Библиографические ссылки

1. Zeybek Ö, Topkaya C, Rotter JM. Requirements for intermediate ring stiffeners placed below the ideal location on discretely supported shells. *Thin-Walled Structures*. 2017;115:21–33. DOI: 10.1016/j.tws.2017.02.003.

2. Старовойтов ЭИ, Нестерович АВ. Неосесимметричное нагружение упругопластической трехслойной пластины в своей плоскости. *Журнал Белорусского государственного университета. Математика. Информатика.* 2022;2:57–69. DOI: 10.33581/2520-6508-2022-2-57-69.

3. Raeesi A, Ghaednia H, Zohrehheydariha J, Das S. Failure analysis of steel silos subject to wind load. *Engineering Failure Analysis*. 2017;79:749–761. DOI: 10.1016/j.engfailanal.2017.04.031.

4. Chen Bo, Zhong Pengpeng, Cheng Weihua, Chen Xinzhong, Yang Qingshan. Correlation and combination factors of wind forces on cylindrical roof structures. *International Journal of Structural Stability and Dynamics*. 2017;17(9):1750104. DOI: 10.1142/S0219455417501048.

5. Yin Caiyu, Jin Zeyu, Chen Yong, Hua Hongxing. Effects of sacrificial coatings on stiffened double cylindrical shells subjected to underwater blasts. *International Journal of Impact Engineering*. 2020;136:103412. DOI: 10.1016/j.ijimpeng.2019.103412.

6. Ren Shaofei, Song Ying, Zhang A-Man, Wang Shiping, Li Pengbo. Experimental study on dynamic buckling of submerged grid-stiffened cylindrical shells under intermediate-velocity impact. *Applied Ocean Research*. 2018;74:237–245. DOI: 10.1016/j.apor.2018.02.018.

7. Seo Jung Kwan, Song Chan Hee, Park Joo Shin, Paik Jeom Kee. Nonlinear structural behaviour and design formulae for calculating the ultimate strength of stiffened curved plates under axial compression. *Thin-Walled Structures*. 2016;107:1–17. DOI: 10.1016/j.tws.2016.05.003.

8. Грачев ВА, Найштут ЮС. Задачи устойчивости тонких упругих оболочек. *Компьютерные исследования и моделирование.* 2018;10(6):775–787. DOI: 10.20537/2076-7633-2018-10-6-775-787.

9. Yankovskii AP. Refined deformation model for metal-composite plates of regular layered structure in bending under conditions of steady-state creep. *Mechanics of Composite Materials*. 2017;52(6):715–732. DOI: 10.1007/s11029-017-9622-7.

10. Stupishin L, Nikitin K, Kolesnikov A. Numerical research orthotropic geometrically nonlinear shell stability using the mixed finite element method. *IOP Conference Series: Materials Science and Engineering*. 2017;201:012019. DOI: 10.1088/1757-899X/201/1/012019.

11. Maksimyuk VA, Storozhuk EA, Chernyshenko IS. Nonlinear deformation of thin isotropic and orthotropic shells of revolution with reinforced holes and rigid inclusions. *International Applied Mechanics*. 2013;49(6):685–692. DOI: 10.1007/s10778-013-0602-x.

12. Huang Sixin, Qiao Pizhong. A new semi-analytical method for nonlinear stability analysis of stiffened laminated composite doubly-curved shallow shells. *Composite Structures*. 2020;251:112526. DOI: 10.1016/j.compstruct.2020.112526.

13. Wang Jingchao, Li Zheng Liang, Yu Wei. Structural similitude for the geometric nonlinear buckling of stiffened orthotropic shallow spherical shells by energy approach. *Thin-Walled Structures*. 2019;138:430–457. DOI: 10.1016/j.tws.2018.02.006.

14. Obodan NI, Gromov VA. The complete bifurcation structure of nonlinear boundary problem for cylindrical panel subjected to uniform external pressure. *Thin-Walled Structures*. 2016;107:612–619. DOI: 10.1016/j.tws.2016.07.020.

15. Евдокимов АА. Методика использования многопоточного программирования для автоматизированных систем многопроцессорных гальванических линий. *Вестник Нижегородского университета имени Н. И. Лобачевского.* 2013;1(3):306–312.

16. Мизин СВ, Махмутов ВС, Максумов ОС, Квашнин АН. Применение многоконвейерного программирования для физического эксперимента. *Краткие сообщения по физике Физического института имени П. Н. Лебедева Российской академии наук.* 2011;2:8–18.

17. Пазников АА, Павский КВ, Павский ВА, Куприянов МС. Моделирование алгоритмов оптимизации выполнения критических секций на выделенных процессорных ядрах в многоядерных вычислительных системах. В: *Труды II Международной научной конференции по проблемам управления в технических системах (CTS-2017); 25–27 октября 2017 г.; Санкт-Петербург, Россия*. Санкт-Петербург: СПбГЭТУ «ЛЭТИ»; 2017. с. 54–57.

18. Кадомский АА, Захаров ВА. Эффективность многопоточных приложений. *Научный журнал.* 2016;7:26–28.

19. Файрушин РН, Якупов ИМ. Повышение эффективности программ с помощью параллельного вычисления задач. *Вестник современных исследований.* 2018;6(3):583–585.

20. Karpov VV, Semenov AA. Refined model of stiffened shells. *International Journal of Solids and Structures.* 2020;199:43–56. DOI: 10.1016/j.ijsolstr.2020.03.019.

21. Бакусов ПА, Семенов АА. Устойчивость сегментов тороидальных оболочек при изменении угла отклонения от вертикальной оси. *Вестник Пермского национального исследовательского политехнического университета. Механика.* 2017;3:17–36. DOI: 10.15593/perm.mech/2017.3.02.

22. Назарков ДА, Телышев АВ. Сравнительный анализ монолитной и микросервисной архитектур информационных систем. *Дневник науки* [Интернет]. 2019 [процитировано 20 февраля 2022 г.];5:48–55. Доступно по: http://www.dnevniknauki.ru/images/publications/2019/5/technics/Nazarkov_Telyshev.pdf.

References

1. Zeybek Ö, Topkaya C, Rotter JM. Requirements for intermediate ring stiffeners placed below the ideal location on discretely supported shells. *Thin-Walled Structures*. 2017;115:21–33. DOI: 10.1016/j.tws.2017.02.003.

2. Starovoitov EI, Nesterovich AV. The non-axisymmetric loading of an elastoplastic three-layer plate in its plane. *Journal of the Belarusian State University. Mathematics and Informatics*. 2022;2:57–69. Russian. DOI: 10.33581/2520-6508-2022-2-57-69.

3. Raeesi A, Ghaednia H, Zohrehheydariha J, Das S. Failure analysis of steel silos subject to wind load. *Engineering Failure Analysis*. 2017;79:749–761. DOI: 10.1016/j.engfailanal.2017.04.031.

4. Chen Bo, Zhong Pengpeng, Cheng Weihua, Chen Xinzhong, Yang Qingshan. Correlation and combination factors of wind forces on cylindrical roof structures. *International Journal of Structural Stability and Dynamics*. 2017;17(9):1750104. DOI: 10.1142/S0219455417501048.

5. Yin Caiyu, Jin Zeyu, Chen Yong, Hua Hongxing. Effects of sacrificial coatings on stiffened double cylindrical shells subjected to underwater blasts. *International Journal of Impact Engineering*. 2020;136:103412. DOI: 10.1016/j.ijimpeng.2019.103412.

6. Ren Shaofei, Song Ying, Zhang A-Man, Wang Shiping, Li Pengbo. Experimental study on dynamic buckling of submerged grid-stiffened cylindrical shells under intermediate-velocity impact. *Applied Ocean Research*. 2018;74:237–245. DOI: 10.1016/j.apor.2018.02.018.

7. Seo Jung Kwan, Song Chan Hee, Park Joo Shin, Paik Jeom Kee. Nonlinear structural behaviour and design formulae for calculating the ultimate strength of stiffened curved plates under axial compression. *Thin-Walled Structures*. 2016;107:1–17. DOI: 10.1016/j.tws.2016.05.003.

8. Grachev VA, Neustadt YuS. Buckling problems of thin elastic shells. *Computer Research and Modeling*. 2018;10(6):775–787. Russian. DOI: 10.20537/2076-7633-2018-10-6-775-787.

9. Yankovskii AP. Refined deformation model for metal-composite plates of regular layered structure in bending under conditions of steady-state creep. *Mechanics of Composite Materials*. 2017;52(6):715–732. DOI: 10.1007/s11029-017-9622-7.

10. Stupishin L, Nikitin K, Kolesnikov A. Numerical research orthotropic geometrically nonlinear shell stability using the mixed finite element method. *IOP Conference Series: Materials Science and Engineering*. 2017;201:012019. DOI: 10.1088/1757-899X/201/1/012019.

11. Maksimyuk VA, Storozhuk EA, Chernyshenko IS. Nonlinear deformation of thin isotropic and orthotropic shells of revolution with reinforced holes and rigid inclusions. *International Applied Mechanics*. 2013;49(6):685–692. DOI: 10.1007/s10778-013-0602-x.

12. Huang Sixin, Qiao Pizhong. A new semi-analytical method for nonlinear stability analysis of stiffened laminated composite doubly-curved shallow shells. *Composite Structures*. 2020;251:112526. DOI: 10.1016/j.compstruct.2020.112526.

13. Wang Jingchao, Li Zheng Liang, Yu Wei. Structural similitude for the geometric nonlinear buckling of stiffened orthotropic shallow spherical shells by energy approach. *Thin-Walled Structures*. 2019;138:430–457. DOI: 10.1016/j.tws.2018.02.006.

14. Obodan NI, Gromov VA. The complete bifurcation structure of nonlinear boundary problem for cylindrical panel subjected to uniform external pressure. *Thin-Walled Structures*. 2016;107:612–619. DOI: 10.1016/j.tws.2016.07.020.

15. Evdokimov AA. Methodology of use multithreaded programming for automated systems multiprocess galvanic lines. *Vestnik of Lobachevsky State University of Nizhny Novgorod*. 2013;1(3):306–312. Russian.

16. Mizin SV, Makhmutov VS, Maksumov OS, Kvashnin AN. [Application of multipipeline programming for a physical experiment]. *Kratkie soobshcheniya po fizike Fizicheskogo instituta imeni P. N. Lebedeva Rossiiskoi akademii nauk*. 2011;2:8–18. Russian.

17. Paznikov AA, Pavsky KV, Pavsky VA, Kupriyanov MS. [Modeling algorithms for optimizing the execution of critical sections on dedicated processor cores in multicore computing systems]. In: *Trudy II Mezhdunarodnoi nauchnoi konferentsii po problemam upravleniya v tekhnicheskikh sistemakh (CTS-2017); 25–27 oktyabrya 2017 g.; Sankt-Peterburg, Rossiya* [Proceedings of the 2nd International scientific conference on control problems in technical systems (CTS-2017); 2017 October 25–27; Saint Petersburg, Russia]. Saint Petersburg: Saint Petersburg Electrotechnical University «LETI»; 2017. p. 54–57. Russian.

18. Kadomskii AA, Zakharov VA. [Efficiency of multithreaded applications]. *Nauchnyi zhurnal*. 2016;7:26–28. Russian.

19. Fairushin RN, Yakupov IM. [Increasing the efficiency of programs using parallel computation of problems]. *Vestnik sovremennykh issledovanii*. 2018;6(3):583–585. Russian.

20. Karpov VV, Semenov AA. Refined model of stiffened shells. *International Journal of Solids and Structures.* 2020;199:43–56. DOI: 10.1016/j.ijsolstr.2020.03.019.

21. Bakusov PA, Semenov AA. Stability of toroidal shell segments at variation of a deflection angle. *PNRPU Mechanics Bulletin*. 2017;3:17–36. Russian. DOI: 10.15593/perm.mech/2017.3.02.

22. Nazarkov DA, Telyshev AV. [Comparative analysis of monolithic and microservice architectures of information systems]. *Dnevnik nauki*. 2019 [cited 2022 February 20];5:48–55. Available from: http://www.dnevniknauki.ru/images/publications/2019/5/technics/Nazarkov_Telyshev.pdf. Russian.

Получена 11.01.2023 / исправлена 19.03.2023 / принята 02.06.2023.
Received 11.01.2023 / revised 19.03.2023 / accepted 02.06.2023.