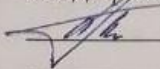


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра прикладной математики

Допустить к защите  
И.о. заведующего кафедрой  
канд. физ.-мат. наук, доц.  
 А.В. Письменский  
2023 г.

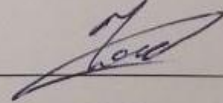
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)

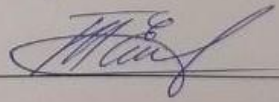
ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ ДЛЯ РАСЧЁТА  
И АНАЛИЗА ВОЛНОВЫХ КОЛЕБАНИЙ ПОРИСТО-УПРУГИХ СРЕД

Работу выполнил  К.В. Лукашкин

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) Программирование и информационные технологии

Научный руководитель  
канд. физ.-мат. наук, доц.  С.И. Фоменко

Нормоконтролер  
преподаватель  Е.С. Троценко

Краснодар  
2023

## РЕФЕРАТ

Выпускная квалификационная работа 56 с., 10 рис., 20 источн., 3 прил.

WEBSTORM, FORTRAN, ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ, NODE.JS, МНОГОСЛОЙНАЯ СРЕДА, ПОРИСТАЯ СРЕДА, ВИЗУАЛИЗАЦИЯ, ELECTRON, SVELTE, BOOTSTRAP, COUCHDB, POUCHDB

Цель выпускной квалификационной работы – создание приложения с графическим интерфейсом для расчета и анализа волновых полей в многослойных пористо-упругих средах

В процессе выполнения работы были добавлены модули расчета параметров и построения графиков дисперсионных кривых.

Реализация выполнена на платформе node.js, языка JavaScript и Fortran с использованием фреймворков: electron, svelte, couchdb, pouchdb, css фреймворка bootstrap.

## СОДЕРЖАНИЕ

Введение.....	4
1 Постановка задачи.....	6
1.1 Общие сведения.....	6
1.2 Математическая модель .....	7
1.3 Реализация математической модели .....	14
2 Программная платформа .....	16
2.1 Язык программирования .....	16
2.2 Используемые фреймворки и вспомогательные инструменты.....	17
3 Среда разработки.....	24
4 Графический интерфейс .....	26
Заключение .....	35
Список использованных источников .....	36
Приложение А Код компилятора Svelte .....	39
Приложение Б Работа с базой данных.....	45
Приложение В Изменение или удаление материала.....	51

## ВВЕДЕНИЕ

В настоящее время наблюдается недостаток программных продуктов с дружественным GUI (графический пользовательский интерфейс), способных проводить расчеты и визуализировать волновые колебания в различных средах. В контексте известных продуктов, можно выделить решение от итальянских авторов – многомерный аналитический инструмент Disperse, предназначенный для проведения анализа данных и визуализации их в различных форматах. Также, стоит отметить программное обеспечение, разработанное Giurgiutiu, которое занимает ведущие позиции на рынке в области инженерного моделирования и симуляции динамических систем. В связи с этим, математическое моделирование нестационарных процессов в водонасыщенных средах, основанное на модели двухкомпонентной среды, представляет собой актуальную и важную проблему. Актуальность данной проблемы обусловлена необходимостью регулирования уровня подземных вод, добычи газа и нефти, создания земляных сооружений, подземных тоннелей и других объектов, обеспечивающих устойчивость грунта [1]. Важным аспектом является также развитие общей теории многокомпонентных сред, которая включает создание математических моделей и аналитических и численных методов решения возникающих краевых задач и представляет интерес для дальнейших исследований в указанных областях.

Кроме того, задачи моделирования волновых процессов важны для развития дистанционного ультразвукового мониторинга на основе регистрации рассеянных бегущих упругих волн. Такие методы использовались с успехом в сейсмологии для определения поглощающих свойств среды, выявления резонансных явлений, оценки мощности источника возбуждения сейсмических волн. В отличие от традиционных методов, основанных на регистрации отраженных упругих волн, методы регистрации рассеянных бегущих упругих волн позволяют определить не только упругие

характеристики среды, но и её диссипативные свойства, снижающие качество изображения и точность оценки параметров среды [2].

Цель данной работы состоит в разработке приложения с графическим интерфейсом для расчета и анализа волновых полей в многослойных пористо-упругих средах. Для достижения указанной цели требуется выполнить разработку модулей, позволяющих осуществлять ввод параметров материалов слоистой пористо-упругой среды, а также разработать графический интерфейс, облегчающий процесс ввода данных для расчета и графической визуализации волновых полей при различных видах входных данных, включая поля смещений и дисперсионные кривые. В процессе работы были использованы такие технологии (фреймворки), как Node.js, Electron, Svelte, Bootstrap, PouchDB, CouchDB.

## **1 Постановка задачи**

### **1.1 Общие сведения**

Необходимость учета многокомпонентности возникает при решении многих прикладных задач, применяемых в различных областях человеческой деятельности. Однако сложности описания протекания различных процессов привели к тому, что до настоящего времени отсутствуют общепринятые модели для насыщенной пористой среды. Даже при значительном упрощении модель пористо-упругой среды существенно усложняется по сравнению с однородным упругим или вязкоупругим континуумом из-за способности жидкости втекать или вытекать в любую область, формируемую порами [3]. Это существенно отличает пористую среду от традиционной упругой среды, где концепция элементарного объема предполагает, что хотя материальные точки упругой среды могут двигаться относительно некоего положения равновесия, они всегда остаются в относительной близости друг к другу. В случае частичек поровой жидкости ситуация иная, поскольку некоторые из них могут удаляться на значительное расстояние от начального положения равновесия. Это особенно важно учитывать в решении задач, связанных с волновыми процессами, так как при росте частоты возрастает вклад динамического поведения жидкости, что значительно усложняет модель среды. Исторически сложилось множество подходов к описанию механики насыщенных пористо-упругих сред, которые активно развиваются и в настоящее время.

Один из подходов к описанию механики насыщенных пористо-упругих сред заключается в применении методов гидродинамики пористой среды и конечных элементов. Метод гидродинамики пористой среды основан на уравнениях Навье-Стокса и Больцмана, которые учитывают динамику движения жидкости внутри поровой структуры. Этот метод широко используется в исследованиях нефтегазовой промышленности и геологии.

Другой подход, основанный на применении методов конечных элементов, позволяет учитывать геометрические особенности пористой структуры и ее механические свойства. Моделирование реальных условий механических процессов в пористых средах позволяет получить более точные результаты и улучшить прогнозирование поведения системы в целом.

Однако, в текущем исследовании будет применена система акустических уравнений, разработанная М.А. Био в 1956 году, которая получила наибольшую известность и широкое распространение в научном сообществе. В работе будет рассмотрено использование данной системы уравнений для анализа волновых колебаний и их взаимодействия с веществом, с учетом соответствующих математических моделей и методов численного моделирования.

Цель данного исследования заключается в разработке и реализации доступного и интуитивно понятного для пользователя интерфейса, предназначенного для моделирования волновых процессов в пористо-упругой среде с учетом различных аспектов распространения волн в данной среде. Вместе с тем при моделировании допускается учет определенных условностей, включая теории, которые учитывают взаимосвязь между микроструктурными и макроструктурными параметрами, что может потребовать определения физических постоянных, которые обычно неизвестны. В рамках данного исследования будет уделено должное внимание разработке удобного и понятного интерфейса, а также выбору подходящих методов численного моделирования и анализа для решения поставленной задачи.

## **1.2 Математическая модель**

Согласно концепции Био пористо-упругой среды, данное вещество представляет собой комплексную систему, состоящую из упругого скелета и взаимосвязанных между собой пор. Пространство скелета занято вязкой,

сжимаемой жидкостью, которая способна передвигаться в поровом пространстве. Пористость данной среды определяется как отношение объема сообщающихся пор к общему объему.

В соответствии с ранней моделью Био, при построении точных измерений следует учитывать несколько условий. В частности, предполагается, что поры в данной среде равномерно распределены и не имеют четкой ориентации относительно друг друга. Кроме того, возможные пульсации смещений в пористом скелете могут быть пренебрежимо малыми. Также полагается, что упругий скелет является изотропным. Кроме того, для описания локальных деформаций в скелете и дилатации жидкости применяется упрощенная модель. Наконец, тензор вязких напряжений в жидкости может считаться пренебрежимо малым, а вязкость учитывается только в рамках межфазового взаимодействия. Следует отметить, что в более поздних работах Био была представлена обобщенная теория распространения волн, которая позволяет учитывать новые переменные. Среди них наиболее важными являются анизотропия, неоднородность пористости и диссипативные эффекты. Эти новые аспекты позволили получить более точные результаты при моделировании физических явлений в пористых средах, таких как распространение упругих и акустических волн в анизотропных слоях или распространение волн в неоднородных средах с различными пористыми структурами. Для учета вязкости следует обратить внимание на силу межсвязного взаимодействия.

Уравнения движения твердой и жидкой фаз имеют вид [4]:

$$\nabla_j \sigma_{ij}^s + \varrho \left( \frac{\partial u_i^f}{\partial t} - \frac{\partial u_i^s}{\partial t} \right) = \rho_{11} \frac{\partial^2 u_i^s}{\partial t^2} + \rho_{12} \frac{\partial^2 u_i^f}{\partial t^2} \quad (1)$$

$$\nabla_j \sigma^f - \varrho \left( \frac{\partial u_i^f}{\partial t} - \frac{\partial u_i^s}{\partial t} \right) = \rho_{12} \frac{\partial^2 u_i^s}{\partial t^2} + \rho_{22} \frac{\partial^2 u_i^f}{\partial t^2} \quad (2)$$

где



$$\sigma_{ij}^s = \lambda \epsilon^s \delta_{ij} + 2\mu \epsilon_{ij}^s + Q \epsilon^f \delta_{ij} \quad (3)$$

$$\sigma^f = Q \epsilon^s + R \epsilon^f \quad (4)$$

$$\sigma^f = -\varepsilon p \quad (5)$$

$$\epsilon^y = \nabla_i u_i^y \quad (6)$$

$$\epsilon_{ij}^s = \frac{1}{2} (\nabla_i u_j^s + \nabla_j u_i^s) \quad (7)$$

$$\rho_{11} = (1 - \varepsilon) \rho_s - \rho_{12} \quad (8)$$

$$\rho_{12} = \varepsilon (1 - \tilde{\alpha}) \rho_f \quad (9)$$

$$\rho_{22} = \varepsilon \rho_f - \rho_{12} \quad (10)$$

$\varrho$  – коэффициент диссипации (переход энергии процессов);

$\lambda, \mu, Q, R$  – константы связей пористой среды;

$\varepsilon$  – коэффициент пористости;

$p$  – давление жидкости;

$\epsilon^s, \epsilon^f$  – тензоры;

$u$  – вектор перемещения (средний по элементарному объему упругого скелета);

$\rho_{11}, \rho_{12}, \rho_{22}$  – инерционные коэффициенты;

$\tilde{\alpha}$  – коэффициент извилистости, характеризующий случайность порового пространства.

Путем подстановки можем получить уравнения Био, описывающие распространения волн в пористых водонасыщенных средах:

$$\nabla[(\lambda + \mu) \operatorname{div} u^s + Q \operatorname{div} u^f] + \mu \Delta u^s = \rho_{11} \frac{\partial^2 u_i^s}{\partial t^2} + \rho_{12} \frac{\partial^2 u_i^f}{\partial t^2} + \varrho \left( \frac{\partial u_i^f}{\partial t} - \frac{\partial u_i^s}{\partial t} \right) \quad (11)$$

$$\nabla[Q \operatorname{div} u^s + R \operatorname{div} u^f] = \rho_{12} \frac{\partial^2 u_i^s}{\partial t^2} + \rho_{22} \frac{\partial^2 u_i^f}{\partial t^2} - \varrho \left( \frac{\partial u_i^f}{\partial t} - \frac{\partial u_i^s}{\partial t} \right) \quad (12)$$

Потенциалы гармонических колебаний в теории Био рассматриваются как функции, которые описывают изменения давления и скорости движения жидкости в порах пористой среды. Решение  $u(x, \omega)$  может быть получено в виде суперпозиции гармонических составляющих [5]:

$$u(x, t) = \frac{1}{\pi} \operatorname{Re} \int_0^{\infty} u(x, \omega) e^{-i\omega t} d\omega \quad (13)$$

Здесь необходимо построить решение относительно комплексной амплитуды колебаний  $u(x, \omega)$ , возбуждаемых источником  $P_0(\omega, x) \exp(-i\omega t)$  (поверхностным или внутренним) относительно частотного спектра их исходного сигнала  $p_0(t, x)$ .  $e^{-i\omega t}$  – гармонический множитель.

Особое внимание стоит уделить модулям среды Био, а именно:

– модуль всестороннего сжатия

$$K_b = \lambda + \frac{2}{3}\mu - \frac{Q^2}{R} \quad (14)$$

– модуль всестороннего сжатия частиц скелета

$$K_s = \frac{K_b}{1 - \varepsilon \frac{Q}{R}} \quad (15)$$

– модуль всестороннего сжатия жидкости

$$K_f = \frac{QK_s}{K_s - \lambda - \frac{2}{3}\mu} \quad (16)$$

Все эти величины могут быть выведены экспериментально, и будут использоваться для выражения модулей упругих связей:

$$Q = \varepsilon(\Lambda_0 - \varepsilon)\Lambda_1 \quad (17)$$

$$R = \varepsilon^2\Lambda_1 \quad (18)$$

$$\lambda = K_b - \frac{2}{3}\mu + \frac{Q^2}{R} \quad (19)$$

$$\Lambda_0 = 1 - \frac{K_b}{K_s} \quad (20)$$

$$\Lambda_1 = \frac{K_s^2 K_f}{K_f(K_s - K_b) + \varepsilon K_s(K_s - K_f)} \quad (21)$$

В разрабатываемом программном комплексе предполагается изучение  $N$ -слойное полупространство  $D: -\infty < x, y < \infty, -\infty < z \leq 0$ . Это пространство состоит из  $N - 1$  слоя  $D_k: z_{k+1} < z \leq z_k$  толщины  $h_k = z_k - z_{k+1}$  и нижнего полупространства  $D_N: -\infty < z \leq z_N$ .

Для упругого слоя нестационарные колебания, характеризуются частотным спектром  $u(x, z, \omega)$  который определяется решением краевой задачи:

$$u(x, z, t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} u(x, z, \omega) e^{-i\omega t} d\omega \quad (22)$$

Для данной ситуации предполагается, что установившиеся колебания  $u(x, z) e^{-i\omega t}$  вызваны нагрузкой  $q(x) e^{-i\omega t}$  приложенной к поверхности ( $z = 0$ ). Здесь  $u(x, z)$  комплексная амплитуда вектора перемещений частиц среды в точке, а  $q(x)$  – функция, задающая распределение поверхности и нагрузки в области ее приложения.

Соответственно, комплексная амплитуда вектора смещений  $u$  удовлетворяет уравнению Ляме (в дальнейшем будет расширено):

$$(\lambda + \mu)\nabla\operatorname{div}u + \mu\Delta u + \rho\omega^2u = 0 \quad (23)$$

где

$\lambda, \mu$  – константы Ляме;

$\rho$  – плотность упругого слоя.

На верхней границе нагрузка приложена к свободной поверхности и соответственные граничные условия:

$$\tau_z(x, 0) = \begin{cases} q(x), & x \in \Omega \\ 0, & x \notin \Omega \end{cases} \quad (24)$$

где

$\Omega$  – ограниченная область.

Для решения задач, связанных с возбуждением гармонического волнового поля  $ue^{-i\omega t}$  заданным источником  $qe^{-i\omega t}$  в волноводе с плоско-параллельными границами (включая полупространства, слои и многослойные полупространства), широко применяется техника интегрального преобразования Фурье по горизонтальным координатам  $F_{xy}$ . Это позволяет получить решение в виде обратного интеграла Фурье от произведения Фурье-символов матрицы Грина среды  $K$  и вектора заданной нагрузки  $Q = F_{xy}[q]$ . Данная методика позволяет получать подробную информацию о механических свойствах среды и ее внутренней структуре [6]:

$$u(x) = F_{xy}^{-1}[KQ] = \frac{1}{\pi} \iint_{\Gamma_1\Gamma_2} K(a_1, a_2, z)Q(a_1, a_2)e^{-i(a_1x+a_2y)}da_1da_2 \quad (25)$$

$$x = (x, y, z)$$

Контурь интегрирования  $\Gamma_1$  и  $\Gamma_2$ , проходят вдоль вещественной оси и отклоняются от нее в комплексной плоскости при обходе вещественных полюсов  $\zeta_n$  элементов матрицы  $K$ . Описываемые вычеты в этих полюсах

играют важную роль в характеристиках поверхностных и каналовых волн, и используются при моделировании волновых процессов:

$$u_n(x) \sim \frac{a_n(\varphi, z)e^{\bar{v}_n r}}{\sqrt{r}}, \frac{r}{l} \gg 1 \quad (26)$$

где

$l$  – длина волны;

$r, \varphi$  – радиус и угол в полярных координатах.

К этим волнам, определяемым полюсами  $\zeta_n$  относятся поверхностные волны Релея и каналовые волны Лэмба.

Стационарное динамическое воздействие моделируется с помощью заданной поверхностной нагрузкой:

$$\tau_z(x, \omega)|_{z=0} = \begin{cases} q_0(x, y, \omega), & (x, y) \in D_0 \\ 0, & (x, y) \notin D_0 \end{cases} \quad (27)$$

где

$q_0(x, y, \omega)$  – комплексная амплитуда нагрузки;

$\tau_z(x, \omega)$  – комплексная амплитуда напряжения;

$D_0$  – приповерхностный слой.

Условия, регулирующие процессы, происходящие на проницаемых границах между пористо-упругими слоями  $D_{k-1}$  и  $D_k$  имеют вид:

$$[u]_k = 0, [\tau]_k = 0, [w]_k = 0, [p^f]_k = 0 \quad (28)$$

Квадратичными скобками обозначен скачок соответствующей функции от  $z$ :

$$[u]_k = \lim_{\delta \rightarrow 0} (u|_{z_k - \delta} - u|_{z_k + \delta}) \quad (29)$$

### 1.3 Реализация математической модели

Основное программное обеспечение, используемое для проведения вычислительных операций, реализовано в динамических библиотеках (или DLL) на языке программирования FORTRAN. В блоках кода, представленных в функциях и процедурах (блок кода, который может быть неоднократно вызван в разных местах программы - с возвратом управления в место вызова после его завершения), выполняется выбор систем исчисления и необходимых параметров для обеспечения эффективного решения задач [7].

Внутри динамических библиотек реализован алгоритм подбора параметров, в соответствии с рисунком 1, который зависит от выбранного типа слоя - упругого или пористого, а также от выбора вектора смещений или построения дисперсионных кривых [8]. Кроме того, эти библиотеки позволяют настраивать систему исчисления на основе заданных единиц или других необходимых значений. В этих же библиотеках производится расчет константных параметров и ключевых характеристик слоев.

```

Select Case(layer_type)
case(1) ! s
  select case(param_model)
  case(2) !vp,vs,rho
    write(str,'("\{ s | Vp = ", e14.5," | Vs = ",e14.5,"
    | Rho = ",e14.5," | hs = ",e14.5,"\}")') param(1),param(2),param(3),h
    res=1;
  end select
case(2) ! p
  select case(param_model)
  case(1) !eps, alpha, rho_f, K_f, rho_s, Ks, Kb, mu
    write(str,'("\{ p | Om = ", e14.5," | alpha = ",e14.5," | RhoF = ",e14.5,"
    | Kf = ",e14.5," | RhoS = ",e14.5," | Ks = ",e14.5," | Kb = ",e14.5," | mu = ",e14.5,"
    | hs = ",e14.5,"\}")') param(1:8),h
    res=1;
  end select
End Select

```

Рисунок 1 – Оператор сабрутины подбора параметров слоев, относительно его типа

В рамках рассматриваемой библиотеки реализованы дополнительные функции, которые позволяют осуществлять нормализацию константных переменных.

## 2 Программная платформа

### 2.1 Язык программирования

Приложение создавалось с помощью Node.js – программной среды, основанной на движке V8, позволяющем переводить язык JavaScript в машинный код. Она способна преобразовывать узкоспециализированный язык JavaScript в язык общего назначения, что расширяет его возможности использования. С помощью своего API, Node.js дает возможность JavaScript взаимодействовать с устройствами ввода-вывода и подключать внешние библиотеки, написанные на разных языках (в том числе Fortran), давая возможность вызывать их из JavaScript-кода. Node.js применяется преимущественно на сервере, где выступает в роли веб-сервера. Однако существуют возможности разработки десктопных оконных приложений, используя NW.js, AppJS или Electron для всех операционных систем. Также возможно программирование микроконтроллеров при помощи tessel, low.js и espruino.

В основу Node.js заложены принципы событийно-ориентированного и асинхронного программирования с неблокирующим вводом/выводом. Это подход, который позволяет задачам выполняться параллельно, что приводит к увеличению производительности приложения. Поэтому Node.js является высокоэффективной средой для создания сетевых приложений и веб-серверов в сфере IT-технологий. Кроме того, она позволяет уменьшить время ответа сервера благодаря асинхронной обработке запросов и встроенному механизму кэширования, что существенно повышает производительность приложений. С помощью Node.js также можно инициировать независимые процессы, что позволяет избежать проблемы остановки всего приложения в случае возникновения ошибки в одном из процессов. Node.js имеет огромное сообщество разработчиков, которые создают и поддерживают библиотеки (например, графические) и пакеты для этой среды. Такие библиотеки



расширяют ее возможности и делают его более гибким и удобным в использовании.

Ключевым преимуществом Node.js является его высокая скорость работы и низкое потребление ресурсов системы. Приложения, созданные в этой программной среде, работают с быстрой скоростью и могут обрабатывать большое количество запросов. Кроме того, Node.js обеспечивает повышенную отказоустойчивость, что является необходимым условием для работы критически важных приложений.

## **2.2 Используемые фреймворки и вспомогательные инструменты**

Node.js является низкоуровневой программной платформой, которая обеспечивает высокую производительность и эффективность при работе с языком JavaScript [9]. Однако для упрощения процесса разработки и облегчения труда программистов были созданы комплекты библиотек и фреймворков, которые расширяют функциональность и сокращают время разработки приложений для Node.js. Ниже приводится краткая информация по каждому используемому фреймворку:

1) Meteor представляет собой мощный фрустек-фреймворк, который реализует изоморфный подход к разработке приложений на языке JavaScript. Он обеспечивает возможность использования кода как на клиентской, так и на серверной стороне приложения, что существенно упрощает и ускоряет процесс разработки. В прошлом Meteor представлял собой полноценный инструмент, который включал в себя все необходимые компоненты и функциональность для успешной разработки приложений. Однако, в настоящее время этот фреймворк существенно трансформировался и интегрировался с популярными фронтенд-библиотеками – React, Vue и Angular. Meteor помимо возможности разрабатывать обычные веб-приложения, может быть использован в мобильной разработке, предоставляя возможность расширенной мобильной разработки на всех устройствах. Таким

образом, являясь мощным и эффективным инструментом, Meteor создает удобные исходные условия для разработки качественных, быстрых и удобных для использования веб-приложений на языке JavaScript [10]. Кроме того, он обладает широким спектром функциональных возможностей, таких как возможность создания в режиме реального времени, интеграции с дополнительными библиотеками и плагинами, возможность делать разные подключения к базам данных и соединяться с любым стеком технологий. Одним из главных преимуществ является его возможность работы с изоморфным подходом, что обеспечивает гладкое взаимодействие между клиентом и сервисом, сокращает время разработки и увеличивает быстродействие и эффективность приложений. Таким образом, Meteor представляет собой современный и развитый инструмент для разработки веб-приложений, который обладает широким спектром функциональных возможностей и возможностью использования в мобильной разработке. Сочетание изоморфного подхода и интеграции с фронтенд-библиотеками позволяет разработчикам создавать быстрые, масштабируемые и современные приложения на языке JavaScript с минимальными затратами времени и усилий.

2) Electron является open source фреймворком, позволяющим разрабатывать графические приложения для операционных систем с использованием веб-технологий. Решение комбинирует функциональность Node.js для работы с back-end и возможности веб-рендеринга библиотеки Chromium [11]. Для создания своих приложений с использованием Electron разработчикам доступен широкий спектр возможностей, таких как использование обширной библиотеки Node.js, возможность обмениваться сообщениями между процессами, а также применение современных веб-технологий, таких как HTML, CSS, JavaScript и другие. В свою очередь, интеграция с библиотекой Chromium позволяет реализовывать передовые функциональные возможности, такие как быстроперемещение рендеринга графических элементов в реальном времени, поддержку современных веб-стандартов, широкий и доступный контроль над элементами интерфейса, а

также многие другие. Таким образом, Electron представляет собой эффективный инструмент для создания мощных графических приложений веб-технологиями. Подробные инструкции и API-документация доступны на официальном сайте, что позволяет разработчикам автоматизировать некоторые процессы в рамках своих проектов и повышать производительность работы в целом. Он является популярным выбором для разработки таких приложений, как текстовые редакторы, IDE, мессенджеры и другие. Он совместим со многими операционными системами, включая Windows, macOS и Linux.

3) Svelte представляет собой принципиально новый подход к созданию пользовательских интерфейсов. В отличие от традиционных фреймворков, которые работают в браузере на этапе выполнения, Svelte выполняет большую часть своей работы на этапе компиляции при сборке приложения [12]. В результате этого, он использует более эффективный подход для обрабатывания изменения состояний пользовательского интерфейса. Вместо традиционных подходов к обновлению DOM-деревьев через сравнение состояний виртуального DOM, данный фреймворк предлагает написание кода точечного обновления DOM при изменениях состояния вашего приложения. Это позволяет значительно снизить нагрузку на производительность, поскольку изменения текста или других элементов на странице приводят к обновлению только измененных компонент. Кроме этого, Svelte обеспечивает более плавный и быстрый опыт работы с пользовательским интерфейсом, так как избегает частых перерисовок страницы при каждом изменении данных. Более того, он предоставляет более компактный и легковесный код, что способствует уменьшению размера и улучшению производительности создаваемых им приложений. Таким образом, Svelte представляет собой мощный инструмент для разработки пользовательских интерфейсов с более эффективной обработкой изменений состояний. Возможности фреймворка позволяют улучшать интерфейсы, создавая легковесные и динамичные страницы со сниженными нагрузками на производительность веб-браузера,

что является ключевым преимуществом и отличительной особенностью фреймворка.

4) Bootstrap – это универсальный фреймворк переднего плана для разработки веб-приложений, который предоставляет широкий набор компонентов и стилей для создания качественных пользовательских интерфейсов [13]. Он создан на основе языков HTML, CSS и JavaScript и имеет свободную лицензию для использования и распространения [14]. Особенность этого фреймворка заключается в том, что он позволяет разработчикам создавать высококачественные пользовательские интерфейсы с минимальным количеством кода. Это достигается благодаря тому, что Bootstrap содержит множество готовых компонентов, таких как кнопки, формы, таблицы, навигационные панели, модальные окна и многие другие, которые можно легко настраивать и сочетать между собой. Другой важной особенностью Bootstrap является его отзывчивость, что делает его идеальным для создания адаптивных веб-приложений, которые хорошо работают на различных устройствах и экранах. Кроме того, в него легко интегрируется с другими фреймворками и библиотеками, что дает возможность разработчикам выбирать наиболее подходящие инструменты для их проектов.

В настоящее время Bootstrap является одним из самых популярных CSS фреймворков для frontend разработки веб-сайтов и интерфейсов. Благодаря использованию готовых классов и компонентов, начинающие веб-разработчики могут создавать качественные сайты без необходимости во глубоких профессиональных знаниях. Также обеспечивает кроссбраузерность и кроссплатформенность, что гарантирует корректное отображение и функционирование веб-сайта во всех поддерживаемых браузерах и операционных системах.

Фреймворк включает в себя различные инструменты и компоненты, такие как обёрточные контейнеры, систему сеток, медиа-объекты, утилитные классы и другие. Для стилизации базового контента, такого как текст, изображения, таблицы и код, Bootstrap предлагает соответствующие классы.

Кроме того, фреймворк включает готовые компоненты, такие как кнопки, формы, навигационные панели, списки и другие, которые были протестированы огромным сообществом веб-разработчиков на различных устройствах.

Преимущества Bootstrap заключаются не только в возможности создания качественной адаптивной вёрстки, но и в возможности настройки фреймворка под свой проект. Это достигается через изменение SCSS переменных и использование миксинов, что позволяет изменить некоторые параметры, такие как количество колонок, цвета, радиус скруглений и отступы между колонками. Он также обеспечивает однородность дизайна и его согласованность между различными компонентами, что позволяет создать сайт в едином стиле. Другим преимуществом фреймворка является наличие огромного сообщества разработчиков и обучающих материалов, что позволяет быстро и эффективно освоить фреймворк, а также получить ответы на любые возникающие вопросы. В целом, использование Bootstrap позволяет значительно ускорить процесс frontend разработки и создать качественный адаптивный веб-сайт

5) PouchDB – база данных, которая написана на языке программирования JavaScript и обладает открытым исходным кодом. При создании данной базы данных было вдохновлено Apache CouchDB, однако, PouchDB имеет преимущества в использовании в браузере. Основная задача этой базы данных заключается в том, чтобы помочь веб-разработчикам в создании приложений, которые работают в автономном режиме и обеспечивают бесперебойное хранение данных, как в Интернете, так и на локальном устройстве [15]. База данных предоставляет удобный интерфейс для создания и управления данными, который легко интегрируется с различными фреймворками и библиотеками. Кроме того, она поддерживает более широкий спектр функций, чем многие другие базы данных JavaScript, такие как индексирование, фильтрация, репликация данных и многие другие. Исключительной особенностью PouchDB является гибкая система

синхронизации, которая может обеспечить синхронизацию данной базы данных с другими локальными и удаленными системами. Такая функциональность предоставляется прямо из «коробки», без необходимости дополнительной настройки и написания большого количества кода. Благодаря этим особенностям PouchDB становится эффективным инструментом, который может значительно облегчить создание масштабируемых и надежных веб-приложений.

б) CouchDB является примером кластерной базы данных, предоставляющей возможность запускать единственный сервер базы данных на любом количестве серверов или виртуальных машинах. Такой подход позволяет наладить настройку с одним узлом, и в результате улучшить пропускную способность и устойчивость системы. В качестве протокола обмена данными, CouchDB использует вездесущий HTTP и формат данных JSON, что обеспечивает легкую интеграцию с любым программным обеспечением, которое поддерживает эти технологии [16]. При этом данная база данных может хорошо работать с различными внешними инструментами, такими как прокси-серверы HTTP и балансировщики нагрузки. В отличие от CouchDB, решения PouchDB и Couchbase Lite были разработаны для использования соответственно в настольных и мобильных веб-браузерах и приложениях iOS и Android. Однако все эти базы данных имеют возможность легко реплицировать данные между другими базами данных. Общей характеристикой всех этих баз данных является открытость и легкость в использовании, что делает их популярным выбором для многих разработчиков веб-приложений.

В качестве вспомогательного инструмента (для полноценного внедрения динамических библиотек) был использован Fable JS – это компактный, но мощный компилятор, который позволяет разработчикам создавать веб-приложения, используя язык программирования F#. Этот инструмент переносит функциональное программирование на веб-платформу, обеспечивая высокую производительность приложений. Fable JS

обеспечивает высокую степень производительности и надежности, благодаря своей способности компилировать код F# в нативный JavaScript, используемый веб-браузерами [17]. Кроме того, компилятор обладает высокой экспрессивностью, что позволяет программистам использовать функциональные паттерны для создания мощных, масштабируемых веб-приложений. Главным преимуществом Fable JS заключается в том, что он обеспечивает совместимость с языками JavaScript и TypeScript, что позволяет использовать уже существующий код и компоненты в новых проектах.

### 3 Среда разработки

Для работы по реализации проекта было выбрано программное обеспечение JetBrains WebStorm, которое является инструментом для разработки web-сайтов и редактирования HTML, CSS и JavaScript кода [18]. Решение предоставляет возможность быстрой навигации по файлам и генерации уведомлений о появлении ошибок в коде в режиме реального времени. Большим преимуществом данной среды является возможность добавлять разметку HTML-документов или элементов SQL непосредственно в JavaScript. WebStorm обеспечивает унифицированный механизм для развертывания и синхронизации проектов через протокол FTP. Благодаря вышеперечисленным функциональным возможностям, использование данной среды разработки позволило значительно ускорить процесс создания приложения и обеспечить более эффективную работу над проектом.

Используя современные технологии кодирования, а именно HTML/XHTML и XML, интегрированная среда разработки WebStorm обеспечивает автоматическое завершение функциональности программного кода, такой как стили, ссылки, атрибуты и другие элементы. Кроме того, при работе с CSS, данное решение также предлагает автоматическое завершение кода классов, номеров HTML, ключевых слов и других атрибутов CSS, тем самым решая проблемы выбора формата, свойств, классов, ссылок на файлы и других атрибутов CSS .

Особенности WebStorm включают поддержку функциональности контроля версий и возможность предварительного просмотра кода, которые фиксируют все изменения и производимые действия. Благодаря созданию истории пользователи могут восстанавливать кодовые выражения, блоки и даже целые файлы. Программное обеспечение также включает в себя Code With Me – сервис для удаленной совместной разработки и парного программирования. Используя этот сервис, пользователи могут не только командно работать и общаться, оставаясь в IDE, но также создавать пул-



реквесты на GitHub, объединять их с целевой веткой, делиться настройками стиля кода и подключаться к баг-трекеру для работы из IDE [19]. В целом, указанные возможности существенно упрощают процесс разработки и обеспечивают более эффективную и быструю работу над проектами.

Кроме того, этот инструмент обеспечивает поддержку TypeScript, CoffeeScript, Dart, и других языков программирования, что позволяет разработчикам использовать различные языки программирования в одном проекте [20]. Это открывает возможности для создания более сложных и функциональных приложений, которые могут интегрироваться с различными платформами и технологиями. Еще он позволяет использовать множество плагинов, которые расширяют функциональность среды разработки. Эти плагины могут добавлять поддержку для различных фреймворков и технологий, а также упрощать процесс работы с файлами и редактирования кода.

В целом, WebStorm предоставляет мощный инструмент для разработки приложений, которые могут быть интегрированы с различными технологиями и фреймворками. Это делает его отличным выбором для работы с программным графическим интерфейсом.

## 4 Графический интерфейс

Согласно выполненной программной реализации, приложение с графическим интерфейсом обладает основным меню, включающим в себя четыре пункта.

При запуске пользователем, по умолчанию, открывается пункт «геометрия», изображено на рисунке 2.

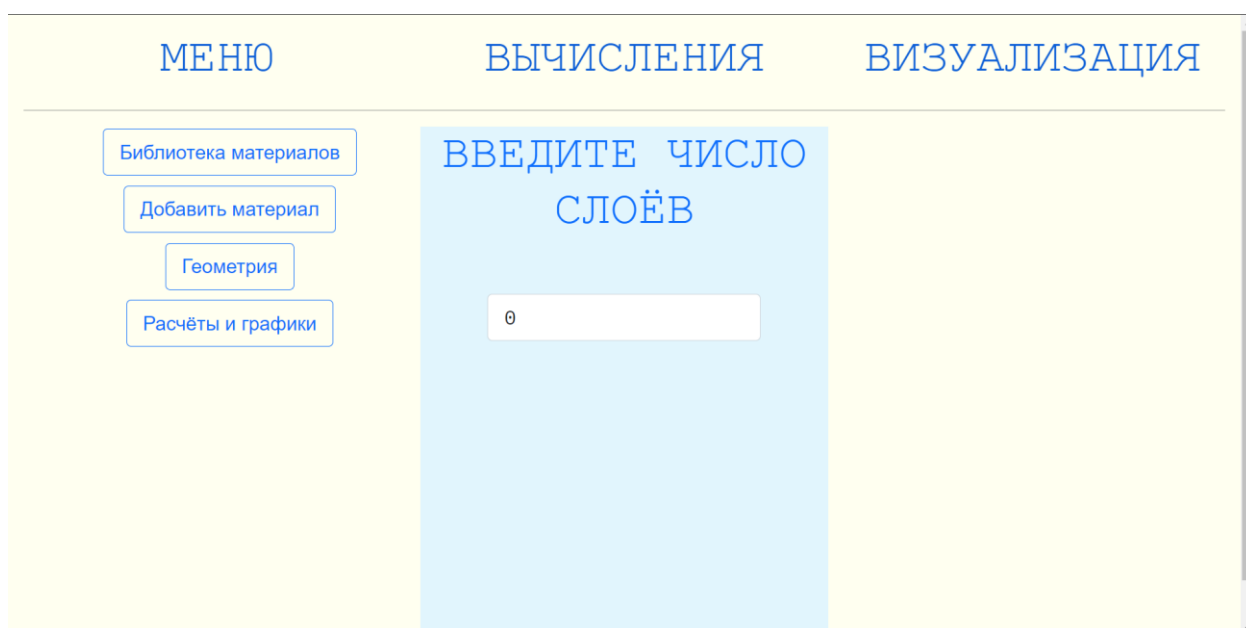


Рисунок 2 – Пункт «геометрия»

Для того чтобы задать модель континуума, необходимо произвести соответствующие настройки приложения. К данному этапу относится ввод количества слоев континуума в соответствующее поле «Введите число слоев». Затем в левой части приложения будет сформирована модель расположения слоев, изображенном на рисунке 3, а в правой части появятся поля для задания ключевых параметров каждого индивидуального слоя, таких как тип (пористый или упругий), материал, а также толщина слоя.

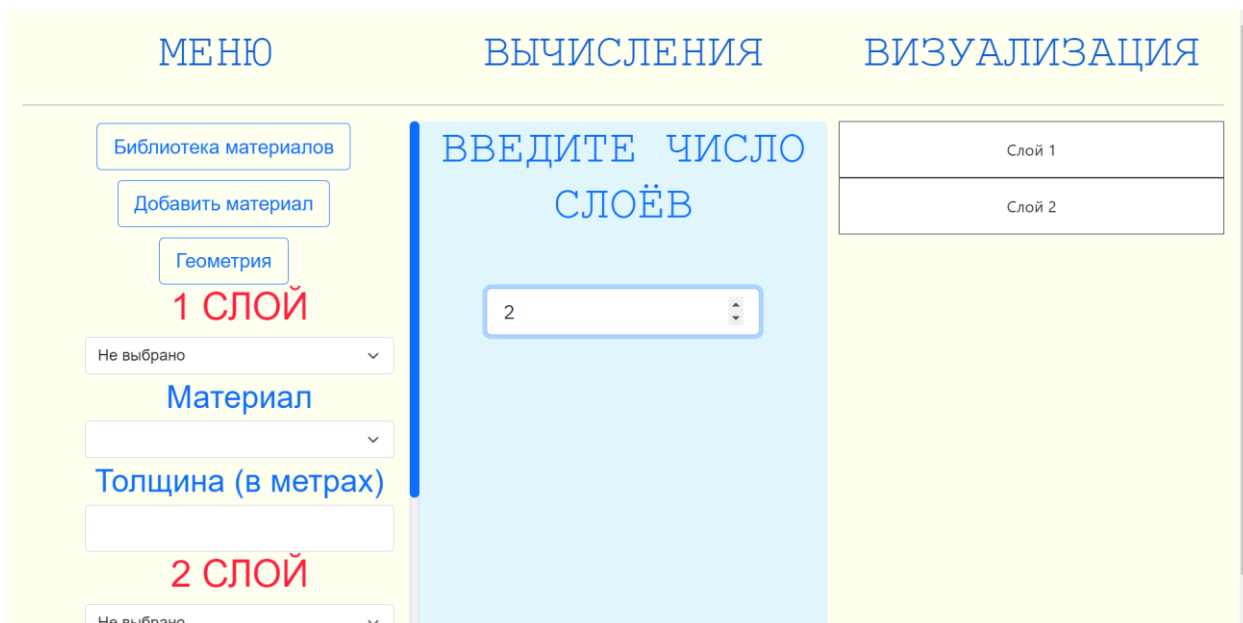


Рисунок 3 – Построение модели

В процессе занесения данных в приложении происходит постепенное формирование модели континуума, которая отображается в левой части интерфейса. При этом каждый тип слоя обозначается своей цветовой индикацией: упругие слои отображаются красным цветом, пористые – синим. В случае, если среда не была выбрана, то слой будет обозначен белым цветом, как на рисунке 4. Для мгновенного формирования отклика на вводимые данные, применяется технология реактивности. Данная функциональность предоставляется платформой Svelte и позволяет автоматически обновлять графический интерфейс при изменении состояния компонента.

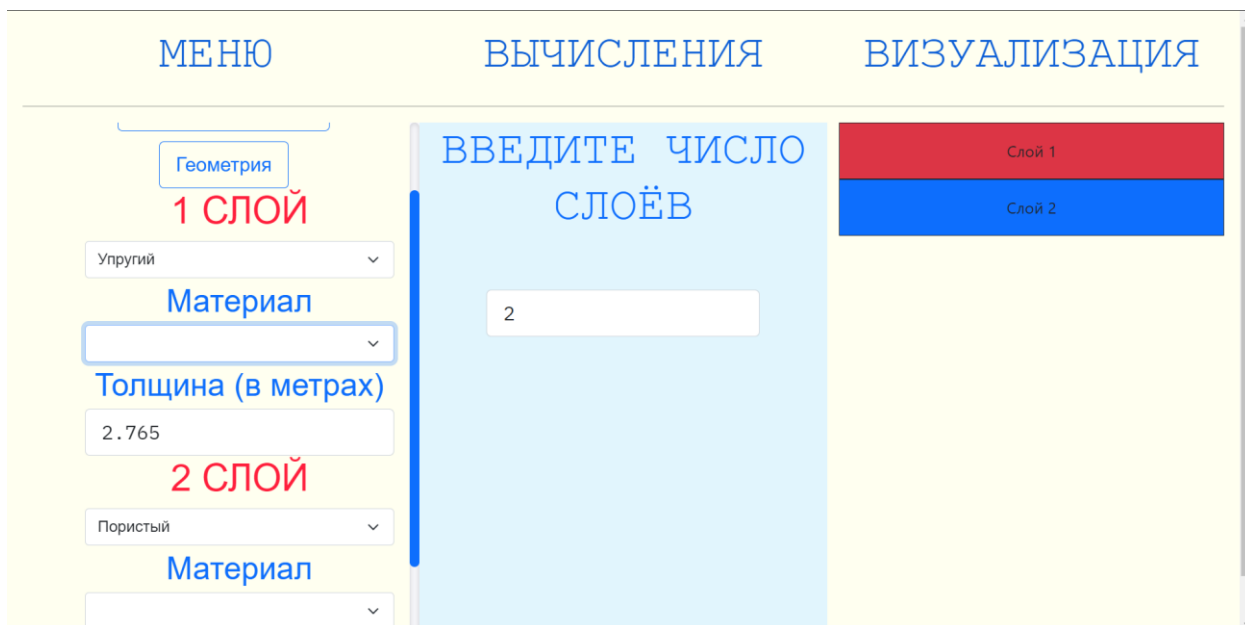


Рисунок 4 – Сформированная модель

В случае необходимости проверки параметров материалов во время создания модели, необходимо перейти в соответствующий пункт меню приложения, обозначенный как «Библиотека материалов», изображено на рисунке 5.

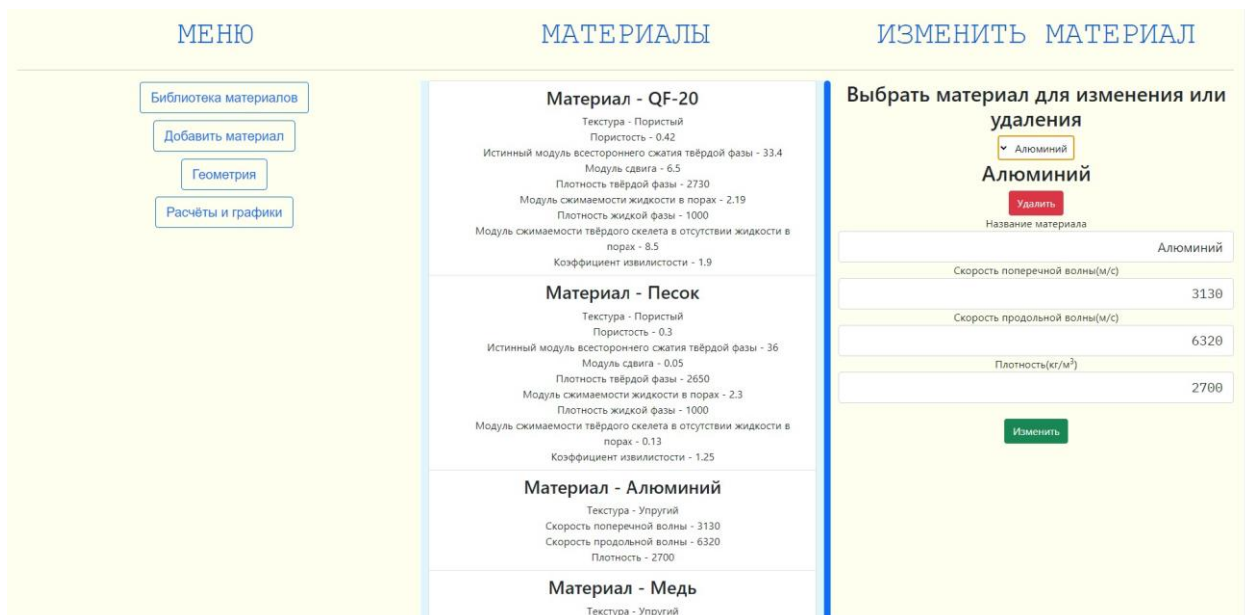


Рисунок 5 – Пункт «Библиотека материалов»

Центральная часть приложения содержит список сохраненных в базе данных PouchDB материалов, который включает набор параметров помимо соответствующего названия каждого материала.

Для упругого:

- скорость поперечной волны;
- скорость продольной волны;
- плотность.

Для пористого:

- пористость;
- истинный модуль всестороннего сжатия твердой фазы;
- модуль сдвига пористого скелета;
- плотность твердой фазы;
- модуль сжатия жидкости в порах;
- плотность жидкой фазы;
- модуль сжимаемости твердого скелета в отсутствии жидкости в порах;
- коэффициент извилистости.

При необходимости внесения изменений в параметры материалов, пользователь может воспользоваться функционалом приложения, расположенном в правой его части. Для этого следует выбрать в поле «Выбрать материал для изменения или удаления» соответствующее наименование в выпадающем списке, отображающем список всех доступных материалов. Возможны изменения любых параметров, в том числе и названия материала. Чтобы сохранить все изменения, нужно нажать кнопку «изменить». В случае, если материал утратил актуальность, его можно удалить, выбрав соответствующий материал из выпадающего списка и нажав кнопку «удалить». Следует отметить, что процедура удаления материала из базы данных носит безвозвратный характер, а также не предполагает возможности восстановления удаленного материала. Обратите внимание: ошибочное удаление материала навсегда лишит пользователя возможности использовать его в дальнейшем.

При возникновении ситуации, когда нужный пользователю материал отсутствует в библиотеке материалов или был удален по ошибке, ему следует воспользоваться пунктом меню «добавить материал», который позволит ему добавить материал и его параметры в базу данных, изображено на рисунке 6.

The screenshot shows a mobile application interface with a yellow background. At the top, there are two tabs: 'МЕНЮ' (Menu) on the left and 'ДОБАВИТЬ МАТЕРИАЛ' (Add Material) on the right. Under the 'МЕНЮ' tab, there are four buttons: 'Библиотека материалов' (Material Library), 'Добавить материал' (Add material), 'Геометрия' (Geometry), and 'Расчёты и графики' (Calculations and graphs). The 'ДОБАВИТЬ МАТЕРИАЛ' tab is active, showing a form for adding a material. The form has a title 'Пористый Упругий' (Porous Elastic) and a subtitle 'Название материала' (Material name). Below the title, there are several input fields with labels: 'Пористость' (Porosity), 'Истинный модуль всестороннего сжатия твёрдой фазы(ГПа)' (True bulk modulus of the solid phase (GPa)), 'Модуль сдвига(ГПа)' (Shear modulus (GPa)), 'Плотность твёрдой фазы(кг/м³)' (Density of the solid phase (kg/m³)), 'Модуль сжимаемости жидкости в порах(ГПа)' (Bulk modulus of liquid in pores (GPa)), 'Плотность жидкой фазы(кг/м³)' (Density of the liquid phase (kg/m³)), 'Модуль сжимаемости твёрдого скелета в отсутствии жидкости в порах(ГПа)' (Bulk modulus of the solid skeleton in the absence of liquid in pores (GPa)), and 'Коэффициент извилистости' (Tortuosity coefficient). Each input field contains the Greek letter theta (θ).

Рисунок 6 – Пункт «Добавить материал»

Возле верхнего края экрана интерфейса приложения пользователю предоставляется возможность выбрать тип материала, после чего ему становятся доступны параметры, соответствующие указанному типу материала. Для того, чтобы добавить новый материал в базу данных, необходимо заполнить все соответствующие параметры и нажать на кнопку «сохранить». После того, как будут успешно выполнены все этапы внесения материала в базу данных, он будет добавлен в библиотеку материалов и доступен для использования.

По завершении процесса моделирования и добавления соответствующих материалов, при условии проведения успешной проверки соответствия параметров, пользователь может переходить к последующему пункту, который назван «расчеты и графики», изображено на рисунке 7.

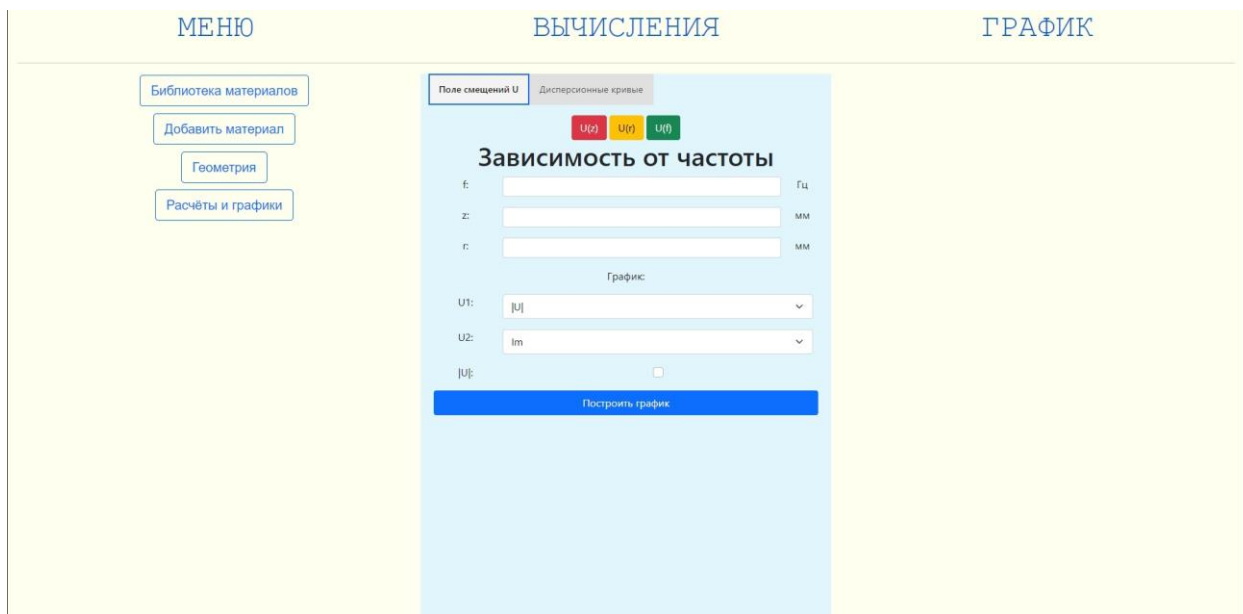


Рисунок 7 – Пункт «Расчеты и геометрия»

На данном шаге необходимо выбрать тип визуального представления данных, такой как построение дисперсионных кривых или графика зависимости компонент вектора смещений. При переходе в данный пункт интерфейса, по умолчанию открывается меню, относящееся к графику зависимости компонент вектора смещений. Здесь необходимо произвести выбор одного из трех вариантов смещения, а также задать частоты, величины смещений и определить тип векторов.

При выборе построения дисперсионных кривых в интерфейсе приложения открывается соответствующее окно, в котором возможно проводить дальнейшие расчеты и визуализировать полученные результаты в виде кривых зависимости различных параметров, изображено на рисунке 8.

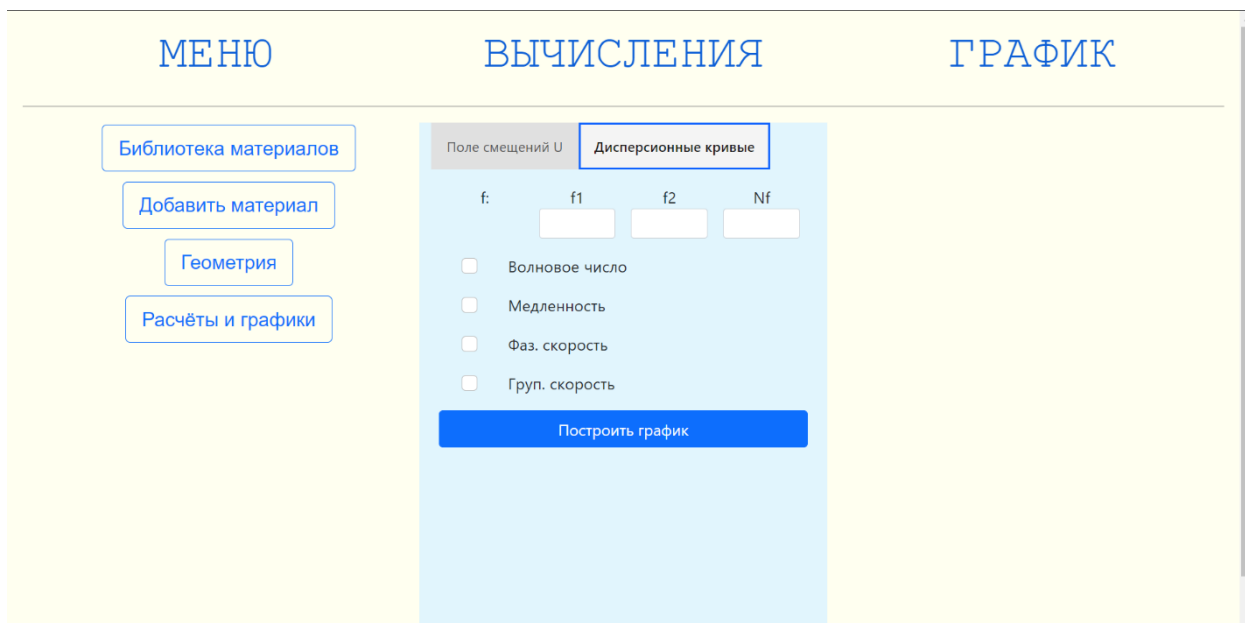


Рисунок 8 – Построение дисперсионных кривых

На данном этапе пользователю предоставляется возможность задать необходимые параметры для построения графика дисперсионных кривых. В частности, необходимо ввести диапазон частоты, задав предельные значения  $f1$  и  $f2$ , а также установить количество точек  $Nf$ , которые будут использованы для построения графика. Помимо этого, необходимо выбрать тип графика, соответствующий требуемой задаче, например, график дисперсионных прямых медленности или график дисперсионных кривых волнового числа.

После задания всех необходимых параметров пользователь должен нажать на кнопку "построить график". По завершении вычислений будет выполнено построение графика, отображающего значение волнового числа, медленности, фазовой скорости, групповой скорости от частоты колебаний. Полученные графические данные могут быть использованы для дальнейшего анализа свойств волновых колебаний.

Для демонстрации эффективности разработанного приложения будет представлен результат построения дисперсионных кривых волнового числа, осуществленный на диапазоне от 0 до 2 герц на основе 20 точек. Для проведения исследования был задан диапазон частоты колебаний, который позволил охватить весь интересующий диапазон значений, а также было



выбрано достаточное количество точек, которые обеспечили высокую точность построения графика дисперсионных кривых. После задания параметров, приложение автоматически произвело необходимые вычисления и построило график, отображающий зависимость волнового числа от частоты колебаний в интересующем диапазоне, изображено на рисунке 9.

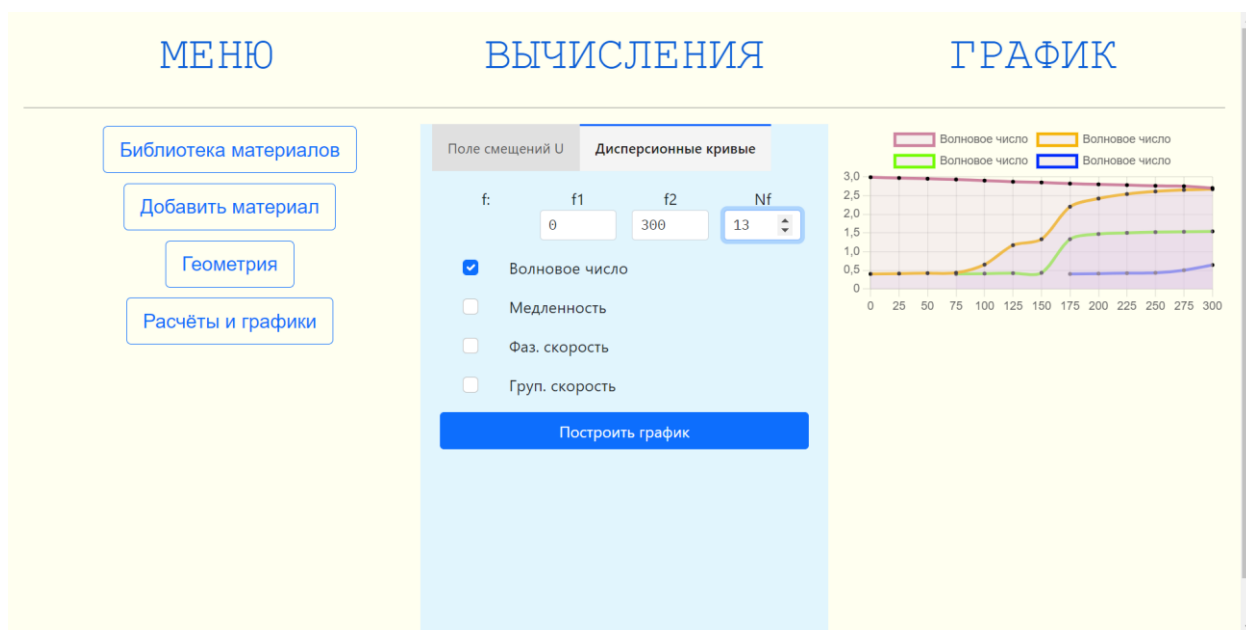


Рисунок 9 – Построение графика

График, полученный в результате построения дисперсионных кривых волнового числа, отображается в правой части рассматриваемого приложения. Каждая точка на графике является результатом обработки соответствующих значений частот. Для дополнительной проверки значений, полученных в процессе вычислений, пользователь приложения может обратиться к выводу значений в консоль приложения, где будут отображаться вычисленные параметры в соответствии с заданными входными условиями. Таким образом, приложение для построения дисперсионных кривых волнового числа обеспечивает удобный и надежный способ оценки свойств упругой среды на основе данных, предоставленных пользователем.

Следует обратить внимание на дополнительные функциональные возможности, реализованные в приложении. Одной из таких возможностей

является предоставление информации о параметрах, соответствующих конкретной точке на графике. При наведении курсора на определенную точку графика, на экране отобразится ее положение относительно оси частот, оси волновых чисел, а также будет предоставлена информация о назначении самого графика, изображено на рисунке 10. Такая функциональность позволяет пользователю получать дополнительную информацию о полученных результатах.

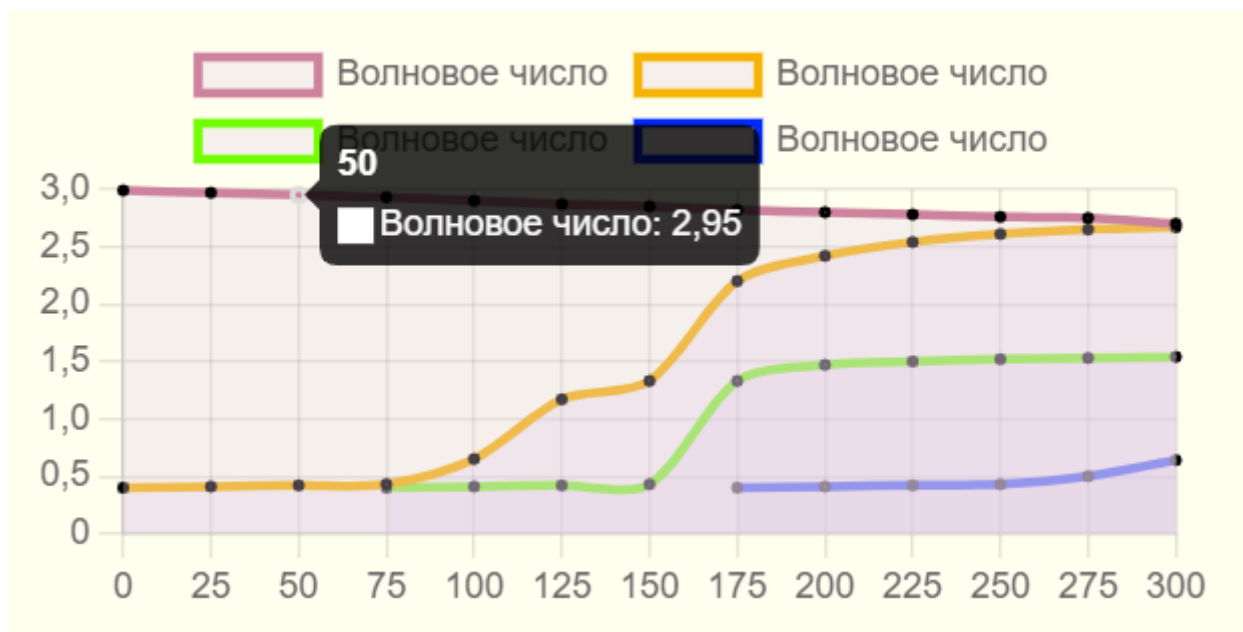


Рисунок 10 – Дополнительный функционал

## **ЗАКЛЮЧЕНИЕ**

В рамках выпускной квалификационной работы были изучены такие фреймворки как Electron, Svelte, Bootstrap для работы с десктопными приложениями. Была изучена среда WebStorm и платформа Node.js.

На основе всех технологий было сконструировано приложение с графическим интерфейсом, включающим модули ввода количества слоев, выбора среды для каждого слоя визуализация модели расположения слоев, ввода данных слоев различных сред, ввода параметров для расчета смещений, модуль расчета и графического представления.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Волны в пористо-упругих насыщенных жидкостью средах: научный журнал / Н. С. Горецкая // Акустичний вісник. – 2007. – № 2. – URL: [http://hydromech.org.ua/content/pdf/av/av-10-2\(43-63\).pdf](http://hydromech.org.ua/content/pdf/av/av-10-2(43-63).pdf). – Дата публикации: 24.04.2007.
2. Энциклопедия по машиностроению XXL: сборник фрагментов и выдержек: сайт. – URL: <https://mash-xxl.info/> (дата обращения: 20.12.2021).
3. Фоменко С. И. Поверхностные волны в материалах с функционально-градиентными покрытиями: научная статья / Е. В. Глушков, Н. В. Глушкова, С. И. Фоменко [и др.] // Акустический журнал. – 2012. – Т.58, № 3. – С. 370-385.
4. disserCat: электронная библиотека диссертаций: исследование волновых процессов в насыщенных упруго – пористых средах: сайт. – URL: <https://www.dissercat.com/content/isledovanie-volnovykh-protsesov-v-nasyshchennykh-uprugogo-poristykh-sredakh> (дата обращения: 30.04.2021).
5. Фоменко, С. И. Распространение и дифракция волн в слоистых пористо-упругих средах с плоскопараллельными и цилиндрическими границами: специальность 01.02.04 «Механика деформируемого твердого тела»: диссертация на соискание ученой степени кандидата физико-математических наук / Фоменко Сергей Иванович; Кубанский государственный университет – Краснодар, 2008. – 130 с. – Место защиты: Кубанский государственный университет. – Библиогр.: с. 14–113.
6. Бабешко, В. А. Динамика неоднородных линейно – упругих сред / В. А. Бабешко, Е. В. Глушков, Ж. Ф. Зинченко – МОСКВА «НАУКА», 1989 – 256 с. – ISBN 5-02-014001-5.
7. Справочное руководство по языку Fortran 95!: учебное пособие: сайт. – URL: <https://math.spbu.ru/user/rus/cluster/Doc/Library/fortran95/langref/> (дата обращения: 30.04.2023).

8. Coder lessons.com: онлайн-сообщество для IT-специалистов и разработчиков: сайт. – URL: <https://coderlessons.com/tutorials/kompiuternoe-programmirovaniye/vyuchit-fortran/fortran-obzor> (дата обращения: 27.04.2023).

9. Node.js Foreign Function Interface for N-API!: официальный сайт. – URL: <https://www.npmjs.com/package/ffi-napi?activeTab=readme> (дата обращения: 30.05.2023).

10. Vue.js: The progressive JavaScript Framework: официальный сайт – URL: <https://vuejs.org/> (дата обращения: 18.12.2021)

11. Electron Build cross-platform desktop apps with JavaScript: HTML и CSS: учебное пособие: сайт. – URL: <https://www.electronjs.org/> (дата обращения: 07.12.2021)

12. Svelte Cybernetically enhanced web apps: официальный сайт. – URL: <https://svelte.dev/> (дата обращения: 07.12.2021)

13. Bootstrap The most popular HTML, CSS and JS library in the world: официальный сайт. – URL: <https://getbootstrap.com/> (дата обращения: 19.12.2021)

14. Основы CSS – Изучение веб-разработки – MDN web Docs: The progressive JavaScript Framework : учебное пособие: сайт. – [https://developer.mozilla.org/ru/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics) (дата обращения: 18.12.2021)

15. PouchDB, the JavaScript Database that Syncs!: официальный сайт. – URL: <https://pouchdb.com/> (дата обращения: 18.04.2022).

16. Apache CouchDB: официальный сайт. – URL: <https://couchdb.apache.org/> (дата обращения: 18.04.2022).

17. Fable JavaScript you can be proud of!: официальный сайт. – URL: <https://fable.io/docs/2-steps/setup.html>(дата обращения: 12.05.2023).

18. Jet Brains возможности WebStorm: официальный сайт. – URL: <https://www.jetbrains.com/ru-ru/webstorm/features/> (дата обращения: 01.12.2021).

19. GitHub: Let's build from here: официальный сайт. – URL: <https://github.com/> (дата обращения: 22.12.2021).

20. TypeScript is JavaScript with syntax for types: учебное пособие: сайт. – URL: <https://www.typescriptlang.org/> (дата обращения: 22.12.2022).

## Приложения

### Приложение А код компилятора Svelte

```
let counter = 0;
  let n = 0;
  let numbers = [];
  let form_data = [];
  let data = [];
  let graph_data;
  let materials;
  async function getAllMaterials() {
    const allMaterials = await local_db.allDocs({
      include_docs: true
    });
    return allMaterials.rows.map(row => row.doc);
  }
function getMain() {
  counter = 0;
}

function getMaterials() {
  counter = 2;
}

function addMaterials() {
  counter = 3;
}
```

```

function getGraphForm() {
    counter = 1;
}

function control_list() {
    numbers = []
    for (let i = 1; i <= n; ++i) {
        numbers.push(i);
    }
    let old_length = form_data.length
    if (old_length > n) {
        for (let i = old_length; i > n; --i) {
            form_data.pop();
        }
    }
    else {
        for (let i = old_length; i < n; ++i) {
            form_data.push('white');
        }
    }
    numbers = numbers;
    form_data = form_data;
}

function get_data_from_slice(number) {
    form_data[number-1] =
document.getElementById(number.toString()).value;
    form_data = form_data;
}

onMount(async() => {
    materials = await getAllMaterials();

```



```

    })
</script>

<main>
  <div class="row">
    <div class="col-4 col-sm-4 col-md-4 col-lg-4">
      <h1>Меню</h1>
    </div>
    <div class="col-4 col-sm-4 col-md-4 col-lg-4">
      {#if counter < 2}
        <h1>Вычисления</h1>
      {:else if counter === 2}
        <h1>Материалы</h1>
      {:else}
        <h1>Добавить материал</h1>
      {/if}
    </div>
    <div class="col-4 col-sm-4 col-md-4 col-lg-4">
      {#if counter < 1}
        <h1>Визуализация</h1>
      {:else if counter === 1}
        <h1>График</h1>
      {:else if counter === 2}
        <h1>Изменить материал</h1>
      {:else }
      {/if}
    </div>
  </div>
  <hr>
  <div class="row">

```

```

<div class="col-4 col-sm-4 col-md-4 col-lg-4" id="element">
  <ul style="padding-right: 4px">
    <li class="pb-2"><button class="btn btn-outline-primary
btn-lg" on:click={getMaterials}>Библиотека материалов</button></li>
    <li class="pb-2"><button class="btn btn-outline-primary
btn-lg" on:click={addMaterials}>Добавить материал</button></li>
    <li class="pb-2">
      <button class="btn btn-outline-primary btn-lg"
on:click={getMain}>Геометрия</button>
    <ul>
      <form method="post">
        {#each numbers as number}
          <li>
            <h2>{number} слой</h2>
            <select id = {number}
class="form-select" aria-label="Default select example" on:change =
{get_data_from_slice(number)}>
              <option selected
value="light">Не выбрано</option>
              <option
value="danger">Упругий</option>
              <option
value="primary">Пористый</option>
            </select>
            <label>Материал</label>
            <select class="form-select" >
              {#each materials as
material, i}
                <option
value={i}>{material.material}</option>

```

```

        {/each}
    </select>
    <label>Толщина (в
метрах)</label>

    <input class="form-control
form-control-lg" id="InputH" type="number step="any">
        </li>
    {/each}
</form>
</ul>
</li>
<li class="pb-2"><button class="btn btn-outline-primary
btn-lg" on:click={ getGraphForm }>Расчёты и графики</button></li>
</ul>
</div>
<div class="col-4 col-sm-4 col-md-4 col-lg-4" id="element"
style="background-color: #e1f5fd">
    {#if counter === 0}
        <h1 class="text-primary">Введите число слоёв</h1>
        <form name="input" method="post" class="pt-5">
            <label><input class="form-control form-control-
lg" type="number bind:value={n} on:change = {control_list}></label>
        </form>
    {:else if counter === 2}
        <GetDB materials={materials}/>
    {:else if counter === 3}
        <AddDB />
    {:else}
        <GraphForm />
    {/if}

```

```

</div>

<div class="dleft col-4 col-sm-4 col-md-4 col-lg-4" id="element">
  {#if counter === 0}
  {#each form_data as color, i}
    <div class="border border-1 border-dark p-3 bg-{color}
text-dark">Слой {i+1}</div>
  {/each}
  {:else if counter === 1}
    <DrawChart />
  {:else if counter === 2}
    <UpdateMatherial materials="{materials}" />
  {:else}
  {/if}
</div>

</div>

</main>

```

## Приложение Б

### работа с базой данных

```
<script>
import PouchDB from 'pouchdb-browser';
const local_db = new PouchDB('db');
const remote_db = new PouchDB("http://localhost:5984/db");
let material = "";
// упругие
let vs = 0;
let vp = 0;
let rho = 0;
// пористые
let eps = 0;
let ks = 0;
let mu = 0;
let rhoS = 0;
let kf = 0;
let rhoF = 0;
let kb = 0;
let alpha = 0;
async function addElasticMaterial(event) {
  const newElasticMaterial = {
    material: material,
    texture: 1,
    params: {
      vs: vs,
      vp: vp,
      rho: rho
    }
  }
}
```

```

    }
    const sendDB = await local_db.post(newElasticMaterial)
  }
  async function addPorousMaterial(event) {
    const newPorousMaterial = {
      material: material,
      texture: 2,
      params: {
        eps: eps,
        ks: ks,
        mu: mu,
        rhoS: rhoS,
        kf: kf,
        rhoF: rhoF,
        kb: kb,
        alpha: alpha
      }
    }
    const sendDB = await local_db.post(newPorousMaterial)
  }
  let count = 1;

  function elastic() {
    count = 0;
  }

  function porous() {
    count = 1;
  }

```

```
</script>
```

```
<div class="form-check">
```

```
  <input class="form-check-input" type="radio" name="flexRadioDefault"
  id="flexRadioDefault1" checked on:click={porous}>
```

```
  <label class="form-check-label" for="flexRadioDefault1">
```

```
    <h2 class="text-primary">Пористый</h2>
```

```
  </label>
```

```
</div>
```

```
<div class="form-check">
```

```
  <input class="form-check-input" type="radio" name="flexRadioDefault"
  id="flexRadioDefault2" on:click={elastic}>
```

```
  <label class="form-check-label" for="flexRadioDefault2">
```

```
    <h2 class="text-primary">Упругий</h2>
```

```
  </label>
```

```
</div>
```

```
{#if count === 0}
```

```
  <form on:submit={addElasticMaterial}>
```

```
    <div class="form-group">
```

```
      <label for="InputMaterial">Название материала</label>
```

```
      <input class="form-control form-control-lg" id="InputMaterial" type="text"
  bind:value={material}>
```

```
    </div>
```

```
    <div class="form-group">
```

```
      <label for="InputVs">Скорость поперечной волны(м/с)</label>
```

```
      <input class="form-control form-control-lg" id="InputVs" type="number"
  step="any" bind:value={vs}>
```

```
    </div>
```

```
    <div class="form-group">
```

```
      <label for="InputVp">Скорость продольной волны(м/с)</label>
```

```

    <input class="form-control form-control-lg" id="InputVp" type="number
step="any" bind:value={ vp }>
  </div>
  <div class="form-group">
    <label for="InputRho">Плотность(кг/м<sup>3</sup></label>
    <input class="form-control form-control-lg" id="InputRho" type="number
step="any" bind:value={ rho }><br>
  </div>
  <button type="submit" class="btn btn-primary">Добавить</button>
</form>
{:else}
<form on:submit={ addPorousMaterial }>
  <div class="form-group">
    <label for="InputMaterial1">Название материала</label>
    <input class="form-control form-control-lg" id="InputMaterial1" type="text
step="any" bind:value={ material }>
  </div>
  <div class="form-group">
    <label for="InputEps">Пористость</label>
    <input class="form-control form-control-lg" id="InputEps" type="number
step="any" bind:value={ eps }>
  </div>
  <div class="form-group">
    <label for="InputKs">Истинный модуль всестороннего сжатия твёрдой
фазы(ГПа)</label>
    <input class="form-control form-control-lg" id="InputKs" type="number
step="any" bind:value={ ks }>
  </div>
  <div class="form-group">
    <label for="InputMu">Модуль сдвига(ГПа)</label>

```



```

    <input class="form-control form-control-lg" id="InputMu" type="number
step="any" bind:value={ mu }><br>
  </div>
  <div class="form-group">
    <label for="InputRhoS">Плотность твёрдой
фазы(кг/м<sup>3</sup>)</label>
    <input class="form-control form-control-lg" id="InputRhoS" type="number
step="any" bind:value={ rhoS }><br>
  </div>
  <div class="form-group">
    <label for="InputKf">Модуль сжимаемости жидкости в
порах(ГПа)</label>
    <input class="form-control form-control-lg" id="InputKf" type="number
step="any" bind:value={ kf }><br>
  </div>
  <div class="form-group">
    <label for="InputRhoF">Плотность жидкой
фазы(кг/м<sup>3</sup>)</label>
    <input class="form-control form-control-lg" id="InputRhoF" type="number
step="any" bind:value={ rhoF }><br>
  </div>
  <div class="form-group">
    <label for="InputKb">Модуль сжимаемости твёрдого скелета в
отсутствии жидкости в порах(ГПа)</label>
    <input class="form-control form-control-lg" id="InputKb" type="number
step="any" bind:value={ kb }><br>
  </div>
  <div class="form-group">
    <label for="InputAlpha">Коэффициент извилистости</label>

```

```
<input class="form-control form-control-lg" id="InputAlpha" type="number
step="any" bind:value={ alpha}><br>
</div>
<button type="submit" class="btn btn-primary">Добавить</button>
</form>
{/if}
```

## Приложение В

### Изменение или удаление материала

script>

```
import PouchDB from 'pouchdb-browser';
export let materials;
const local_db = new PouchDB('db');
const remote_db = new PouchDB("http://localhost:5984/db");
let selected = false;
let type = 0;

let selected_material;
let m_id;
function select_material() {
  selected = true;
  m_id = selected_material._doc_id_rev
  type = selected_material.texture
  type=type;
}
async function removeMaterial(event) {
  local_db.get(m_id).then(function (doc) {
    return local_db.remove(doc);
  });
}
async function addPorousMaterial(event) {
  const newPorousMaterial = {
    material: selected_material.material,
    texture: 2,
```

```

    params: {
      eps: selected_material.params.eps,
      ks: selected_material.params.ks,
      mu: selected_material.params.mu,
      rhoS: selected_material.params.rhoS,
      kf: selected_material.params.kf,
      rhoF: selected_material.params.rhoF,
      kb: selected_material.params.kb,
      alpha: selected_material.params.alpha
    }
  }
  const sendDB = await local_db.put(newPorousMaterial)
}
async function addElasticMaterial(event) {
  const newElasticMaterial = {
    material: selected_material.material,
    texture: 1,
    params: {
      vs: selected_material.params.vs,
      vp: selected_material.params.vp,
      rho: selected_material.params.rho
    }
  }
  const sendDB = await local_db.put(newElasticMaterial)
}

```

</script>

<h2>Выбрать материал для изменения или удаления</h2>

```

<form>
  <select bind:value={selected_material} on:change="{select_material}">
    {#each materials as material}
      <option value={material}>
        {material.material}
      </option>
    {/each}
  </select>
</form>
  {#if selected}
    <h2>{selected_material.material}</h2>
    <button on:click={removeMaterial} class="btn btn-
danger">УДАЛИТЬ</button>
    {#if type === 0}
    {:else if type === 1}
      <form on:submit={addElasticMaterial}>
        <div class="form-group">
          <label for="InputMaterial">Название материала</label>
          <input class="form-control form-control-lg" id="InputMaterial"
type=text bind:value={selected_material.material}>
        </div>
        <div class="form-group">
          <label for="InputVs">Скорость поперечной волны(м/с)</label>
          <input class="form-control form-control-lg" id="InputVs" type=number
step="any" bind:value={selected_material.params.vs}>
        </div>
        <div class="form-group">
          <label for="InputVp">Скорость продольной волны(м/с)</label>

```

```
<input class="form-control form-control-lg" id="InputVp" type="number"
step="any" bind:value={selected_material.params.vp}>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputRho">Плотность(кг/м3)</label>
```

```
<input class="form-control form-control-lg" id="InputRho"
type="number" step="any" bind:value={selected_material.params.rho}><br>
```

```
</div>
```

```
<button type="submit" class="btn btn-success">Изменить</button>
```

```
</form>
```

```
{:else}
```

```
<form on:submit={addPorousMaterial}>
```

```
<div class="form-group">
```

```
<label for="InputMaterial1">Название материала</label>
```

```
<input class="form-control form-control-lg" id="InputMaterial1"
type="text" step="any" bind:value={selected_material.material}>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputEps">Пористость</label>
```

```
<input class="form-control form-control-lg" id="InputEps" type="number"
step="any" bind:value={selected_material.params.eps}>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputKs">Истинный модуль всестороннего сжатия
твёрдой фазы(ГПа)</label>
```

```
<input class="form-control form-control-lg" id="InputKs" type="number"
step="any" bind:value={selected_material.params.ks}>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputMu">Модуль сдвига(ГПа)</label>
```

```
<input class="form-control form-control-lg" id="InputMu" type="number" step="any" bind:value={selected_material.params.mu}><br>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputRhoS">Плотность твёрдой фазы(кг/м3)</label>
```

```
<input class="form-control form-control-lg" id="InputRhoS" type="number" step="any" bind:value={selected_material.params.rhoS}><br>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputKf">Модуль сжимаемости жидкости в порах(ГПа)</label>
```

```
<input class="form-control form-control-lg" id="InputKf" type="number" step="any" bind:value={selected_material.params.kf}><br>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputRhoF">Плотность жидкой фазы(кг/м3)</label>
```

```
<input class="form-control form-control-lg" id="InputRhoF" type="number" step="any" bind:value={selected_material.params.rhoF}><br>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputKb">Модуль сжимаемости твёрдого скелета в отсутствии жидкости в порах(ГПа)</label>
```

```
<input class="form-control form-control-lg" id="InputKb" type="number" step="any" bind:value={selected_material.params.kb}><br>
```

```
</div>
```

```
<div class="form-group">
```

```
<label for="InputAlpha">Коэффициент извилистости</label>
```

```
<input class="form-control form-control-lg" id="InputAlpha" type="number" step="any" bind:value={selected_material.params.alpha}><br>
```

```
</div>  
<button type="submit" class="btn btn-success">Изменить</button>  
</form>  
{/if}  
{/if}
```



## СПРАВКА

о результатах проверки текстового документа  
на наличие заимствований

Кубанский Государственный университет

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Лукашкин К. В.  
Самоцитирование  
рассчитано для: Лукашкин К. В.  
Название работы: ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ ДЛЯ РАСЧЁТА И АНАЛИЗА ВОЛНОВЫХ  
КОЛЕБАНИЙ ПОРИСТО-УПРУГИХ СРЕД  
Тип работы: Выпускная квалификационная работа  
Подразделение: ФКТиПМ, кафедра прикладной математики

### РЕЗУЛЬТАТЫ

■ ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ: НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

СОВПАДЕНИЯ	18.87%	СОВПАДЕНИЯ	18.87%
ОРИГИНАЛЬНОСТЬ	75.18%	ОРИГИНАЛЬНОСТЬ	75.18%
ЦИТИРОВАНИЯ	5.95%	ЦИТИРОВАНИЯ	5.95%
САМОЦИТИРОВАНИЯ	0%	САМОЦИТИРОВАНИЯ	0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 14.06.2023

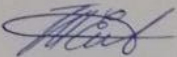
ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 14.06.2023 09:31

Структура документа: Проверенные разделы: титульный лист с.1, основная часть с.2, 4-33, содержание с.3, библиография с.34-36, приложение с.37-54  
Модули поиска: ИПС Адилет; Библиография; Сводная коллекция ЭБС; Интернет Плюс\*; Сводная коллекция РГБ; Цитирование; Переводные заимствования (RuEn); Переводные заимствования по eLIBRARY.RU (EnRu); Переводные заимствования по Интернету (EnRu); Переводные заимствования издательства Wiley; eLIBRARY.RU; СПС ГАРАНТ: аналитика; СПС ГАРАНТ: нормативно-правовая документация; Модуль поиска "КубГУ"; Медицина; Диссертации НББ; Коллекция НБУ; Перефразирования по eLIBRARY.RU; Перефразирования по СПС ГАРАНТ: аналитика\*; Перефразирования по Интернету; Перефразирования по Интернету (EN); Перефразирования по коллекции издательства Wiley; Патенты СССР, РФ, СНГ; СМИ России и СНГ; Шаблонные фразы; Кольцо вузов; Издательство Wiley; Переводные заимствования

Работу проверил: Троценко Екатерина Сергеевна

ФИО проверяющего

Дата подписи: 14.06.2023

  
Подпись проверяющего



Чтобы убедиться  
в подлинности справки, используйте QR-код,  
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование  
корректным, система оставляет на усмотрение проверяющего.  
Предоставленная информация не подлежит использованию  
в коммерческих целях.