

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“Московский Государственный Университет
Имени М. В. Ломоносова”**

**Механико-Математический Факультет
Кафедра: “Вычислительная Механика”**

**Курсовая Работа: “Реализация задачи Лэмба с помощью программного
обеспечения CAE Fidesys. Ее приложение к распространению
сейсмических волн”**

**“Implementation of Lamb's task using CAE Fidesys software. Its application to
the propagation of seismic waves.”**

Выполнил студент:

426 группа

Уханов Евгений Михайлович

Научный руководитель:

Доктор физико-математических наук

Анатолий Викторович Вершинин

Москва.

2025г.

Оглавление:

- 1) Вступление.
- 2) Историческая справка.
- 3) Актуальность Задачи Лэмба.
- 4) Научная новизна Задачи Лэмба.
- 5) Практическая ценность Задачи Лэмба.
- 6) Постановка целей и задач работы.
- 7) Преобразования Лэмба.
- 8) Аналитическое решение задачи Лэмба.
 1. Физическая постановка задачи.
 2. Математическая постановка задачи.
 3. Вычисление аналитического решения задачи Лэмба.
- 9) Моделирование задачи Лэмба с помощью МСЭ.
 1. Свойства геометрической модели в CAE Fidesys.
 2. Код программы, строящий модель для программного обеспечения CAE Fidesys.
 3. Галерея распространения волн для различных порядков (100 сетка)
 4. График сравнения численного и аналитического решений (100 сетка)
 5. Разница численных и аналитического решений (100 сетка)
 6. График времени вычисления в моделях различных порядков (100 сетка)
 7. Код программы, добавляющий демпфирующий слой в модель
 8. Галерея распространения волн для различных порядков (50)
 9. График сравнения численного и аналитического решений (50)
 10. Разница численных и аналитического решений (50 сетка)
 11. График времени вычисления в моделях различных порядков (50 сетка)
 12. Галерея распространения волн для различных порядков (25)
 13. График сравнения численного и аналитического решений (25)
 14. Разница численных и аналитического решений (25 сетка)
 15. График времени вычисления в моделях различных порядков
- 10) Расчёты числа точек на длину волны (PPW).
- 11) Результаты работы и Выводы.
- 12) Литература и источники.

Раздел №1: Вступление.

Вопросы динамического взаимодействия упругих тел с внешними нагрузками играют важную роль в механике сплошных сред и инженерных приложениях. Одной из классических задач в этой области является задача Лэмба — исследование отклика упругого полупространства на действие сосредоточенной силы, приложенной к его границе.

Задача Лэмба — это задача о динамическом воздействии сосредоточенной силы на границу полупространства или полуплоскости, меняющейся по времени (**внешняя задача Лэмба**), и аналогичная задача с силой, меняющейся во времени и приложенной внутри упругого полупространства или полуплоскости (**внутренняя задача Лэмба**). Внутренняя и внешняя задачи Лэмба представляют огромный интерес для геофизики. Причиной этого стало то, что данная задача широко используется при моделировании различных волновых процессов, сопровождающих, например, взрывы или землетрясения.

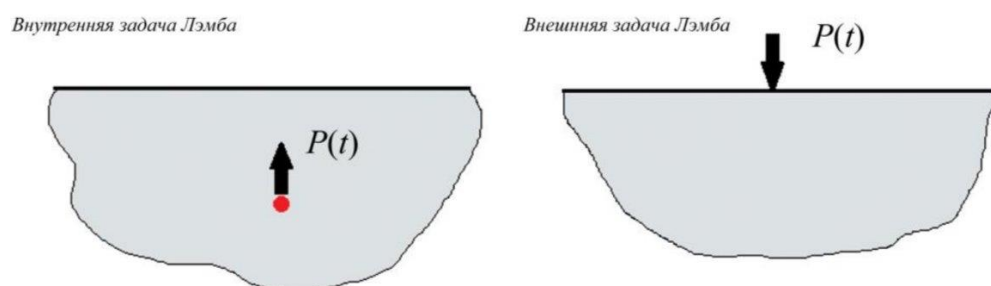


Рисунок 1: Внутренняя и внешняя задачи Лэмба.

Задача Лэмба имеет фундаментальное значение как в теоретической, так и в прикладной механике. Ее изучение важно для понимания распространения сейсмических волн, анализа поведения оснований под воздействием динамических нагрузок, проектирования конструкций, взаимодействующих с упругими средами, а также в области неразрушающего контроля материалов.

В данной задаче источник действует как импульс, приложенный в точке свободной поверхности твердого полупространства по нормали к ней. При таком воздействии будут наблюдаться три волновых фронта — это **волна Рэлея**, а также **продольная** и **поперечная волны**.

Однако, термин "Задача Лэмба" в данный момент времени относят и к более общему случаю произвольного источника в среде с одной границей.

В данной работе рассматривается постановка задачи Лэмба, основные методы ее решения, а также анализ особенностей распространения волн в упругом полупространстве при динамическом воздействии сосредоточенной силы.

Раздел №2: Историческая Справка:

Задача о динамическом воздействии сосредоточенной силы на границу упругого полупространства была впервые рассмотрена британским математиком и физиком **Горацием Лэмбом** (Horace Lamb) в 1904 году. В своей работе [1] он исследовал проблему распространения волн в упругих средах и вывел аналитическое решение для возмущений, возникающих при приложении мгновенной силы к поверхности полупространства.

Теория, необходимая для объяснения сейсмических данных, принадлежит одной из двух категорий: изучаются либо асимптотические приближения для волн, распространяющихся в реалистических моделях Земли, либо точные решения для крайне идеализированных сред. В 1904 г. Лэмб [2] дал точное решение задачи второго типа. Работы Лэмба [1, 2] содержали большинство важных для моделирования распространения сейсмических волн в упругой среде элементов.

Работы Лэмба основывались на более ранних исследованиях лорда Рэлея [3], который впервые описал поверхностные волны, распространяющиеся вдоль границы упругого тела. Волны Рэлея играют важную роль в сейсмологии, поскольку они ответственны за значительную часть разрушительных эффектов землетрясений.

Джон Уильям Стретт, третий барон Рэлей (12 ноября 1842–30 июня 1919) — британский физик и механик, который в 1855 году открыл волну, которая может распространяться вдоль плоской границы твердого упругого полупространства. Но распространяется данная волна медленнее, чем поперечная объемная волна, а также ее амплитуда быстро затухает с глубиной. Данная волна впоследствии была названа в его честь волной Рэлея. Также он смог определить, что скорость этой волны:

$$C_R \approx 0,6 C_S, \text{ где } C_S - \text{ скорость поперечной волны.}$$

Волны Рэлея образуются при динамическом воздействии на поверхности упругих тел. Для определения скорости поверхностной волны Рэлея, учёный получил следующий полином шестой степени:

$$\alpha^6 - 8\alpha^4 + 8(3 - 2\beta^2)\alpha^2 - 16(1 - \beta^2) = 0$$

Где, α и β — это **фазовые скорости продольных и поперечных волн**. Они выражаются через **волновые числа продольных и поперечных волн** следующим образом:

$$\alpha = \frac{k_t}{k} = \frac{c}{c_t} \quad \text{и} \quad \beta = \frac{k_l}{k_t} = \frac{c_t}{c_l}$$

Лэмб рассматривал воздействие точечной нагрузки на плоскую поверхность, исследуя динамические возмущения в твердом теле. В своей работе от 1904 года Лэмб получил сложные интегральные уравнения для определения распространения колебаний по поверхности твердого тела. В дальнейшем решение данных задач было сведено к интегральным уравнениям относительно скалярного и векторного потенциалов.

Однако, полученные Лэмбом интегральные представления оказались довольно сложными: в замкнутом виде решение получили лишь для гармонической нагрузки, а также в пространстве изображений в случае произвольной зависимости от времени.

В 1930 году **Раймон Каньяр (Raymond Cagniard)** защитил кандидатскую диссертацию, в которой показал, что можно вычислять интегральные представления, полученные Лэмбом, считая частоту не действительной величиной, а комплексной. Это позволяет обходить особенности в знаменателе интегралов, используя метод, известный как **метод Каньяра** [4].

Метод Каньяра начал широко применяться для решения всевозможных задач, связанных с внешними и внутренними задачами Лэмба. Он позволяет формировать решение во временной области непосредственно из интегрального представления в частотной области без использования обратного преобразования Фурье [5].

Работы Лэмба и Каньяра заложили основу для дальнейших исследований в области сейсмологии, динамики грунтов и механики сплошных сред. Их результаты оказались особенно важны для понимания природы сейсмических волн, включая поверхностные волны Рэлея, которые играют ключевую роль в распространении землетрясений.

Например, в 1984 году с помощью похожей техники В.Б. Поручиков [6] получил формулы для определения горизонтального и вертикального смещения на расстоянии «х» от точки приложения силы, если сила является дельта импульсом по времени:

$$U_y(x, \theta, t) = -\frac{P}{\pi \mu C_S^2 x} \operatorname{Im} \left(\frac{\sqrt{C_P^{-2} - \theta^2}}{S(\theta^2)} \right);$$

$$U_x(x, \theta, t) = -\frac{P\theta}{\pi \mu x} \operatorname{Im} \left(\frac{2\sqrt{C_P^{-2} - \theta^2} \cdot \sqrt{C_S^{-2} - \theta^2} - C_S^{-2} + 2\theta^2}{S(\theta^2)} \right);$$

Где: $S(\theta^2) = (C_S^{-2} - 2\theta^2)^2 + 4\theta^2 \cdot \sqrt{C_P^{-2} - \theta^2} \cdot \sqrt{C_S^{-2} - \theta^2}$

В дальнейшем исследования динамического воздействия на полупространство активно развивались. Значительный вклад в изучение поверхностных акустических волн внесли И.А. Викторов [7] и А.М. Антонов [8], которые рассмотрели особенности распространения волн в упругих средах с различными физическими свойствами.

Современные исследования направлены на анализ распространения волн в сложных средах, включая пористые и градиентно-упругие материалы [9]. Численные методы, такие как метод конечных элементов, активно применяются для расчёта параметров акустических волн в пьезоэлектрических материалах [10].

Волновые процессы в грунтах также имеют важное практическое значение, например, при оценке состояния оснований зданий и тоннелей по данным сейсмотомографии [11]. Кроме того, устройства на поверхностных акустических волнах используются в современных системах связи [12].

Раздел №3: Актуальность Задачи Лэмба:

На мой взгляд, актуальность задачи Лэмба в механике связана с необходимостью понимания поведения упругих волн при различных динамических нагрузках, что имеет важное значение в сейсмологии, строительной механике и акустике, например, для проектирования и анализа различных конструкций, инженерных систем, зданий и т. д.

Например, в инженерии задача Лэмба может быть использована для анализа динамического поведения сооружений, таких как мосты, здания, дамбы, а также для проектирования и диагностики различных механических систем, обеспечения безопасности инфраструктуры и т. д.

Также, внутренняя и внешняя задачи Лэмба представляют исключительный интерес для геофизики, поскольку широко используются при моделировании волновых процессов, сопровождающих землетрясения. Задача Лэмба имеет применение в геофизике и для предсказания поведения горных пород при землетрясениях. В геодезии задача Лэмба может быть использована для анализа деформаций в земной коре и исследования ее структуры.

В медицине и биологии задача Лэмба может быть применена для изучения взаимодействия упругих волн с тканями организма, разработки методов обнаружения и лечения заболеваний, а также для улучшения качества изображений в медицинской диагностике (например, в ультразвуковых технологиях).

Раздел №4: Научная Новизна Задачи Лэмба:

Изначально предложенная Горацием Лэмбом в начале XX века, задача получила дальнейшее развитие благодаря новым математическим методам и численным подходам. Современные исследования расширяют класс рассматриваемых материалов и условий нагружения, что углубляет понимание динамического поведения сложных систем.

Научная новизна задачи Лэмба в механике может проявляться в разработке более точных моделей распространения упругих волн, учете нелинейных эффектов, разработке новых методов численного моделирования и анализа.

Также возможно изучение влияния различных параметров на поведение упругих волн в изучаемом пространстве. На мой взгляд, новые методы численного моделирования упругих волн в полупространстве могут обеспечить более точное и эффективное моделирование различных физических процессов.

Раздел №5: Практическая ценность Задачи Лэмба:

Практическая ценность задачи Лэмба в механике заключается в ее применении для анализа, проектирования и улучшения различных инженерных систем, конструкций и механических устройств. Ниже приведены основные области, где задача Лэмба имеет практическое значение:

- 1) **Проектирование конструкций:** за счет анализа упругих волн и динамического воздействия, задача Лэмба позволяет оптимизировать проектирование механических систем и конструкций, учитывая их поведение при различных динамических нагрузках.
- 2) **Диагностика и контроль:** задача Лэмба может применяться для диагностики состояния инженерных систем, обнаружения дефектов, трещин и деформаций, а также для контроля качества материалов и конструкций. Это позволяет прогнозировать и предотвращать возможные повреждения конструкций, обеспечивая их долговечность и безопасность.
- 3) **Безопасность и надежность:** изучение упругих волн и их воздействия на конструкции помогает повысить безопасность сооружений и механических устройств, предотвращать аварии, а также обеспечить их долговечность и надежность.
- 4) **Медицинская диагностика:** в медицине задача Лэмба может быть применена для улучшения качества изображений при ультразвуковых исследованиях и диагностике различных заболеваний. Также она может сыграть роль в разработке новых методов терапии и лечения.
- 5) **Геофизика и сейсмология:** задача Лэмба имеет применение в изучении сейсмических явлений, предсказании землетрясений и оценке влияния упругих волн на поведение горных пород и земной коры.

Раздел №6: Постановка Целей и Задач Работы:

- 1) Прodelать интегральные преобразования Лэмба. Получить неоднородные уравнения Гельмгольца для скалярного и векторного потенциалов.
- 2) Получить формулы аналитического решения задачи Лэмба. Построить графики зависимости перемещений от времени.
- 3) Написать код, строящий модель в CAE Fidesys для численных расчетов задачи Лэмба.
- 4) Реализовать численное решение задачи Лэмба. Построить графики изменения перемещения от времени.
- 5) Сравнить численные решения для различных порядков с аналитическим решением.
- 6) Написать код, строящий демпфирующий слой для f_c модели.
- 7) Проанализировать аналитическое и численное решения задачи Лэмба для трех сеток (25, 50, 100). Добиться совпадения численного и аналитического решений для какой-нибудь из сеток, для какого-нибудь порядка.
- 8) Провести сравнение аналитического и численного решений задачи Лэмба. Для численного решения задачи Лэмба на различных порядках МСЭ (вплоть до 9-го) добиться разницы с аналитическим решением, не превышающей 1%. Выполнить анализ зависимости числа точек на длину волны Рэлея (points per wavelength, PPW) от порядка аппроксимации.
- 9) Заключительные выводы.

Раздел №7: Преобразования Лэмба.

1. Уравнения движения:

Запишем уравнения движения Навье в изотропной среде для построения решений в форме Ламе — Клайперона:

$$(\lambda + 2\mu)\nabla\text{div}u - \mu\text{rot}\text{rot}u + b = \rho u''$$

где λ и μ — константы Ламе; u — поле перемещений; b — поле массовых сил; ρ — плотность среды.

2. Разложение полей в потенциалы:

Воспользуемся представлением Гельмгольца для поля массовых сил:

$$b = -\nabla\alpha - \text{rot}\beta$$

где α и β — скалярный и векторный потенциалы.

И представлением Ламе — Грина для динамического поля перемещений:

$$u = \nabla\phi + \text{rot}\psi$$

где ϕ и ψ — скалярный и векторный потенциалы.

3. Подстановка в уравнения движения:

Подставим последние два выражения в уравнения движения и применим преобразования Лапласа по переменной времени. Получим неоднородные уравнения Гельмгольца для скалярного и векторного потенциалов:

$$(\lambda + 2\mu)\nabla^2\text{div}u + \nabla b = \rho \frac{\partial^2}{\partial t^2}\text{div}u$$

Так как $\text{div}u = \nabla^2\phi$ и $\nabla b = -\nabla^2\alpha$ получаем уравнения:

$$(\lambda + 2\mu)\nabla^2\nabla^2\phi - \nabla^2\alpha = \rho \frac{\partial^2}{\partial t^2}\nabla^2\phi$$

Если массовые силы отсутствуют $b = 0 \Rightarrow \alpha = 0$ и $\beta = 0$. Тогда упрощая:

$$\frac{\partial^2\phi}{\partial t^2} = c_p^2\nabla^2\phi \quad \text{где } c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}}$$

Аналогично, применяя оператор ротор к уравнению движения:

$$\mu\nabla^2\text{rot}u - \text{rot}b = \rho \frac{\partial^2}{\partial t^2}\text{rot}u$$

С учетом: $\text{rot}u = \text{rot}\psi$ и $\text{rot}b = -\nabla^2\beta$ получаем: $\mu\nabla^2\nabla^2\psi - \text{rot}\beta = \rho \frac{\partial^2}{\partial t^2}\nabla^2\psi$. Откуда

$$\frac{\partial^2\psi}{\partial t^2} = c_s^2\nabla^2\psi \quad \text{где } c_s = \sqrt{\frac{\mu}{\rho}}$$

Раздел №8: Аналитическое решение задачи Лэмба.

1) Физическая постановка задачи

Среда: рассматривается полупространство (или полуплоскость) с постоянной плотностью ρ , модулем Юнга E , и коэффициентом Пуассона ν , что соответствует линейной, однородной и изотропной упругой среде.

Механизм воздействия: внешняя сила действует на поверхность полупространства, создавая волновое поле, которое распространяется вглубь материала. Величина силы зависит от времени и может быть задана по закону Берлаге, создавая импульсное воздействие.

Необходимо найти поле перемещений $u(x, y, z, t)$ на некотором расстоянии от источника сейсмической нагрузки. Сосредоточенная сила приложена в центре свободной полуплоскости.

Решение задачи: для анализа реакции среды на эту силу используется теория упругих волн, описываемая системой уравнений движения для смещений u_x, u_y, u_z , которые зависят от координат и времени.

2) Математическая постановка задачи

Для упругой среды, поддерживающей волновые процессы, уравнения движения для смещения (вектор $u = (u_x, u_y, u_z)$) имеют вид:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \nabla \cdot \sigma_i$$

где σ_i - тензор напряжений, связанный с деформациями среды через закон Гука для упругих материалов:

$$\sigma_i = \lambda \cdot \text{tr}(\varepsilon)I + 2\mu\varepsilon_i$$

где ε — тензор деформаций, λ и μ — параметры Ламе, которые связаны с модулем Юнга E и коэффициентом Пуассона ν через следующие соотношения:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \text{ и } \mu = \frac{E}{2(1+\nu)}$$

Таким образом, для данной задачи сила, действующая на поверхность полупространства, приводит к волновым возбуждениям, которые распространяются внутрь среды, и мы можем решить задачу для каждой компоненты смещения.

Сила, действующая на полупространство, задана по закону Берлаге:

$$f(t) = A \cdot \frac{\omega_1 \cdot e^{\omega_1 t}}{4} \left(\sin(\omega_0 t) \left(-\frac{t^2}{\omega_1} + \frac{t}{\omega_1^2} + \frac{1}{\omega_1^3} \right) - \cos(\omega_0 t) \sqrt{3} \left(\frac{t^2}{\omega_1} + \frac{t}{\omega_1^2} \right) \right)$$

где: A – амплитуда силы, $\omega_1 = \frac{\omega_0}{\sqrt{3}}$ – параметр, связанный с частотой затухания и $\omega_0 = 2\pi\omega$ – частота силы.

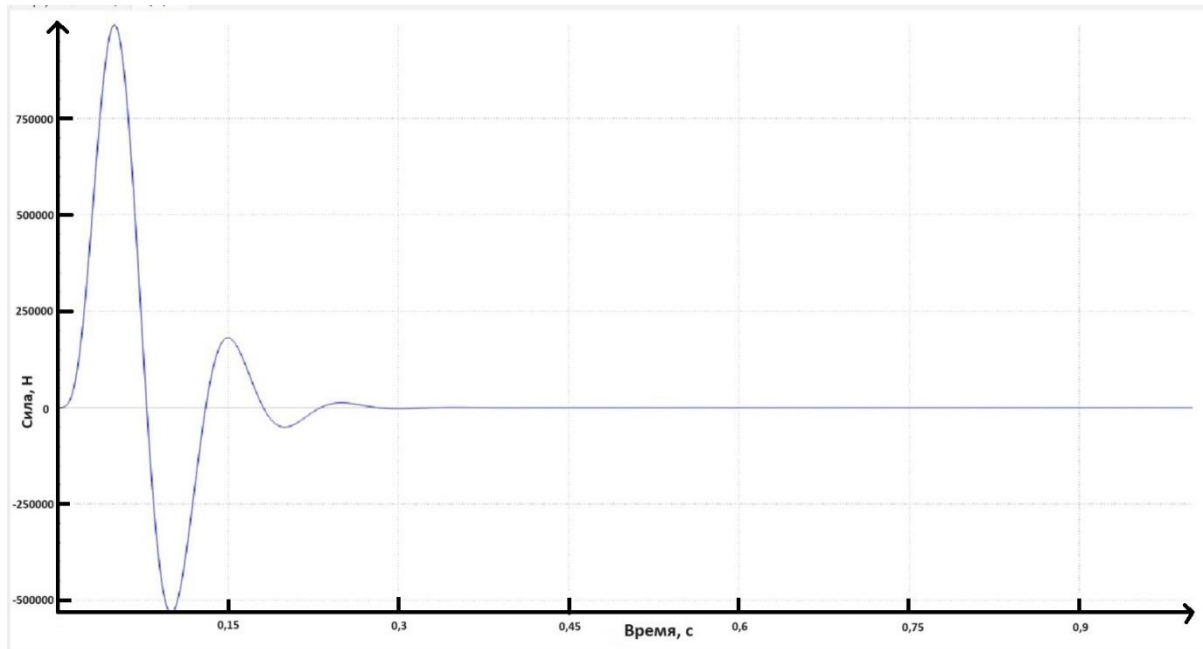


Рисунок 2: График силы Берлаге.

Эта сила вызывает импульсное воздействие на полупространство, создавая волны, которые распространяются в среде. Такие волны, в зависимости от её свойств, могут быть как продольными, так и поперечными.

3) Вычисление аналитического решения задачи Лэмба.

Для решения задачи Лэмба мы вычисляем смещения в точке (r, z) полупространства, используя интеграл по волновым числам:

$$u_x(r, z, t) = \int_0^\infty (R_P \cdot \exp(-\alpha_P z) + R_S \cdot \exp(-\alpha_S z)) \cdot J_1(kr) \cdot F(t) dk \quad (1)$$

где: R_P и R_S – амплитуды для продольных и поперечных волн, $J_1(kr)$ – функция Бесселя первого рода, $F(t)$ – сила, действующая на поверхность.

Интегрируя по всем волновым числам k , получаем решение для смещения в любой момент времени.

1) Запишем уравнения движения:

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \sum_j \frac{\partial \sigma_{ij}}{\partial x_j} + f_i$$

где, ρ – это плотность, t – время, а σ_{ij} – компоненты тензора напряжений:

$$\sigma_{ij} = \lambda \cdot \delta_{ij} \nabla \cdot u + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

2) Решение через потенциалы Лэмба:

Введение продольного ϕ и ψ поперечного потенциалов в виде:

$$u = \nabla\phi + \nabla \times \psi$$

приводит к волновым уравнениям:

$$\frac{\partial^2 \phi}{\partial t^2} = c_p^2 \nabla^2 \phi \quad \text{и} \quad \frac{\partial^2 \psi}{\partial t^2} = c_s^2 \nabla^2 \psi$$

где $c_p = \sqrt{\frac{\lambda+2\mu}{\rho}} = \sqrt{\frac{K+\frac{4}{3}\mu}{\rho}}$ - скорость продольной волны, $c_s = \sqrt{\frac{\mu}{\rho}}$ - скорость поперечной волны, $c_R = \frac{0,87+1,12 \cdot \nu}{1+\nu} \cdot c_s$ - скорость волны Рэлея.

3) Интегральное представление решения:

Мы хотим найти решения для смещений в виде:

$$u_i(r, z, t) = \int_0^\infty H_i(k, z, t) \cdot J_n(kr) dk \quad (2)$$

где $H_i(k, z, t)$ определяется граничными условиями и типами волн.

Классическое решение Лэмба включает интегральные представления для продольных (P) и поперечных (S) волн. Для каждой компоненты смещения u_x , u_y , u_z суммируются

вклады продольных (P) $c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}$, поперечных (S) $c_s = \sqrt{\frac{\mu}{\rho}}$ и поверхностных волн (R):

$$u_x(r, z, t) = \int_0^\infty H_p^x(k, z, t) \cdot J_1(kr) dk + \int_0^\infty H_s^x(k, z, t) \cdot J_1(kr) dk$$

$$u_y(r, z, t) = \int_0^\infty H_p^y(k, z, t) \cdot J_1(kr) dk + \int_0^\infty H_s^y(k, z, t) \cdot J_1(kr) dk \quad (3)$$

$$u_z(r, z, t) = \int_0^\infty H_p^z(k, z, t) \cdot J_0(kr) dk + \int_0^\infty (H_s^z(k, z, t) + H_R^z(k, z, t)) \cdot J_0(kr) dk$$

4) Влияние временной зависимости силы

Чтобы получить конечные выражения, нужно взять преобразование Фурье от $f(t)$:

$$F(\omega) = \int_0^\infty f(t) \cdot \exp(i\omega t) dt$$

Подставляя заданную функцию $f(t)$, получаем:

$$F(\omega) = A \cdot \frac{\omega_1}{4} \int_0^\infty e^{\omega_1 t} \cdot e^{i\omega t} \left(\sin(\omega_0 t) \left(-\frac{t^2}{\omega_1} + \frac{t}{\omega_1^2} + \frac{1}{\omega_1^3} \right) - \cos(\omega_0 t) \sqrt{3} \left(\frac{t^2}{\omega_1} + \frac{t}{\omega_1^2} \right) \right) dt$$

После его нахождения интеграла можно получить выражения для $H_i(k, z, t)$, а затем для u_x , u_y , u_z с использованием обратного преобразования Фурье.

Обратное преобразование Фурье можно вычислить с помощью языка Python:

```
import sympy as sp

# Определяем переменные
t, omega, omega_0, omega_1, A = sp.symbols('t omega omega_0 omega_1 A', real=True,
positive=True)

# Функция f(t) по закону Берлаге
f_t = (A * (omega_1 * sp.exp(omega_1 * t)) / 4) * (
    sp.sin(omega_0 * t) * (-t**2 / omega_1 + t / omega_1**2 + 1 / omega_1**3) -
    sp.cos(omega_0 * t) * sp.sqrt(3) * (t**2 / omega_1 + t / omega_1**2) )

# Вычисляем преобразование Фурье F(omega)
F_omega = sp.integrate(f_t * sp.exp(-1j * omega * t), (t, 0, sp.oo))

# Упрощаем выражение
F_omega = sp.simplify(F_omega)

print(F_omega)
```

Полученное преобразование Фурье $F(\omega)$ имеет два выражения в зависимости от условий на аргумент:

$F(\omega) =$

$$= \begin{cases} \frac{A}{4\omega_1^2} \cdot \frac{-2\omega_0\omega_1^2(-\omega_0^2 + 3(i\omega - \omega_1)^2) + 2\omega_0\omega_1(\omega_0^2 + (i\omega - \omega_1)^2)(i\omega - \omega_1)}{(\omega_0^2 + (i\omega - \omega_1)^2)^3} + \\ + \frac{\omega_0(\omega_0^2 + (i\omega - \omega_1)^2)^2 - 2\sqrt{3}\omega_1^2(-3\omega_0^2 + (i\omega - \omega_1)^2)(i\omega - \omega_1)}{(\omega_0^2 + (i\omega - \omega_1)^2)^3} + \\ + \frac{\sqrt{3}\omega_1(\omega_0^2 - (i\omega - \omega_1)^2)(\omega_0^2 + (i\omega - \omega_1)^2)}{(\omega_0^2 + (i\omega - \omega_1)^2)^3} \\ \frac{A}{4\omega_1^2} \int_0^\infty (-\sqrt{3}\omega_1 t(\omega_1 t + 1) \cos(\omega_0 t) + (-\omega_1^2 t^2 + \omega_1 t + 1) \sin(\omega_0 t)) \cdot \exp(t(-i\omega + \omega_1)) dt \end{cases}$$

Первое при $\left| 2 \arg \left(-\omega_0 e^{\frac{i\pi}{2}} + \omega_1 \right) + 2\pi \right| < \pi$, второе в остальных случаях.

5) $H_i(k, z, t)$ в (2) связаны с продольными (P) $c_P = \sqrt{\frac{\lambda + 2\mu}{\rho}}$, поперечными (S) $c_S = \sqrt{\frac{\mu}{\rho}}$ и поверхностными волнами (R):

$$H_P(k, z, t) = R_P(k) e^{-\alpha_P z} F(\omega)$$

$$H_S(k, z, t) = R_S(k) e^{-\alpha_S z} F(\omega)$$

$$H_R(k, z, t) = R_R(k)e^{-\alpha_R z}F(\omega)$$

Здесь R_P, R_S, R_R — коэффициенты отражения волн, а параметры $\alpha_P, \alpha_S, \alpha_R$ зависят от свойств среды.

Воспользуемся классическими решениями (1) и (3) для трехмерного полупространства:

$$u_x(r, z, t) = \int_0^\infty (R_P^x(k) \cdot e^{-\alpha_P z} + R_S^x(k) \cdot e^{-\alpha_S z}) \cdot F(\omega) \cdot J_1(kr) dk$$

$$u_y(r, z, t) = \int_0^\infty (R_P^y(k) \cdot e^{-\alpha_P z} + R_S^y(k) \cdot e^{-\alpha_S z}) \cdot F(\omega) \cdot J_1(kr) dk$$

$$u_z(r, z, t) = \int_0^\infty (R_P^z(k) \cdot e^{-\alpha_P z} + R_S^z(k) \cdot e^{-\alpha_S z} + R_R^z(k) \cdot e^{-\alpha_R z}) \cdot F(\omega) \cdot J_0(kr) dk$$

Коэффициенты отражения определяются через соотношения скоростей продольных и поперечных волн:

$$R_P(k) = \frac{2\mu k}{\rho(c_P^2 k^2 - \omega^2)}$$

$$R_S(k) = \frac{2\mu k}{\rho(c_S^2 k^2 - \omega^2)}$$

$$R_R(k) \approx e^{-kz}$$

где $c_P = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}}$ и $c_S = \sqrt{\frac{E}{2\rho(1+\nu)}}$

Интегралы для смещений в аналитическом виде оказываются достаточно сложными, чтобы выразить их напрямую.

6) Численные параметры среды:

1. Используемые величины:

$E = 2 \cdot 10^8$ (Па) - **модуль Юнга** (физическая величина, характеризующая способность материала сопротивляться растяжению, сжатию при упругой деформации)

$\nu = 0,3$ - **коэффициент Пуассона** (это мера **эффекта Пуассона**, явления, при котором материал стремится расширяться в направлениях, перпендикулярных направлению сжатия)

Плотность материала: $\rho = 1900 \left(\frac{\text{кг}}{\text{м}^3}\right)$.

2. Тогда:

$\mu = \frac{E}{2(1+\nu)} = 7,692 \cdot 10^7$ (Па) - **модуль сдвига** (физическая величина, характеризующая упругие свойства материалов и их способность сопротивляться сдвигающим деформациям)

$K = \frac{E}{3(1-2\nu)} = 1,667 \cdot 10^8$ (Па) - **модуль всестороннего сжатия** (характеристика способности вещества сопротивляться всестороннему сжатию)

Параметры Ламе, λ и μ (названные в честь Габриэля Ламе) — материальные константы, характеризующие упругие деформации изотропных твёрдых тел, то есть модули упругости.

$$C_P = \sqrt{\frac{\lambda + 2\mu}{\rho}} = \sqrt{\frac{K + \frac{4}{3}\mu}{\rho}} = 376.451 \left(\frac{\text{м}}{\text{с}} \right) - \text{скорость продольной волны}$$

$$C_S = \sqrt{\frac{\mu}{\rho}} = 201,211 \left(\frac{\text{м}}{\text{с}} \right) - \text{скорость поперечной волны}$$

$$C_R = \frac{0,87 + 1,12 \cdot \nu}{1 + \nu} \cdot C_S = 186,662 \left(\frac{\text{м}}{\text{с}} \right) - \text{скорость волны Рэлея.}$$

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.special import jv

# Заданные параметры
rho = 1900 # кг/м³
nu = 0.3
E = 2e8 # Па
A = 10**8 # Амплитуда
omega_0 = 10 # Гц
omega_1 = 5 # Гц
cp = np.sqrt(E / (rho * (1 - nu**2))) # Скорость продольных волн
cs = cp / np.sqrt(3) # Скорость поперечных волн

# Функция силы Берлаге (создающая импульс)
def F_t(t):
    return (A * (omega_1 * np.exp(-omega_1 * t)) / 4) * (
        np.sin(omega_0 * t) * (-t**2 / omega_1 + t / (omega_1**2) + 1 / (omega_1**3))
        - np.cos(omega_0 * t) * np.sqrt(3) * (t**2 / omega_1 + t / (omega_1**2))
    )

# Численный расчет интегралов для смещений
def u_x(r, z, t):
    def integrand(k):
        alpha_P = k / cp
        alpha_S = k / cs
        R_P = 2 * rho * k / (rho * (cp**2 * k**2 - omega_0**2))
        R_S = 2 * rho * k / (rho * (cs**2 * k**2 - omega_0**2))
        return (R_P * np.exp(-alpha_P * z) + R_S * np.exp(-alpha_S * z)) * jv(1, k * r)

    integral, _ = quad(integrand, 0, np.inf)
    return integral * F_t(t)
```

```

def u_y(r, z, t):
    return u_x(r, z, t) # Симметрия задачи

def u_z(r, z, t):
    def integrand(k):
        alpha_P = k / cp
        alpha_S = k / cs
        R_P = 2 * rho * k / (rho * (cp**2 * k**2 - omega_0**2))
        R_S = 2 * rho * k / (rho * (cs**2 * k**2 - omega_0**2))
        R_R = np.exp(-k * z)
        return (R_P * np.exp(-alpha_P * z) + R_S * np.exp(-alpha_S * z) + R_R) * jv(0, k * r)

    integral, _ = quad(integrand, 0, np.inf)
    return integral * F_t(t)

# Построение графиков
t_vals = np.linspace(0, 10, 500) # Увеличен диапазон времени до 10 секунд с 500
точками
r, z = 250, 0 # Точка наблюдения

# Вычисление смещений для всех точек времени
u_x_vals = [u_x(r, z, t) for t in t_vals]
u_y_vals = [u_y(r, z, t) for t in t_vals]
u_z_vals = [u_z(r, z, t) for t in t_vals]

# Построение графиков
plt.figure(figsize=(12, 8))

plt.subplot(3, 1, 1)
plt.plot(t_vals, u_x_vals, label="u_x")
plt.xlabel("Время (с)")
plt.ylabel("Смещение u_x")
plt.legend()

plt.subplot(3, 1, 2)
plt.plot(t_vals, u_y_vals, label="u_y", color="r")
plt.xlabel("Время (с)")
plt.ylabel("Смещение u_y")
plt.legend()

plt.subplot(3, 1, 3)
plt.plot(t_vals, u_z_vals, label="u_z", color="g")
plt.xlabel("Время (с)")

```

```
plt.ylabel("Смещение u_z")
plt.legend()

plt.tight_layout()
plt.show()
```

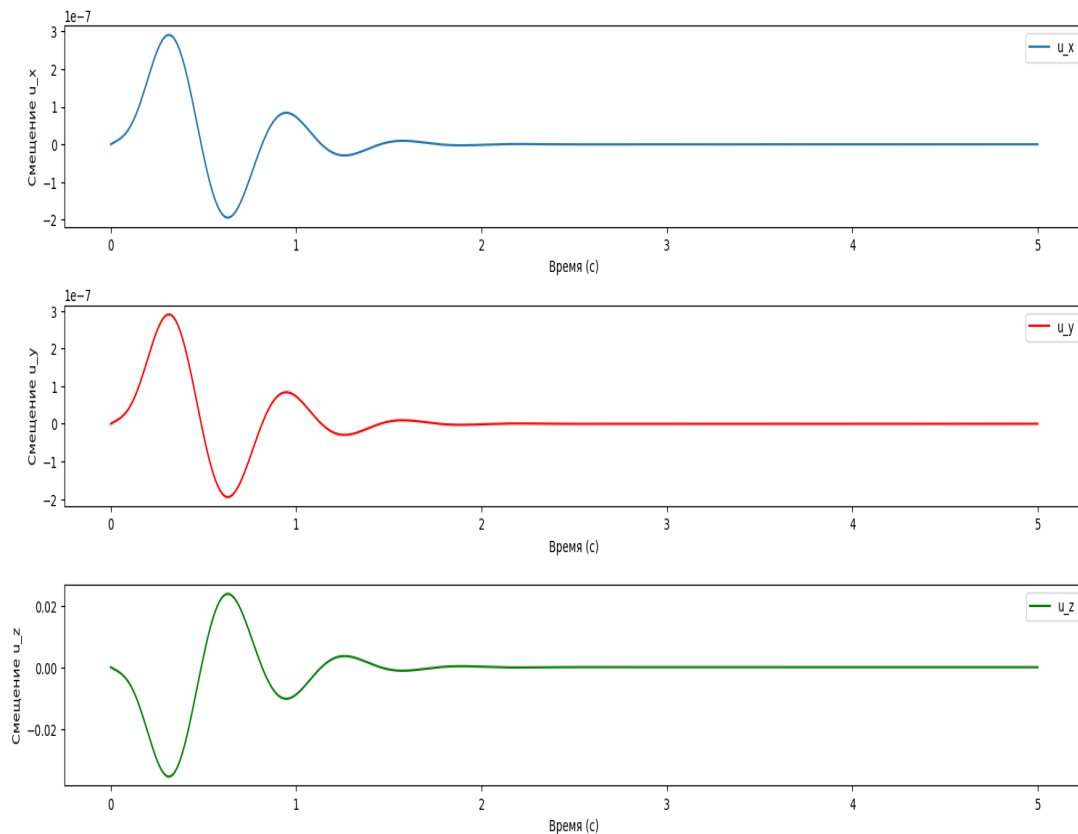


Рисунок 3: График смещения $u(x,y,z,t)$

Анализ: от точечного источника в полупространстве распространяются: **объемные Р и S волны**, и **волны Рэлея**. Волны Рэлея — это разновидность поверхностных акустических волн, которые распространяются вдоль поверхности твердых тел.

Продольной (Р) волной называется волна, при распространении которой частицы среды совершают колебания вдоль направления распространения волны. При землетрясениях и взрывах они приходят первыми.

Поперечной (S) волной называется волна, при распространении которой частицы среды совершают колебания перпендикулярно направлению распространения волны. Данные волны приходят позже продольных волн.

Колебания, связанные с S – волнами, могут происходить в двух взаимно перпендикулярных плоскостях. В результате чего их можно разделить на два вида: параллельную земной поверхности и перпендикулярную ей. Они существуют раздельно и называются **плоскополяризованными SH – и SV – волнами**.

Раздел №9: Численное решение задачи Лэмба.

Результаты численного решения были получены с помощью программного обеспечения CAE Fidesys. Расчет распространения волн в программе происходил на основе метода спектральных элементов, который дает более быструю сходимость и высокую точность по сравнению с методом конечных элементов. Сетка для геометрической модели, характеристики которой представлены ниже, строилась на основе метода конечных элементов, а далее при помощи алгоритма внутри программы преобразовывалась в сетку от 1-го до 9-го порядка.

1) Свойства геометрической модели в CAE Fidesys.

Свойства материала и линейный размеры модели:

- 1) Длина: 800. Ширина: 600. Высота: 200.
- 2) Модуль Юнга: $E = 2 \cdot 10^8$ (Па)
- 3) Коэффициент Пуассона: $\nu = 0,3$
- 4) Плотность: $\rho = 1900 \left(\frac{\text{кг}}{\text{м}^3} \right)$

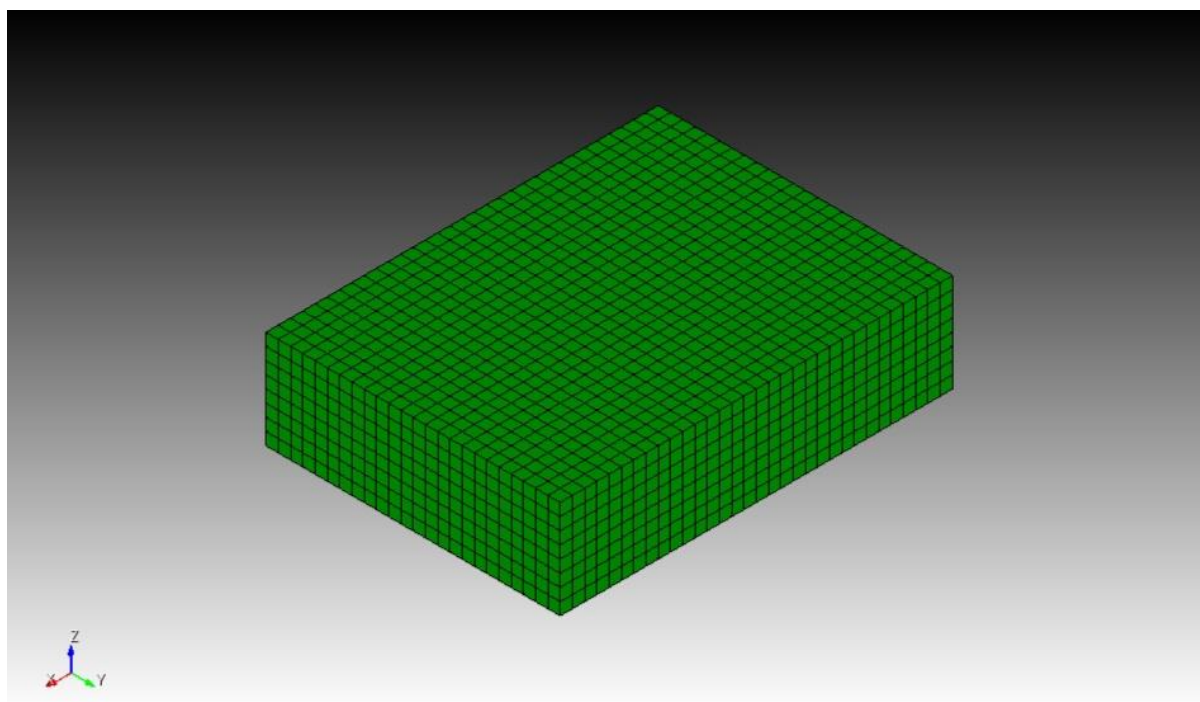


Рисунок 4: Модель в CAE Fidesys.

Сетка:

- 1) Размер одного элемента сетки: 100.
- 2) Порядок точности МСЭ: от 2 до 9-го для различных сеток.

Граничные условия:

Амплитуда силы: $A = 10^8$ (м). Частота: $\omega = 5$ (Гц). Сила задана по закону Берлаге. На боковых и нижней поверхности модели заданы неотражающие условия.

Максимальное время наблюдения - 2 секунд.

Максимальное число шагов - 15000.

Считаю, что $|V_q - V_{ан}| < 1\%$

2) Код программы, строящий модель для программного обеспечения CAE Fidesys:

Следующий код создает трехмерную конечно-элементную модель, задавая сетку узлов и элементов, а также свойства материала, свойства и настройки численного расчета. Он определяет сетку элементов, назначает узлы, задает свойства материалов, создает приемники и прикладывает силу.

В начале кода импортируются библиотеки, необходимые для работы с массивами (numpy), визуализации (matplotlib), численного интегрирования (scipy), работы с файлами SEG-Y (segvio) и конечно-элементной модели (fc_model).

```
import numpy as np
from fc_model import FCModel, FCReciver, FCMaterial, FCMaterialProperty, FCElem,
FCBlock, FCCoordinateSystem, FCNode, FC_ELEMENT_TYPES

import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.special import jv
import segvio
```

Далее задаются размеры элементов и вычисляется количество элементов вдоль каждой оси.

```
ELEMS_X_SIZE = 100 # Размер одного эл-та по оси X
ELEMS_Y_SIZE = 100
ELEMS_Z_SIZE = 100

ELEMS_X_COUNT = int(800/ELEMS_X_SIZE) # Число эл-в по оси X
ELEMS_Y_COUNT = int(600/ELEMS_Y_SIZE) # Число эл-в по оси Y
ELEMS_Z_COUNT = int(200/ELEMS_Z_SIZE) # Число эл-в по оси Z

print(ELEMS_X_COUNT,ELEMS_Y_COUNT,ELEMS_Z_COUNT)
NODES_X_COUNT = ELEMS_X_COUNT+1 # Число узлов по оси X
NODES_Y_COUNT = ELEMS_Y_COUNT+1
NODES_Z_COUNT = ELEMS_Z_COUNT+1
ORDER = 1 # Порядок элементов в модели (варьируется)
```

Определяются координаты начала сетки и параметры материала.

```
X_START = 0
Y_START = 0
```

```
Z_START = 0
```

```
Young_module = 2e+08
```

```
Poison_ratio = 0.3
```

```
Density = 1900
```

Далее следуют функции для преобразования индексов. Эти функции позволяют работать с индексами элементов и узлов в трехмерной сетке.

```
def elems_id(i, j, k):  
    return (i * (NODES_Y_COUNT - 1) + j) * (NODES_Z_COUNT - 1) + k + 1
```

```
def xyz2id(i, j, k):  
    return (i*NODES_Y_COUNT+j)*(NODES_Z_COUNT) + k + 1
```

```
def id2xyz(id):  
  
    k = (id-1) % NODES_Z_COUNT  
    j = ((id - 1) // NODES_Z_COUNT) % NODES_Y_COUNT  
    i = ((id - 1) // NODES_Z_COUNT) // NODES_Y_COUNT  
  
    return i, j, k
```

Создание модели и узлов: создаётся объект `fc_model` и добавляются узлы сетки.

```
fc_model = FCModel()  
  
for i in range(NODES_X_COUNT):    # Создаем узлы  
    for j in range(NODES_Y_COUNT):  
        for k in range(NODES_Z_COUNT):  
  
            node_id = xyz2id(i, j, k) # Преоб-ет трёх-е коор-ты (i, j, k) в уникальный  
одномерный идентификатор ('node_id') для каждого узла  
            #print("node_id:", node_id)  
            #print('x ', i, 'y ', j, 'z ', k)  
            node: FCNode = {  
                "id": node_id,  
                "xyz": np.array([  
                    ELEMS_X_SIZE*i + X_START,  
                    ELEMS_Y_SIZE*j + Y_START,  
                    ELEMS_Z_SIZE*k + Z_START  
                ])  
            }  
  
            fc_model.nodes.add(node)
```

```
print("Число узлов в модели:", node_id)
```

Создание элементов: здесь создаются элементы (объемные ячейки) модели.

```
elements_numpy_id = [] # Массив, хранящий id всех эл-в

for i in range(ELEMS_X_COUNT): # Создаем элементы в модели
    for j in range(ELEMS_Y_COUNT):
        for k in range(ELEMS_Z_COUNT):

            elem_id = elems_id(i, j, k) # Получаем id элемента
            elements_numpy_id.append(elem_id) # Добавляем id эл-та в массив

            element: FCElem = { # Создаем класс FCElem эл-в модели
                "id": elem_id,
                "block": 1,
                "order": ORDER,
                "parent_id": 1,
                "type": FC_ELEMENT_TYPES[3],
                "nodes": [
                    xyz2id(i,j,k),
                    xyz2id(i+1,j,k),
                    xyz2id(i+1,j+1,k),
                    xyz2id(i,j+1,k),
                    xyz2id(i,j,k+1),
                    xyz2id(i+1,j,k+1),
                    xyz2id(i+1,j+1,k+1),
                    xyz2id(i,j+1,k+1),
                ],
            }

            fc_model.elems.add(element)

print("Число элементов в модели:", elem_id)
```

Далее идет определение блоков, материалов и координатной системы. Добавляются блоки и координатная система.

```
block: FCBlock = { # Создаем блок
    "id": 1, # id блока
    "cs_id": 1,
    "material_id": 0, # указываем, что нет материала
    "property_id": -1
}
```

```

fc_model.blocks.add(block)

coordinate_system: FCCoordinateSystem = {
    'id': 1,
    'dir1': np.array([1., 0., 0.], np.float64),
    'dir2': np.array([0., 1., 0.]),
    'origin': np.array([0., 0., 0.]),
    'name': 'cartesian',
    'type': 'cartesian'
}

fc_model.coordinate_systems.add(coordinate_system)

```

Добавляю заготовки таблиц, в которых хранятся данные о свойствах материала в модели.

```

E_prop: FCMaterialProperty = {
    'name': 0,
    'type': 0,
    'data': np.array([11,12,13], np.float64),
    'dependency': [{
        'type': 10, # elems
        'data': np.array([1,2,3], np.float64),
    }]
}

Nu_prop :FCMaterialProperty = {
    'name': 1,
    'type': 0,
    'data': np.array([11,12,13], np.float64),
    'dependency': [{
        'type': 10, # elems
        'data': np.array([1,2,3], np.float64),
    }]
}

Pho_prop: FCMaterialProperty = {
    'name': 0,
    'type': 0,
    'data': np.array([11,12,13], np.float64),
    'dependency': [{
        'type': 10, # elems
        'data': np.array([1,2,3], np.float64),
    }]
}

```

```

material: FCMaterial = {
    "id" : 1,
    "name": 'Mat1',
    "properties": {
        "elasticity": [
            E_prop, Nu_prop
        ],
        "common": [
            Pho_prop
        ]
    }
}

fc_model.materials.add(material)

```

Создаются линии приемников для регистрации данных.

```

x_receiver_start = 0
y_receiver_start = ELEMS_Y_COUNT // 2
z_receiver_start = ELEMS_Z_COUNT

receiver_numpy_id = []
receiver_coord = []

for x_receiver_start in range (NODES_X_COUNT):
    receiver_id = xyz2id(x_receiver_start, y_receiver_start, z_receiver_start)
    receiver_numpy_id.append(receiver_id)
    receiver_coord.append([x_receiver_start*ELEMS_X_SIZE,
y_receiver_start*ELEMS_Y_SIZE, z_receiver_start*ELEMS_Z_SIZE])

print("Линии приемников", receiver_numpy_id)
print("Координаты узлов на линии приемников: ", receiver_coord)
receiver_per: FCReciver = {
    'apply_to': np.array(receiver_numpy_id, np.int32),
    'dofs': [1,1,1],
    'type': 0,
    'name': 'Rec1',
    'id': 1
}

fc_model.receivers.add(receiver_per)

```

Определяется точка, в которую будет приложена сила.

```

x_sila = ELEMS_X_COUNT/2;
y_sila = ELEMS_Y_COUNT/2;

```

```
z_sila = ELEMS_Z_COUNT;
node_sila = xyz2id(x_sila,y_sila,z_sila)
print("Сила приложена в узле:", node_sila)
```

Настройки модели: определяются параметры численного решения.

```
fc_model.settings = {
    'bc_tolerance': 0.0005,
    'dimensions': '3D',
    'dynamics': {
        'courant': 0.8,
        'max_steps_count': 10000,
        'max_time': 2.0,
        'method': 'full_solution',
        'result_output_iter': 1,
        'scheme': 'explicit'
    },
    'eigen_solver': None,
    'elasticity': True,
    'environment': {
        'temperature': 0.0
    },
    'finite_deformations': False,
    'heat_transfer': False,
    'incompressibility': False,
    'linear_solver': {
        'iter_opts': {
            'epsilon': 1e-09,
            'max_iterations': 50000,
            'preconditioner': 'auto',
            'stopping_criteria': 0.0
        },
        'method': 'auto',
        'on_fail': True,
        'use_uzawa': 'auto'
    },
    'nonlinear_solver': {
        'arc_method': False,
        'line_search': False,
        'max_iterations': 100,
        'max_load_steps': 1000,
        'min_load_steps': 1,
        'start_load_steps': 1,
        'target_iter': 5,
```

```

        'tolerance': 0.001
    },
    'output': {
        'energy': False,
        'full_periodic': False,
        'intermediate_results': True,
        'log': True,
        'material': False,
        'normal_force': False,
        'record3d': False,
        'vtu': True,
        'without_smoothing': False
    },
    'permission_write': True,
    'plasticity': False,
    'porefluid_transfer': False,
    'preload': False,
    'radiation_among_surfaces': False,
    'slm': False,
    'thermal_gap_settings': {
        'end_temp': 3000.0,
        'impact_eps': 0.01,
        'scatangle_eps': 0.01,
        'start_temp': 0.0,
        'step_temp': 1.0,
        'velocity_eps': 0.01
    },
    'type': 'dynamic'
}

```

Следующий фрагмент кода выполняет настройку параметров материала (модуль Юнга, коэффициент Пуассона и плотность) для каждого конечного элемента модели.

Извлечение свойств материала.

```

E_md = fc_model.materials[1]['properties']['elasticity'][0]      # Модуль Юнга
Nu_md = fc_model.materials[1]['properties']['elasticity'][1]    # Коэффициент Пуассона
Density_md = fc_model.materials[1]['properties']['common'][0]   # Плотность

```

Создание массивов для хранения свойств материала

```

length_numpy_elements = len(elements_numpy_id) # Длина массива с id узлами
empty_numpy_array_E_md = np.empty(length_numpy_elements, dtype=np.float64)
empty_numpy_array_Nu_md = np.empty(length_numpy_elements, dtype=np.float64)
empty_numpy_array_Density = np.empty(length_numpy_elements, dtype=np.float64)

```


Настройка зависимости свойств от элементов и установка значений для всех элементов.

```
E_m_data_dependency = fc_model.materials[1]['properties']['elasticity'][0]['dependency']

for dep in E_m_data_dependency:
    dep['data'] = np.array(elements_numpy_id, dtype=np.float64) # Добавляем массив id
    dep['type'] = 10 # Устанавливаем тип элемента

empty_numpy_array_E_md[:] = Young_module
E_md['data'] = empty_numpy_array_E_md # Устанавливаю модуль Юнга

empty_numpy_array_Nu_md[:] = Poison_ratio
Nu_md['data'] = empty_numpy_array_Nu_md # Устанавливаю к-т Пуассона
Nu_md['dependency'] = E_md['dependency'] #Перекачиваю в него id

empty_numpy_array_Density[:] = Density
Density_md['data'] = empty_numpy_array_Density # Устанавливаю плотность
Density_md['dependency'] = E_md['dependency']

fc_model.save('var100.fc')
```

Далее необходимо в CAE Fidesys задать неотражающие граничные условия, силу и ее зависимость, а также присвоить материал блоку.

```
create absorption on surface 1 2 3 5 6
create force on node 96 force value 1 direction 0 0 -1
bcdep force 1 value 'berlage(1e+08, 5, time)
block 1 material 1 cs 1 element solid order 1

dynamic results everytime 0.1
output nodalforce off energy off record3d off material off without_smoothing off
fullperiodic off\analysis type dynamic elasticity dim3 preload off
dynamic method full_solution scheme explicit maxsteps 10000 maxtime 2
```

3) Галерея распространения волн для различных порядков (100 сетка).

В данной галереи продемонстрировано распространение волн на свободной поверхности модели в задаче Лэмба. На представленных изображениях показано, что изменение порядка модели влияет на численное решение. По мере увеличения порядка заметно уменьшение шумов при распространении волн.

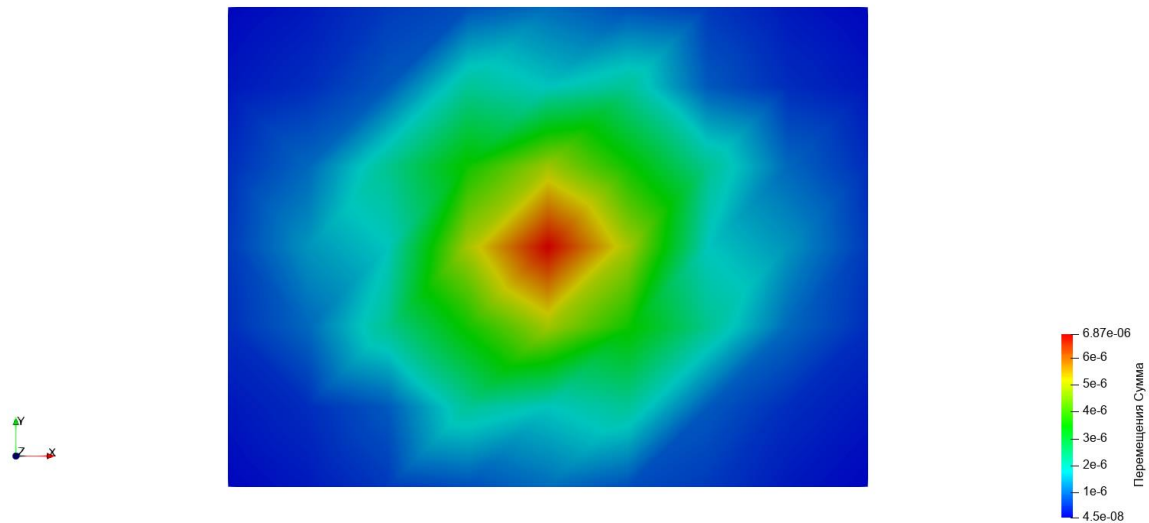


Рисунок 5: 1 порядок 100 сетка

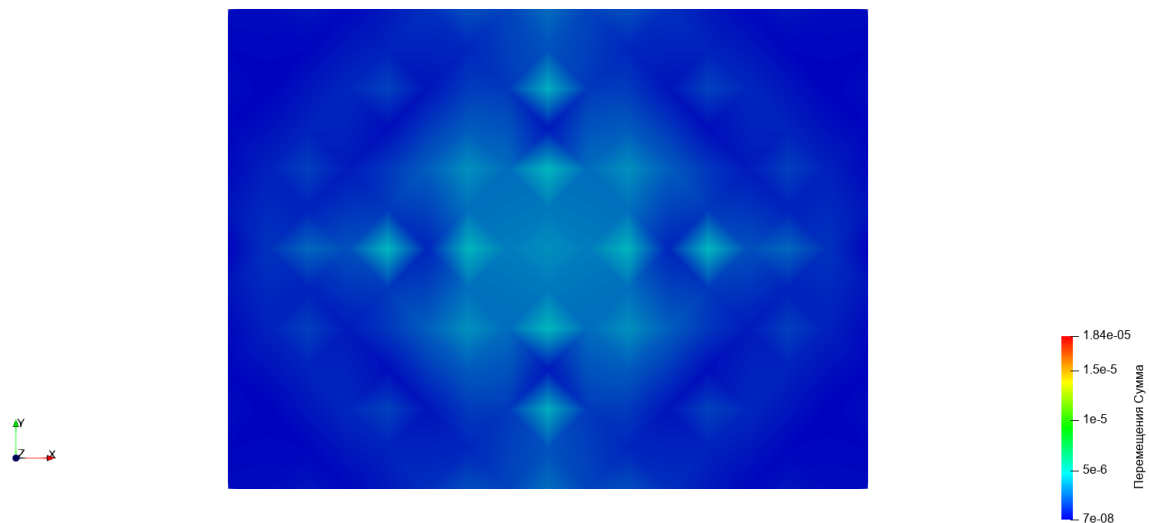


Рисунок 6: 2 порядок 100 сетка

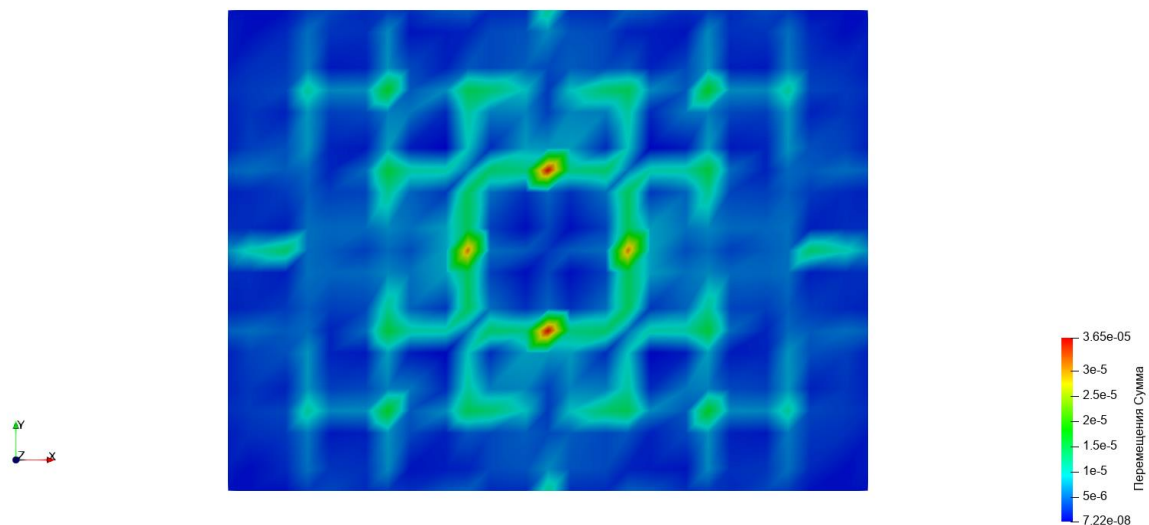


Рисунок 7: 3 порядок 100 сетка

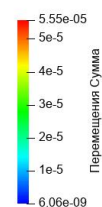
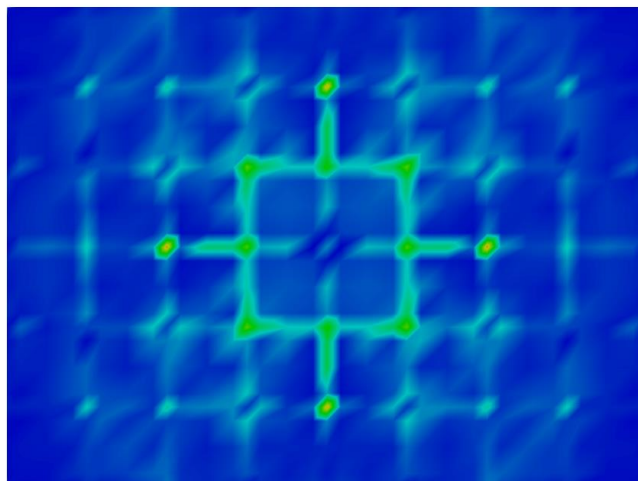


Рисунок 8: 4 порядок 100 сетка

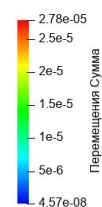
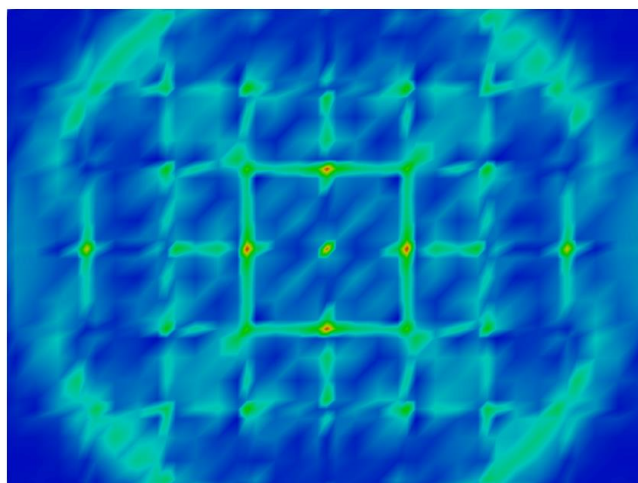


Рисунок 9: 5 порядок 100 сетка

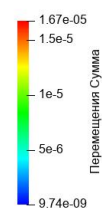
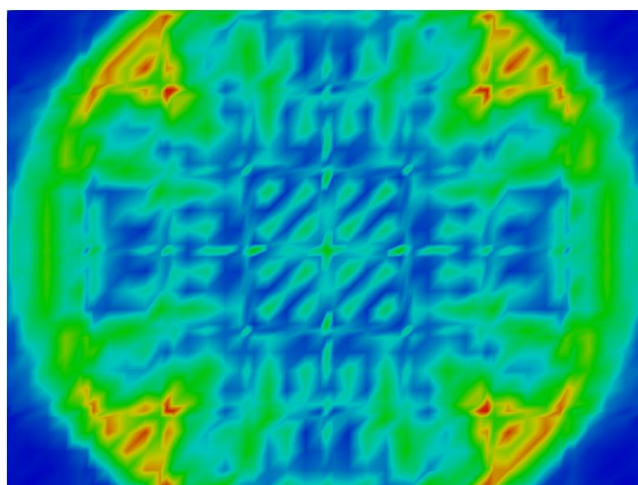


Рисунок 10: 6 порядок 100 сетка

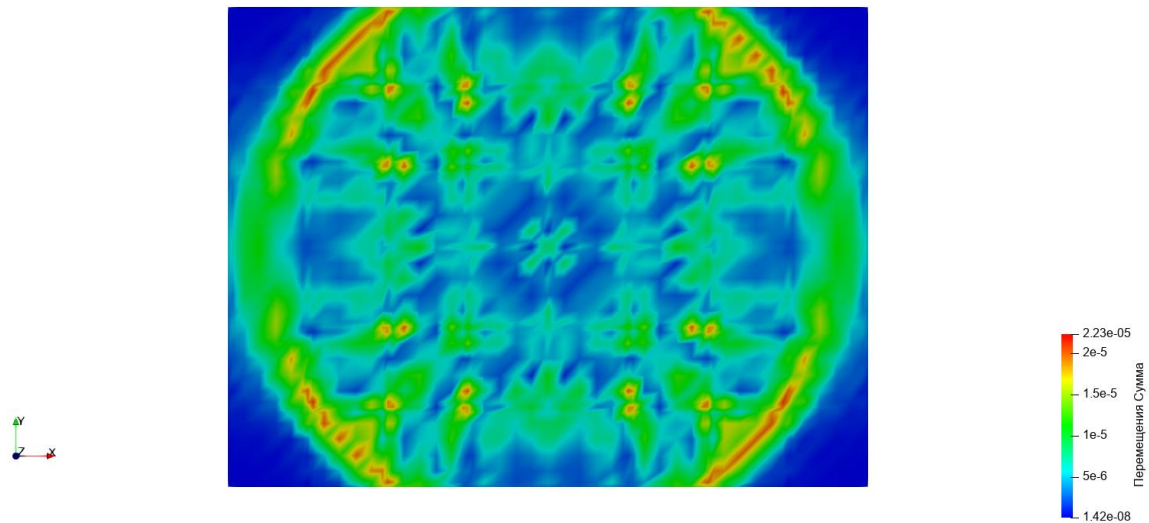


Рисунок 11: 7 порядок 100 сетка

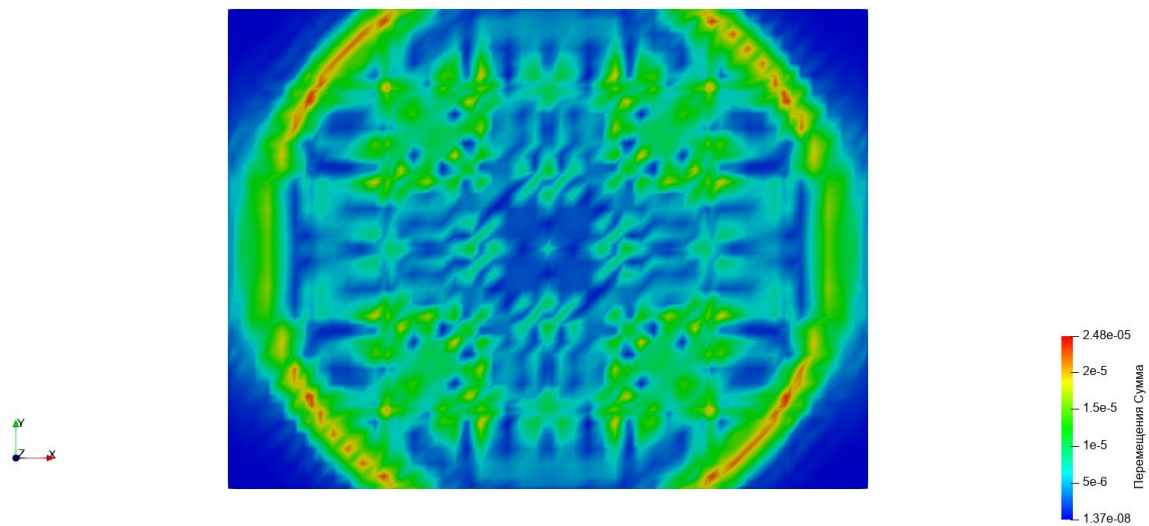


Рисунок 12: 8 порядок 100 сетка

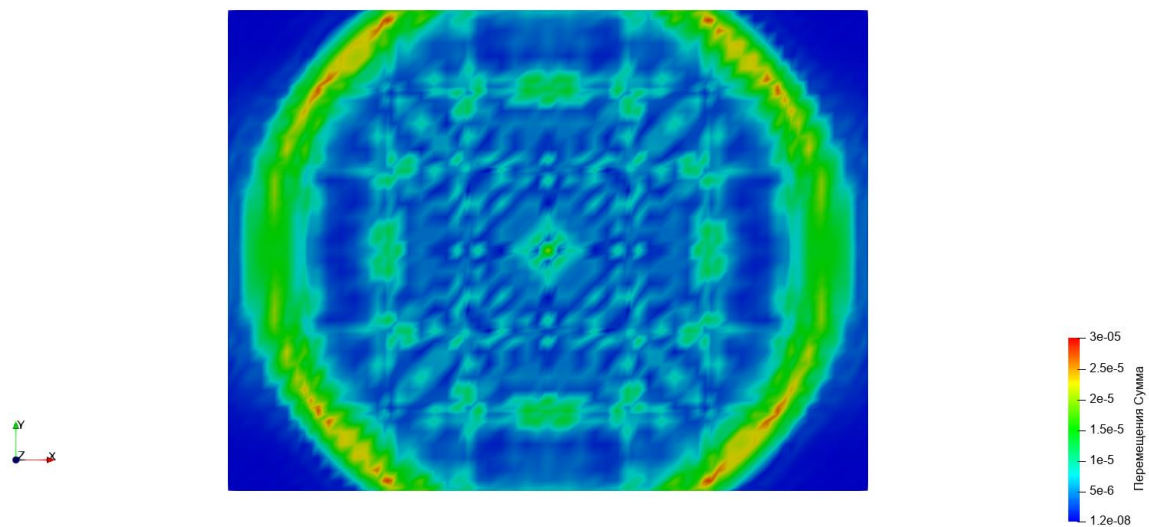


Рисунок 13: 9 порядок 100 сетка

С увеличением порядка элемента численного метода шумы в модели уменьшаются, что приводит к более точному описанию процесса распространения волн.

4) График сравнения численного и аналитического решений (100 сетка)

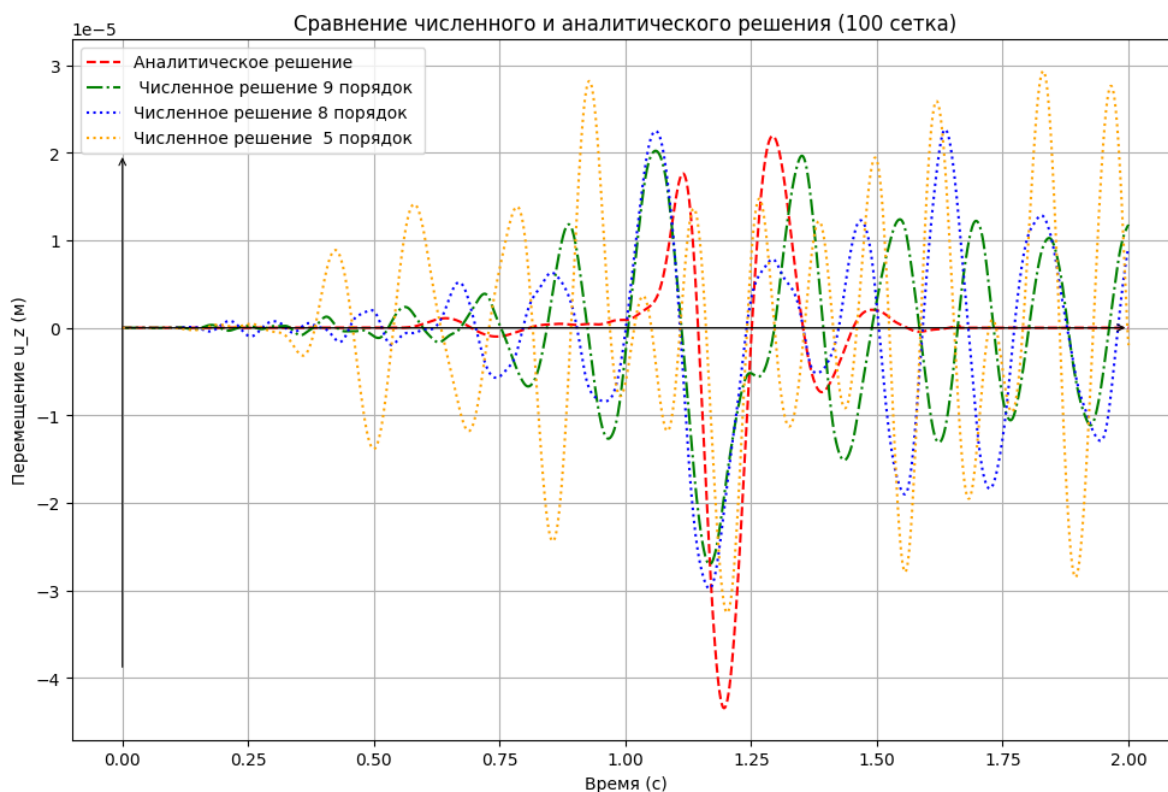


Рисунок 14: Сравнение аналитического решения и решений различных порядков

5) Разница численных и аналитического решений (100 сетка)

$$\text{Ошибка} = \frac{\max(V_{\text{числ.}}(t) - V_{\text{анал.}}(t))}{\max|V_{\text{анал.}}(t)|} * 100\%$$

Разница аналитического решения и решения 1-го порядка = 120.1%

Разница аналитического решения и решения 2-го порядка = 100.0%

Разница аналитического решения и решения 3-го порядка = 97.6%

Разница аналитического решения и решения 4-го порядка = 91.0%

Разница аналитического решения и решения 5-го порядка = 81.6%

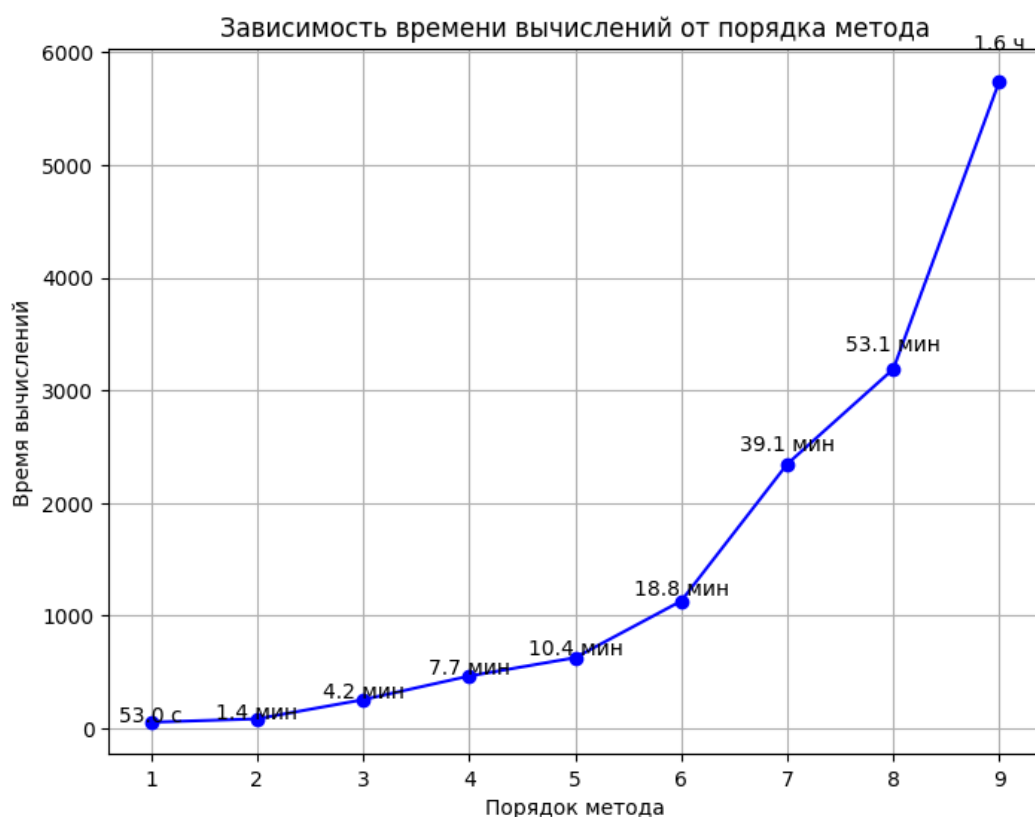
Разница аналитического решения и решения 6-го порядка = 75.5%

Разница аналитического решения и решения 7-го порядка = 67.2%

Разница аналитического решения и решения 8-го порядка = 62.9%

Разница аналитического решения и решения 9-го порядка = 52.9%

6) График времени вычисления в моделях различных порядков (100 сетка)



Сравнение численных решений различных порядков (от 1 до 9) на сетке размера 100 с аналитическим решением не дало ожидаемых результатов в 1 процент разницы аналитического и численного решения. Поэтому было решено проделать два следующих шага:

1) Мельчить сетку до достижения разницы между аналитическим и численным решениями в 1 процент.

2) Создать модель, у которой есть не только неотражающие граничные условия, но и демпфирующий слой.

3) Проводить расчёты для нее.

Демпфирующий слой в модели — это область, предназначенная для поглощения волн и предотвращения отражений от границ расчетной области. Он особенно важен в численных методах решения волновых уравнений, например, при моделировании распространения сейсмических или акустических волн.

7) Код программы, добавляющий демпфирующий слой в модель

Чтобы добавить демпфирующий слой в модель, на программном языке Python был написан следующий код, преобразующий `fc` модель для CAE Fidesys.


```

k = 2
# Массивы для хранения id узлов
found_ids = [] # Текущие узлы слоя
f = []         # Все найденные узлы

flag_break = 0 # Флаг для прерывания
# Инициализация пустых массивов
empty_numpy_array = np.array([], dtype=np.float64)
new_data = np.array([], dtype=np.float64)

# Вычисление коэффициентов массового демпфирования
koef = round(1 / k, 2)

iter_koef = 0
for iter_mass in range(k):
    if (iter_mass == 0):
        Mass_damping.append(1)
    else:
        print(iter_mass)
        Mass_damping.append(round(iter_mass*koef,2))
Mass_damping.append(0)
print("Коэффициент массового демпфирования:", Mass_damping)

# Основной цикл по слоям
for iter in range(k+1):

    x = ELEMES_X_COUNT - iter # Правая граница по X
    y = ELEMES_Y_COUNT       # Верхняя граница по Y
    iter_koef += 1

    # Поиск узлов на 1-й и 2-й сторонах слоя
    while (x - iter >= 0):

        z = ELEMES_Z_COUNT     # Верхний узел в столбце
        while (z - iter >= 0):  # Идём вниз

            node_id = ijk2id(x, y - iter, z) # Запоминаем k-ты узла на 1-й стороне
            found_ids.append(node_id)        # Добавляем k-ты узла на 1-й стороне в массив

            if (z == ELEMES_Z_COUNT):
                print(f"ID узла на 1-й стороне в верхней строчке {iter + 1}-го слоя с
координатами {(x*ELEMES_X_SIZE, y*ELEMES_Y_SIZE, z*ELEMES_Z_SIZE)}: {node_id}")
                if (xyz2id(x, y - iter, z) != xyz2id(x, iter, z)):

```

```

node_id = xyz2id(x, iter, z)      # Запоминаем к-ты узла на 2-й стороне
found_ids.append(node_id)        # Добавляем к-ты узла на 2-й стороне в массив

if (xyz2id(x, y - iter, z) == xyz2id(x, iter, z)):
    flag_break = 1
if (z == ELEMS_Z_COUNT):
    print(f"ID узла на 2-й стороне в верхней строчке {iter + 1}-го слоя с координатами
    {(x * ELEMS_X_SIZE, iter * ELEMS_Y_SIZE, z * ELEMS_Z_SIZE)}: {node_id}")
    z -= 1          # Сдвигаемся вниз
x -= 1             # Сдвигаемся вправо

# Поиск узлов на 3-й и 4-й сторонах слоя

z = ELEMS_Z_COUNT
x = ELEMS_X_COUNT
y = ELEMS_Y_COUNT - 1 - iter      # На один узел меньше по Y

while (y - iter >= 1):

    z = ELEMS_Z_COUNT              # Начинаем с верхнего узла
    while (z - iter >= 0):          # Идём вниз
        node_id = xyz2id(x-iter, y, z) # Запоминаем к-ты узла на 3-й стороне
        found_ids.append(node_id)      # Добавляем к-ты узла на 3-й стороне в массив

        if (z == ELEMS_Z_COUNT):
            print(f"ID узла на 3-й стороне в верхней строчке {iter + 1}-го слоя с координатами
            {(x-iter) * ELEMS_X_SIZE, y * ELEMS_Y_SIZE, z * ELEMS_Z_SIZE)}: {node_id}")

        if (xyz2id(x-iter, y, z) != xyz2id(iter, y, z)):
            node_id = xyz2id(iter, y, z) # Запоминаем к-ты узла на 4-й стороне
            found_ids.append(node_id)    # Добавляем к-ты узла на 4-й стороне в массив

        if (z == ELEMS_Z_COUNT):
            print(f"ID узла на 4-й стороне в верхней строчке {iter + 1}-го слоя с координатами
            {(iter * ELEMS_X_SIZE, y * ELEMS_Y_SIZE, z * ELEMS_Z_SIZE)}: {node_id}")
            z -= 1          # Сдвигаемся вниз
        y -= 1             # Сдвигаемся влево

# Поиск узлов на нижней стороне слоя

z = iter          # Двигаемся с iter-го узла вверх
x_start = iter + 1 # Определяем границы нижней области
x_end = ELEMS_X_COUNT - iter
y_start = iter + 1
y_end = ELEMS_Y_COUNT - iter

```



```

for x in range(x_start, x_end):
    for y in range(y_start, y_end):
        node_id = xyz2id(x, y, z)      # Запоминаем координаты узла на нижней стороне
        found_ids.append(node_id)
        print(f"ID узла на нижней стороне в верхней строчке {iter + 1}-го слоя с
координатами {(x* ELEMS_X_SIZE, y* ELEMS_Y_SIZE, z* ELEMS_Z_SIZE): {node_id}}")
print(f"Число узлов на {iter + 1}-м слое : {len(found_ids)}")

# Обновление параметров демпфирования

dns_md = fc_model.materials[1]['properties']['common'][1]      # Коэф-т масс-го демпф-я

f.append(found_ids[:])    # Добавляем массив id узлов на iter-м слое в общий массив с
id узлов
length = len(f[iter])      # Длина массива с id узлами
data_dns = np.full(length, Mass_damping[iter])

empty_numpy_array = np.concatenate((empty_numpy_array, data_dns))
dns_md['data'] = empty_numpy_array

# Обновление зависимости демпфирования
dns_data_dependency = fc_model.materials[1]['properties']['common'][1]['dependency']

for dep in dns_data_dependency:

    new_data = np.concatenate((new_data, found_ids))    # Модифицируем new_data
    dep['data'] = new_data      # Заменяем старый массив новым
    new_type = 11              # Меняю тип на элемент
    dep['type'] = new_type

found_ids.clear()

if (flag_break == 1):
    break

# Сохранение модели
fc_model.save('demp25.fc')

```

Коэффициент массового демпфирования (иногда его называют коэффициентом демпфирования в массовой матрице) — это параметр, который вводится в уравнения движения для учета потерь энергии в системе. Он связан с так называемым пропорциональным демпфированием (демппированием Релея), когда матрица

демпфирования C представляется в виде: $C = \alpha M + \beta K$

где: M — матрица масс, K — матрица жесткости, α — коэффициент массового демпфирования, β — коэффициент жесткостного демпфирования.

В программе устанавливается коэффициент массового демпфирования для граничного слоя в 1, а затем уменьшается вглубь модели пропорционально количеству слоев так, чтобы у последнего слоя коэффициент массового демпфирования был равен 0.

8) Галерея распространения волн для различных порядков (50 сетка).

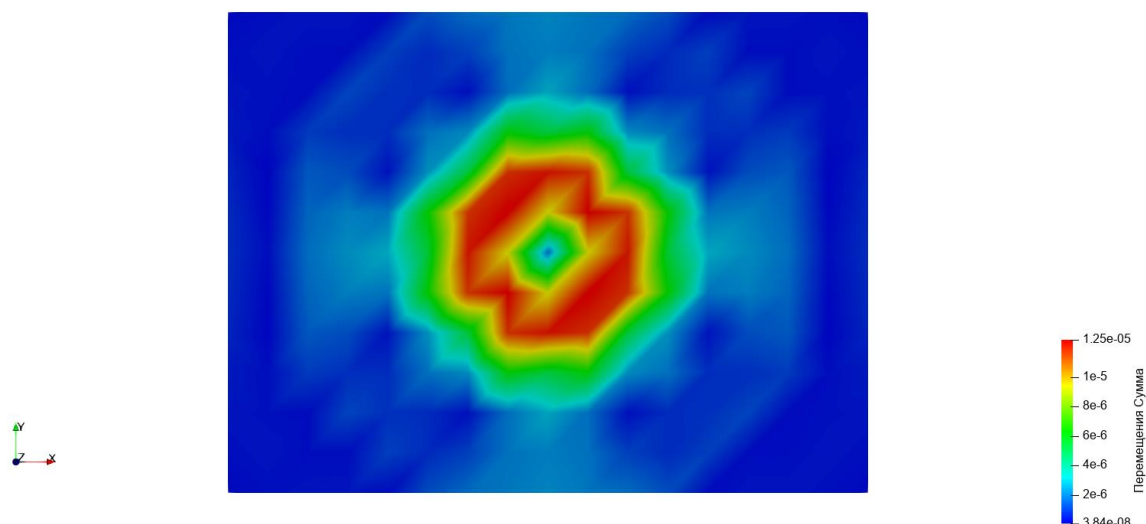


Рисунок 15: 1 порядок 50 сетка

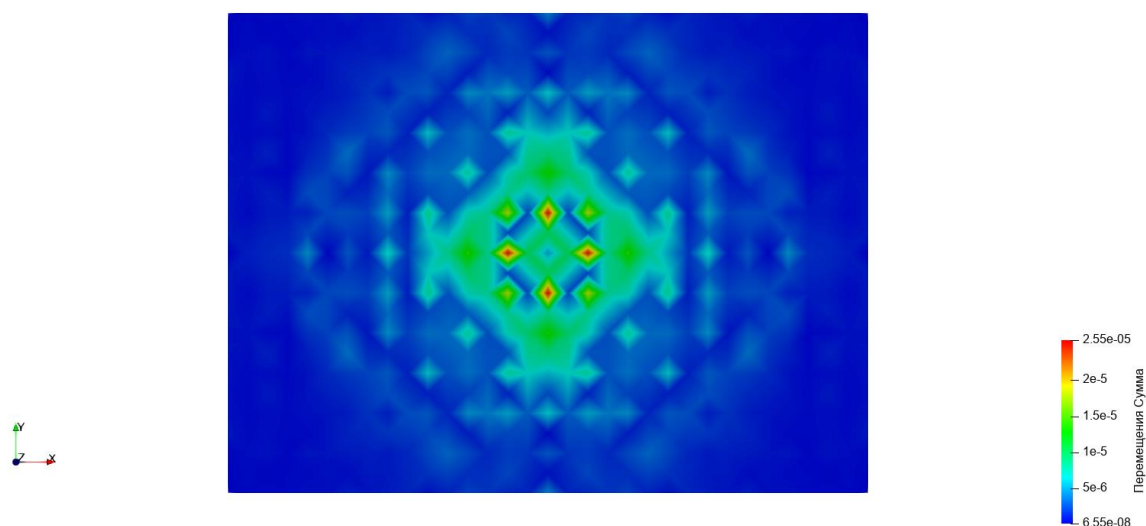


Рисунок 16: 2 порядок 50 сетка

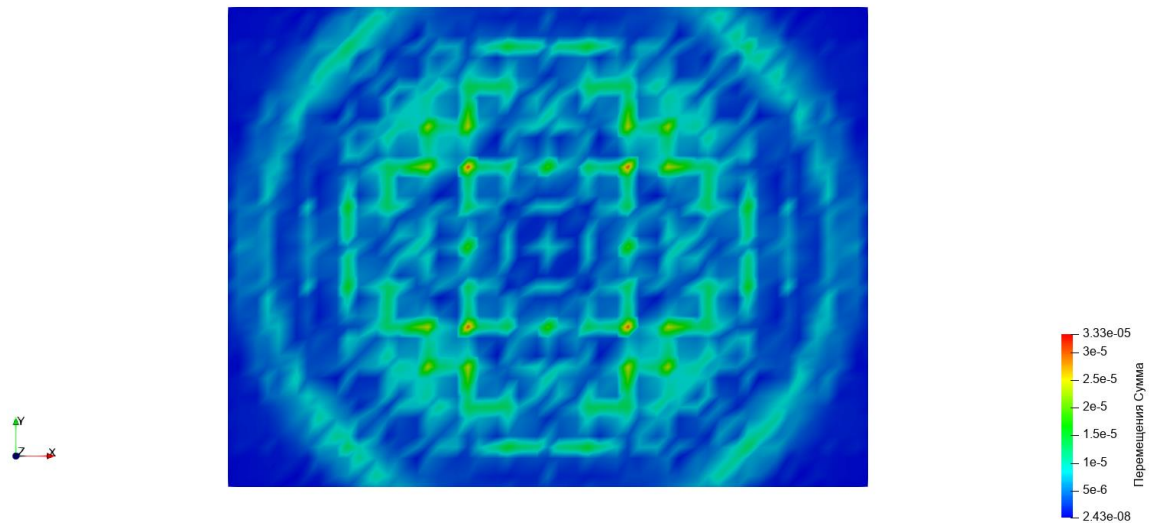


Рисунок 17: 3 порядок и 50 сетка

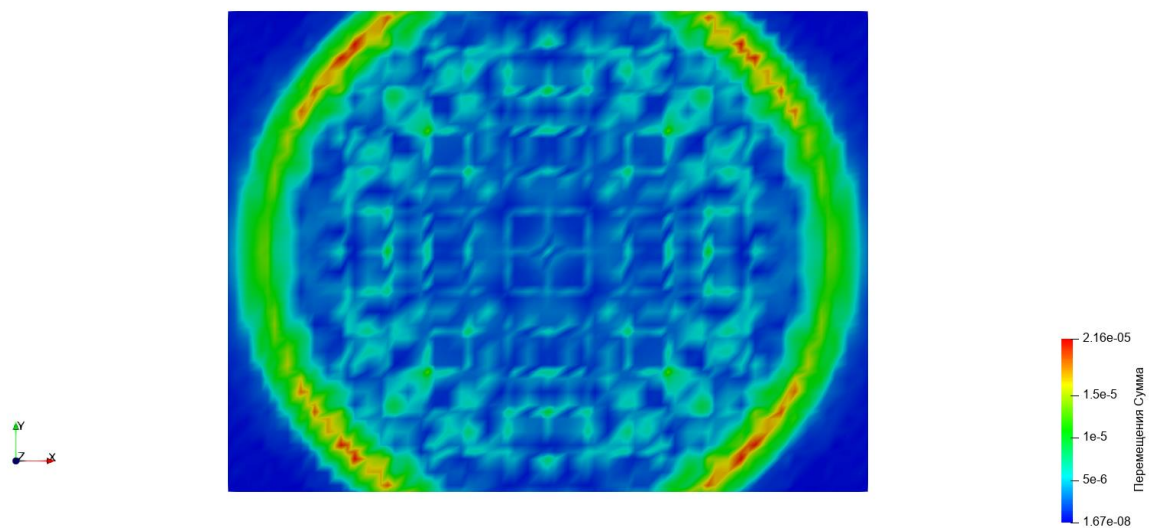


Рисунок 18: 4 порядок 50 сетка

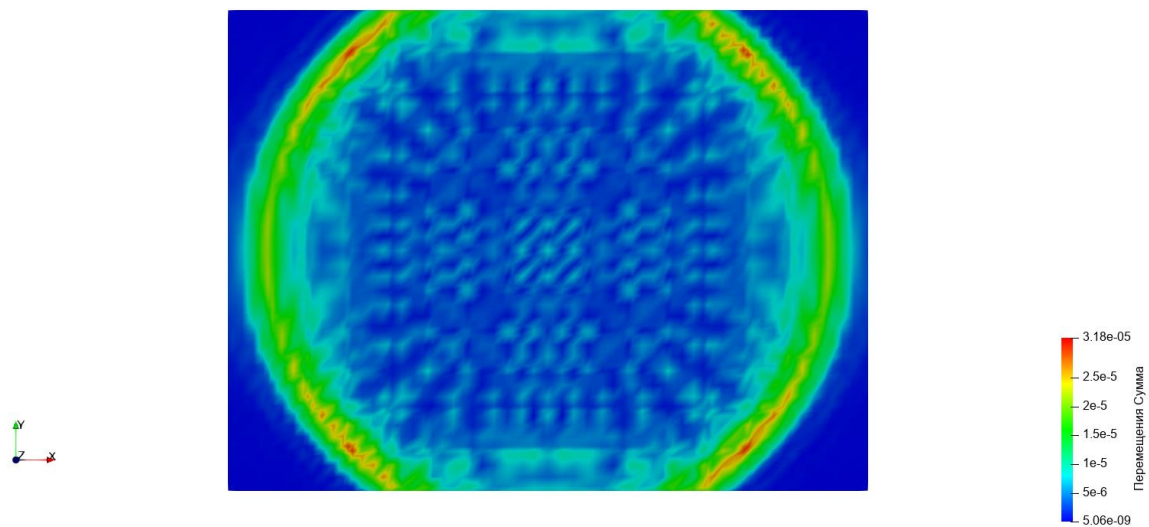


Рисунок 19: 5 порядок 50 сетка

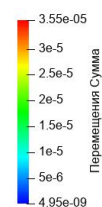
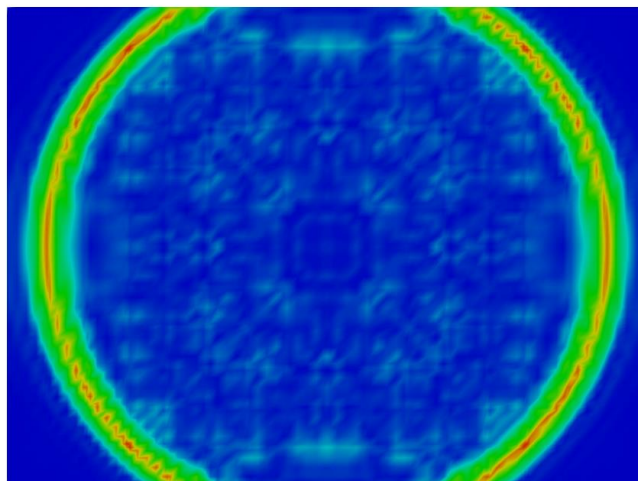


Рисунок 20: 6 порядок 50 сетка

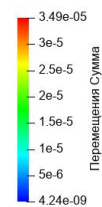
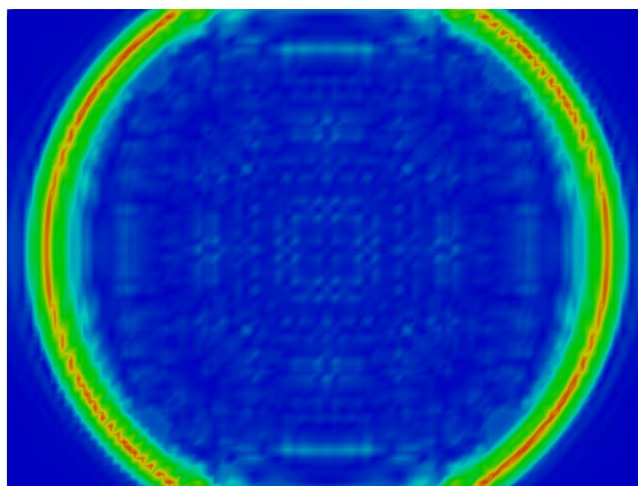


Рисунок 21: 7 порядок 50 сетка

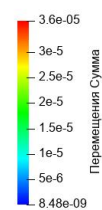
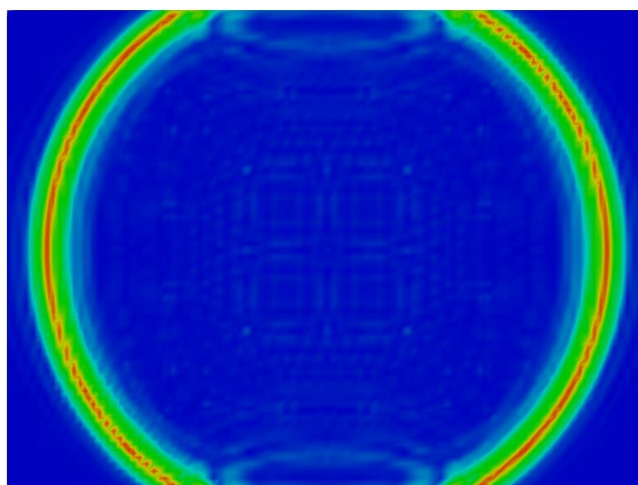


Рисунок 22: 8 порядок 50 сетка

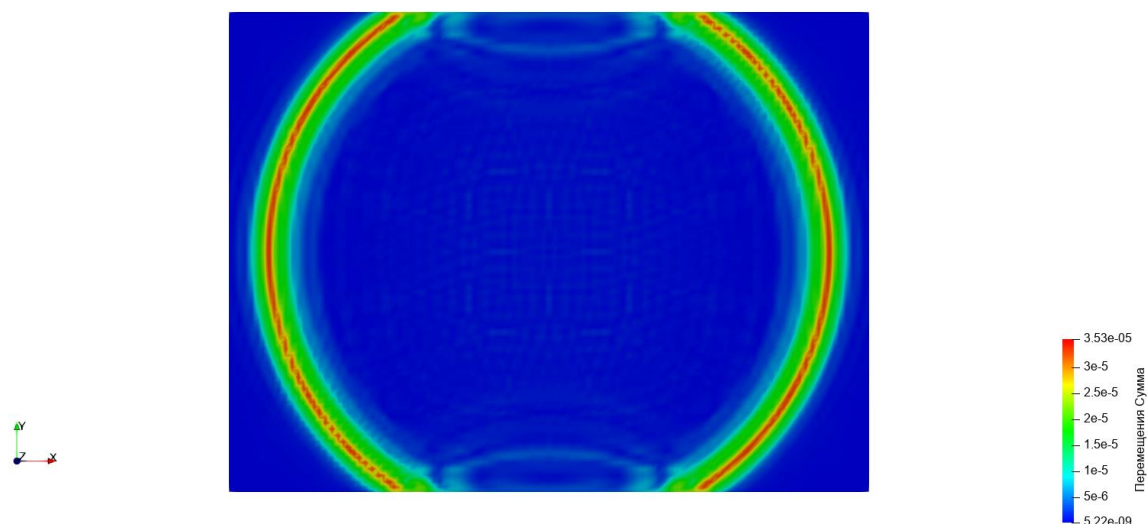
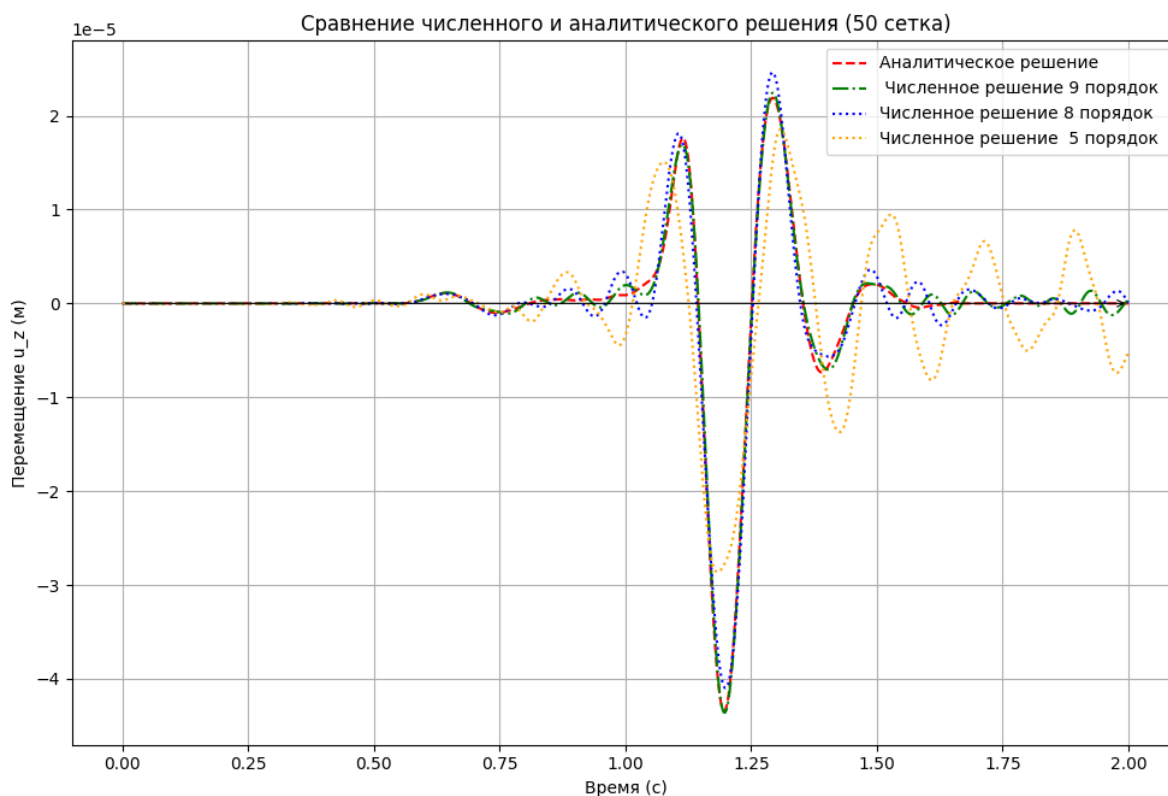


Рисунок 23: 9 порядок 50 сетка

9) График сравнения численного и аналитического решений (50 сетка)



10) Разница численных и аналитического решений (50 сетка)

Разница аналитического решения и решения 1-го порядка = 104.0%

Разница аналитического решения и решения 2-го порядка = 96.5%

Разница аналитического решения и решения 3-го порядка = 80.8%

Разница аналитического решения и решения 4-го порядка = 58.4%

Разница аналитического решения и решения 5-го порядка = 39.0%

Разница аналитического решения и решения 6-го порядка = 22.2%

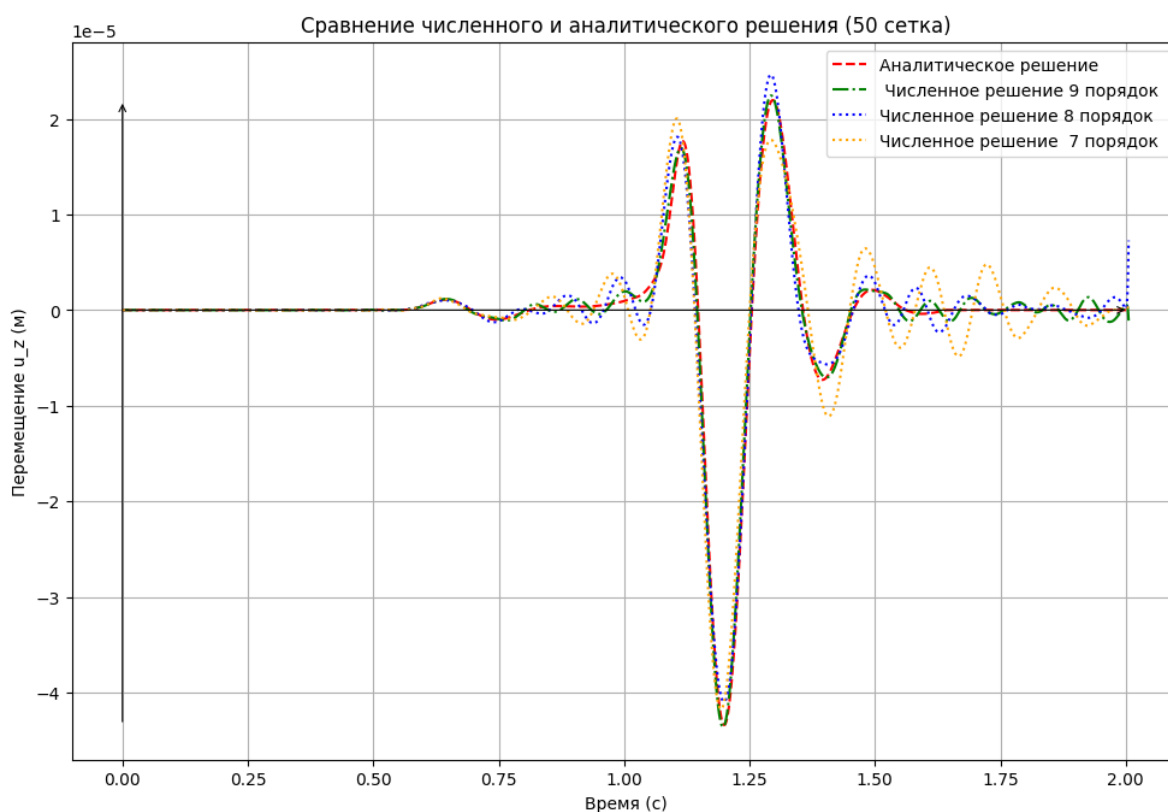
Разница аналитического решения и решения 7-го порядка = 18.2%

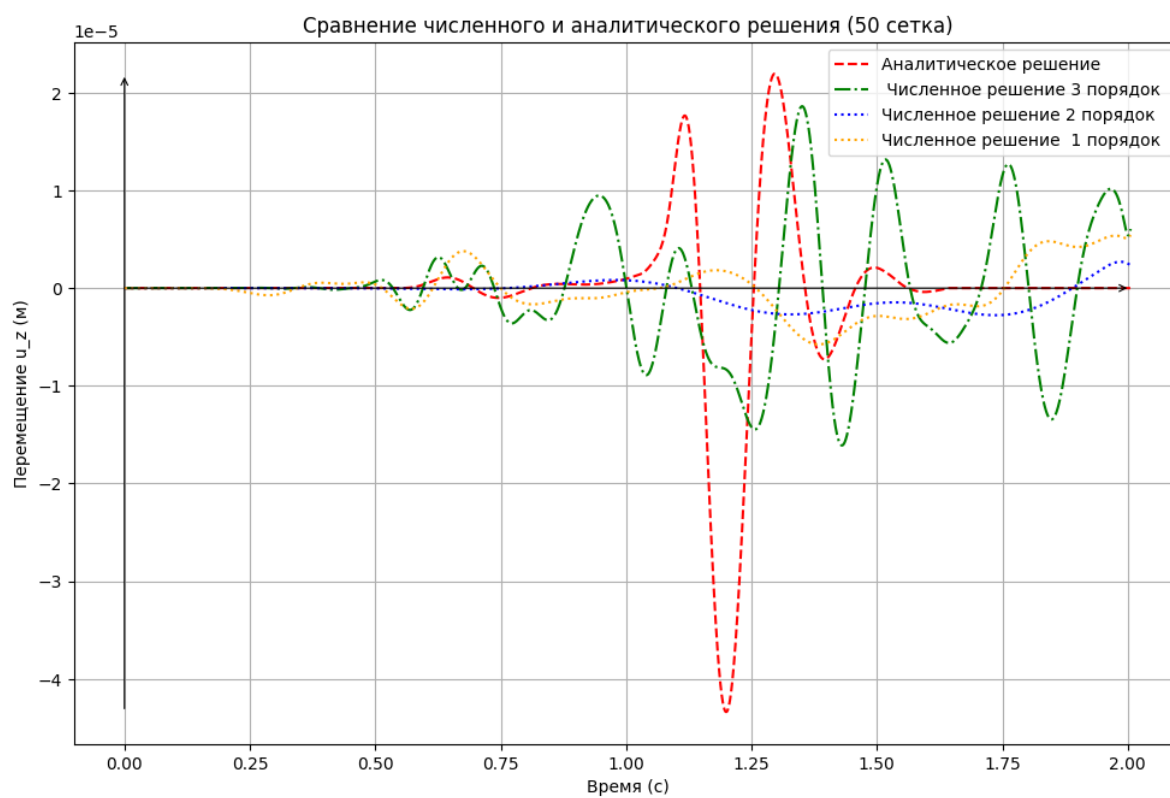
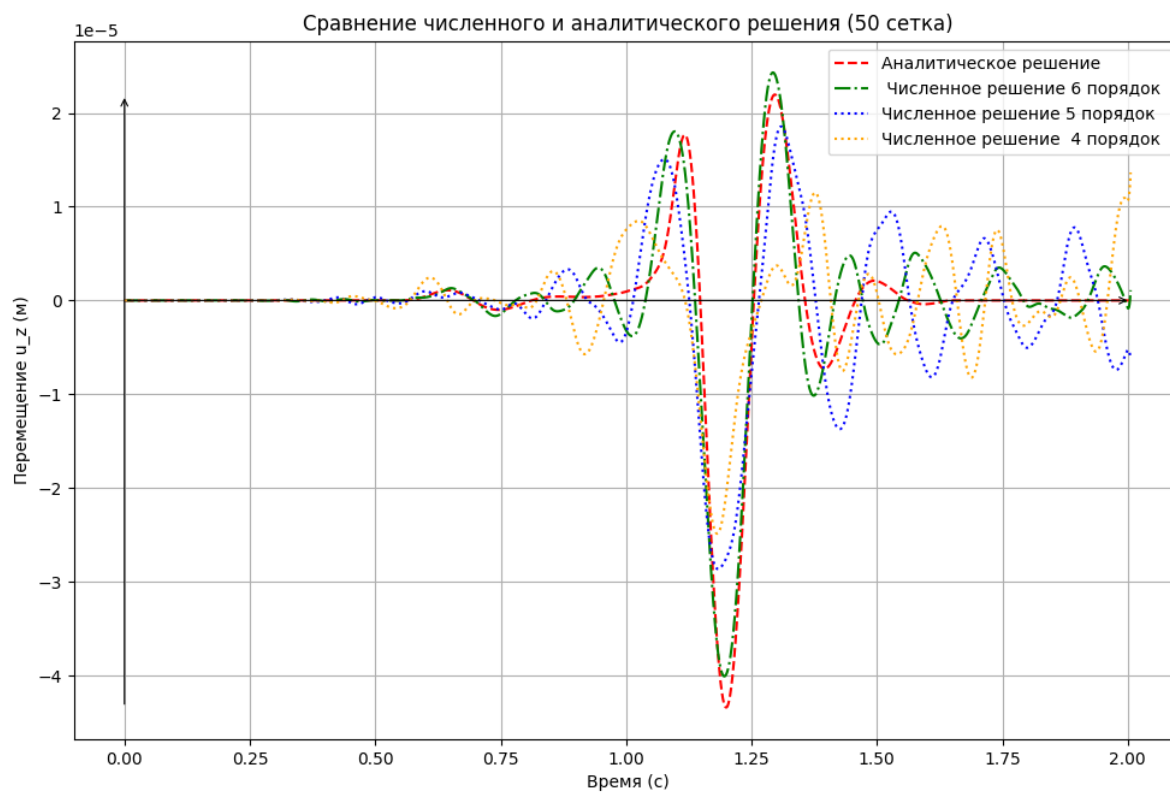
Разница аналитического решения и решения 8-го порядка = 16.9%

Разница аналитического решения и решения 9-го порядка = 5.7%

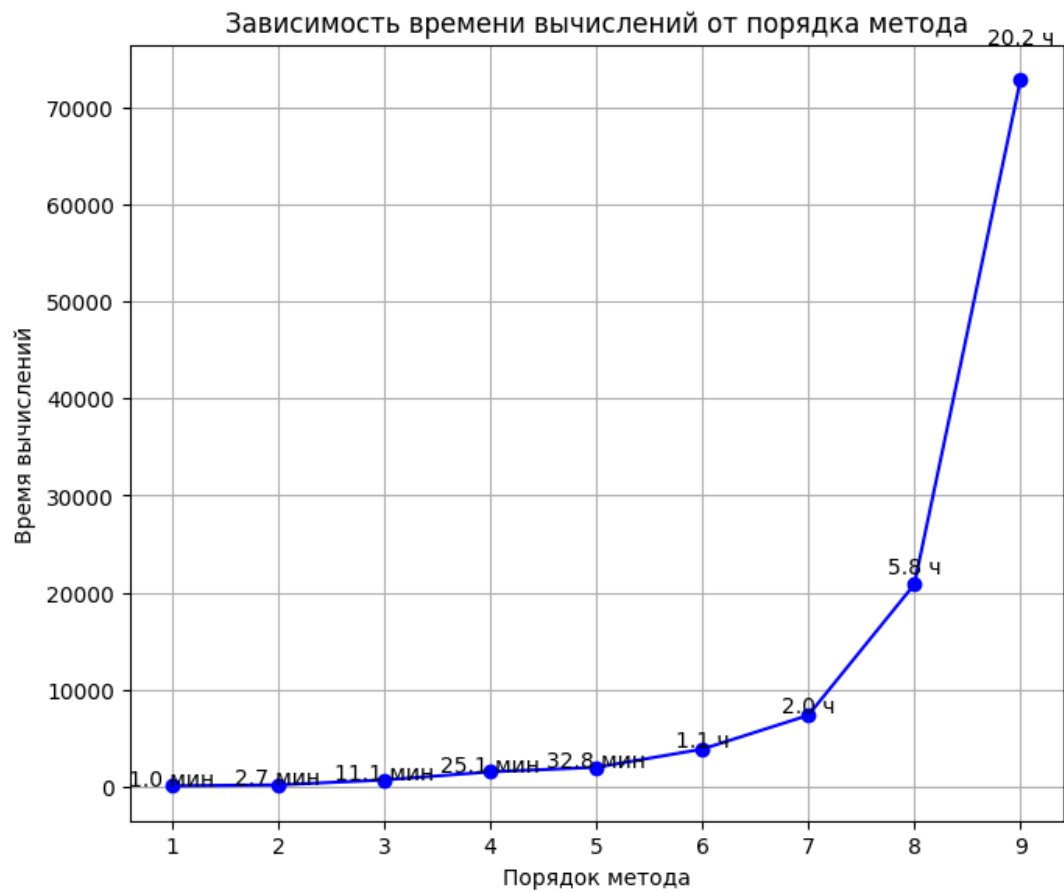
Видно, что если сравнить разницу между аналитическим и численным решениями одинаковых порядков для сеток 50 и 100, то ошибка между аналитическим и численным решениями становится меньше при уменьшении ячейки сетки.

Ниже представлены рисунки сравнения аналитического решения со всеми порядками численного решения для сетки 50.





11) График времени вычисления в моделях различных порядков (50 сетка)



12) Галерея распространения волн для различных порядков (25 сетка).

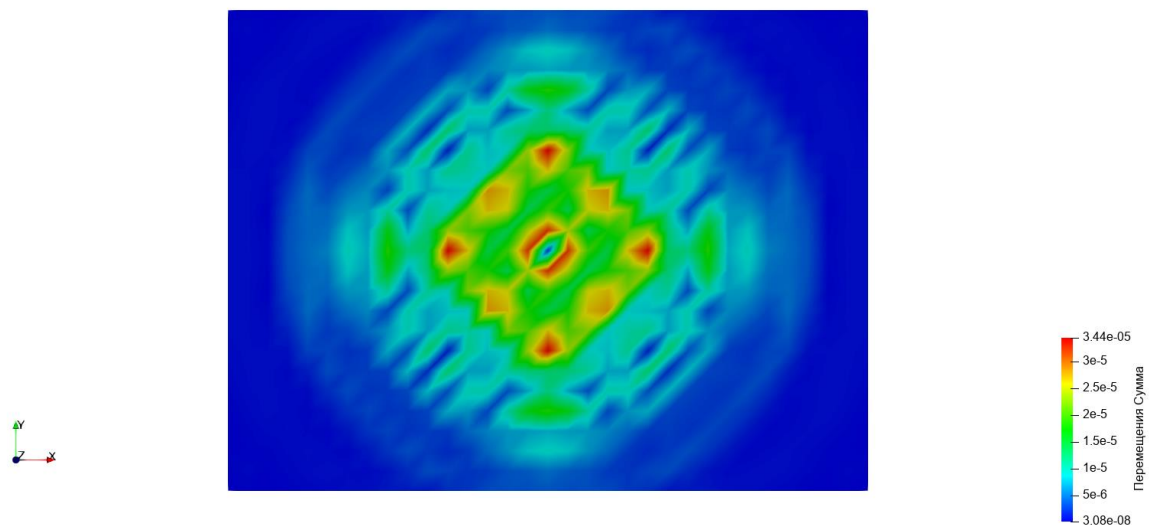


Рисунок 24: 1 порядок 25 сетка

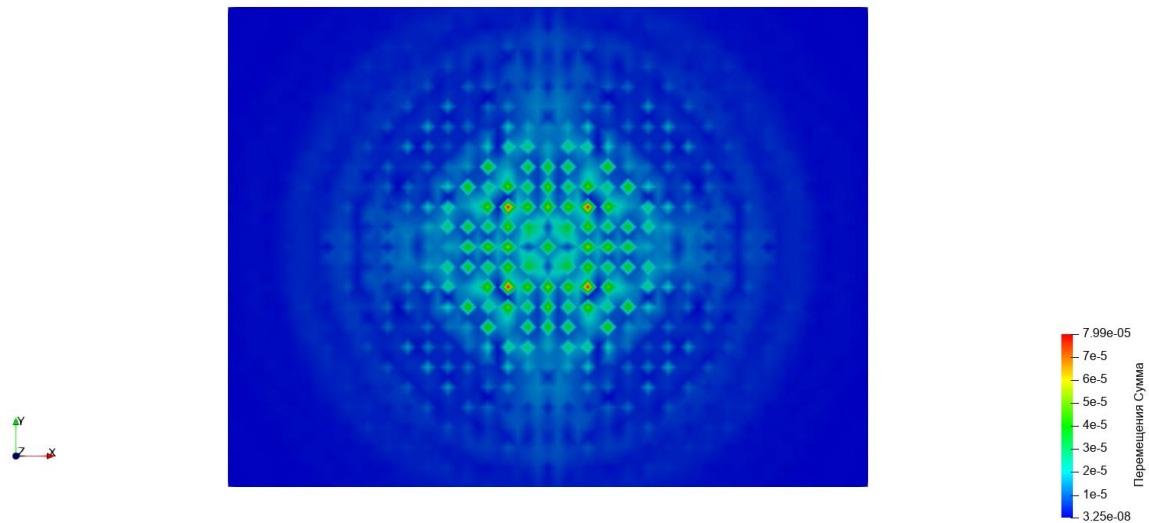


Рисунок 25: 2 порядок 25 сетка

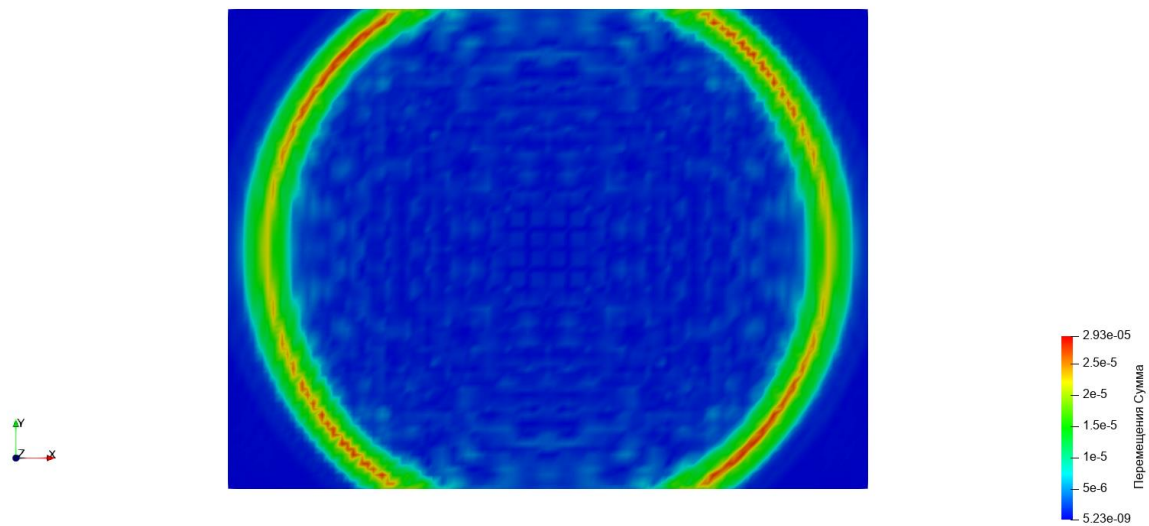


Рисунок 26: 3 порядок 25 сетка

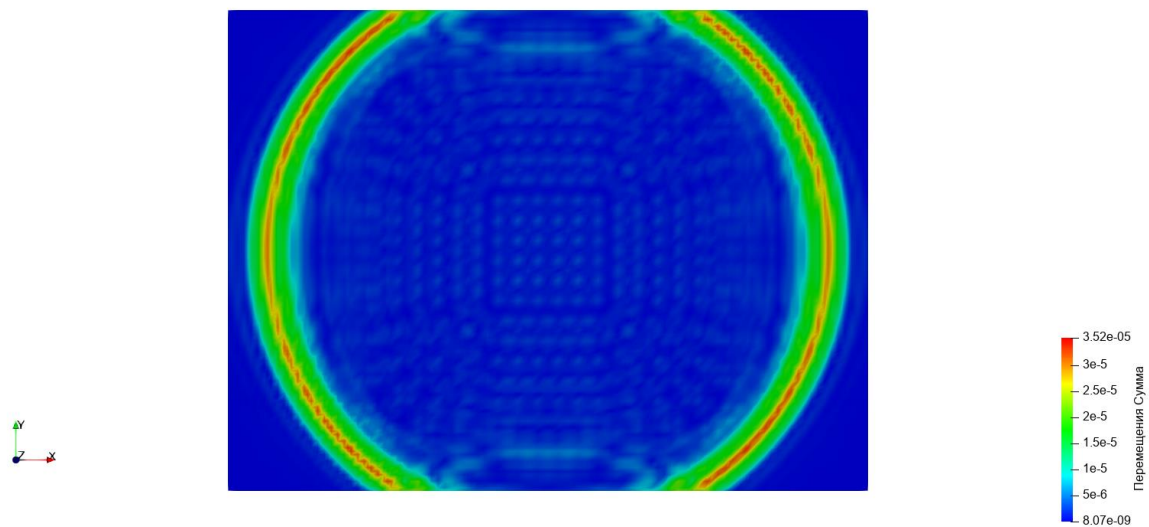


Рисунок 27: 4 порядок 25 сетка

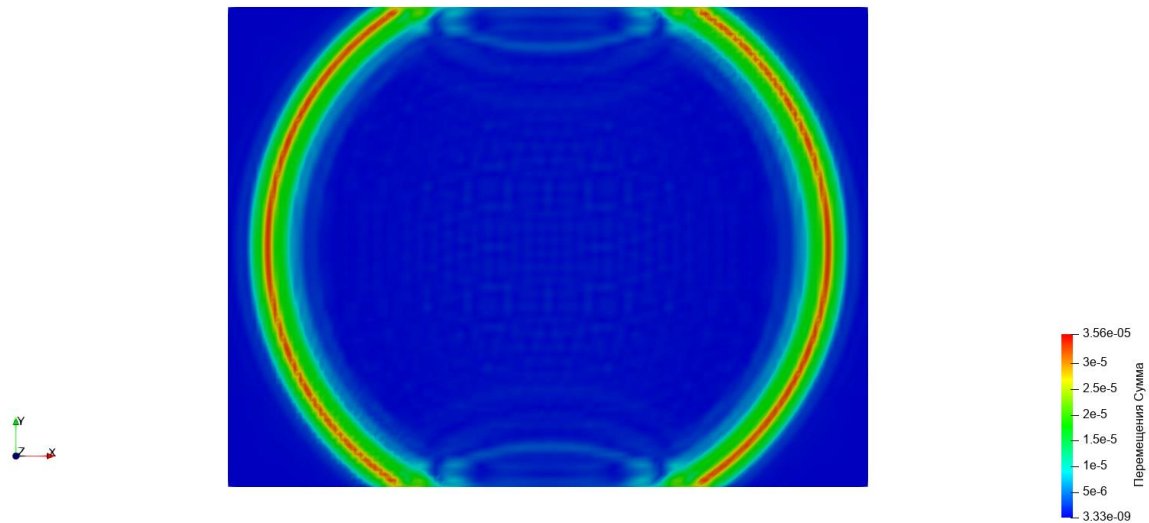


Рисунок 28: 5 порядок 25 сетка

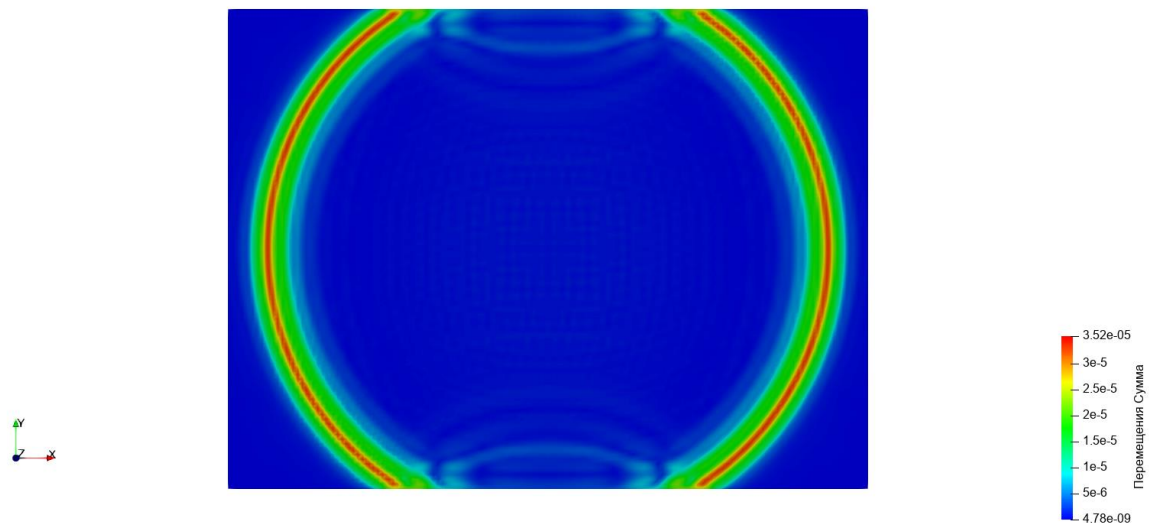


Рисунок 29: 6 порядок 25 сетка

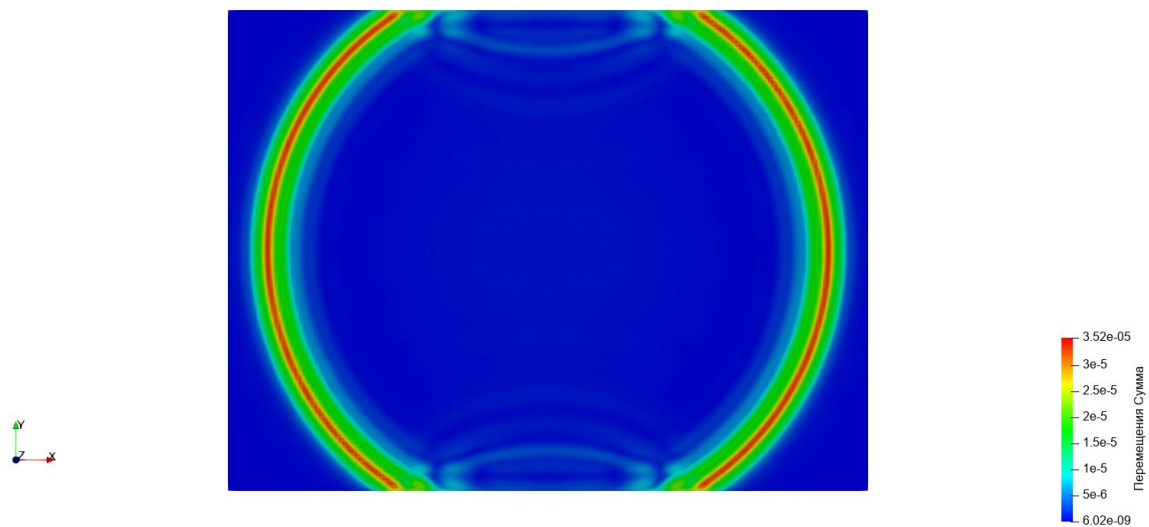


Рисунок 30: 7 порядок 25 сетка

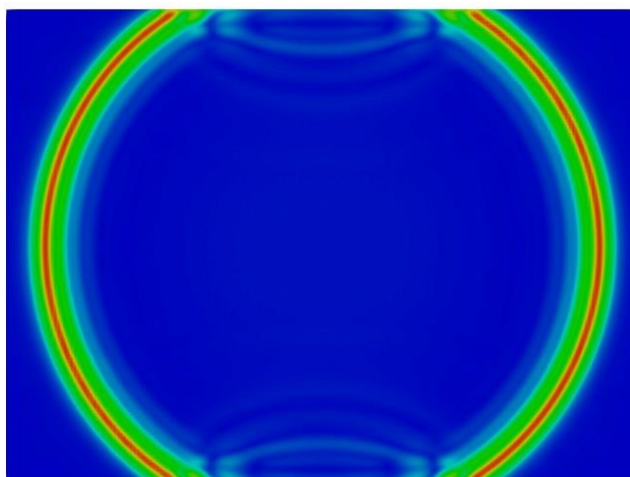


Рисунок 31: 8 порядок 25 сетка

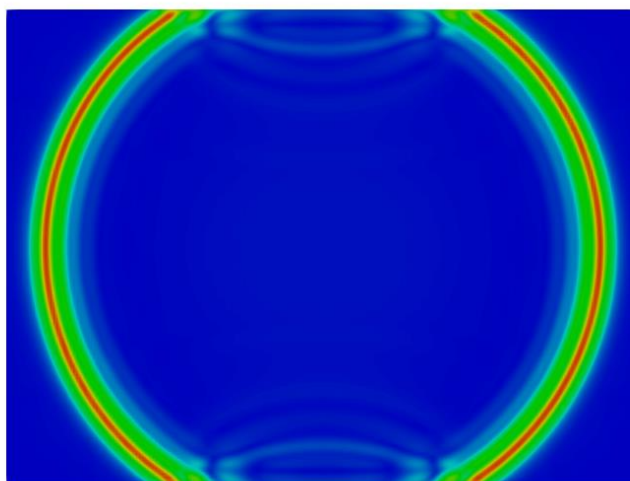
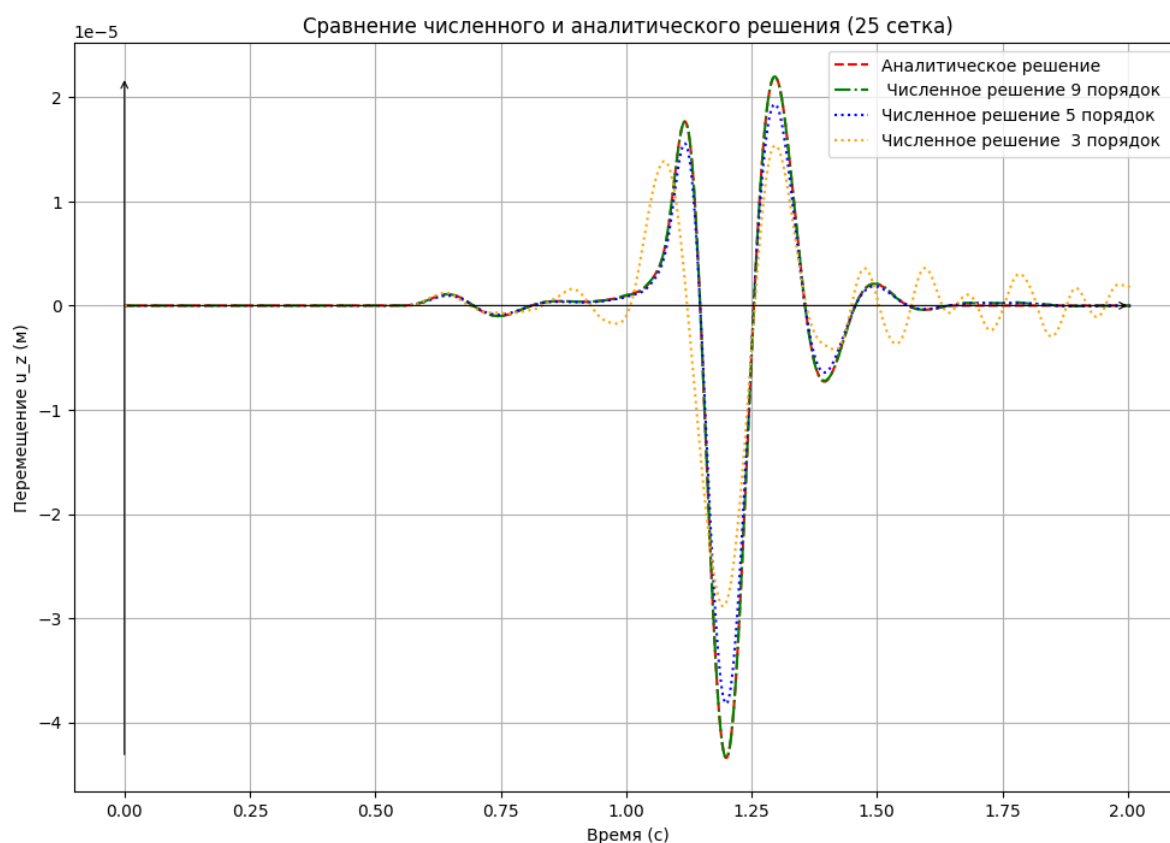


Рисунок 32: 9 порядок

13) График сравнения численного и аналитического решений (25 сетка)



14) Разница численных и аналитического решений (25 сетка)

Разница аналитического решения и решения 1-го порядка = 109.5%

Разница аналитического решения и решения 2-го порядка = 88.7%

Разница аналитического решения и решения 3-го порядка = 43.5%

Разница аналитического решения и решения 4-го порядка = 18.0%

Разница аналитического решения и решения 5-го порядка = 12.0%

Разница аналитического решения и решения 6-го порядка = 9.2%

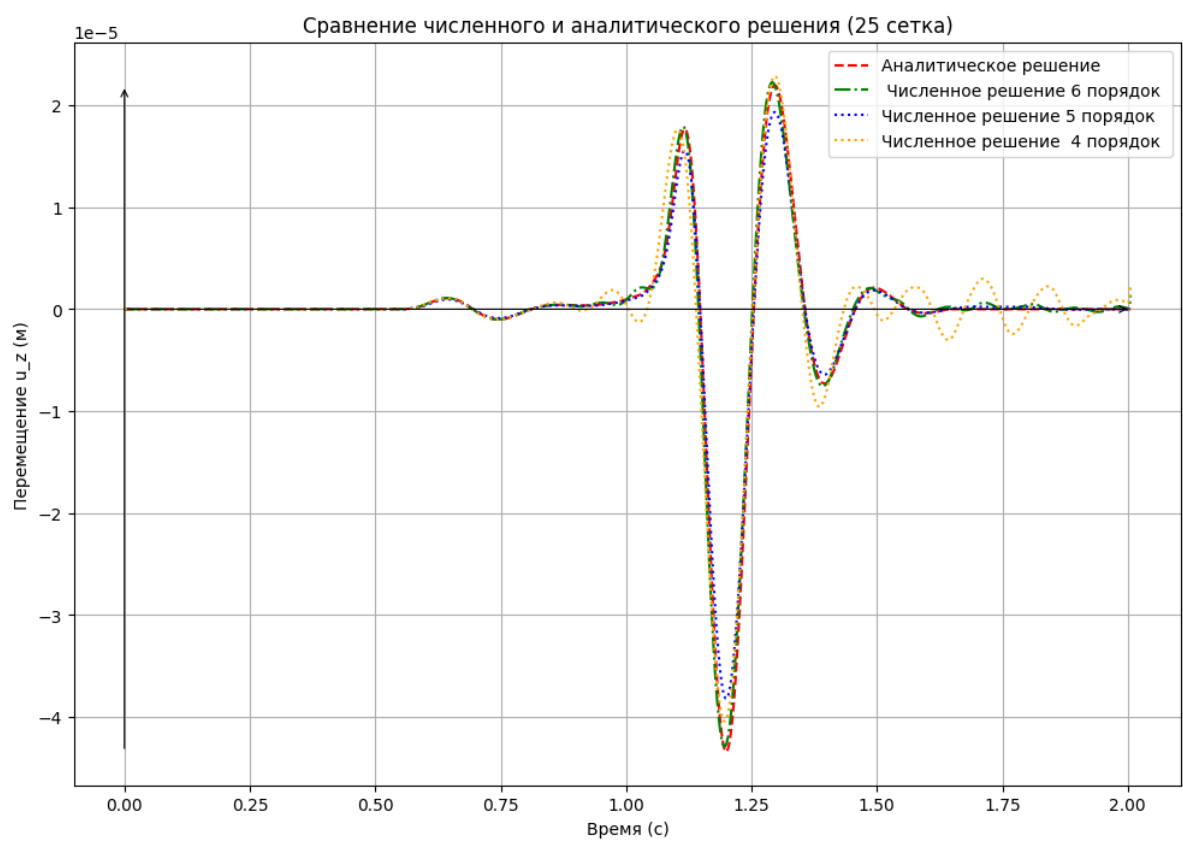
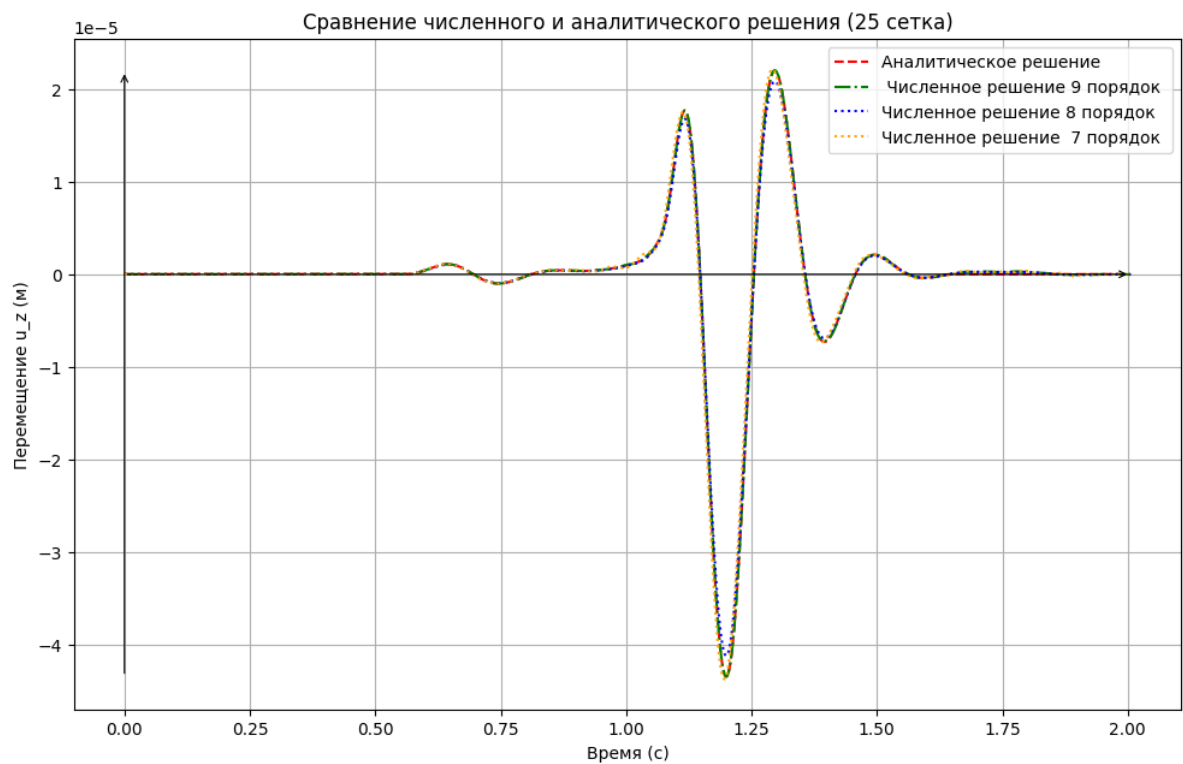
Разница аналитического решения и решения 7-го порядка = 8.5%

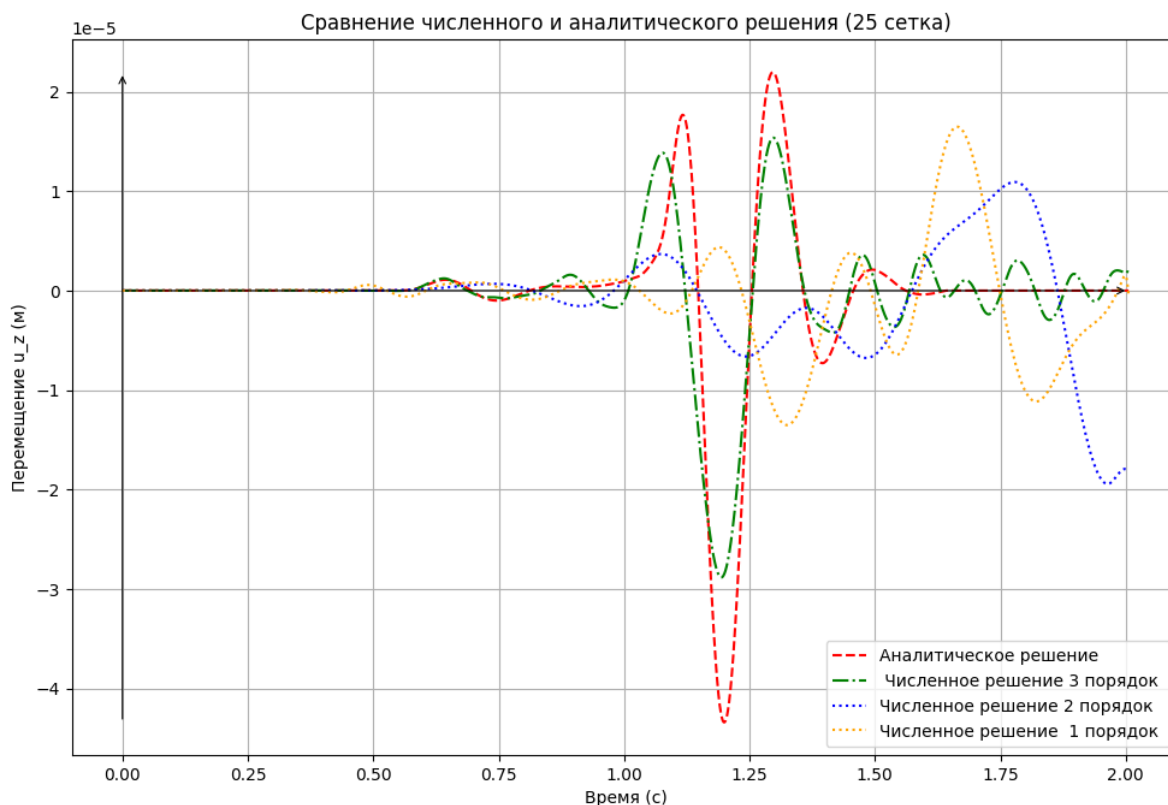
Разница аналитического решения и решения 8-го порядка = 5.2%

Разница аналитического решения и решения 9-го порядка = 0.2%

Для сетки размера 25 удалось добиться разницы между численным и аналитическим решением в 1 процент для 9-го порядка.

Ниже представлены рисунки сравнения аналитического решения со всеми порядками численного решения для сетки 25.

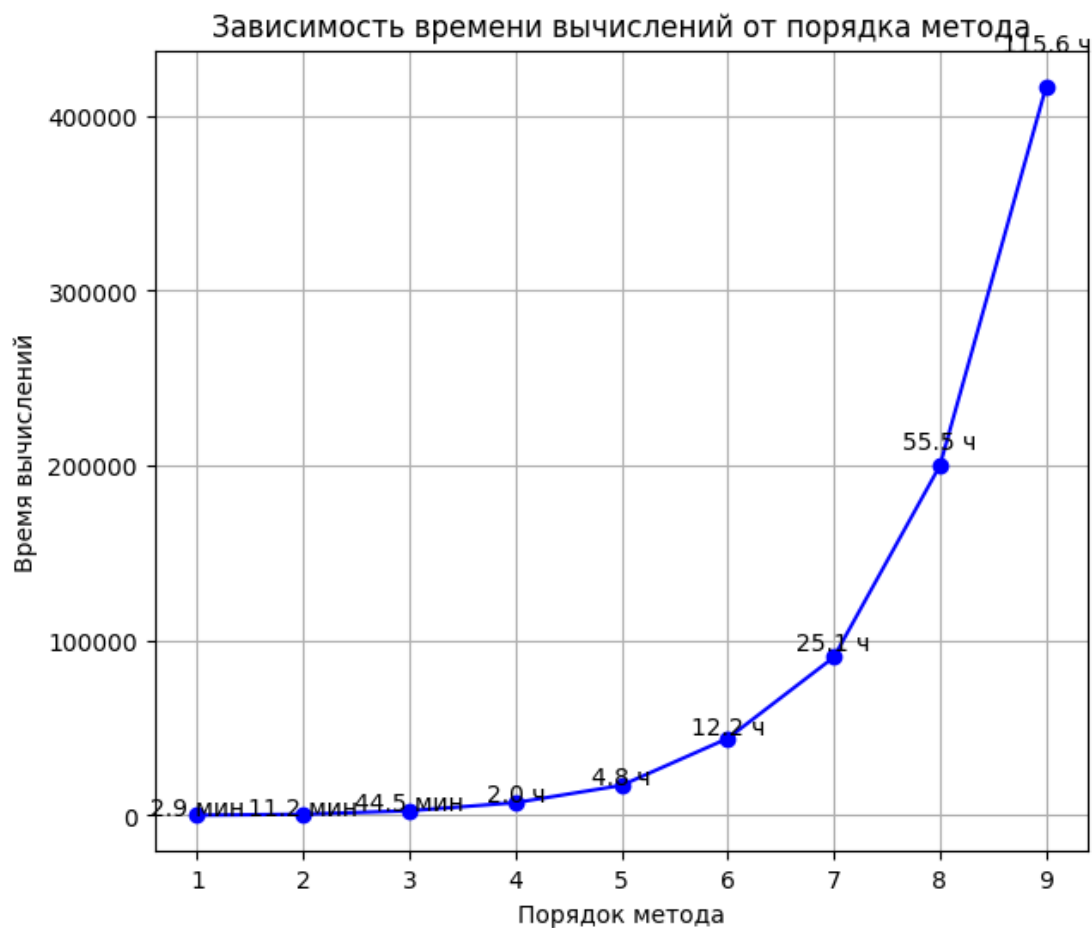




Вывод: на графиках приведено аналитическое и численное решение задачи Лэмба. При их анализе можно увидеть, что в точке тела, удаленной от точки приложения нагрузки, материал не деформируется до момента прихода продольной волны, которая вызывает скачкообразные перемещения. Далее прибывает поперечная волна. Пересечение продольной волны со свободной поверхностью полуплоскости вызывает волну Рэлея, которая оказывает наибольшее влияние. После прохождения рэлеевской волны устанавливается статическое распределение перемещений поверхности. Также, наблюдается практически полное совпадение аналитического решения и конечноэлементного решения 9-го порядка во всем исследовавшемся временном диапазоне.

Вывод: на графике видно, что с повышением порядка элемента решение сходится и приближается к аналитическому графику. На рисунке выделяется решение задачи Лэмба в **CAE Fidesys** методом конечных элементов (например, пятый порядок), для которого используемое разбиение сетки недостаточно (ошибка 12%): требуется измельчить сетку, что потребует, в свою очередь, больше расчетного времени. Следовательно, удобнее и выгоднее применять МСЭ как достаточно точный, современный и менее загруженный по вычислительному времени численный метод.

15) График времени вычисления в моделях различных порядков (25 сетка)



1. Что означает порядок элемента?

В МКЭ и МСЭ внутри каждого конечного элемента мы аппроксимируем искомую функцию (например, смещение, давление, температуру) с помощью базисных функций.

- 1-й порядок (линейные элементы): Функция аппроксимируется линейно между узлами элемента (например, для треугольного элемента — линейная интерполяция).
- 2-й порядок (квадратичные элементы): Добавляются дополнительные узлы внутри элемента, а функция аппроксимируется квадратичным полиномом.
- 3-й порядок и выше: Используются полиномы третьей, четвёртой и более высокой степени.

2. Влияние порядка на точность и вычислительную сложность

- Чем выше порядок, тем точнее решение, так как функция лучше интерполируется внутри элемента.
- Но с ростом порядка увеличивается число степеней свободы (узлов), что приводит к увеличению объёма вычислений.

3. Порядок от 1 до 9 в МСЭ

В методе спектральных элементов (МСЭ) обычно используют высокие порядки (до 9 и выше). Это связано с тем, что в МСЭ применяются полиномы Чебышева или Лагранжа высокой степени для интерполяции решения внутри элемента.

- При порядке 9 внутри одного элемента может быть 10 узлов (по схеме Гаусса-Лобатто), а функция аппроксимируется полиномом 9-й степени.

Раздел №10: Зависимость числа узлов/элементов от порядка.

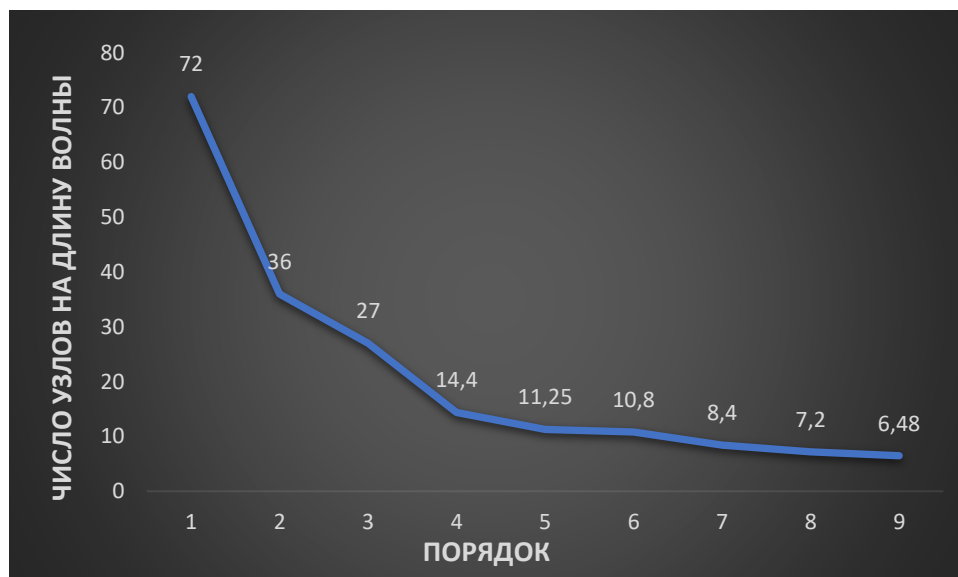
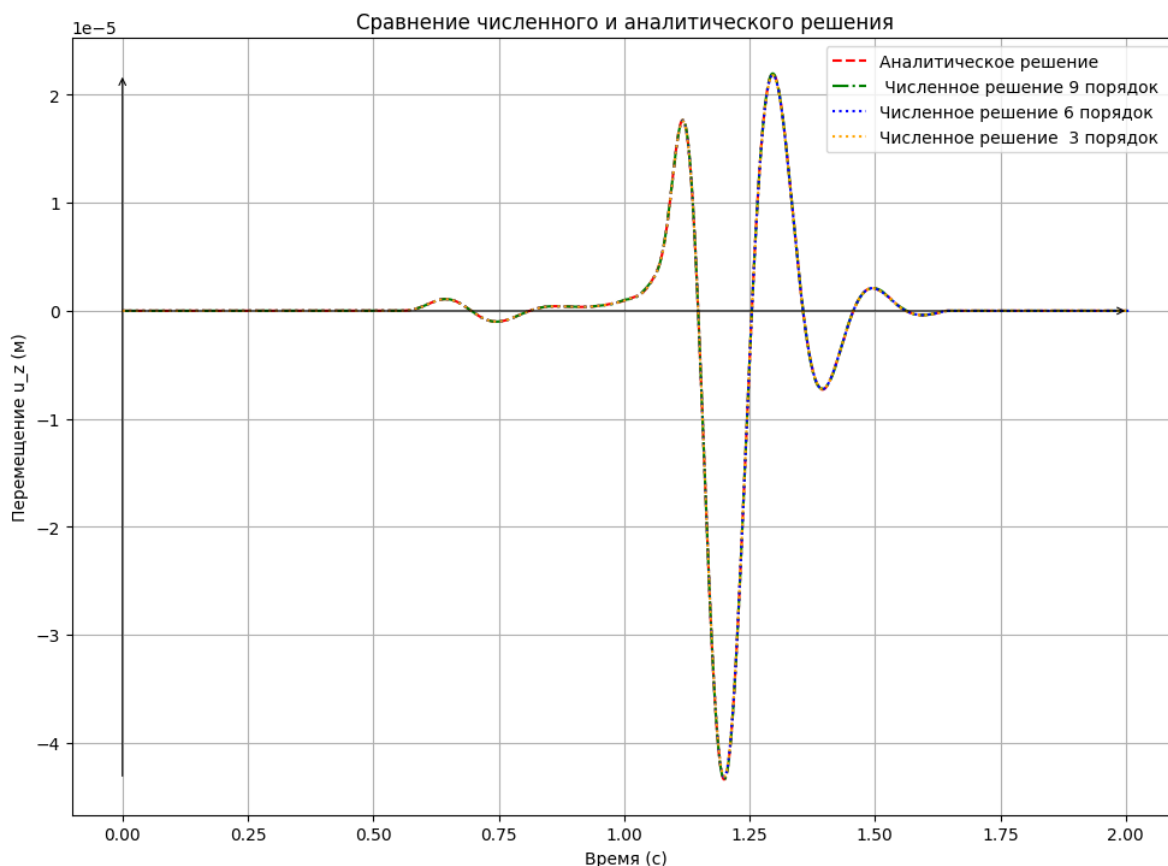


Рис. 33. Зависимость числа узлов/элементов на длину волны, необходимых для достижения погрешности в 1% от порядка МСЭ.

Порядок	Размер элемента	Число элементов на длину волны (EPW)	Число точек на длину волны (PPW)	Погрешность %
1	0,25	72	72	0.82
2	1	18	36	0.7
3	2	9	27	0.67
4	5	3,6	14,4	0.6
5	8	2,25	11,25	0.54
6	10	1,8	10,8	0.5
7	15	1,2	8,4	0.48
8	20	0,9	7,2	0.35
9	25	0,72	6,48	0.22

Вывод: с увеличением порядка уменьшается число узлов в сетке, необходимых для достижения погрешности в 1%.



Раздел №11: Результаты работы и выводы.

1) Выполнены интегральные преобразования Лэмба, в результате которых получены неоднородные уравнения Гельмгольца для скалярного и векторного потенциалов. Получены формулы аналитического решения задачи Лэмба для нестационарной волны. Построены графики изменения компонент перемещений во времени.

2) Решение задачи Лэмба получено также методом конечных элементов. Проведено сравнение численного решения с аналитическим решением. Вывод: наблюдается полное совпадение аналитического решения и конечноэлементного решения во всем исследовавшемся временном диапазоне. Установлено, что при увеличении порядка аппроксимации снижается необходимое число узлов в сетке для достижения заданной точности (погрешность не более 1%). На основе разработанного алгоритма (скрипта) для программы CAE Fidesys становится возможным провести численные расчеты для многослойных сред, где аналитическое решение невозможно.

3) Проведено детальное сравнение аналитического и численного решений задачи Лэмба. Для численного решения задачи Лэмба на различных порядках МСЭ (вплоть до 9-го) удалось добиться разницы с аналитическим решением, не превышающей 1%. В работе проведен анализ зависимости числа точек на длину волны Рэлея (points per wavelength – PPW) для различных порядков МСЭ на примере задачи Лэмба для достижения погрешности не более 1% в сравнении с аналитическим решением. Показано, что с увеличением порядка МСЭ PPW резко сокращается, что

свидетельствует о преимуществе использования МСЭ высоких порядков в задачах полноволнового моделирования.

Раздел №12: Литература и источники:

- 1) Lamb H. On the Propagation of Tremors over the Surface of an Elastic Solid. Philosophical Transactions of the Royal Society of London. Ser. A. 1904, vol. 203, pp. 1—42.
- 2) Lamb H. On waves due to a travelling disturbance, with an application to waves in superposed fluids. Philosophical Magazine, 1916, vol. 13, pp. 386—399, 539—548.
- 3) Lord Rayleigh. "On waves propagated along the plane surface of an elastic solid." Proceedings of the London Mathematical Society, 1885.
- 4) Cagniard R. Réflexion et réfraction des ondes séismiques progressives. Gauthier-Villars, 1939.
- 5) Братов В.В. Метод Каньяра в решении задач распространения волн в упругих средах. Прикладная математика и механика, 2022.
- 6) Поручиков В.Б. Методы динамической теории упругости в задачах сейсмологии. М.: Наука, 1986. П. МИР, Москва, 360 стр., УДК: 53+55
- 7) Викторов И.А. Поверхностные акустические волны в твёрдых телах. Наука, 1981.
- 8) Антонов А.М. Волна Рэлея на границе градиентно-упругого полупространства. Вестник МГТУ им. Н.Э. Баумана, 2015.
- 9) Городецкая Н.С., Соболев Т.В., Зубарева Л.П. Волны на границе пористо-упругого полупространства. Акустический вестник, 2008.
- 10) Осетров А.В., Нгуен Ван Шо. Расчёт параметров поверхностных акустических волн в пьезоэлектриках методом конечных элементов. Вычислительная механика сплошных сред, 2011.
- 11) Бойко О.В. Оценка свойств и состояния грунтов за обделкой транспортных тоннелей по данным 2D-сейсмотомографии. Геофизический журнал, 2010.
- 12) Багдасарян А. Устройства на поверхностных акустических волнах в системах и средствах связи. Радиотехника и электроника, 2005.