

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
**МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**  
**Кафедра теоретической и прикладной механики**

ДРОБЫШЕВСКИЙ

Василий Александрович

**РАСПАРАЛЛЕЛЕННЫЙ ГРАНИЧНО-ЭЛЕМЕНТНЫЙ РАСЧЁТ ТРЕХ-  
МЕРНОГО НАПРЯЖЕННОГО СОСТОЯНИЯ**

Дипломная работа

Научный руководитель:  
доктор физ.-мат. наук,  
профессор С. С. Щербаков

Допущен к защите

«\_\_\_» \_\_\_\_\_ 2022 г.

Зав. кафедрой теоретической и прикладной механики

доктор физ.-мат. наук, профессор М. А. Журавков

Минск, 2022

## ОГЛАВЛЕНИЕ

<b>РЕФЕРАТ.....</b>	3
<b>РЭФЕРАТ.....</b>	4
<b>THESIS.....</b>	5
<b>ВВЕДЕНИЕ.....</b>	6
<b>ГЛАВА 1. АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ НАГРУЖЕНИЯ ПОЛУПРОСТРАНСТВА.....</b>	7
1.1 Решение для сосредоточенной силы.....	7
1.2 Решение для распределённой по прямоугольнику нагрузки.....	8
<b>ГЛАВА 2. ПОСЛЕДОВАТЕЛЬНОЕ РЕШЕНИЕ О ВДАВЛИВАНИИ ПЛОСКОГО ЭЛЛИПТИЧЕСКОГО ШТАМПА В ПОЛУПРОСТРАНСТВО.....</b>	24
2.1 Постановка задачи.....	24
2.2 Дискретизация функции поверхностного распределения.....	25
2.3 Получение распределения напряжений .....	28
<b>ГЛАВА 3. РАСПАРАЛЛЕЛИВАНИЕ ПОСЛЕДОВАТЕЛЬНОГО РЕШЕНИЯ ЗАДАЧИ О ВДАВЛИВАНИИ ПЛОСКОГО ЭЛЛИПТИЧЕСКОГО ШТАМПА В ПОЛУПРОСТРАНСТВО.....</b>	32
3.1 Некоторые сведения о технологии NVIDIA CUDA.....	32
3.2 Распараллеливание последовательного решения и сравнение результатов.....	34
<b>ЗАКЛЮЧЕНИЕ .....</b>	36
<b>СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....</b>	37
<b>Приложение А: листинг в программе Wolfram Mathematica для параллельного расчета статического случая неравномерного нагружения полупространства .....</b>	38

## **РЕФЕРАТ**

В дипломной работе 29 страниц, 28 рисунков, 9 источников, 1 приложение.

**КОМПЬЮТЕРНАЯ МЕХАНИКА, ГРАНИЧНО-ЭЛЕМЕНТНОЕ МОДЕЛИРОВАНИЕ, РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЙ, ТЕХНОЛОГИЯ NVIDIA CUDA, ПРОГРАММИРОВАНИЕ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ.**

В данной дипломной работе была смоделирована задача о неравномерном нагружении полупространства. Задача была решена аналитически и численно, при решении использовалась вычислительная программа Wolfram Mathematica, с установленным пакетом CUDA Development Kit 11.7.

По итогам работы было получено многократное ускорение расчётного времени (до 100 раз). Результаты были достигнуты при помощи встроенных в пакет Wolfram Mathematica функций, написанных с использованием технологии CUDA.

Работа является актуальной в связи с увеличением сложности задач, рассматриваемых в рамках компьютерной механики и ограниченностью вычислительных возможностей центральных процессоров. Актуальность полученных результатов будет возрастать со временем.

## **РЭФЕРАТ**

У дыпломнай работе 29 старонак, 28 малюнкаў, 9 крыніц, 2 дадатка.

**КАМПУТАРНАЯ МЕХАНІКА, МЕЖАВА-ЭЛЕМЕНТНАЕ МАДЭЛЯ-  
ВАННЕ, РАСПАРАЛЕЛЬВАННЕ ВЫЛІЧЭННАЯ, ТЭХНАЛОГІЯ NVIDIA  
CUDA, ПРАГРАМАВАННЕ ГРАФІЧНЫХ ПАСКАРАЛЬNIКАЎ.**

У дыпломнай працы была змадэдлявана задача пра нераўнамернае нагруженне паупространства. Задача была вырашана аналітычна і лікава, пры рашэнні выкарыстоўвалась вылічальная праграма Wolfram Mathematica, з устаноўленым пакетам CUDA Developer Toolkit 11.7.

Па выніках працы было атрымана шматразовае скарачэнне разліковага часу (да 1500 раз). Вынікі былі дасягнуты з дапамогай убудаваных у пакет Wolfram Mathematica функцый, напісаных з выкарыстаннем тэхнологіі CUDA.

Праца з'яўляецца актуальнай у сувязі з павеличэннем складанасці задач, разгляданых у рамках кампьютарнай механікі і абмежаванасцю вылічальных магчымасцей цэнтральных працэсараў. Актуальнасць атрыманых вынікаў будзе ўзрастаць з часам.

# **THESIS**

Diploma contains 29 pages, 28 pictures, 9 sources, 1 appendix.

**COMPUTER MECHANICS, BOUNDARY-ELEMENT MODELING, PARALLELIZING OF COMPUTATIONS, NVIDIA CUDA TECHNOLOGY, PROGRAMMING OF GRAPHICS PROCESSORS**

Problem of non-linear loading of a half-space was modeled in this diploma. This problem was solved analytically and numerically, solution was done using Wolfram Mathematica program and installed CUDA SDK 11.7.

Based on results of the solved problem, a multiple reduction in estimating time (up to 100) was obtained. Results was achieved using built-in in Wolfram Mathematica functions, which written using CUDA technology.

This work is relevant in connection with growth of problems difficulty under computer mechanics consideration and limitation of computational possibilities of central processors. The relevance of the results obtained will increase over time.

## ВВЕДЕНИЕ

Современные задачи деформируемого твёрдого тела, которые компьютерная механика ставит перед инженерами, требуют качественного анализа сложных систем. Такой анализ может быть ограничен временем расчёта напряженно-деформированного состояния рассматриваемой системы для заданного набора входных параметров. Например, расчёт режущего аппарата сельскохозяйственного комбайна или системы типа колесо-рельс [1,2] с помощью распространённых пакетов конечно-элементного моделирования, таких как Ansys, SOLID-WORKS, NX NASTRAN и т.д. может длится около недели.

Так как при использовании наиболее популярного метода численного моделирования подобных задач — Метода Конечных Элементов особенностью аппроксимации является связь, между соседними узлами расчётной сетки, то распараллеливание расчёта на ядрах центрального процессора обеспечивает сокращение времени вычислений не более чем в 2-4 раза. Такой вывод приводит к необходимости поиска новых возможностей для ускорения вычислений.

При гранично-элементной аппроксимации рассматриваются граничные интегральные уравнения, которые позволяют производить расчёт взаимных влияний граничных элементов (расчётных узлов) независимо друг от друга, таким образом формируя матрицу системы линейных алгебраических уравнений [3 — 5]. Следовательно, при моделировании задач гранично-элементным методом, все искомые значения могут быть определены параллельно [6].

Целью работы является решение задачи о вдавливании плоского эллиптического штампа в полупространство методом граничных элементов последовательно, распараллеливание этого решения и сравнение времени, необходимо на оба решения.

# ГЛАВА 1

## АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ НАГРУЖЕНИЯ ПОЛУ- ПРОСТРАНСТВА

### 1.1 Решение для сосредоточенной силы

Рассмотрим задачу нагружения трёхмерного полупространства, которое является однородной изотропной упругой средой, нагрузкой, распределённой по прямоугольной области, при этом нагрузка действует вдоль вертикальной оси. Необходимо аналитически определить напряженно-деформированное состояние среды, получить общую формулу для расчёта элементов тензора напряжений и вектора перемещений.

Рассмотрим для начала аналогичную задачу для сосредоточенной силы. После чего полученные решения проинтегрируем по заданному прямоугольнику.

Полученное Кельвином решение для сосредоточенной силы, действующей в данной среде, имеет вид:

$$u_i(x) = G_{ij}(x, \xi) * e_j(\xi),$$

где  $e_j(\xi)$  — единичные силы, приложенные в точке  $\xi_j$  и направленные по осям декартовой системы координат  $x_i$ , причём

$$G_{ij}(x, \xi) = \frac{1}{16\pi\mu(1-\nu)} * \frac{1}{r} \left( (3 - 4\nu)\delta_{ij} + \frac{y_i y_j}{r^2} \right),$$

Здесь  $\mu$  и  $\nu$  — модуль сдвига и коэффициент Пуассона среды соответственно и

$$\begin{aligned} y_i &= x_i - \xi_i, r^2 = y_i * y_i, \\ r &= \sqrt{x_1^2 + x_2^2 + x_3^2}. \end{aligned}$$

Компоненты тензора напряжений определяются следующим образом:

$$\sigma_{ij}(x) = T_{ijk}(x, \xi) e_k(\xi), \text{ где}$$

$$T_{ijk}(x, \xi) = -\frac{1}{8\pi(1-\nu)} \frac{1}{r^2} \left( (\delta_{ik} \frac{y_j}{r} + \delta_{jk} \frac{y_i}{r} - \delta_{ij} \frac{y_k}{r})(1 - 2\nu) + \frac{3y_i y_j y_k}{r^3} \right).$$

## 1.2 Решение для распределённой по прямоугольнику нагрузки

После того, как нами было получено решение задачи о точечном нагружении полупространства, рассмотрим аналогичную задачу, с тем лишь изменением, что теперь будем прикладывать к данной среде нагрузку, распределённую по прямоугольнику. Для этого воспользуемся решениями, полученными в предыдущем пункте. Дважды проинтегрируем решения Кельвина для компонент тензора напряжений и вектора перемещений.

Решение для  $\sigma_{3,3}$ :

$$\begin{aligned} \sigma_{3,3}^I(x, y, z, a, b) &= \int_{-a}^a \sigma_{3,3}(x - p, y, z, a, b) dp, \\ \sigma_{3,3}^{II}(x, y, z, a, b) &= \int_{-b}^b \sigma_{3,3}^I(x, y - k, z, a, b) dk = \\ &\frac{p_0 x_3 (\lambda + \mu)}{4\pi(\lambda + 2\mu)} \left( -\frac{x_1 x_2 (x_1^2 + x_2^2 + 2x_3^2)}{(x_1^2 + x_3^2)(x_2^2 + x_3^2)\sqrt{x_1^2 + x_2^2 + x_3^2}} + \right. \\ &\frac{(-a + x_1)x_2((-a + x_1)^2 + x_2^2 + 2x_3^2)}{((-a + x_1)^2 + x_3^2)(x_2^2 + x_3^2)\sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}} + \frac{x_1(-b + x_2)(x_1^2 + (-b + x_2)^2 + 2x_3^2)}{(x_1^2 + x_3^2)((-b + x_2)^2 + x_3^2)\sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}} + \\ &\frac{(a - x_1)(-b + x_2)((-a + x_1)^2 + (-b + x_2)^2 + 2x_3^2)}{((-a + x_1)^2 + x_3^2)((-b + x_2)^2 + x_3^2)\sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}} + \frac{p_0}{4\pi} \left( -arctg \left[ \frac{x_1 x_2}{x_3 \sqrt{x_1^2 + x_2^2 + x_3^2}} \right] + \right. \\ &arctg \left[ \frac{(-a + x_1)x_2}{x_3 \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}} \right] + arctg \left[ \frac{x_1(-b + x_2)}{x_3 \sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}} \right] - \\ &\left. arctg \left[ \frac{(-a + x_1)(-b + x_2)}{x_3 \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}} \right] \right). \end{aligned}$$

Теперь проинициализируем постоянные, описывающие поведение среды и построим графики напряжений при фиксированных  $x_3$  и  $x_2$ :

$$E = 10^{10}; \nu = 0,3; \lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}; \mu = \frac{E}{2(1 + \nu)};$$

Для  $a = 1; b = 0,7; p_0 = 100; x \in (-1, 1); y \in (-1, 1)$ ; график распределения напряжений будет выглядеть следующим образом.

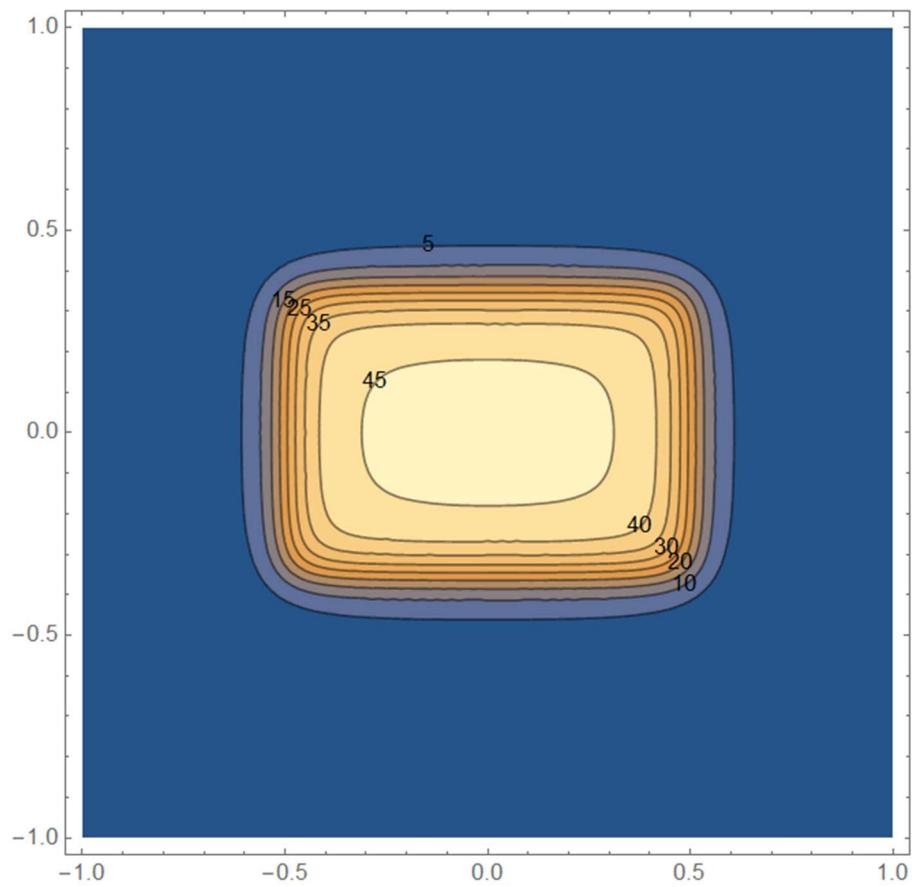


Рисунок 1.1 Распределение  $\sigma_{3,3}$  в полупространстве при фиксированном  $\chi_3$

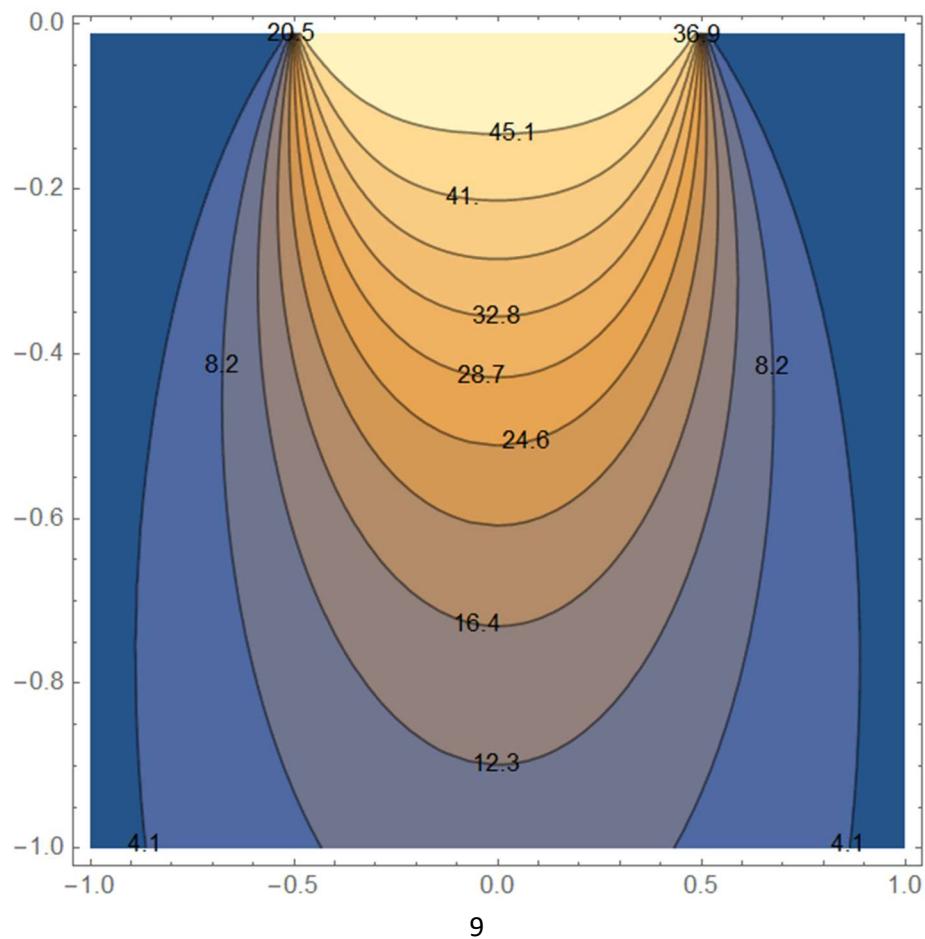


Рисунок 1.2 Распределение  $\sigma_{3,3}$  в полупространстве при фиксированном  $x_2$

Решение для  $\sigma_{1,1}$ :

$$\sigma_{1,1}^I(x, y, z, a, b) = \int_{-a}^a \sigma_{1,1}(x - p, y, z, a, b) dp,$$

$$\begin{aligned} \sigma_{1,1}^{II}(x, y, z, a, b) &= \int_{-b}^b \sigma_{1,1}^I(x, y - k, z, a, b) dk = \frac{p_0}{4\pi(\lambda+2\mu)} * (x_3(\lambda + \\ &\mu)\left(\frac{x_1x_2}{(x_1^2+x_3^2)\sqrt{x_1^2+x_2^2+x_3^2}} - \frac{(-a+x_1)x_2}{((-a+x_1)^2+x_3^2)\sqrt{(-a+x_1)^2+x_2^2+x_3^2}} - \right. \\ &\left. \frac{x_1(-b+x_2)}{(x_1^2+x_3^2)\sqrt{x_1^2+(-b+x_2)^2+x_3^2}} + \frac{(-a+x_1)(-b+x_2)}{((-a+x_1)^2+x_3^2)\sqrt{(-a+x_1)^2+(-b+x_2)^2+x_3^2}}\right) + \\ &\lambda(-\arctg[\frac{x_1x_2}{x_3\sqrt{x_1^2+x_2^2+x_3^2}}] + \arctg[\frac{(-a+x_1)x_2}{x_3\sqrt{(-a+x_1)^2+x_2^2+x_3^2}}] + \\ &\arctg[\frac{x_1(-b+x_2)}{x_3\sqrt{x_1^2+(-b+x_2)^2+x_3^2}}] - \arctg[\frac{(-a+x_1)(-b+x_2)}{x_3\sqrt{(-a+x_1)^2+(-b+x_2)^2+x_3^2}}])) . \end{aligned}$$

Построим графики напряжений при фиксированных  $x_3$  и  $x_2$ :

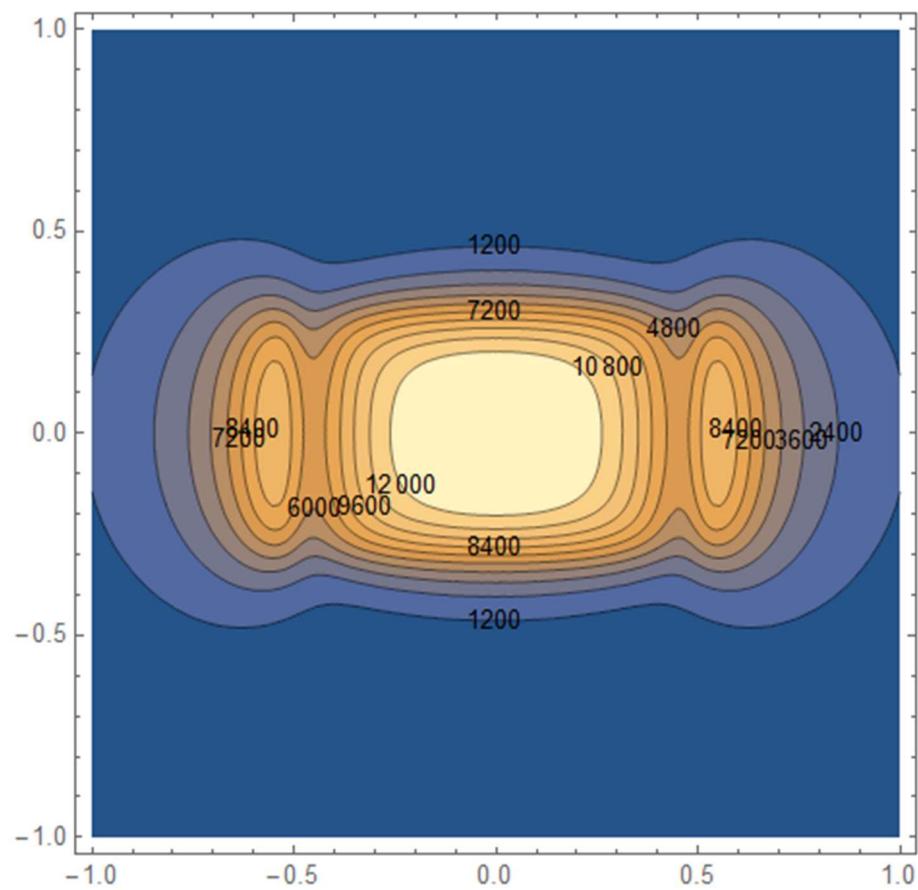


Рисунок 1.3 Распределение  $\sigma_{1,1}$  в полупространстве при фиксированном  $x_3$

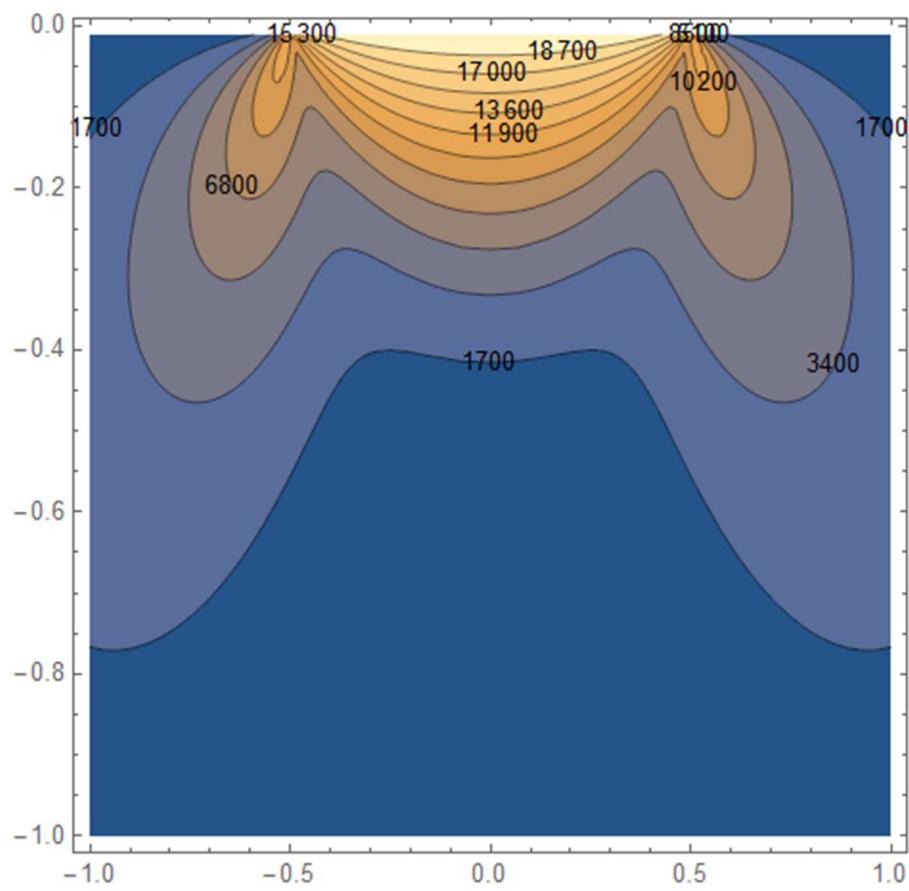


Рисунок 1.4 Распределение  $\sigma_{1,1}$  в полупространстве при фиксированном  $x_2$

Решение для  $\sigma_{1,2}$ :

$$\sigma_{1,2}^I(x, y, z, a, b) = \int_{-a}^a \sigma_{1,2}(x - p, y, z, a, b) dp,$$

$$\begin{aligned} \sigma_{1,2}^{II}(x, y, z, a, b) &= \int_{-b}^b \sigma_{1,2}^I(x, y - k, z, a, b) dk = \frac{p_0 x_3 (\lambda + \mu)}{4\pi(\lambda + 2\mu)} * \left( -\frac{1}{\sqrt{x_1^2 + x_2^2 + x_3^2}} + \right. \\ &\quad \left. \frac{1}{\sqrt{(-a+x_1)^2 + x_2^2 + x_3^2}} + \frac{1}{\sqrt{x_1^2 + (-b+x_2)^2 + x_3^2}} - \frac{1}{\sqrt{(-a+x_1)^2 + (-b+x_2)^2 + x_3^2}} \right). \end{aligned}$$

Построим графики напряжений при фиксированных  $x_3$  и  $x_2$ :

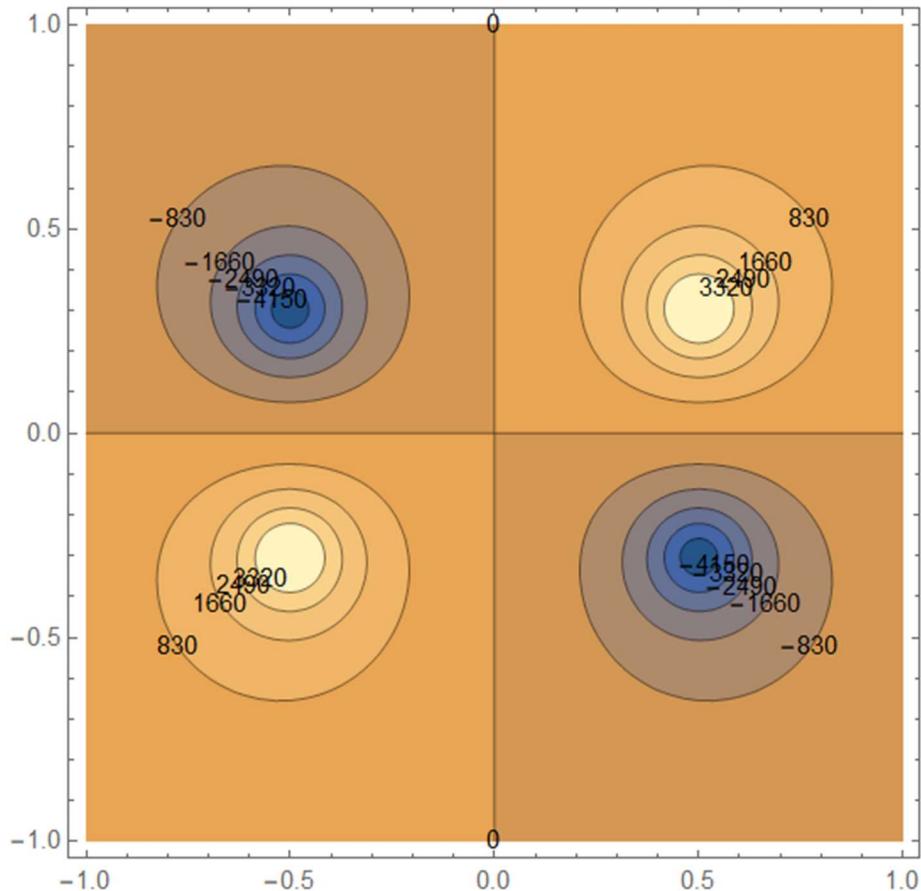


Рисунок 1.5 Распределение  $\sigma_{1,2}$  в полупространстве при фиксированном  $x_3$

Решение для  $\sigma_{1,3}$ :

$$\sigma_{1,3}^I(x, y, z, a, b) = \int_{-a}^a \sigma_{1,3}(x - p, y, z, a, b) dp,$$

$$\sigma_{1,3}^{II}(x, y, z, a, b) = \int_{-b}^b \sigma_{1,3}^I(x, y - k, z, a, b) dk = \frac{p_0}{4\pi(\lambda+2\mu)} ((\lambda + \mu) * \\ \left( \frac{x_2 x_3^2}{(x_1^2 + x_3^2) \sqrt{x_1^2 + x_2^2 + x_3^2}} - \frac{x_2 x_3^2}{((-a + x_1)^2 + x_3^2) \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}} - \right. \\ \left. \frac{(-b + x_2) x_3^2}{(x_1^2 + x_3^2) \sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}} + \frac{(-b + x_2) x_3^2}{((-a + x_1)^2 + x_3^2) \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}} \right) + \mu (\log[x_2 + \\ \sqrt{x_1^2 + x_2^2 + x_3^2}] - \log[x_2 + \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}] - \log[-b + x_2 + \\ \sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}] + \log[-b + x_2 + \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}])).$$

Построим графики напряжений при фиксированных  $x_3$  и  $x_2$ :

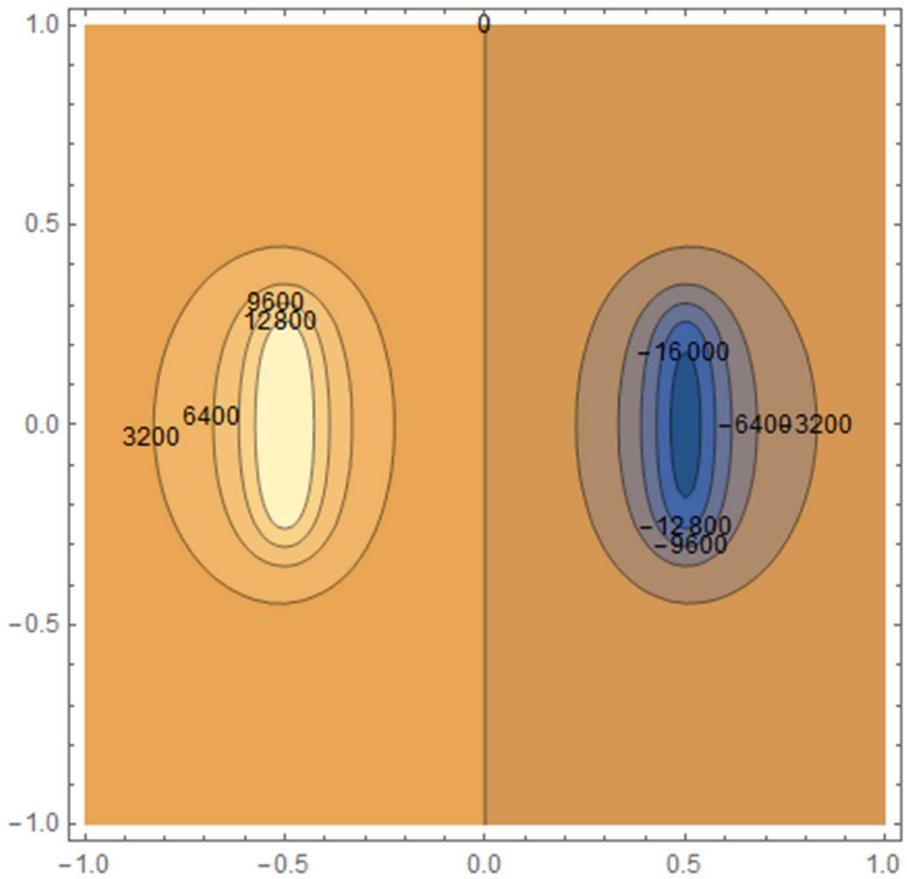


Рисунок 1.6 Распределение  $\sigma_{1,3}$  в полупространстве при фиксированном  $x_3$

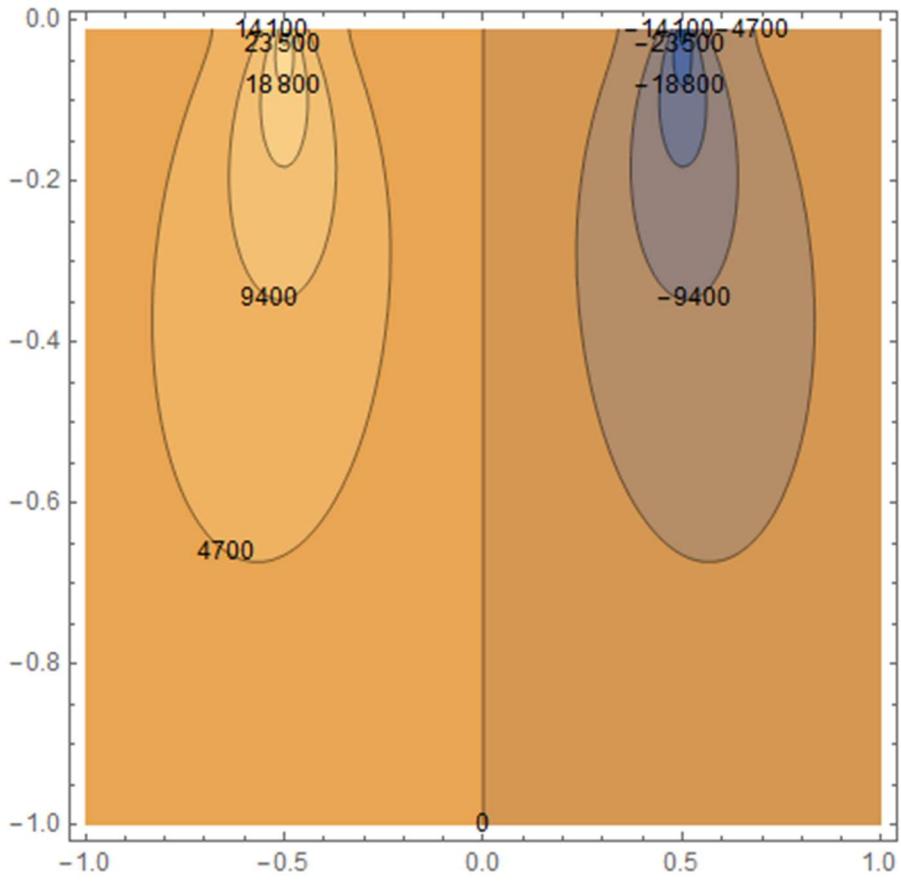


Рисунок 1.7 Распределение  $\sigma_{1,3}$  в полупространстве при фиксированном  $x_2$

Решение для  $\sigma_{2,2}$ :

$$\sigma_{2,2}^I(x, y, z, a, b) = \int_{-a}^a \sigma_{2,2}(x - p, y, z, a, b) dp,$$

$$\begin{aligned} \sigma_{2,2}^{II}(x, y, z, a, b) &= \int_{-b}^b \sigma_{2,2}^I(x, y - k, z, a, b) dk = \frac{p_0}{4\pi(\lambda+2\mu)} * (x_3(\lambda + \\ &\mu)\left(\frac{x_1x_2}{(x_2^2+x_3^2)\sqrt{x_1^2+x_2^2+x_3^2}} - \frac{(-a+x_1)x_2}{(x_2^2+x_3^2)\sqrt{(-a+x_1)^2+x_2^2+x_3^2}} - \right. \\ &\left. \frac{x_1(-b+x_2)}{((-b+x_2)^2+x_3^2)\sqrt{x_1^2+(-b+x_2)^2+x_3^2}} + \frac{(-a+x_1)(-b+x_2)}{((-b+x_2)^2+x_3^2)\sqrt{(-a+x_1)^2+(-b+x_2)^2+x_3^2}}\right) + \\ &\lambda(-\arctg[\frac{x_1x_2}{x_3\sqrt{x_1^2+x_2^2+x_3^2}}] + \arctg[\frac{(-a+x_1)x_2}{x_3\sqrt{(-a+x_1)^2+x_2^2+x_3^2}}] + \\ &\arctg[\frac{x_1(-b+x_2)}{x_3\sqrt{x_1^2+(-b+x_2)^2+x_3^2}}] - \arctg[\frac{(-a+x_1)(-b+x_2)}{x_3\sqrt{(-a+x_1)^2+(-b+x_2)^2+x_3^2}}])). \end{aligned}$$

Построим графики напряжений при фиксированных  $x_3$  и  $x_2$ :

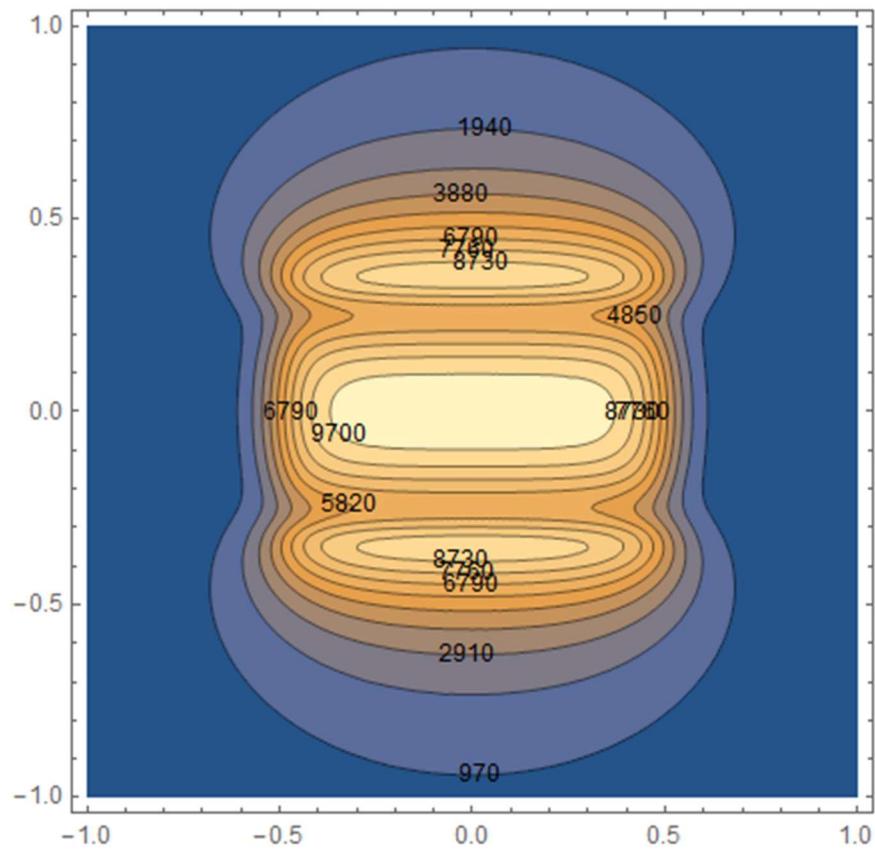


Рисунок 1.8 Распределение  $\sigma_{2,2}$  в полупространстве при фиксированном  $x_3$

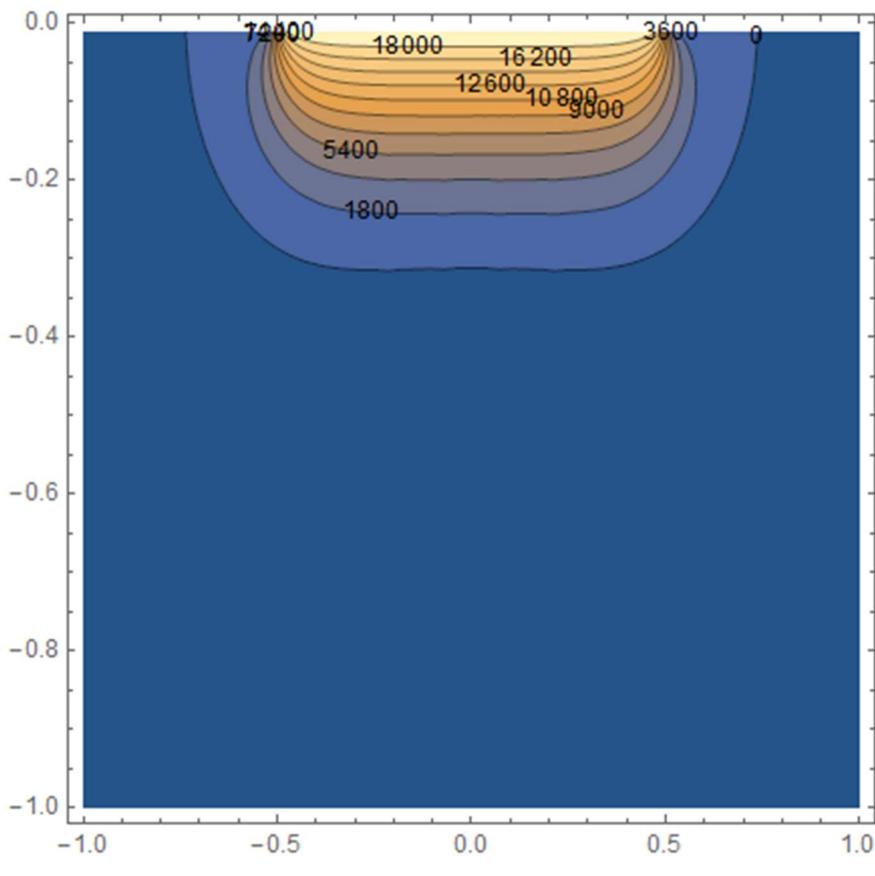


Рисунок 1.9 Распределение  $\sigma_{2,2}$  в полупространстве при фиксированном  $x_2$

Решение для  $\sigma_{2,3}$ :

$$\begin{aligned} \sigma_{2,3}^I(x, y, z, a, b) &= \int_{-a}^a \sigma_{2,3}(x - p, y, z, a, b) dp, \\ \sigma_{2,3}^{II}(x, y, z, a, b) &= \int_{-b}^b \sigma_{2,3}^I(x, y - k, z, a, b) dk = \frac{p_0}{4\pi(\lambda+2\mu)} * ((\lambda + \\ \mu) \left( \frac{x_1 x_3^2}{(x_2^2+x_3^2)\sqrt{x_1^2+x_2^2+x_3^2}} - \frac{(-a+x_1)x_3^2}{(x_2^2+x_3^2)\sqrt{(-a+x_1)^2+x_2^2+x_3^2}} - \right. \\ \left. \frac{x_1 x_3^2}{((-b+x_2)^2+x_3^2)\sqrt{x_1^2+(-b+x_2)^2+x_3^2}} + \frac{(-a+x_1)x_3^2}{((-b+x_2)^2+x_3^2)\sqrt{(-a+x_1)^2+(-b+x_2)^2+x_3^2}} \right) + \\ \mu (\log[x_1 + \sqrt{x_1^2 + x_2^2 + x_3^2}] - \log[-a + x_1 + \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}] - \\ \log[x_1 + \sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}] + \log[-a + x_1 + \\ \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}]). \end{aligned}$$

Построим графики напряжений при фиксированных  $x_3$  и  $x_2$ :

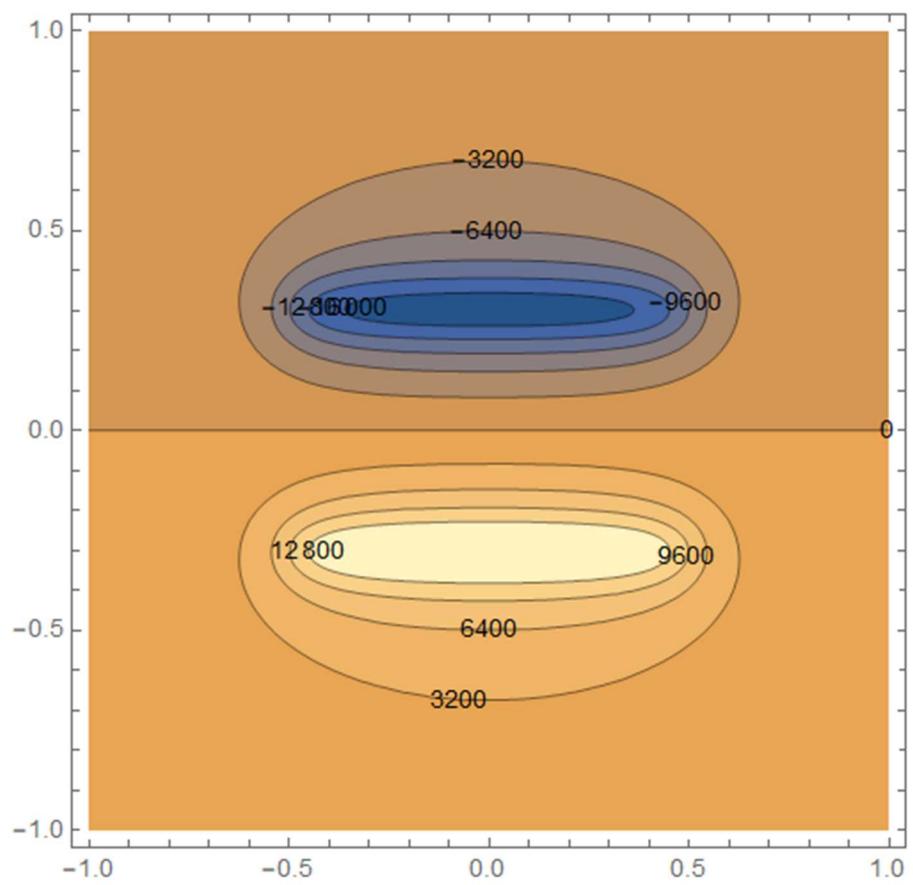


Рисунок 1.10 Распределение  $\sigma_{2,3}$  в полупространстве при фиксированном  $x_3$

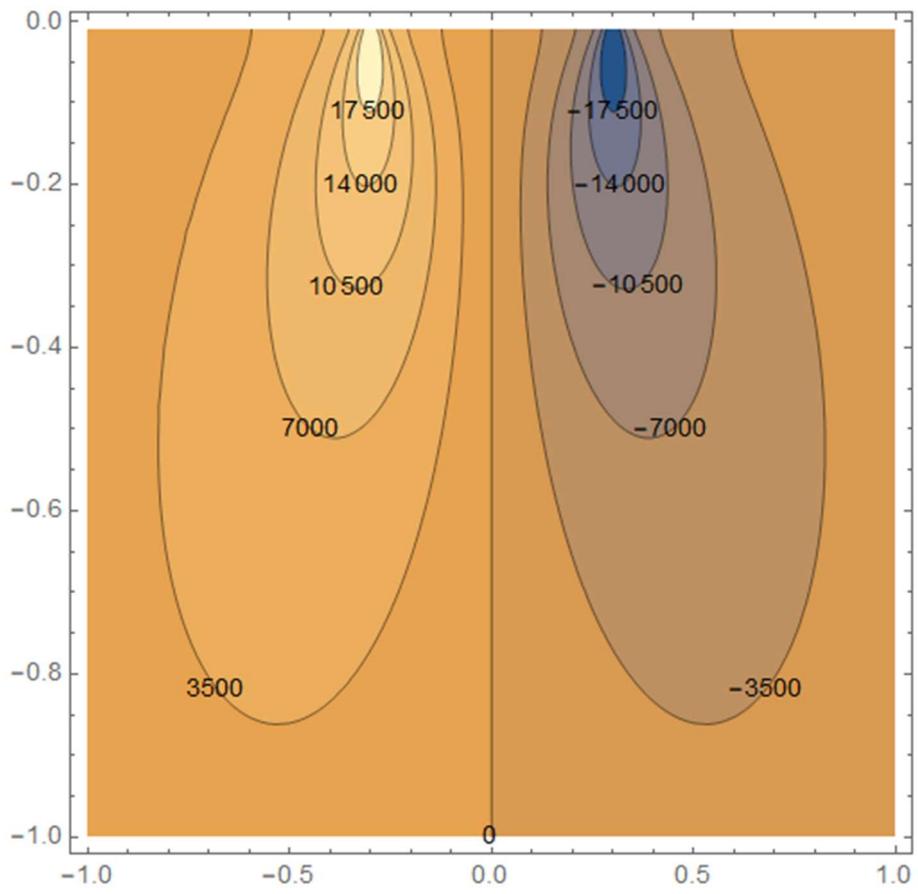


Рисунок 1.11 Распределение  $\sigma_{2,3}$  в полупространстве при фиксированном  $x_2$

Решение для  $U_1$ :

$$U_1^I(x, y, z, a, b) = \int_{-a}^a U_1(x - p, y, z, a, b) dp,$$

$$U_1^{II}(x, y, z, a, b) = \int_{-b}^b U_1^I(x, y - k, z, a, b) dk = -\frac{p_0 x_3 (\lambda + \mu)}{8\pi\mu(\lambda + 2\mu)} * (\log[x_2 +$$

$$\sqrt{x_1^2 + x_2^2 + x_3^2}] - \log[x_2 + \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}] - \log[-b + x_2 +$$

$$\sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}] + \log[-b + x_2 + \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}]).$$

Построим графики перемещений при фиксированных  $x_3$  и  $x_2$ :

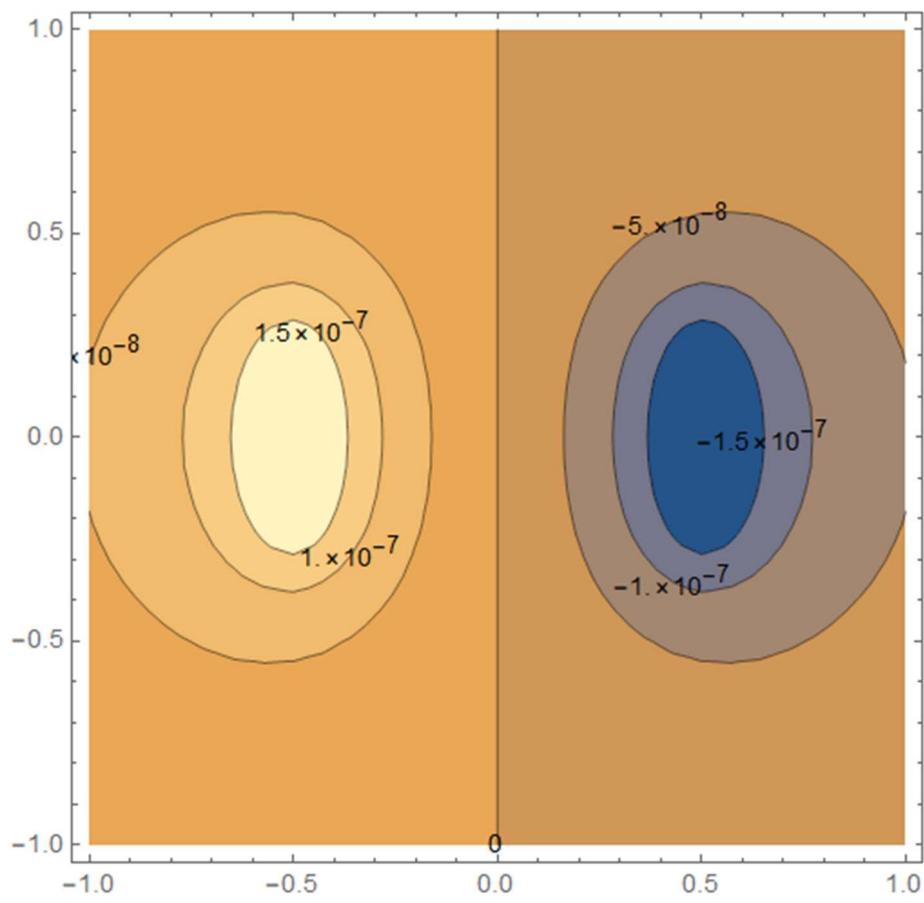


Рисунок 1.12 Распределение  $U_1$  в полупространстве при фиксированном  $x_3$

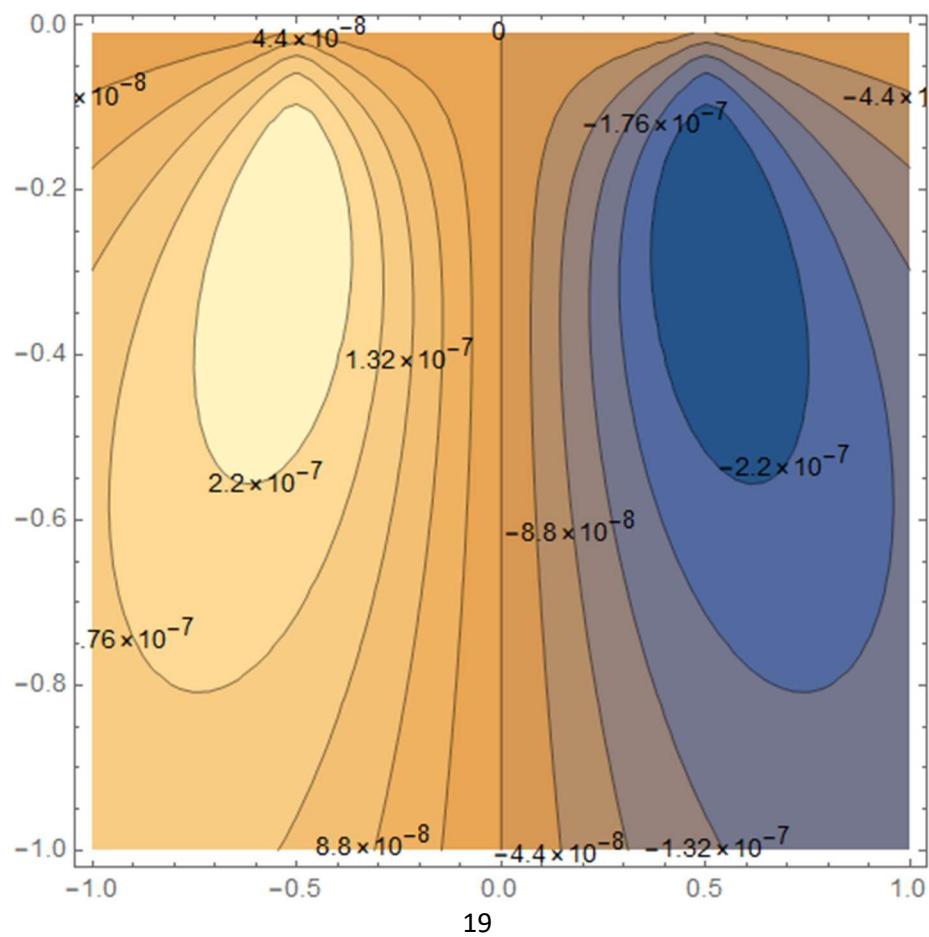


Рисунок 1.13 Распределение  $U_1$  в полупространстве при фиксированном  $x_2$

Решение для  $U_2$ :

$$U_2^I(x, y, z, a, b) = \int_{-a}^a U_2(x - p, y, z, a, b) dp,$$

$$U_2^{II}(x, y, z, a, b) = \int_{-b}^b U_2^I(x, y - k, z, a, b) dk = -\frac{p_0 x_3(\lambda + \mu)}{8\pi\mu(\lambda + 2\mu)} * (\log[x_1 +$$

$$\sqrt{x_1^2 + x_2^2 + x_3^2}] - \log[-a + x_1 + \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}] - \log[x_1 +$$

$$\sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}] + \log[-a + x_1 + \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}]).$$

Построим графики перемещений при фиксированных  $x_3$  и  $x_1$ :

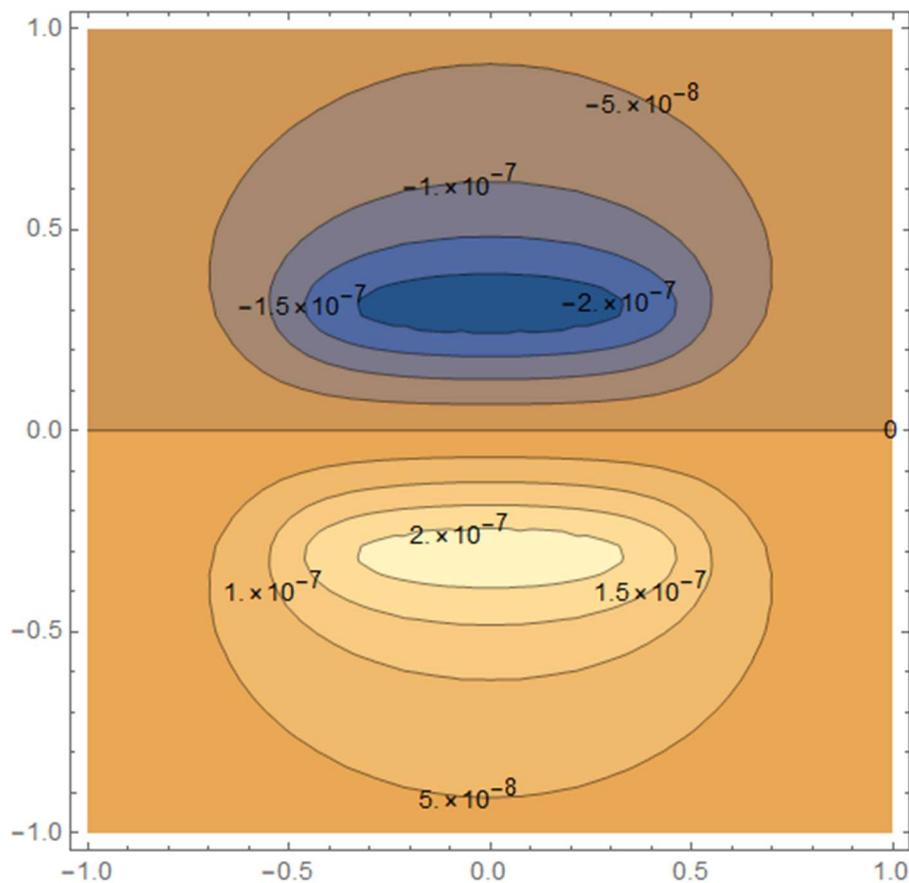


Рисунок 1.14 Распределение  $U_2$  в полупространстве при фиксированном  $x_3$

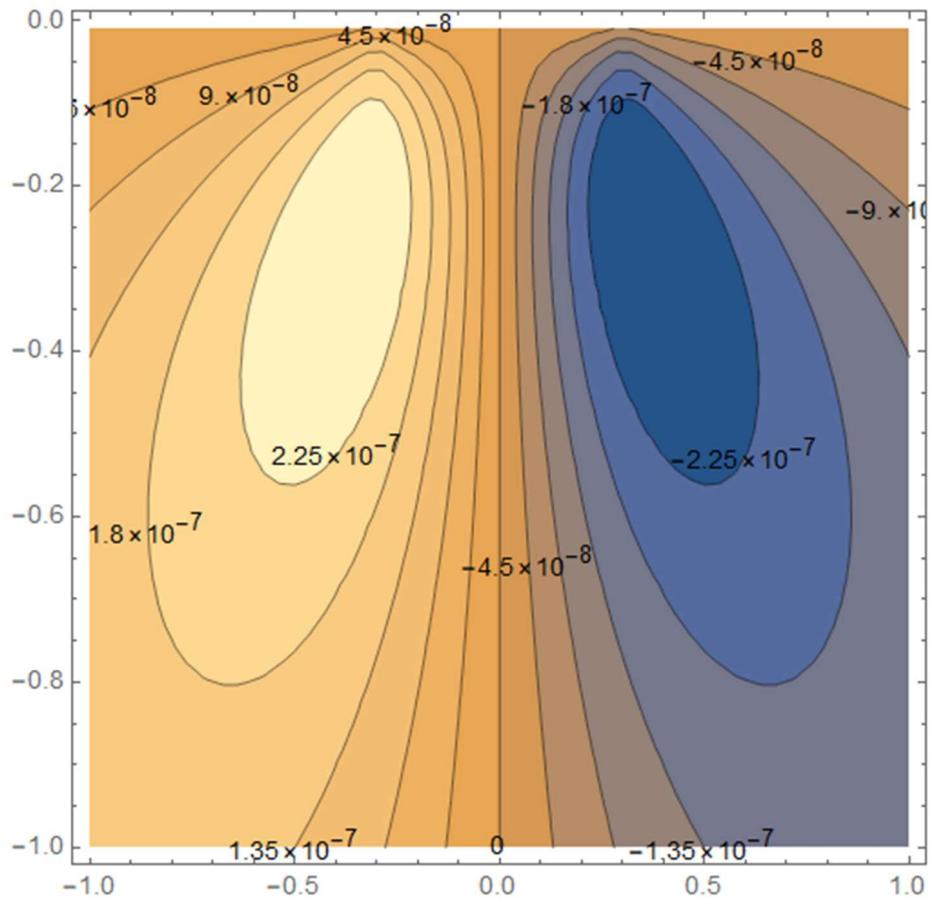


Рисунок 1.15 Распределение  $U_2$  в полупространстве при фиксированном  $x_1$

Решение для  $U_3$ :

$$U_3^I(x, y, z, a, b) = \int_{-a}^a U_3(x - p, y, z, a, b) dp,$$

$$U_3^{II}(x, y, z, a, b) = \int_{-b}^b U_3^I(x, y - k, z, a, b) dk = \frac{p_0}{8\pi (\lambda + 2\mu)} *$$

$$(2x_3\mu(-\operatorname{arctg}\left[\frac{x_1x_2}{x_3\sqrt{x_1^2+x_2^2+x_3^2}}\right] + \operatorname{arctg}\left[\frac{(-a+x_1)x_2}{x_3\sqrt{(-a+x_1)^2+x_2^2+x_3^2}}\right] +$$

$$\operatorname{arctg}\left[\frac{x_1(-b+x_2)}{x_3\sqrt{x_1^2+(-b+x_2)^2+x_3^2}}\right] - \operatorname{arctg}\left[\frac{(-a+x_1)(-b+x_2)}{x_3\sqrt{(-a+x_1)^2+(-b+x_2)^2+x_3^2}}\right]) + (\lambda +$$

$$3\mu)(x_2(\log[x_1 + \sqrt{x_1^2 + x_2^2 + x_3^2}] - \log[-a + x_1 +$$

$$\sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}) + x_1(\log[x_2 + \sqrt{x_1^2 + x_2^2 + x_3^2}] - \log[-b + x_2 +$$

$$\begin{aligned}
& \sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}]]) + (-b + x_2)(-\log[x_1 + \sqrt{x_1^2 + (-b + x_2)^2 + x_3^2}]] + \\
& \log[-a + x_1 + \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}]) + (-a + x_1)(-\text{Log}[x_2 + \\
& \sqrt{(-a + x_1)^2 + x_2^2 + x_3^2}] + \log[-b + x_2 + \\
& \sqrt{(-a + x_1)^2 + (-b + x_2)^2 + x_3^2}]))).
\end{aligned}$$

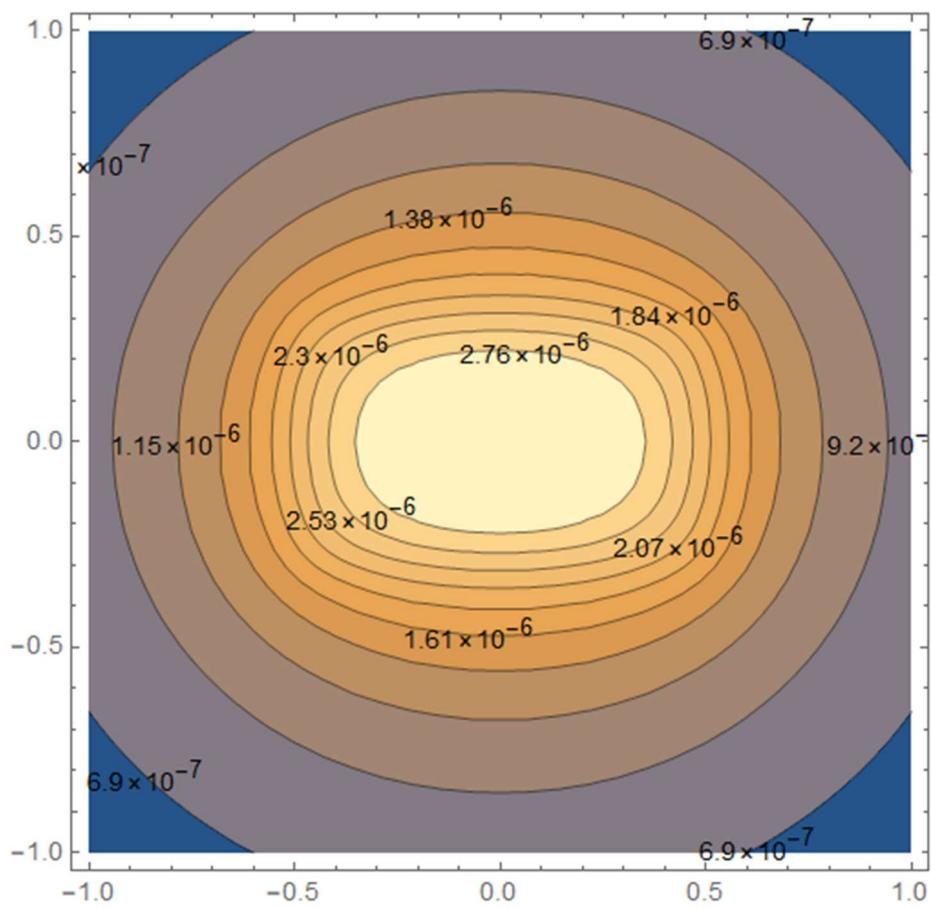


Рисунок 1.16 Распределение  $U_3$  в полупространстве при фиксированном  $x_3$

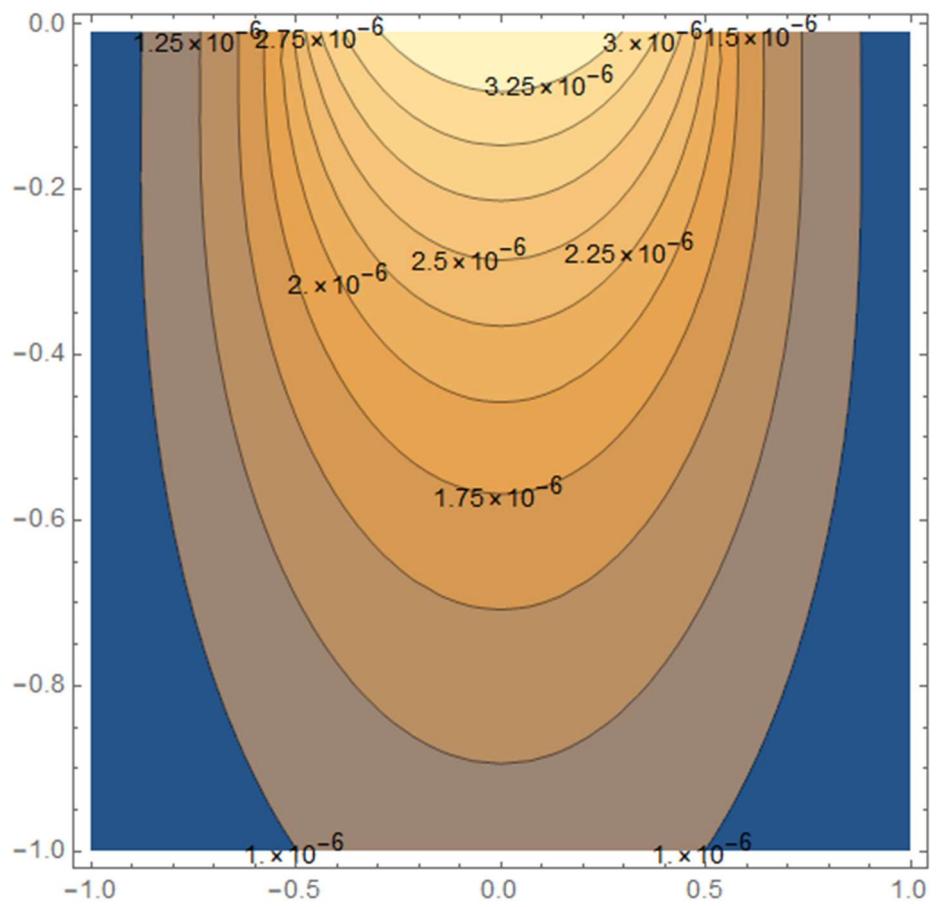


Рисунок 1.17 Распределение  $U_3$  в полупространстве при фиксированном  $x_2$

## ГЛАВА 2

# ПОСЛЕДОВАТЕЛЬНОЕ РЕШЕНИЕ ЗАДАЧИ О ВДАВЛИВАНИИ ПЛОСКОГО ЭЛЛИПТИЧЕСКОГО ШТАМПА В ПОЛУПРОСТРАНСТВО

## 2.1 Постановка задачи

Теперь рассмотрим случай нагружения полупространства по эллиптической области, задающееся следующим образом (рис 2.1):

$$\bar{U}(x, y, a, b, U_0) = U_0, \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 \leq 1$$

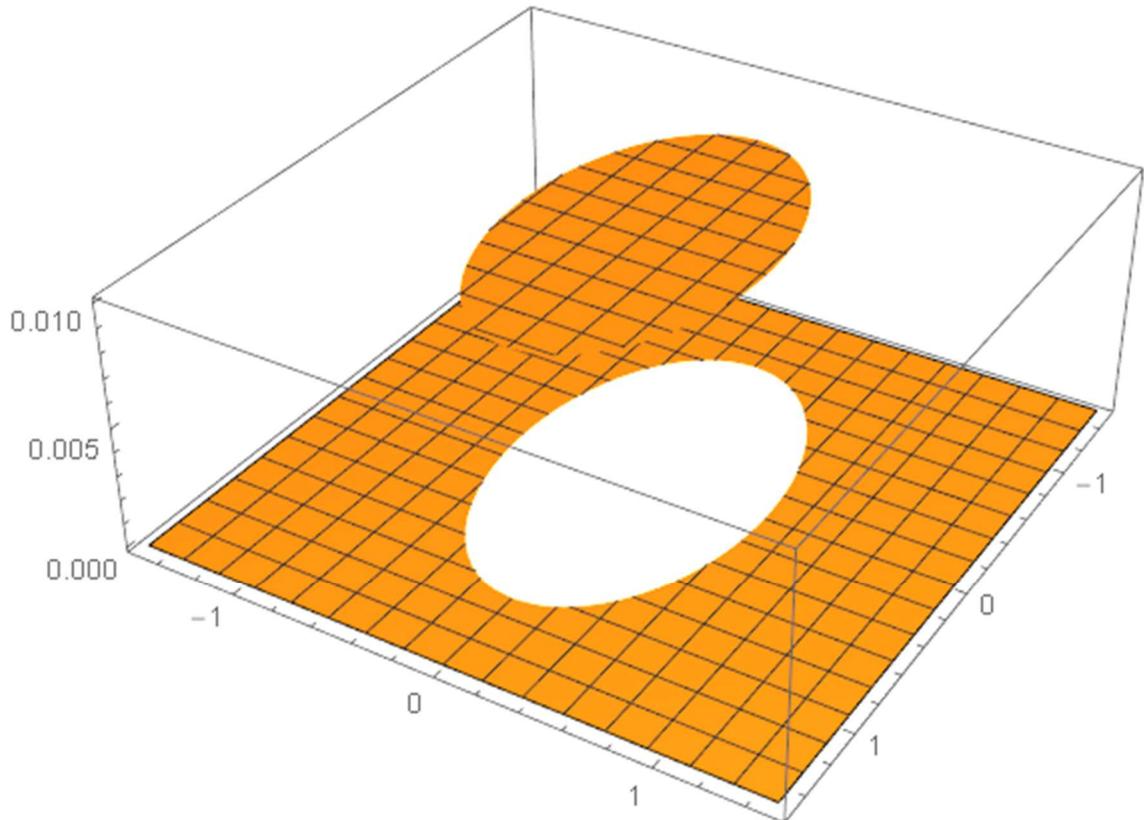


Рисунок 2.1 Функция распределения перемещений

Решение данной граничной задачи сводится к определению фиктивных усилий  $\bar{s}^*$ , приложив которые на границе полупространства, удовлетворим граничные условия. Решим данную задачу используя с помощью Метода

Границных Элементов. Для этого дискретизируем полученную функцию перемещения на граничные элементы.

## 2.2 Дискретизация функции поверхностного распределения

Как и при любой дискретизации, очевидно, что общее количество элементов будет зависеть от размера элемента, учитывая, что выбрана равномерная сетка, с одинаковыми размерами каждого граничного элемента, это количество будет выражаться следующим образом.

$$n = \frac{2s}{h} + 1,$$

где  $n$  – количество элементов вдоль одной оси,  $s$  – размер расчётной области,  $h$  – размер граничного элемента,  $N = n^2$  – общее количество элементов.

При решении данной задачи, для того чтобы в будущем иметь возможность проводить сравнительный анализ решений, рассмотрим 4 различных случая разбиения функции распределения перемещений:  $n = 20$ ,  $n = 40$ ,  $n = 60$ ,  $n = 80$ . Определение массива элементов вдоль одной оси происходит динамически: сначала задаётся размер граничного элемента, после чего определяется их количество и в конце производится разбиение расчётной области с шагом, равным размеру граничного элемента. Таким образом вся область разбита на сетку. Так же приведём графики дискретизации функции поверхностного распределения для всех случаев.

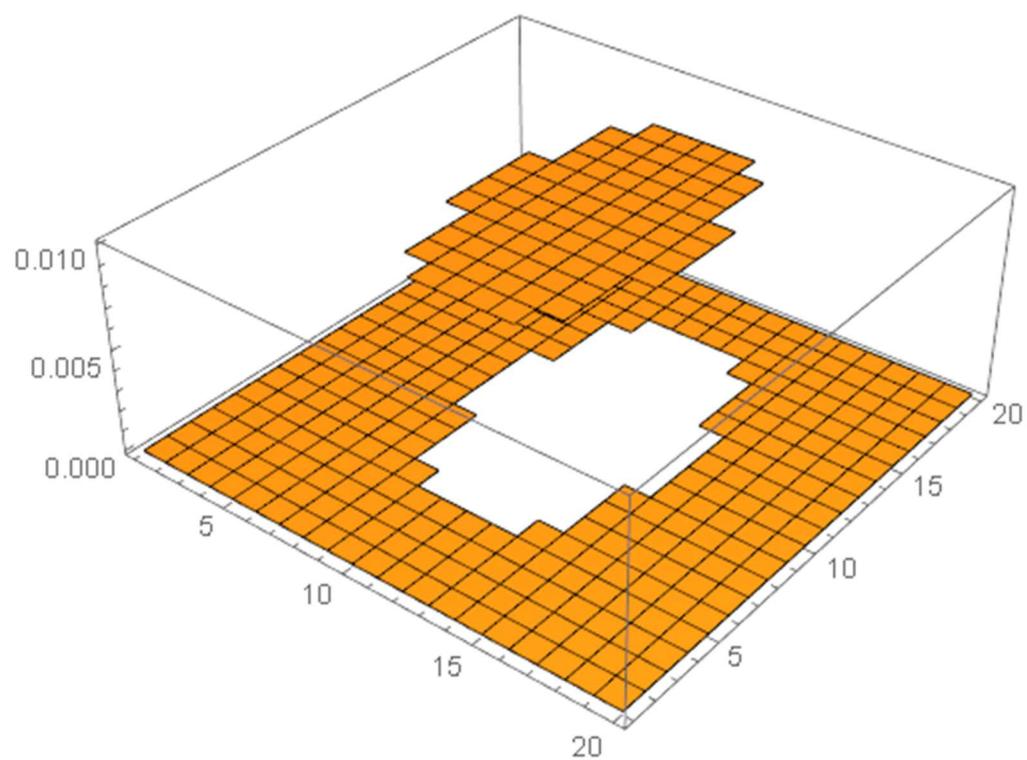


Рисунок 2.2 Дискретизация функции поверхностного распределения при  
 $n = 20$

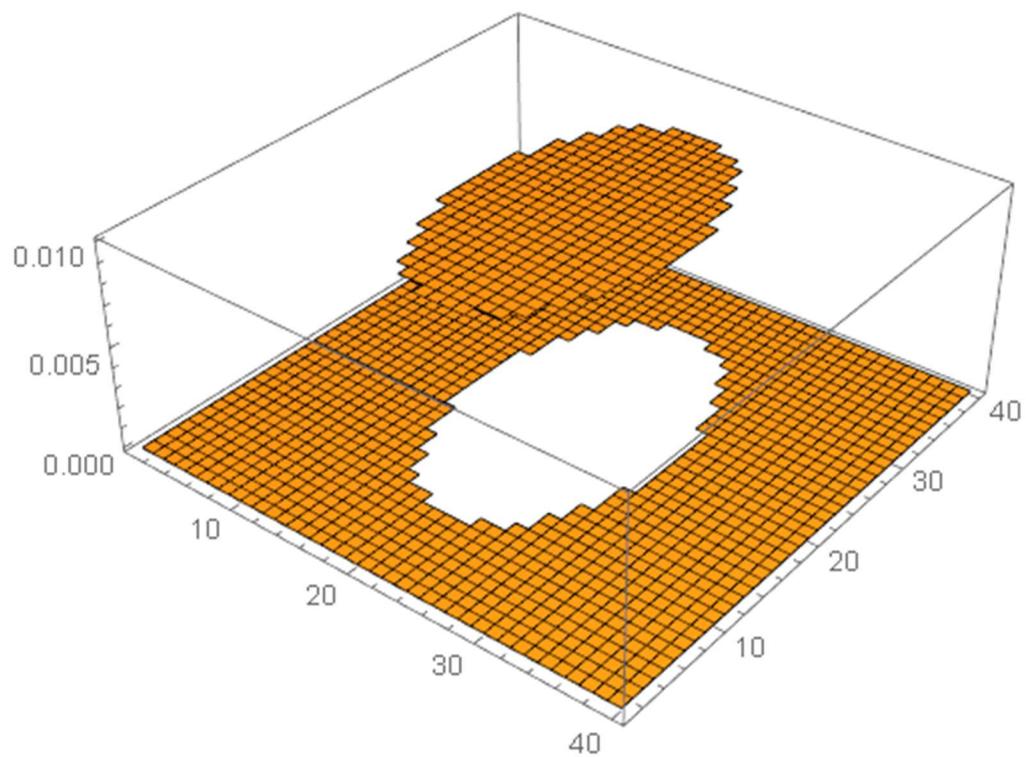


Рисунок 2.3 Дискретизация функции поверхностного распределения при  
 $n = 40$

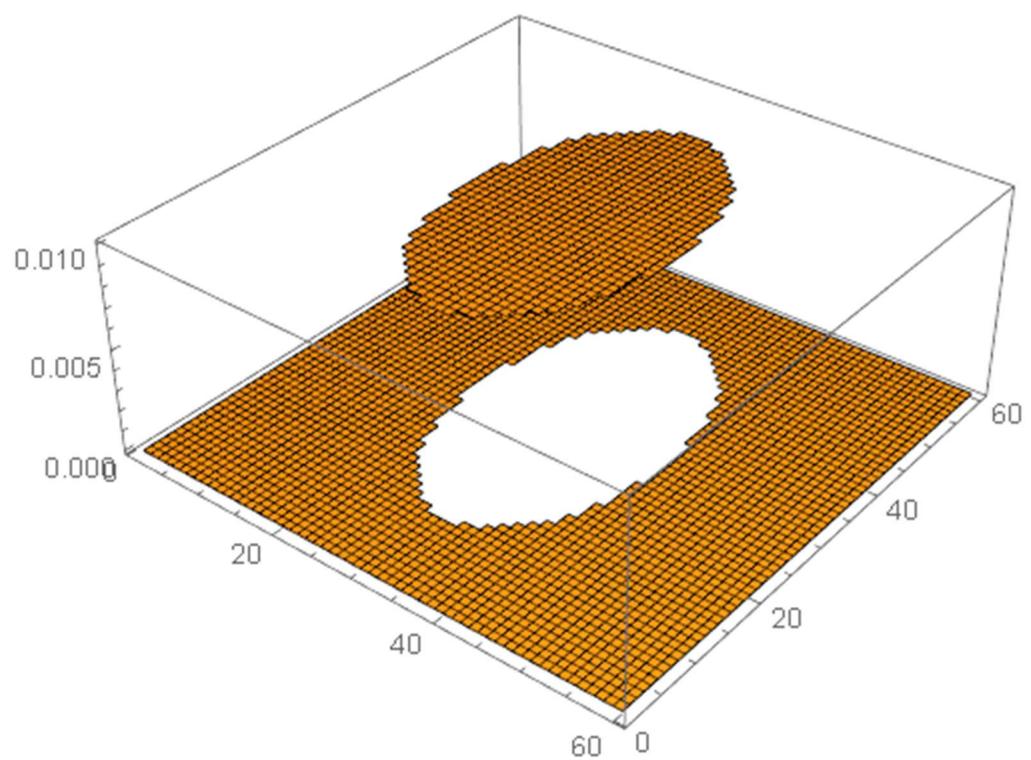


Рисунок 2.4 Дискретизация функции поверхностного распределения при  
 $n = 60$

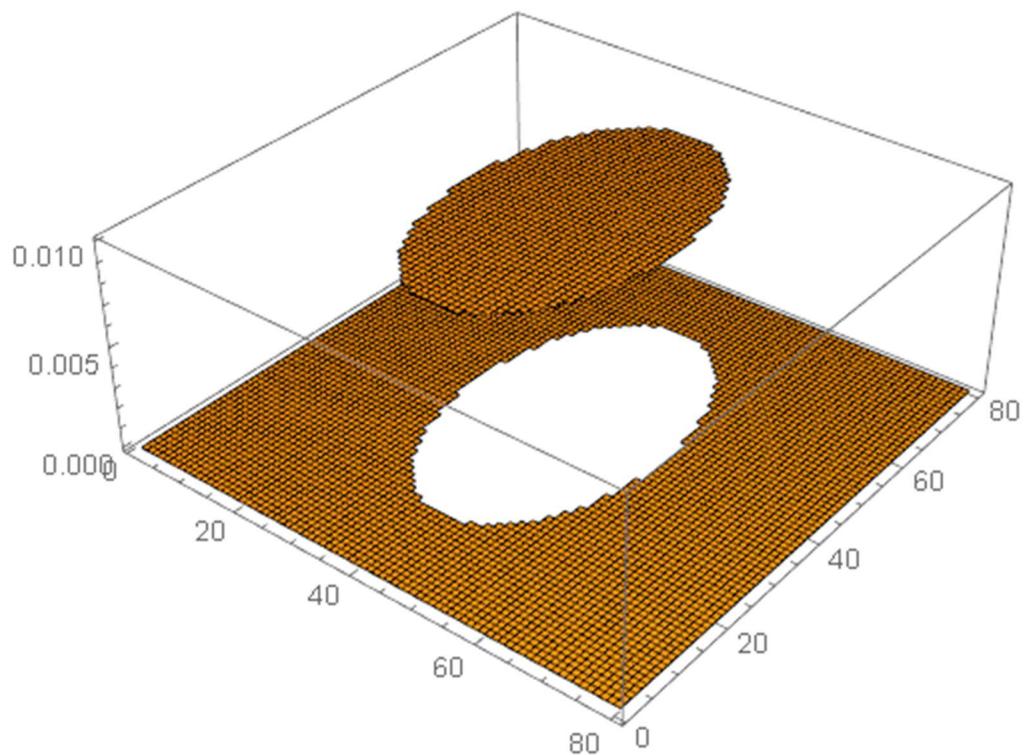


Рисунок 2.3 Дискретизация функции поверхностного распределения при  
 $n = 80$

## 2.3 Получение распределения напряжений

Получив значение функции перемещений, внутри эллиптической области, необходимо получить значение давлений в точке, как суперпозицию перемещений от каждого граничного элемента. Значение будем искать в виде суммы всех действующих на точку граничных элементов.

$$p(x_i, y_j) = \sum_{i=1}^k p_j^* * U_3^{II}(x - x_i, y - y_j, z, a, b),$$

где  $p_j^*$  — искомые значения,  $k$  — количество элементов внутри эллипса.

Исходя из этого построим систему линейных алгебраических уравнений, состоящую из  $k$  уравнений и имеющую  $k$  переменных, подставляя в функцию  $p(x_i, y_j)$  значения координат узлов граничных элементов.

Решим эту систему матричным методом, для этого составим уравнение вида

$$Ax = B,$$

где  $A$  — матрица, составленная из функций  $p(x_i, y_j)$ ,  $x$  — столбец, состоящий из искомых значений усилий и  $B$  — столбец, полученный уплощением матрицы дискретизации функции перемещений. Отобразим на рисунках картины распределения фиктивных усилий для двух вариантов дискретизации полупространства:

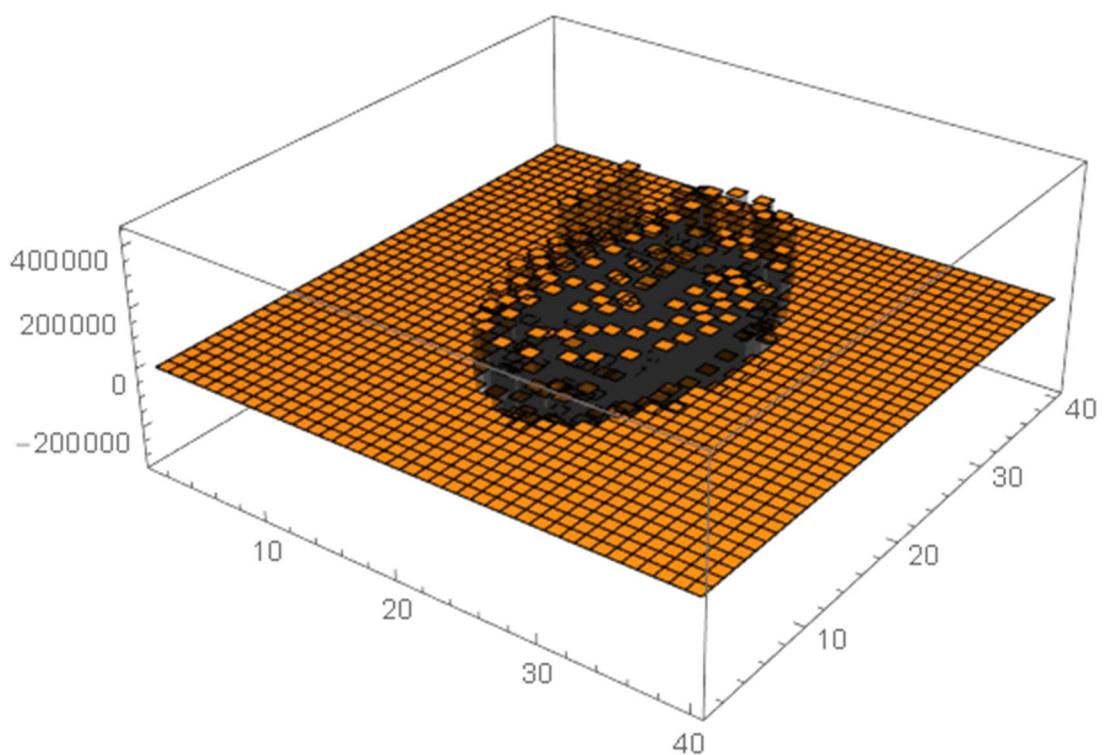


Рисунок 2.4 Распределение фиктивных усилий при  $n = 40$

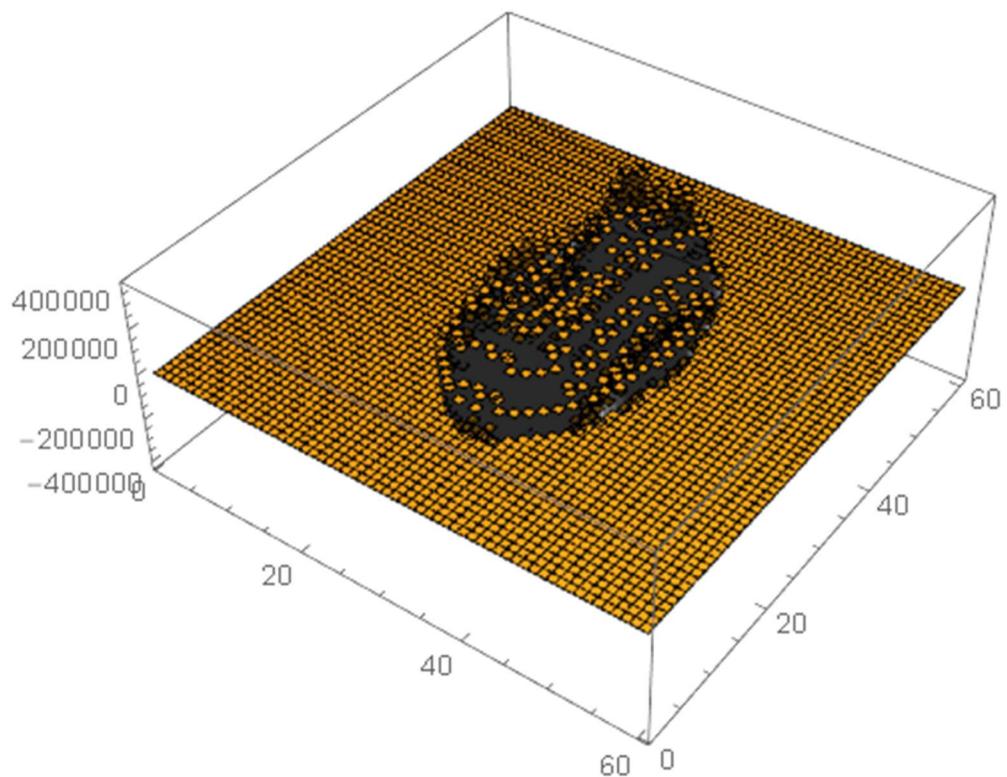


Рисунок 2.4 Распределение фиктивных усилий при  $n = 60$

Далее построим распределение перемещений на поверхности полупространства, приложив к нему фиктивные усилия

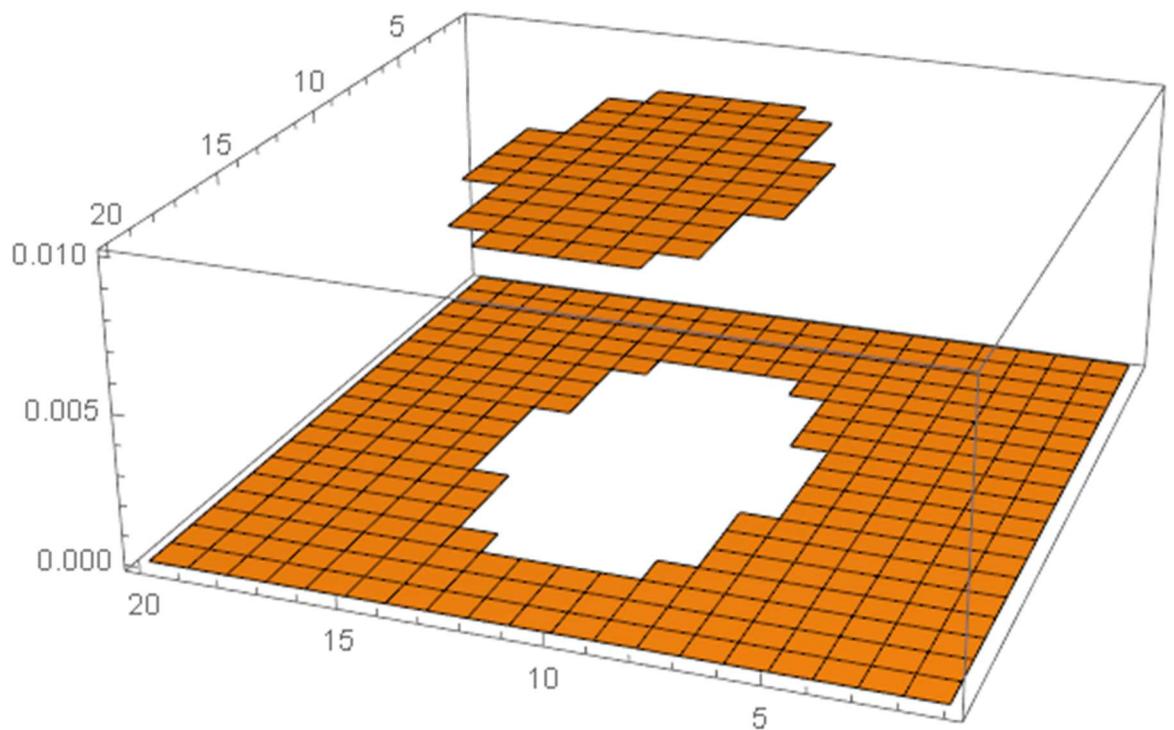


Рисунок 2.5 Решение на поверхности полупространства, полученное с использованием фиктивных усилий.

## ГЛАВА 3

# РАСПАРАЛЛЕЛИВАНИЕ ПОСЛЕДОВАТЕЛЬНОГО РЕШЕНИЯ ЗАДАЧИ О ВДАВЛИВАНИИ ПЛОСКОГО ЭЛЛИПТИЧЕСКОГО ШТАМПА В ПОЛУПРОСТРАНСТВО

## 3.1 Некоторые сведения о технологии NVIDIA CUDA

Благодаря оптимизации, современное программное обеспечение способно показать значительные результаты в области сокращения временных затрат на выполнение математических операций и вычислений, однако даже самые мощные процессоры постепенно приближаются к своему пределу, связанному с ограниченностью размера техпроцесса. Единственным решением сложных и громоздких задач является объединение таких процессоров в кластеры для их одновременной работы. Однако для обычных пользователей существуют ограничения, связанные с невозможностью использования таких кластеров или мощных самих по себе центральных процессоров, которые используются в качестве основных вычислительных единиц.

Одним из немногих оптимальных решений проблемы ускорения вычислений становятся графические процессоры, в частности те, которые поддерживают работу с технологией NVIDIA CUDA. Эта технология предоставляет возможность взаимодействия с ядрами и памятью GPU (графического ускорителя), с целью ускорения вычислений.

Первоначальная версия CUDA Software Development Kit была представлена в феврале 2007 года. Основой API (Интерфейс Программирования Приложений) CUDA стал язык Си, с необходимыми расширениями. В состав CUDA входит собственный Си-компилятор командной строки nvcc, созданный компанией Nvidia на основе компилятора Open64. Данная технология применима не только для производительных графических вычислений: помимо всего прочего, CUDA используют в различных областях: моделирование динамики жидкостей, астрофизика, электромагнитные взаимодействия, химия, вычислительная биология и т.д. Отдельно стоит упомянуть, что работу с CUDA поддерживают такие операционные системы как Linux, Windows и Mac OS, что делает её кроссплатформенной технологией.

CUDA как платформа впервые появилась на рынке с выходом чипа восьмого поколения G80, после чего стала появляться во всех последующих сериях графических чипах, используемых в семействе GeForce, Tesla и Quadro. Графических процессор организует аппаратную многопоточность, позволяя использовать больше ресурсов графического ускорителя. Так стало возможно делегировать функции физического ускорителя на графический, примером такой реализации стал PhysX.

Единицами вычисления графический ускоритель являются сеть, блок и нить. Наивысшая единица вычислительной архитектуры — сеть, она состоит из блоков, а те, в свою очередь, из нитей. Нить — самая маленькая вычислительная единица. Следовательно, используя графический процессор, можно запустить  $N_{\text{блоков}} \times N_{\text{нитей}}$  в одном блоке параллельных вычислений. Сеть обычного GPU может состоять из 65535 блоков, в каждом из которых можно запустить 1024 нити. Произведя подсчёт, получим возможность запустить 6710784 параллельно работающих единиц. Для дальнейшей работы с вычислительными единицами существует возможность получить следующую информацию о конфигурации нитей, блоков и сети:

- $\text{gridDim.x}, \text{gridDim.y}, \text{gridDim.z}$  — размерность сети блоков вдоль каждого из измерений соответственно;
- $\text{blockDim.x}, \text{blockDim.y}, \text{blockDim.z}$  — размерность блока вдоль каждого из измерений соответственно;
- $\text{blockIdx.x}, \text{blockIdx.y}, \text{blockIdx.z}$  — индекс блока внутри сети вдоль каждого из измерений соответственно;
- $\text{threadIdx.x}, \text{threadIdx.y}, \text{threadIdx.z}$  — индекс нити внутри блока вдоль каждого из измерений соответственно.

# CUDA Grid

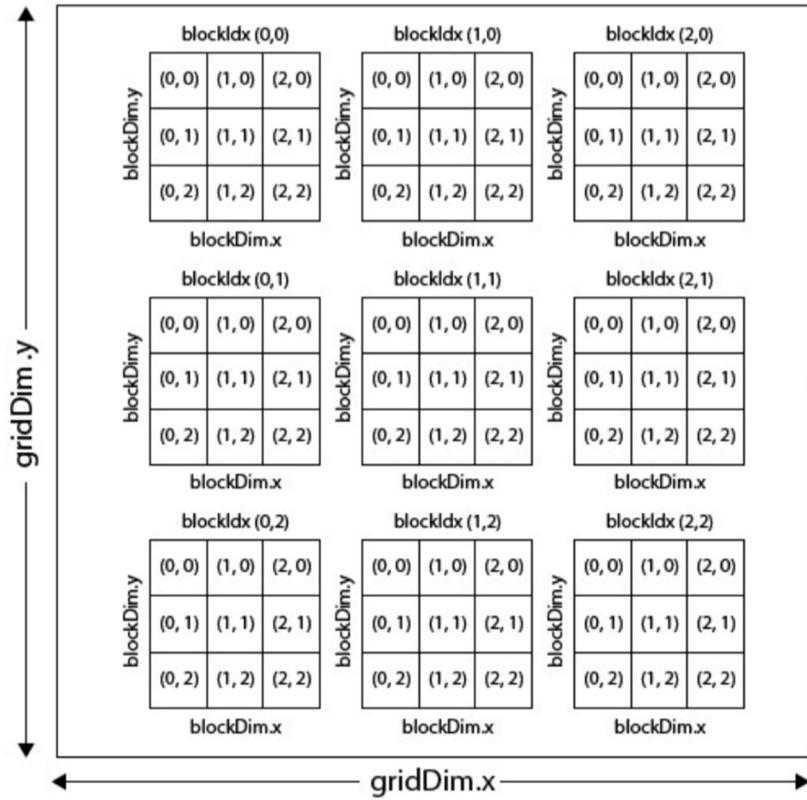


Рисунок 3.1 Организация вычислительных единиц на графическом процессоре

## 3.2 Распараллеливание последовательного решения и сравнение результатов

Последовательное решение задачи, рассмотренное в предыдущей главе возможно распараллелить несколькими способами. Например, при решении СЛАУ матричным методом воспользуемся функцией `CUDADot[]`, которая позволяет перемножить два двумерных массива, предварительно загруженных в память графического ускорителя, для загрузки в память будем использовать функцию `CUDAMemoryLoad[]`. Также решим задачу о вдавливании эллиптического штампа в полупространство для всех дискретизаций функции поверхностного распределения и проведём сравнительный анализ скорости решения.

При выполнении параллельных расчётов были выбраны следующие значения размера граничных элементов для дискретизации ими полупространства:

$$\frac{3}{19}, \frac{3}{39}, \frac{3}{59}, \frac{3}{79}$$

Соответственно, расчёты проводились для 20, 40, 60 и 80 граничных элементов, а также для 400, 1600, 3600, 6400 расчётных узлов. Для вычислений использовался компьютер на базе центрального процессора Intel Core i5 8400, с тактовой частотой 3800 МГц, объёмом оперативной памяти DDR3 8 Гб и графического адаптера NVIDIA GeForce GTX 1050Ti.

Из рисунка видно, что при решении СЛАУ ускорение параллельных вычислений растёт практически линейно по сравнению с последовательными, с увеличением количества граничных элементов. Для 80 граничных элементов ускорение решения СЛАУ составило примерно 64 раза (для 6400 граничных элементов в пространстве)

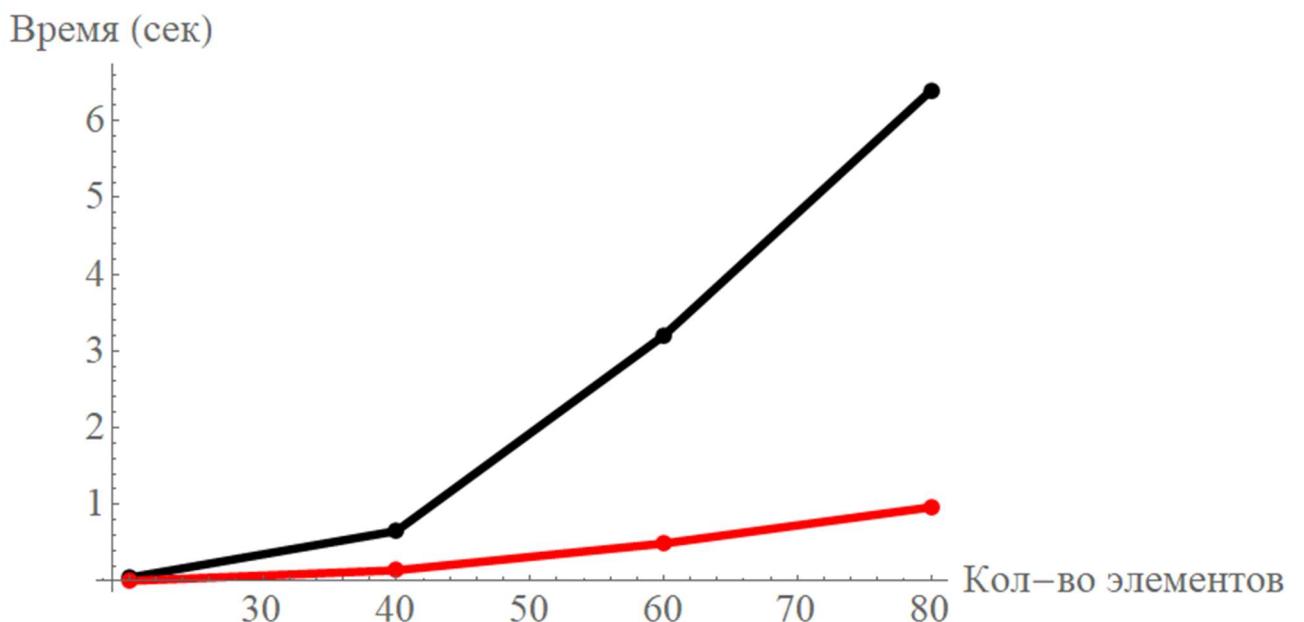


Рисунок 3.2 Сравнивание временных затрат при решении СЛАУ матричным методом последовательно (чёрная линия) и с помощью CUDA Dot[] параллельно (красная линия)

Получив данные о затраченном на решение задачи времени, построим график ускорения расчёта фиктивных усилий в полупространстве.

Коэффициент ускорения

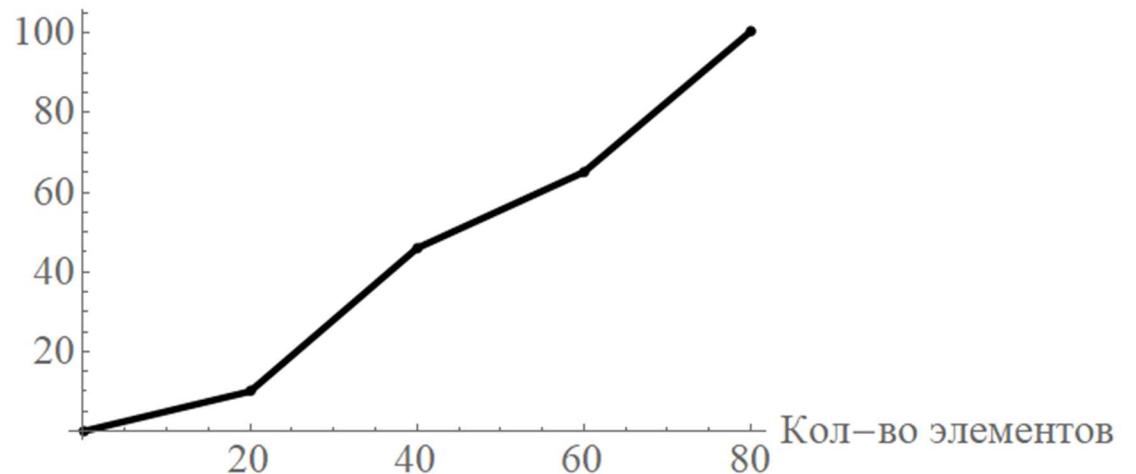


Рисунок 3.3 Коэффициенты ускорения расчётов при решении СЛАУ матричным методом.

## ЗАКЛЮЧЕНИЕ

Методом граничных элементов была решена задача о неравномерном стационарном нагружении поверхности полупространства давлением с эллиптическим распределением. Так же были продемонстрированы результаты для нагружения полупространства по прямоугольной области, были приведены графики шести независимых компонент тензора напряжений и 3 компонент вектора перемещений.

Распределение усилий полупространства было получено на основе последовательных и параллельных вычислений с использованием технологии Nvidia CUDA.

В результате было получено уменьшение времени расчетов. Для 6400 граничных элементов было получено ускорение последовательных вычислений в 101 раз при расчёте распределения давления на поверхности полупространства.

## **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Щербаков, С.С., Напряженно-деформированное состояние и объемная повреждаемость в окрестности контактного взаимодействия в трибофатической системе колесо / рельс с учетом неконтактного деформирования рельса / С.С. Щербаков, О.А. Насань // Вестник БелГУТА: Наука и транспорт. – 2016. – № 1 (32). – С. 234–247
2. Журавков, М.А. Объемная повреждаемость динамически нагруженных элементов режущего инструмента сельскохозяйственного комбайна / Журавков М.А., Щербаков С.С., Насань О.А. // Теоретическая и прикладная механика: Межведомственный сборник научно-методических статей. – Вып.30. –Минск: БНТУ, –2015. – С. 297 – 305.
3. Бенерджи, П. Метод граничных элементов в прикладных науках / П. Бенерджи, Р. Баттерфилд – М: Мир, 1984. – 494 с.
4. Крауч, С Методы граничных элементов в механике твердого тела / С. Крауч, А. Старфилд – М:Мир, 1987. – 328 с.
5. Beer, G. Boundary element method with programming / G. Beer, I. Smith, Duenser C. – Springer, 2008. – 494 р.
6. Журавков, М.А. Гранично-элементное моделирование с использованием элементов с квадратичным распределением усилий и распараллеливанием вычислений / М.А. Журавков, А.В. Круподеров, С.С. Щербаков // Теоретическая и прикладная механика: Межведомственный сборник научно-методических статей. –Вып.29. –Минск: БНТУ, –2014. –С.105-110.
7. [http://www.nvidia.ru/object/cuda\\_home\\_new.html](http://www.nvidia.ru/object/cuda_home_new.html)
8. Jason Sanders. CUDA by Example: An Introduction to General-Purpose GPU Programming 1st Edition.
9. Щербаков С.С., Сергеев В.Э. Применение технологии CUDA для распараллеливания гранично-элементных вычислений. // Теоретическая и прикладная механика. Международный научно-технический сборник. Вып.32. Минск: БНТУ, 2017. С. 330 — 335.

# Приложение А: листинг кода в программе Wolfram Mathematica для расчёта задачи о вдавливании эллиптического штампа в полуупространство

## Инициализация

```

In[1]:= ME = 1010; ν = 0.3; λ = ν ME / (1 + ν) (1 - 2 ν); μ = ME / (2 (1 + ν)); dz = -0.1;
U0 = 0.01;
p0 = P0 = 10000;
b = 0.6; a = 1;
bx = 1; by = 1;

Pressure[x_, y_, a_, b_, p0_] = If[(x/a)2 + (y/b)2 ≤ 1, Re[P0 Sqrt[1 - (x/a)2 - (y/b)2]], 0];
(*Pressure[x_,y_,a_,b_,p0_]:=If[(x/a)^2+(y/b)^2≤1,P0,0];*)

s = 1.5 * a; (*размер расчетной области*)
(*Установить h = 3a/19 для n = 20*)
h = 3 a / 19; (*размер ГЭ*)
n = 2 s / h + 1; (*number of elementary units*)
(*array of X-coordinates*)
X = Table[xh, {xh, -s, s, h}];
(*таблица значений*)
Y = Table[yh, {yh, -s, s, h}];
(*таблица значений*)

Plot3D[Displacement[x, y, a, b, U0], {x, -s, s}, {y, -s, s}, PlotRange → All];
(*график функции 2-х переменных*)

```

## Графики функции поверхностного распределения и её дискретизации

```

In[26]:= DisplacementTab = Table[Displacement[X[[xi]], Y[[yj]], a, b, U0], {xi, 1, n}, {yj, 1, n}];
(*таблица значений*)

PressureTab = Table[Pressure[X[[xi]], Y[[yj]], a, b, P0] // N, {xi, 1, n}, {yj, 1, n}];
(*таблица значений*)
(*численное приближение*)

ListPlot3D[DisplacementTab, InterpolationOrder → 0, Filling → None, Mesh → None,
(*3-мерная диаграмма разброса*)
PlotRange → All];
(*отображает*)

```

In[29]:= BoundaryElements = Table[{X[[xi]], Y[[yj]], DisplacementTab[[xi, yj]]}, {xi, 1, n},
(\*таблица значений\*)
{yj, 1, n}];

### 3)Перемещения внутри эллипса заданы

```
In[37]:= DisplacementTab = Table[Displacement[X[[xi]], Y[[yj]], a, b, U0], {xi, 1, n}, {yj, 1, n}];  
(*таблица значений*)  
amount = 0;  
(*Матрица позиций эллипса*)  
Mp = {};  
For[i = 1, i <= n, i++,  
|цикл для  
For[j = 1, j <= n, j++,  
|цикл для  
If[DisplacementTab[[i, j]] ≠ 0,  
|условный оператор  
amount++;  
AppendTo[Mp, {i, j}];,  
|добавить в конец к  
a;  
(*Print["i: ", i, " j: ", j]*);  
  
]  
Print["Amount of Boundary elements inside ellipse is: ", amount]  
|печатать  
Ak = Table[qk, {k, 1, amount}];  
|таблица значений  
DisV[xj_, yj_] :=  
Sum[U[3, 3] [X[[Mp[[m, 1]]]] - xj, h, Y[[Mp[[m, 2]]]] - yj, h, U0, p0, λ, μ] * Ak[[m]],  
|сумма  
{m, 1, amount}];  
Amount of Boundary elements inside ellipse is: 76  
  
AbsoluteTiming[A = {};  
|длительность по настенным часам  
For[i = 1, i < n + 1, i++,  
|цикл для  
For[j = 1, j < n + 1, j++,  
|цикл для  
AppendTo[A, Flatten[DisV[X[[i]], Y[[j]]]]];]  
|добавить в к*** |уплотнить  
]  
InvertedMatrix = Inverse[A];  
|обратная матрица  
  
cudaMemInvertedMatrix = CUDAMemoryLoad[InvertedMatrix];  
cudaMemAppliedForce = CUDAMemoryLoad[FPT];  
  
AbsoluteTiming[SearchForceWithCuda = CUDADot[cudaMemInvertedMatrix, cudaMemAppliedForce]];  
|длительность по настенным часам  
  
AbsoluteTiming[PLS1 = Dot[InvertedMatrix, FPT]];  
|длительность по настен*** |скалярное произведение
```

```

SolutionValues = Table[PLS1[[1, i]], {i, 1, amount}];
|таблица значений

(*добавление столбца со значениями, выполнять только один раз*)
Tmp = Transpose[Mp];
|транспозиция

AppendTo[Tmp, SolutionValues];
|добавить в конец к

Mp = Transpose[Tmp];
|транспозиция

FinalList = Table[0, {i, 1, n}, {j, 1, n}];
|таблица значений

Table[FinalList[[Mp[[i, 1]], Mp[[i, 2]]]] = Mp[[i, 3]], {i, 1, amount}];
|таблица значений

(*в необходимые места подставляем полученные значения, остальные приравниваем 0,
для того, чтобы построить 3-х мерный график*)

ListPlot3D[FinalList, InterpolationOrder → 0, Filling → Axis, Mesh → None,
|3-мерная диаграмма раз... [порядок интерполяции] |заливка |ось |сетка |ни одного
|отображает... |всё

PlotRange → All]
|отображает... |всё

SeqSet = {{20, 0.0098949}, {40, 0.134317}, {60, 0.799889}, {80, 3.5948636563}};
ParalelSet = {{20, 0.0015787}, {40, 0.0082962}, {60, 0.0214812}, {80, 0.0563654}};

ListLinePlot[{SeqSet, ParalelSet, SeqSet, ParalelSet},
|линейный график данных

PlotRange → All, PlotStyle → {{Black, Thickness[0.01]}, {Red, Thickness[0.01]}},
|отображает... |всё |стиль графика |чёрный |толщина |красный |толщина

AxesLabel → {"Кол-во элементов", "Время (сек)" },
|обозначения на осях

LabelStyle → {FontFamily → "Times New Roman", 20},
|стиль отметки |семейство шрифта

Joined → {True, True, False, False}]
|соединяет... |ист... |ист... |ложь |ложь

BoostSet = {{0, 0}, {20, 6.267751947805157}, {40, 16.190183457486558},
{60, 37.2366999981379}, {80, 63.777843434}};

ListPlot[{BoostSet, BoostSet},
|диаграмма разброса данных

PlotRange → All, PlotStyle → {{Black, Thickness[0.01]}, {Black, PointSize[0.015]}},
|отображает... |всё |стиль графика |чёрный |толщина |чёрный |размер точки

AxesLabel → {"Кол-во элементов", "Коэффициент ускорения"},
|обозначения на осях

LabelStyle → {FontFamily → "Times New Roman", 20},
|стиль отметки |семейство шрифта

Joined → {True, False}]
|соединяет... |ист... |ложь

```