ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ОЛЕСЯ ГОНЧАРА

Факультет прикладної математики

Кафедра обчислювальної математики та математичної кібернетики

Дипломна робота

Перший (бакалаврський) рівень вищої освіти

Спеціальність 113 Прикладна математика

АЛГОРИТМ РОЗВ’ЗАННЯ КРАЄВОЇ ЗАДАЧІ ДЛЯ ТОНКОСТІННОЇ СИСТЕМИ ПРИ ЗМІННИХ УМОВАХ НА КОНТУРІ

Виконавець

студент групи ПА–16–1

Ф.О. Мостовий

Керівник

доцент кафедри обчислювальної математики та математичної кібернетики,  
канд. фіз.-мат. наук

Ю.Я. Годес

Завідувач кафедри обчислювальної математики та математичної кіберне­тики,  
канд. фіз.-мат. наук

В.А. Турчина

2020

Зміст

[ВСТУП 3](#_Toc41638837)

[1. ОГЛЯД ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ 3](#_Toc41638838)

[1.1. ТЕОРЕТИЧНІ ТА МЕТОДОЛОГІЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ 4](#_Toc41638839)

[1.2. МАТЕМАТИЧНІ МЕТОДИ ТА МОДЕЛІ У ДОСЛІДЖУВАНІЙ ГАЛУЗІ 5](#_Toc41638840)

[Метод скінченних різниць (метод сіток) 5](#_Toc41638841)

[Метод Бубнова-Гальоркіна 6](#_Toc41638842)

[Метод Рітца-Тимошенко 7](#_Toc41638843)

[Метод Кантаровича-Власова 8](#_Toc41638844)

[1.3. ФІЗИЧНА ПОСТАНОВКА ЗАДАЧІ 9](#_Toc41638845)

[1.4. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ 10](#_Toc41638846)

[2. МЕТОДИКА РОЗВ’ЯЗАННЯ ЗАДАЧІ 11](#_Toc41638847)

[2.1. РОЗВ’ЯЗОК ЗАДАЧІ ДЛЯ ВІЛЬНО ОПЕРТОЇ ПЛАСТИНИ ПІД ГІДРОСТАТИЧНИМ НАВАНТАЖЕННЯМ 12](#_Toc41638848)

[2.2. ПРЯМОКУТНА ПЛАСТИНА ПІД ГІДРОСТАТИЧНИМ НАВАНТАЖЕННЯМ ДВІ ПРОТИЛЕЖНІ ГРАНИЦІ ВІЛЬНО ОПЕРТІ ІНШІ ДВІ ЗАЩЕМЛЕНІ 15](#_Toc41638849)

[2.3. ПРЯМОКУТНА ПЛАСТИНА ПІД ГІДРОСТАТИЧНИМ НАВАНТАЖЕННЯМ ДВІ ПРОТИЛЕЖНІ ГРАНИЦІ ВІЛЬНО ОПЕРТІ ІНШІ ДВІ ВІЛЬНІ 17](#_Toc41638850)

[3. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ТА РЕЗУЛЬТАТИ ЧИСЕЛЬНОГО МОДЕЛЮВАННЯ 20](#_Toc41638851)

[3.1. ОСОБЛИВОСТІ ЧИСЕЛЬНОЇ РЕАЛІЗАЦІЇ 20](#_Toc41638852)

[3.2. СТРУКТУРА ПРОГРАМНОГО ПРОДУКТУ 20](#_Toc41638853)

[3.3. РЕЗУЛЬТАТИ 21](#_Toc41638854)

[ВИСНОВКИ 24](#_Toc41638855)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 25](#_Toc41638856)

[ДОДАТОК 26](#_Toc41638857)

# ВСТУП

Пластини або ж, інакше, тонкі плити є важливими конструкторськими елементами багатьох споруджень. Розрахунок під дією різних типів навантажень має широкий спектр застосування в залежності від положення, розміщення приладів, місця експлуатації, та багатьох інших факторів, панелі виготовляються з різних матеріалів, починаючи з дерева, закінчуючи композитними матеріалами. При проектуванні приладових панелей значна кількість зусиль витрачається на розрахунки жорсткості та міцності майбутнього виробу. В ході цих розрахунків розраховують вигини під навантаженнями, аналізують отримані результати на основі яких приймаються подальші рішення в розробці.

На сьогоднішній день однією з найбільш розповсюджених у техніці конструктивних деталей є пластина. Одним із багатьох методів використання пластин є створення, наприклад, приладова панелей, власне для закріплення приборів. Такі панелі досить розповсюджені в усіх промислових, і не тільки, галузях. Починаючи від воєнної техніки та промислового машинобудування, закінчуючи побутовими товарами різних масштабів.

Теорія розрахунку пластинок існує вже досить довго, проте більша частина типових задач була вирішена наприкінці минулого століття більшою мірою тоді ще радянськими вченими серед яких в першу чергу мають бути названі І. Г. Бубнов, С. П. Тимошенко, Б. Г. Гальоркін, Ю. А. Шиманський, П. Ф. Папкович та В. З. Власов.

Метою даної роботи було:

* Побудова фізичної та математичної моделі для пластинки з защемленими протилежними кінцями.
* Опис алгоритму розв’язку крайової задачі для визначених граничних умов закріплення тонкої однорідної гнучкої пластини.
* Програмна реалізація описаного алгоритму для розв’язання крайової задачі з довільними вхідними параметрами.
* Графічне відображення отриманих результатів.

# 1. ОГЛЯД ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

## 1.1. ТЕОРЕТИЧНІ ТА МЕТОДОЛОГІЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

Пластиною або плитою – називають тіло призматичної або циліндричної форми, один з розмірів якої (*h* – товщина пластини) є меншим порівняно з її іншими розмірами.

Тонкою пластиною – називають таку пластину, для якої виконується наступна нерівність  ( − характерний розмір основи).

Серединною площиною – називають площину, рівновіддалену від обох поверхностей пластини, розділяючу тіло на дві рівні частини. [1]

Теорія розрахунку пластинок базується на ряді гіпотез (гіпотезах Кіргхофа), які дозволяють звести трьох вимірну задачу до двох вимірної або одновимірної [2]:

* *Гіпотеза прямої нормалі*, згідно з якою нормаль до серединної площини залишається прямою і до вигнутої серединної площини, довжина її не змінна.
* *Гіпотеза незмінності довжини серединної площини*. На основі цієї гіпотези вважають, що серединна площина не змінює розмірів та форми при вигині.

## 1.2. МАТЕМАТИЧНІ МЕТОДИ ТА МОДЕЛІ У ДОСЛІДЖУВАНІЙ ГАЛУЗІ

Кажучи про задачу математичної фізики, будемо мати на увазі математичну модель якогось фізичного або іншого явища, що має в своєму складі одне або більше диференційних рівнянь разом з додатковими співвідношеннями – початковими та/або кінцевими умовами.

Повна оцінка будь-якого методу має включати його переваги та недоліки, оскільки порівняння методів математичної фізики не може виділити не один із них – жоден не може бути абсолютно досконалим для вирішення усіх можливих задач.

### Метод скінченних різниць (метод сіток)

Метод Скінченних Різниць (МСР) виділяється своєю універсальністю, котра є значно вищою, аніж у будь-якого аналітичного метода. Використання цього методу часто характеризується відносною простотою побудови вирішального алгоритму та його програмної реалізації: майже завжди можна розбити алгоритм на паралельні задачі.  
До недоліків методу можна віднести його використання на нерегулярних сітках, швидке зростання трудомісткості, як наслідок зростання розмірності задачі (кількості змінних), а також складність аналітичного досліду властивостей різницевої схеми.

Суть МСР полягає в заміні початкової (безперервної) задачі математичної фізики на її дискретний аналог (різницеву схему) с подальшим використанням спеціальних алгоритмів вирішення дискретної задачі.

Зазвичай МСР складається із чотирьох кроків:

1. **Дискретизація**: уся ділянка безперервної зміни аргументів заміняється на кінцевий або полічений набір точок, що звуться вузлами. Уся множина вузлів називається сіткою. Замість функцій безперервних аргументів розглядають функції, що визначені на сітці (сітковій площині) та називаються сітковими функціями. Рівняння та умови, що входять до опису задачі математичної фізики, заміняються на їх дискретні аналоги. Так будується різницева схема.
2. **Аналітичне дослідження схеми**: теоретичне дослідження головних властивостей різницевої схеми, тобто апроксимації, стійкості та сходження. Тут визначаються порядки сходження схеми відносно параметрів дискретизації.
3. **Алгоритмізація**: розроблюється алгоритм рішення дискретної задачі, лістинг комп’ютерної програми, що реалізує алгоритм та зневаження самої комп’ютерної програми.
4. **Експериментальне дослідження**: формуються тестові завдання, котрі можна обчислити з досить високою точністю й альтернативними методами. Потім за допомогою розробленої програми досліджується сходження сіткових рішень тестових задач до високоточних при зменшенні шагу сітки. Таким чином перевіряють відповідність фактичних порядків сходження теоретичним. Експериментальне дослідження є значним але не абсолютним доказом практичності розробленої схеми для вирішення початкової задачі.

### Метод Бубнова-Гальоркіна

Свою назву метод набув завдяки Б. Г. Гальоркіну, котрий визначив метод вирішення крайових задач як для звичайних диференційних рівнянь, так і для рівнянь в частинних похідних. А також завдяки І. Г. Бубнову, котрий вперше у 1913р. скористався цим методом для вирішення задачі по викривленню пластини у будівництві човнів.

В загальному вигляді задача по методу Гальоркіна формулюється як знаходження рішення диференційного рівняння з крайовими умовами:*D(U) = 0; B(U) = 0;*Де потрібно знайти таку функцію *U*, котра задовольняє наданим диференційним рівнянням та крайовим умовам. Таку задачу зазвичай неможливо вирішити аналітичними методами.

Приклад кроків знаходження рішення по методу Бубнова-Гальоркіна.

1. **Визначення наближеної функції** деякого ряду, що може замінити функцію ***U*** на якомусь відрізку.
2. **Добір вагових функцій типу w(x),** котрі визначають, що наближена функція частково задовольняє рішенню, а отже ми можемо побудувати кінцеву функцію ***u(x)***.
3. **Побудова системи лінійних рівнянь**, за допомогою котрих знаходяться усі потрібні коефіцієнти.
4. **Приблизне рішення для нашої наближеної функції** дає нам апроксимоване рішення, що нам потрібно.
5. **Рішення диференційного рівняння** з крайовими умовами для перевірки.

### Метод Рітца-Тимошенко

Про метод стало відомо з роботи С.П. Тимошенко (на той час декана КПІ) під назвою «Про стійкість пружних систем», яка була подана для отримання премії імені Журавського Д. І. у 1910 р. В цій роботі Тимошенко дає рішення великої кількості задач на стійкість суцільних, решітчастих та тонкостінних стержнів разом з тонкими пластинами за різних умов та навантажень. Усі рішення використовували варіант енергетичного метода.

За два роки до Тимошенко Вальтер Рітц у своїй роботі «Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik» (Про новий метод вирішення певних варіаційних задач у математичній фізиці) 1909 р. описав метод мінімизуючої послідовності функцій, для знаходження мінімуму функціонала. Пізніше ці ідеї були застосовані для рішення прикладних задач С. П. Тимошенко, що в результаті і дало назву методу математичної фізики.

Варіаційний метод Рітца-Тимошенко дозволяє вирішувати не лише лінійні задачі, але й геометричні та фізичні нелінійні задачі різної складності особливо у випадках, коли запис диференційних рівнянь неможливий або надто складний. Метод базується на використанні відомого із курсу теоретичної механіки принципу можливих переміщень, котрий можливо проілюструвати так:

***δA - δU = 0,*** де  ***δA*** – можлива робота зовнішніх сил на протязі деякого можливого переміщення  
 ***δU*** – можлива робота внутрішніх сил, як прирощення потенційної енергії.

З прикладної точки зору важливо розуміти, що швидкість сходження (будь-якого) метода Рітца залежить від вибору системи базисних функцій. Хоча при досить вдалому виборі для досягнення бажаної точності може бути достатньо 3-4 складника в лінійній комбінації.

### Метод Кантаровича-Власова

Л. В. Канторович у 1933 р. запропонував метод наближеного рішення задачі про мінімум подвійного інтегралу, котрий дозволяє звести двомірну задачу до одномірної. Пізніше у 1946 р. В. З. Власов застосував метод Канторовича у вирішенні задач будівельної механіки пластин та оболонок. Аби замінити двомірну задачу викривлення пластин та оболонок до одномірної, функція прогину задається у вигляді суми множення функцій; серед цих функцій одна змінна вважається відомою, тобто вона задається, а інша – по іншій змінній – має бути знайдена.

Власов також скористався варіаційним рівнянням Лагранжа – початком можливих переміщень. На відміну від метода Бубнова-Гальоркіна, де рішення диференційного рівняння зводиться до рішення системи алгебраїчних рівнянь, цей метод заміщає інтегрування диференційного рівняння в частинних похідних на систему звичайних диференційних рівнянь. Якщо задача лінійна, то виходить система звичайних лінійних рівнянь з постійними коефіцієнтами.

Сходження методу Кантаровича-Власова, як і у випадку із методом Рітца-Тимошенко, дуже залежить від початкових умов: від того скільки приближень буде прийнято спочатку. Хоча треба зазначити, що цей метод дає одні з найточніших результатів поряд разом із найшвидшим сходженням.

## 1.3. ФІЗИЧНА ПОСТАНОВКА ЗАДАЧІ

На кінцях осі *х* (*х = 0* та *х = а*) пластинка защемлена, таке кріплення не дозволяє жодних зміщень. На ребрах *у = -b/2* та *у = b/2* пластина вільна від будь-яких кріплень. Навантаження розподіллено по гідростатичному закону, як продемонстровано на рисунку нижче (рис. 1).

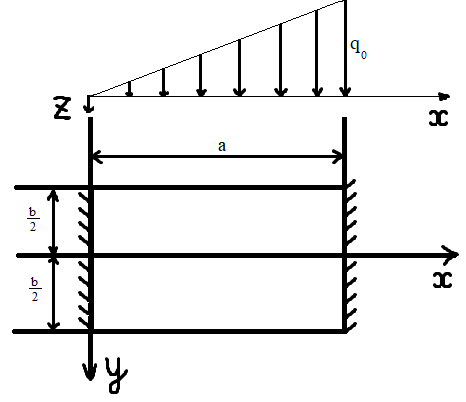


Рис. 1

Механічні властивості досліджуваної тонкої, гнучкої, однорідної пластини виражаються коефіцієнтом - *циліндрична жорсткість,* позначається *D* та визначається за формулою:

де:

– коефіцієнт Юнга,

– висота пластинки,

– коефіцієнт Пуассона.

## 1.4. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ

В математичному плані задача зводить до знаходження розв’язку крайової задачі. У даному випадку маємо наступні граничні умови (див. рис.1):

1. На осі *y*:

1. На осі *х:*

Також слід зауважити, що оскільки пластинка є тонкою, то має місце співвідношення:

Крім того представлений розв’язок задачі не входить визначення критичних значень навантажень, що призводять до розриву або ж сильних деформацій пластини. Доцільне використання представленого нижче алгоритму для знаходження поля прогинів, максимальне значення якого знаходиться в околі (де - є товщиною представленої пластини).

# 2. МЕТОДИКА РОЗВ’ЯЗАННЯ ЗАДАЧІ

Загалом пошук поля прогинів тонких пластин під поперечним навантаженням зводиться до інтегрування диференційного рівняння четвертого порядку у частинних похідних, що носить назву *рівняння Софі Жермен*:

|  |  |
| --- | --- |
|  |  |

При інтегруванні рівняння Софі Жермен з’являться довільні константи, які й мусять визначатись із умов кріплення пластини по контуру, які в свою чергу визначаються з характеру закріплення границь пластини. Найбільш універсальними, хоча й наближеними аналітичними методами розрахунку пластин є варіаційні методи, що коротко характеризувалися у попередньому параграфі. Проте в даній роботі застосовувався метод *«Суперпозицій»*. Цей метод полягає в тому, що прогин пластини з заданими граничними умовами кріплення можна розглядати як результат впливу кожного кріплення узагалі й зокрема. За цим принципом результуючий ефект для зазначених умов визначається як сума ефектів, які впливають кожним діянням окремо:

|  |  |
| --- | --- |
|  |  |

Досить важливо підкреслити, що метод «Суперпозицій» може бути застосований лише до систем, поведінка яких описується лінійними рівняннями.

Увесь подальший розв’язок побудований на базовому рішенні задачі знаходження функції прогинів шарнірно опертої по контуру пластини під гідростатичним навантаженням та накладанням на цей розв’язок обмежуючих функцій прогинів для різних кріплень на кінцях. Суперпозиція усіх окремих рішень і буде розв’язком цільової задачі.

## 2.1. РОЗВ’ЯЗОК ЗАДАЧІ ДЛЯ ВІЛЬНО ОПЕРТОЇ ПЛАСТИНИ ПІД ГІДРОСТАТИЧНИМ НАВАНТАЖЕННЯМ

Для початку необхідно розглянути розв’язок поставленої задачі для більш простого випадку – шарнірно опертої по контуру пластини під довільним навантаженням розподіленим по гідростатичному закону (рис. 2).

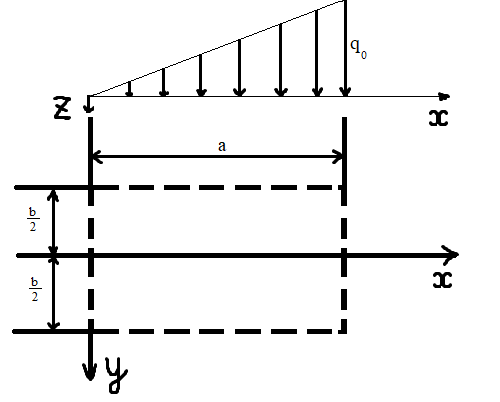


Рис. 2

Досить витончене рішення описав М. Леві, запропонувавши записати рішення у вигляді ряду наступним чином:

|  |  |
| --- | --- |
|  |  |

Де *Ym* є функцією лише однієї змінної *у*. При цьому оскільки границі *х=0* та *х=а* теж вільно оперті, тому кожен член ряду (3) задовольняє на цих двох краях граничним умовам . Тож залишається лише визначити *Ym* так, щоб задовольнити граничним умовам на кінцях , а також рівнянню вигнутої поверхності:

|  |  |
| --- | --- |
|  |  |

У якості спрощення рішення для даного методу, приймемо загальне рішення у вигляді:

|  |  |
| --- | --- |
|  |  |

Де [1]

|  |  |
| --- | --- |
|  |  |

визначає прогин однієї умовної лінії під навантаженням, розподіленим по закону трикутника. Ця рівність задовольняє наступне диференційне рівняння:

|  |  |
| --- | --- |
|  |  |

Частина *w2* візьмемо у вигляді ряду:

|  |  |
| --- | --- |
|  |  |

Підставивши рівняння (8) та (6) у рівність (5) отримаємо:

|  |  |
| --- | --- |
|  |  |

Де константи *Am* та *Bm* визначають із граничних умов на кінцях звідки отримаємо :

Для скорочення запису була використана наступна заміна:

Розв’язуючи рівняння вище, знаходимо:

|  |  |
| --- | --- |
|  |  |
|  |  |

Підставивши віднайдені константи (10) та (11) у рівняння (9) отримаємо шукану функцію прогинів шарнірно опертої по контуру пластинки під гідростатичним навантаженням:

|  |  |
| --- | --- |
|  |  |

## 2.2. ПРЯМОКУТНА ПЛАСТИНА ПІД ГІДРОСТАТИЧНИМ НАВАНТАЖЕННЯМ ДВІ ПРОТИЛЕЖНІ СТОРОНИ ВІЛЬНО ОПЕРТІ ІНШІ ДВІ ЗАЩЕМЛЕНІ

Як було зазначено раніше, прогин пластини під будь-яким поперечним навантаженням можна отримати, спочатку вирішивши задачу в припущені, що всі грані пластинки шарнірно оперті, а далі користуючись методом «Суперпозиції» накласти необхідні умови кожної з відрізних границь. На цьому кроці як продемонстровано на (рис. 3), *х=0* та *х=а* – защемлені. Тобто це означає, що необхідно з цих сторін додати такі згинальні моменти за величиною такі, щоб ліквідувати повороти здійснювані на цих границях дією поперечним навантаженням.

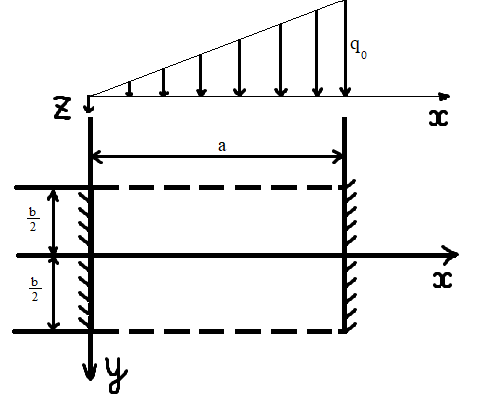


Рис. 3

Перш за все необхідно визначити нахили пластинки на границях *х=0* та *х=а*. Достатньо продиференціювати функцію прогинів по відповідній змінній:

|  |  |
| --- | --- |
|  |  |

Цей нахил необхідно ліквідувати тим самим задовільнивши граничним умовам (рис. 3). Визначимо згинаючі моменти по межам *х=0* та *х=а* наступним чином:

|  |  |
| --- | --- |
|  |  |

Визначимо коефіцієнти *Em* таким чином, щоб здійснюваний нахил цими моментами був рівним по величині нахилу, визначеному у рівності (13), проте мав протилежний до нього знак. При цьому отримаємо:

|  |  |
| --- | --- |
|  |  |

Відповідно підставивши коефіцієнти *Em* (15) у ряд (14) отримаємо:

|  |  |
| --- | --- |
|  |  |

По середині защемлених сторін, де *y = 0*, згідно до розташування початку відліку координат (рис. 3), члени ряду (16) з парними *m* зникають і сума ряду, як це й має бути, дорівнює половині відповідної суми для рівномірно навантаженої пластинки. Ряд досить швидко сходиться, і значення згинаючого моменту з легкістю можна отримати для будь якої точки затисненої границі.

## 2.3. ПРЯМОКУТНА ПЛАСТИНА ПІД ГІДРОСТАТИЧНИМ НАВАНТАЖЕННЯМ ДВІ ПРОТИЛЕЖНІ СТОРОНИ ВІЛЬНО ОПЕРТІ ІНШІ ДВІ ВІЛЬНІ

Залишилось розглянути останню окрему задачу, розв’язок якої входить до суми розв’язку цільової. Розглядаємо призматичну пластинку під гідростатичним тиском у якої *х=0* та *х=а* – вільно оперті, а – вільні від будь яких кріплень (рис. 4).

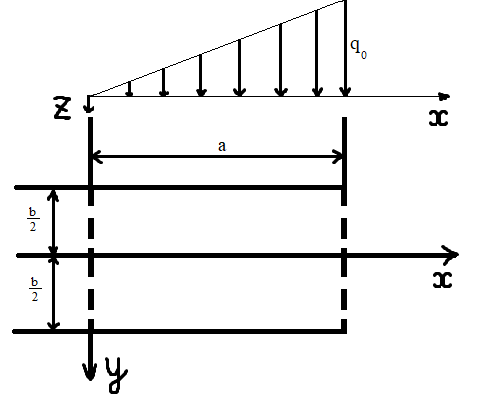


Рис. 4

Важливо підкреслити, що, оскільки, вибраний закон розподілення навантаження на пластинку симетричний відносно осі *х*, тому достатньо буде прийняти до уваги лише одну границю осі *у* (рис. 4). Наступні розрахунки будуть робитись для границі *y = b/2*. Почнемо з граничних умов кріплення пластинки по осі ординат:

|  |  |
| --- | --- |
|  |  |

Таким же чином як і при розв’язку попередньої задачі, скористаємося методом «Суперпозицій» - задаємо рівняння прогинів у вигляді (5) де:

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

Оскільки навантаження на пластинку симетричний відносно осі *х*, то *Ym* матиме такий вид:

|  |  |
| --- | --- |
|  |  |

Як і в задачі про визначення функції прогинів для шарнірно опертої по контуру пластини під гідростатичним навантаженням, коефіцієнти *Am* та *Bm* визначаються із граничних умов (17), визначених при спрощенні:

Розглядаючи граничні умови кріплення пластини на кінці *y = b/2*, отримаємо систему двох рівнянь відносно невідомих коефіцієнтів:

Вирішивши дану систему двох рівнянь, отримаємо шукані коефіцієнти:

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

Площина прогинів пластинки визначається функцією – результатом підстановки у (5), віднайдених функцій (18) та (19):

|  |  |
| --- | --- |
|  |  |

# 3. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМУ ТА РЕЗУЛЬТАТИ ЧИСЕЛЬНОГО МОДЕЛЮВАННЯ

## 3.1. ОСОБЛИВОСТІ ЧИСЕЛЬНОЇ РЕАЛІЗАЦІЇ

Як видно з параграфів попереднього пункту, кожне з окремих рішень є безкінечним функціональним рядом. Не складно здогадатися, що для підрахування суперпозиції цих рішень, необхідно знайти суму кожного ряду. Тут необхідно зазначити, що всі розглянуті окремі розв’язки побудовані опираючись в основі своїй на роботу С. П. Тимошенко та С. Войновського-Кригера «Пластинки та oболонки» [1]. В зазначеній вище роботі, окрім дослідження на збіжність результуючих рядів, приклав чимало зусиль, аби ряди збігались досить швидко. Тому при програмній реалізації не виникло жодних складностей. В середньому на обчислення одного члену функціонального ряду потребується від чотирьох до восьми ітерацій циклу.

## 3.2. СТРУКТУРА ПРОГРАМНОГО ПРОДУКТУ

Стек технологій, використаних для програмної реалізації в даній роботі, обраний з метою уникнення будь-якої несумісності або залежності від операційної системи, її версій тощо. Єдиними вимогами для перегляду програмної реалізації є доступ до мережі Інтернет та наявність сучасного браузера.

Алгоритм обчислення поверхності прогинів реалізовано на ECMAScript 6 (версія 2015 р.) далі - JavaScript. Враховуючи додаткові вимоги до точності обчислень рівнянь математичної фізики, котрі звичайний JavaScript не може забезпечити внаслідок обмежень технічного характеру, використана спеціальна JavaScript бібліотека з відкритим сирцевим кодом - *Math.js* (сайт проекту - <https://mathjs.org>).

Графічна візуалізація обчислень реалізована за допомогою іншої вільно доступної бібліотеки *Vis.js* (сайт проекту - <https://visjs.org>). Використовуючи можливості трьох вимірної візуалізації даних цієї бібліотеки побудовано інтерактивний інтерфейс моделі та, власне, саму математичну модель деформованої пластини.

Програма вбудована в HTML сторінку, що має додаткове меню, за допомогою котрого можливо задавати або змінювати параметри моделі та одночасно спостерігати її зовнішні зміни. Саму веб-сторінку створено за допомогою технологій HTML5, CSS3 та розробницького пакету WebStorm (сайт розробника - <https://www.jetbrains.com/webstorm>).

Перегляд демонстраційної версії математичної моделі доступно за посиланням: <https://tm.mova.club>

## 3.3. РЕЗУЛЬТАТИ

Нижче будуть представлені результати обчислення програмного комплексу при різних вхідних параметрах (різних геометричних та механічних властивостей пластин). Розпочнемо з тонкої металевої пластини прямокутної форми. Геометричні та механічні властивості (вхідні параметри) представлені зліва:

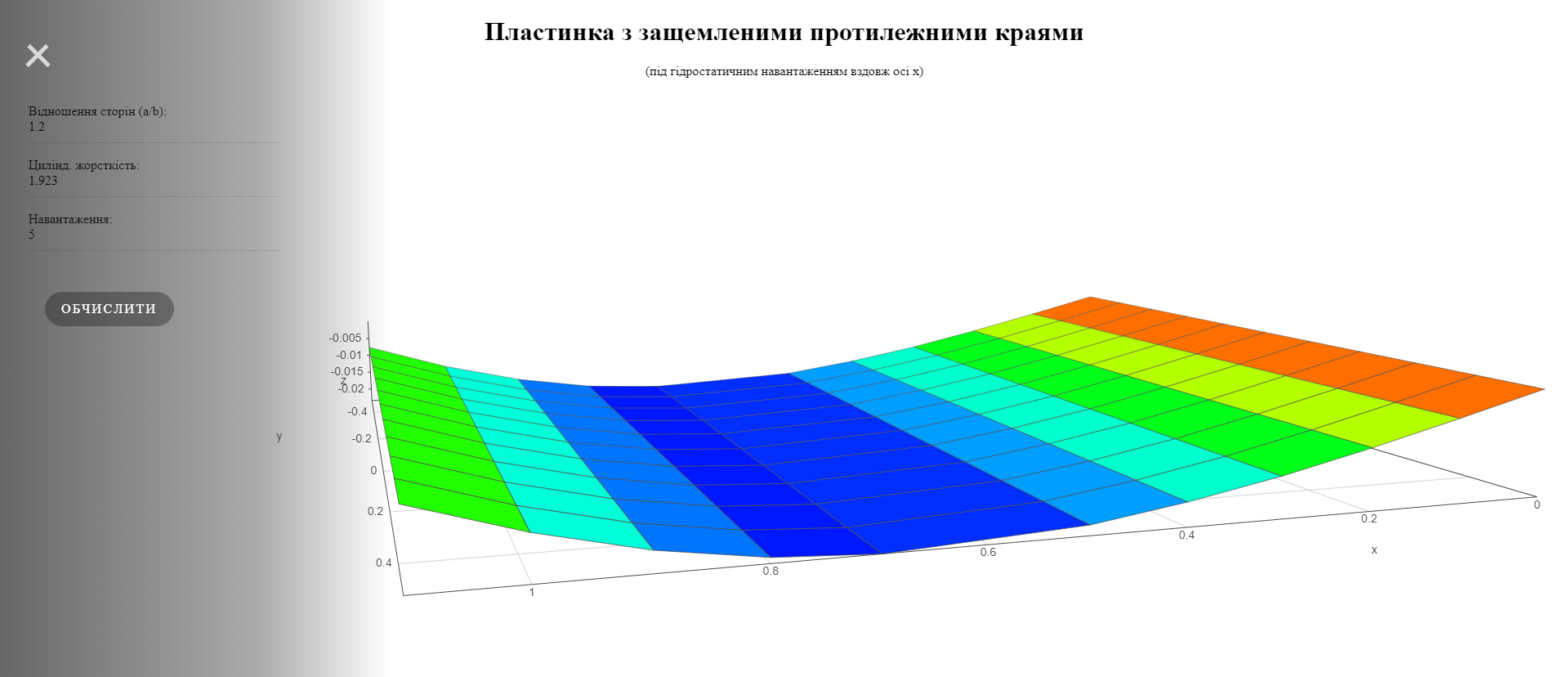


Рис. 5

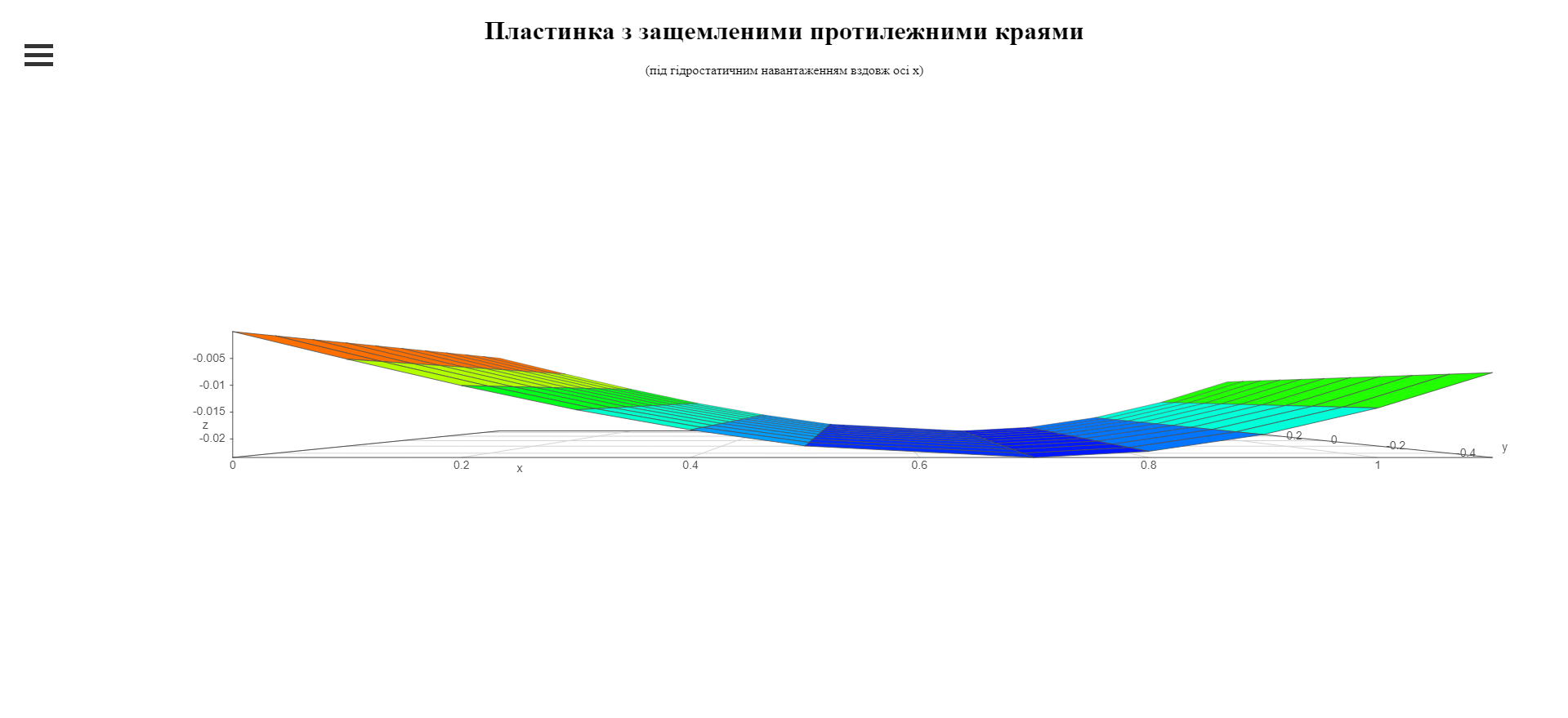


Рис. 6

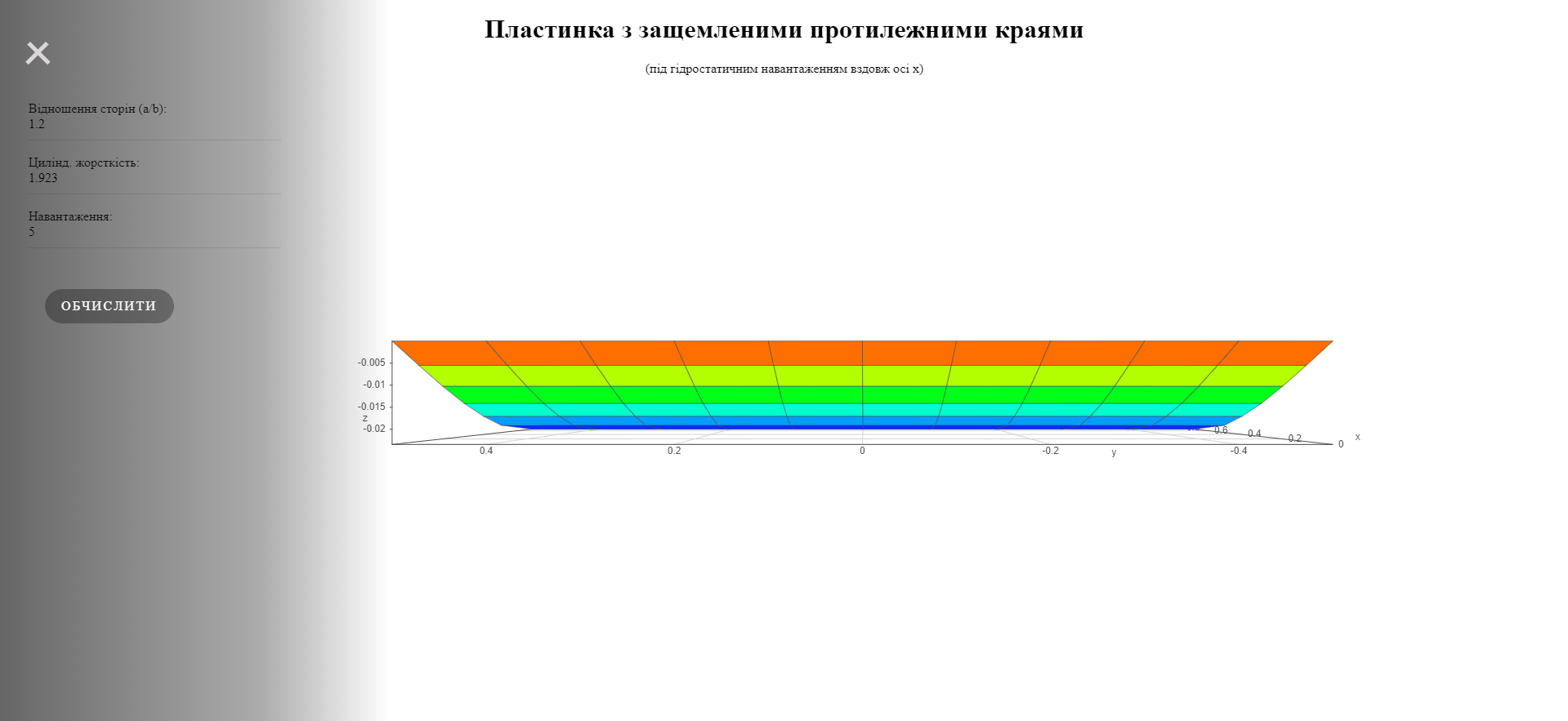


Рис. 7

# ВИСНОВКИ

Основні результати роботи такі:

* Сформульована фізична та математична постановка задачі про визначення згину прямокутної тонкої пластини з протилежно защемленими краями під гідростатичним навантаженням.
* Пошук та огляд типових методів та алгоритмів розв’язання схожих задач.
* Аналіз задачі з точки зору граничних умов та обрання оптимального методу рішення.
* Загальний опис розв’язання крайової задачі.
* Розробка програмного забезпечення на основі описаного розв’язку.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Тимошенко С. П. Пластинки и оболочки. М.: Наука, 1966.
2. Львов Г. И. Основы теории пластинок и оболочек. М.: Наука, 2003.
3. Иващенко Н. А., Фомін В. М. Численные методы расчётов в енерго машиностроение. М.: Наука, 1988.
4. Гоц А. Н. Численные методы расчёта машиностроения. Т. 1. М.: Гостройиздат, 1969.
5. Гоц А. Н. Численные методы расчёта машиностроения. Т. 2. М.: Гостройиздат, 1971.
6. Жемочкин Б. Н. Теория упругости. М.: Гостройиздат, 1957.
7. Гольденвейзер А. Н. Теория тонких упругих оболочек. М.: Гостройиздат, 1953.
8. Чорних К. Ф. Линейная теория оболочек. Т.1. Ленинград: ЛГУ, 1962.
9. Чорних К. Ф. Линейная теория оболочек. Т.2. Ленинград: ЛГУ, 1965.
10. Болотін В. В. Неконсервативные задачи теории упругой устойчивости. М.: Физматгиз, 1961.

# ДОДАТОК

main.css  
/\* Main area (graph) \*/  
.main-area {  
 height: 100%;  
 width: 100%;  
 position: fixed;  
 top: 0;  
 right: 0;  
  
 text-align: center;  
}  
.main-area #graph{  
 position: fixed;  
 top: 0;  
 left: 0;  
 right: 0;  
 bottom: 0;  
 z-index: -1;  
}  
/\* Menu side panel \*/  
.side-menu {  
 height: 100%;  
 width: 0;  
 position: fixed;  
 z-index: 1;  
 top: 0;  
 left: 0;  
 background: rgb(0,0,0);  
 background: linear-gradient(90deg, rgba(0,0,0,0.6) 0%, rgba(17,17,17,0.35) 67%, rgba(255,255,255,0.01) 100%);  
 overflow-x: hidden;  
 padding-top: 140px;  
 transition: 0.5s;  
 display: flex;  
 flex-direction: column;  
 align-items: flex-start;  
}  
.side-menu-change {  
 width: 23%;  
 padding-top: 120px;  
 padding-left: 35px;  
}  
  
/\* Floatin label & inputs \*/  
  
input {  
 font-size: 16px;  
 padding: 20px 0px;  
 height: 56px;  
 border: none;  
 border-bottom: solid 1px rgba(0, 0, 0, .1);  
 background:transparent;  
 min-width: 180px;  
 width: 70%;  
 box-sizing: border-box;  
 transition: all 0.3s linear;  
 color: #000;  
 font-weight: 400;  
 -webkit-appearance: none;  
 z-index: 5;  
}  
input:focus {  
 border-bottom: solid 1px #000000;  
 outline: 0;  
 box-shadow: 0 2px 6px -8px rgba(182, 157, 230, .45);  
}  
::placeholder{  
 color: #000;  
}  
.floating-label {  
 position: relative;  
 width: 100%;  
 margin-bottom: 10px;  
}  
.floating-label label {  
 position: absolute;  
 top: calc(45% - 7px);  
 left: 0;  
 opacity: 0;  
 z-index: 1;  
 transition: all 0.3s ease;  
}  
.floating-label input:not(:placeholder-shown) {  
 padding: 28px 0px 12px 0px;  
}  
.floating-label input:not(:placeholder-shown) + label {  
 transform: translateY(-10px);  
 opacity: 0.9;  
}  
/\* Menu btn \*/  
.btn-container {  
 display: inline-block;  
 cursor: pointer;  
 z-index: 12;  
 position: absolute;  
 top: 50px;  
 left: 30px;  
}  
.bar1,  
.bar2,  
.bar3 {  
 width: 35px;  
 height: 5px;  
 background-color: #333;  
 margin: 6px 0;  
 transition: 0.4s;  
}  
.menu-view-change .bar1 {  
 background-color: rgb(218, 215, 215);  
 -webkit-transform: rotate(-45deg) translate(-9px, 6px);  
 -ms-transform: rotate(-45deg) translate(-9px, 6px);  
 transform: rotate(-45deg) translate(-9px, 6px);  
 -webkit-transform: background-color 0.5s linear;  
 -ms-transform: background-color 0.5s linear;  
 transform: background-color 0.5s linear;}  
.menu-view-change .bar2 {  
 opacity: 0;  
}  
.menu-view-change .bar3 {  
 background-color: rgb(218, 215, 215);  
 -webkit-transform: rotate(45deg) translate(-8px, -8px);  
 transform: rotate(45deg) translate(-8px, -8px);  
  
 -webkit-transform: background-color 0.5s linear;  
 -ms-transform: background-color 0.5s linear;  
 transform: background-color 0.5s linear;  
}  
.calc-btn-area {  
 display: flex;  
 flex-direction: row;  
 align-items: center;  
 justify-content: start;  
 padding-top: 40px;  
 width: 100%;  
}  
.button {  
 display: inline-block;  
 padding: 0.75rem 1.25rem;  
 margin-left: 20px;  
 border-radius: 10rem;  
 color: #fff;  
 background-color: transparent;  
 border:none;  
 text-transform: uppercase;  
 font-size: 1rem;  
 letter-spacing: 0.15rem;  
 transition: all 0.3s;  
 position: relative;  
 overflow: hidden;  
 z-index: 1;  
}  
.button:after {  
 content: '';  
 position: absolute;  
 bottom: 0;  
 left: 0;  
 width: 100%;  
 height: 100%;  
 background-color: rgba(17, 17, 17, 0.35);  
 border-radius: 10rem;  
 z-index: -2;  
}  
.button:before {  
 content: '';  
 position: absolute;  
 bottom: 0;  
 left: 0;  
 width: 0%;  
 height: 100%;  
 background-color: rgba(0, 0, 0, 0.35);  
 transition: all 0.3s;  
 border-radius: 10rem;  
 z-index: -1;  
}  
.button:hover {  
 color: #fff;  
}  
.button:hover:before {  
 width: 100%;  
}  
.botton:focus {  
 border:none;  
 outline: 0**!important**;  
}

index.html

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="utf-8">  
 <title>Plate</title>  
 <link rel="stylesheet" type="text/css" href="normilize.css" />  
 <link rel="stylesheet" type="text/css" href="main.css" />  
<!-- <script src="https://unpkg.com/mathjs@7.0.0/dist/math.min.js"></script>-->  
 <script src="vis-4.21.0/dist/vis.js"></script>  
 <script src="math.min.js"></script>  
 <script src="app.js"></script>  
</head>  
<body onload=" toggleButton( ***document***.getElementsByClassName('btn-container')[0]); init() ">  
 <div>  
 <div class="btn-container" onclick="toggleButton(this)">  
 <div class="bar1"></div>  
 <div class="bar2"></div>  
 <div class="bar3"></div>  
 </div>  
 <div class="side-menu">  
 <div class="floating-label">  
 <input placeholder="Довжина (м)" type="text" id="length" autocomplete="off" onchange="onChange\_a( this.value )">  
 <label for="length">Довжина (м):</label>  
 </div>  
 <div class="floating-label">  
 <input placeholder="Ширина (м)" type="text" id="width" autocomplete="off" onchange="onChange\_b( this.value )">  
 <label for="width">Ширина (м):</label>  
 </div>  
 <div class="floating-label">  
 <input placeholder="Товщина (м)" type="text" id="height" autocomplete="off" onchange="onChange\_h( this.value )">  
 <label for="height">Товщина (м):</label>  
 </div>  
 <div class="floating-label">  
 <input placeholder="модуль Юнга (Па)" type="text" id="ung" autocomplete="off" onchange="onChange\_E( this.value )">  
 <label for="ung">модуль Юнга (Па):</label>  
 </div>  
 <div class="floating-label">  
 <input placeholder="коеф. Пуассона" type="text" id="mu" autocomplete="off" onchange="onChange\_mu( this.value )">  
 <label for="mu">коеф. Пуассона:</label>  
 </div>  
 <div class="floating-label">  
 <input placeholder="Цилінд. жорсткість (Н\*м)" type="text" id="D" autocomplete="off" disabled>  
 <label for="D">Цилінд. жорсткість (Н\*м):</label>  
 </div>  
 <div class="floating-label">  
 <input placeholder="Максимальне навантаження (Н)" type="text" id="q0" autocomplete="off" onchange="onChange\_q0( this.value )">  
 <label for="q0">Максимальне навантаження (Н):</label>  
 </div>  
 <div class="calc-btn-area">  
 <button class="button" onclick="start()" style="outline: 0;">Обчислити</button>  
 </div>  
 </div>  
 <div class="main-area">  
 <h1>Пластинка з защемленими протилежними кінцями</h1>  
 <span>(під гідростатичним навантаженням вздовж осі х)</span>  
 <div id="graph"></div>  
 </div>  
 </div>  
</body>  
</html>

app.js

const ***EPS*** = 1e-8;  
let ***q0*** = 5; // Hydrostatic max pressure  
let ***a*** = 2; // length  
let ***b*** = 1; // width  
let ***h*** = 0.31;  
let ***E*** = 21000;  
let ***mu*** = 0.3;  
let ***D*** = 0; // 1.923; //2.4038461538461538461538461538462e-4;  
  
function calc\_D() {  
 ***D*** = ( ***E*** \* math.pow(***h***, 3) ) / ( 12 \* (1 - math.pow(***mu***, 2)));  
 ***document***.getElementById('D').value = ***D***.toString();  
}  
  
function onChange\_q0( value ) {  
 ***q0*** = parseFloat( value.replace(',', '.') );  
}  
function onChange\_a( value ) {  
 ***a*** = parseFloat( value.replace(',', '.') );  
}  
function onChange\_b( value ) {  
 ***b*** = parseFloat( value.replace(',', '.') );  
}  
function onChange\_h( value ) {  
 ***h*** = parseFloat( value.replace(',', '.') );  
 calc\_D();  
}  
function onChange\_E( value ) {  
 ***E*** = parseFloat( value.replace(',', '.') );  
 calc\_D();  
}  
function onChange\_mu( value ) {  
 ***mu*** = parseFloat( value.replace(',', '.') );  
 calc\_D();  
}  
function preshure(x) {  
 return (***q0*** \* x) / ***a***;  
}  
function alpha(m) {  
 return ((m \* math.pi \* ***b***) / (2 \* ***a***));  
}  
  
function sagging(pos) {  
 const multiplier = (***q0*** \* math.pow(***a***, 4)) / (***D*** \* math.pow(math.pi, 4));  
 let m = 0;  
 let step;  
 let res = 0;  
 do {  
 m++;  
 const alph = alpha(m);  
 step = math.chain(math.pow(-1, m + 1) / math.pow(m, 5))  
 .multiply(2 - (((2 + alph \* math.tanh(alph)) / math.cosh(alph)) \* math.cosh(m \* math.pi \* pos.y / ***a***)) + ((1 / math.cosh(alph) \* (m \* math.pi \* pos.y / ***a***) \* (math.sinh(m \* math.pi \* pos.y / ***a***)))))  
 .multiply(math.sin(m \* math.pi \* pos.x / ***a***))  
 .done();  
 res = math.add(res, step);   
 } while (math.abs(step) > ***EPS***);  
 return (multiplier \* res);  
}  
function calc() {  
 var data = new vis.DataSet();  
  
 return new ***Promise***((resolve, reject) => {  
 for (var x = 0; x <= ***a***; x+=0.1) {  
 if( x === ***a***/2 ) x += 0.1;  
 for (var y = -***b***/2; y <= ***b***/2; y+=0.1) {  
 x = ***Math***.round(x \* 10) / 10;  
 y = ***Math***.round(y \* 10) / 10;  
 let value = sagging({x, y:0.0});  
 value = ( math.abs(value) <= ***EPS*** ) ? 0 : -value;  
 data.add({  
 x: x,  
 y: y,  
 z: value,  
 style: value  
 });  
 }  
 }  
  
 resolve( data );  
 });  
}  
  
// ######################### UI Functions ########################  
  
function toggleButton(btnContainer) {  
 btnContainer.classList.toggle('menu-view-change');  
 ***document***.getElementsByClassName('side-menu')[0].classList.toggle('side-menu-change');  
}  
  
function checkAndBuild() {  
 var options = {  
 width: '100%',  
 height: '100%',  
 style: 'surface',  
 showPerspective: true,  
 showGrid: true,  
 showShadow: false,  
 keepAspectRatio: true,  
 verticalRatio: 0.1  
 };  
  
 calc().then( data => {  
 let gr = new vis.Graph3d(***document***.getElementById('graph'), data, options);  
 });  
}  
  
function init() {  
 ***document***.getElementById('length').value = ***a***.toString();  
 ***document***.getElementById('width').value = ***b***.toString();  
 ***document***.getElementById('height').value = ***h***.toString();  
 ***document***.getElementById('ung').value = ***E***.toString();  
 ***document***.getElementById('mu').value = ***mu***.toString();  
 ***document***.getElementById('q0').value = ***q0***.toString();  
 calc\_D();  
 checkAndBuild();  
}  
  
function start() {  
 if ( ***a*** > 0 && ***b*** > 0 && ***h*** > 0 && ***E*** > 0 && ***mu*** > 0 && ***D*** > 0 ) {  
 let r = ***Math***.round(***h***/***a*** \* 10000) / 10000;  
 if ( (1/80 <= r) && (r <= 1/5) ) {  
 toggleButton( ***document***.getElementsByClassName('btn-container')[0]);  
 init();  
 } else {  
 alert( 'Пластника не є тонкою! (1/80 <= h/a <= 1/5) ' );  
 }  
 } else {  
 alert( 'Ви допустили помилку при введені вхідних параметрів!' );  
 }  
}