

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Факультет математики, фізики та інформаційних технологій

(повне найменування інституту/факультету)

Кафедра механіки, автоматизації та інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «магістр»

«Інформаційна технологія оцінювання критичного діаметру
теплоізоляції багатoshарових стін»

«Information technology for evaluating the critical diameter of thermal
insulation of multilayer walls»

Виконав: здобувач денної форми навчання

спеціальності 126 Інформаційні системи та технології

Освітня програма Інформаційні системи та технології

Григор'єв Вадим Віталійович

(прізвище, ім'я, по-батькові здобувача)

Керівник к.ф.-м.н., доцент Рачинська А.Л.

(науковий ступінь, вчене звання, прізвище та ініціали, підпис)

Рецензент д.т.н., професор Волков В.Е.

(науковий ступінь, вчене звання, прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ _____ від _____ 2023 р.

Захищено на засіданні ЕК №

протокол № _____ від _____ 2023 р.

Оцінка _____ / _____ / _____
(за національною шкалою, шкалою ECTS, бали)

Завідувачка
кафедри

Голова ЕК

Алла РАЧИНСЬКА

(підпис)

Володимир ВИЧУЖАНІН

(підпис)

Одеса – 2023

АНОТАЦІЯ

У дипломній роботі розробляється тема «Інформаційна технологія оцінювання критичного діаметру теплоізоляції багатошарових стін».

Метою роботи є знаходження температурного поля багатошарової конструкції та діаметру теплоізоляції циліндричної труби.

У роботі представлено розроблену інформаційну технологію, що дозволяє фахівцю проводити розрахунок температурного поля для багатошарових циліндричних труб. При проведенні розрахунків немає необхідності задавати характеристики матеріалів труб та ізоляції, оскільки всі дані зберігаються у БД ІТ. Після проведення розрахунків усі побудовані розрахункові моделі та отримані числові моделі результатів так само зберігаються у БД. Це дозволяє в разі потреби не проводити повторний розрахунок, а використовувати дані, що вже є.

Для аналізу результатів розроблений інструментарій дозволяє будувати графіки зміни температури багатошарової стінки. При цьому вибір необхідних розрахункових моделей користувач здійснює самостійно з даних БД ІТ. Для візуальної наочності розрахунків числової моделі розроблено алгоритм побудови кольорового температурного поля згідно з проведеними розрахунками.

Особливістю запропонованої інформаційної технології є те, що її функціонал дозволяє працювати з інструментарієм трьом користувачам із різним рівнем доступу: адміністратору БД, інженеру побудови математичних моделей та фахівцеві аналізу числових моделей.

ABSTRACT

The topic "Information technology for evaluating the critical diameter of thermal insulation of multilayer walls" is developed in the thesis.

The purpose of the work is to find the temperature field of the multilayer structure and the diameter of the thermal insulation of the cylindrical pipe.

The work presents the developed information technology that allows a specialist to calculate the temperature field for multilayer cylindrical pipes. When performing calculations, there is no need to specify the characteristics of pipe and insulation materials, since all data is stored in the IT database. After the calculations, all the built calculation models and the received numerical models of the results are also stored in the database. This allows you not to recalculate if necessary, but to use the data that is already available.

To analyze the results, the developed toolkit allows you to plot temperature changes of a multilayer wall. At the same time, the user selects the necessary calculation models independently from IT database data. For the visual clarity of the calculations of the numerical model, an algorithm for constructing the color temperature field according to the calculations has been developed.

A feature of the proposed information technology is that its functionality allows three users with different levels of access to work with the toolkit: a database administrator, a mathematical model construction engineer, and a numerical model analysis specialist.

ЗМІСТ

	стор.
ВСТУП	5
1 МЕТОДИ ДОСЛІДЖЕННЯ ТЕПЛОПРОВІДНОСТІ ПОВЕРХНОСТЕЙ	7
1.1 Огляд літератури по темі дипломної роботи	7
1.2 Критичний діаметр ізоляції.....	17
2 ПРОЕКТУВАННЯ СИСТЕМИ.....	21
2.1 Архітектура системи.....	21
2.2 Інформаційне моделювання.....	22
2.3 Визначення користувачів інструментарію системи	24
2.4 Проектування візуальної моделі температурного поля	25
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДОСЛІДЖЕННЯ	27
3.1 Реалізація функціоналу модулів системи.....	27
3.2 Реалізація роботи з БД.....	33
3.3 Візуалізація температурного поля багатошарової конструкції.....	35
4 РОБОТА СИСТЕМИ ДОСЛІДЖЕННЯ.....	38
4.1 Тестування роботи	38
4.2 Аналіз результатів.....	45
ВИСНОВОК.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТОК А. ІНТЕРФЕЙСНИЙ МОДУЛЬ	52
ДОДАТОК Б. РОЗРАХУНКОВИЙ МОДУЛЬ	53
ДОДАТОК В. МОДУЛЬ АДМІНІСТРАТОРА БД	58
ДОДАТОК Г. МОДУЛЬ ВІЗУАЛІЗАЦІЇ ТЕМПЕРАТУРНОГО ПОЛЯ	60
ДОДАТОК Д. МОДУЛЬ АНАЛІЗУ ДАНИХ.....	62
ДОДАТОК Ж. ДОВІДКА ПРО ВПРОВАДЖЕННЯ	65

ВСТУП

Актуальність теми. Розрахунок процесів теплопровідності є одним із важливих розділів сучасних інженерних досліджень у технологічних процесах. Широке застосування у всіх галузях промисловості та господарської діяльності людини має теплова ізоляція. Її правильне застосування значною мірою визначає можливість підтримки необхідних технологічних процесів і зрештою робить вирішальний вплив на показники енергоефективності та експлуатації.

В умовах сьогодення проблема ізоляції поверхонь є дуже актуальною для широких кіл населення України. Всім відомо, що ізоляція поверхні дозволяє зменшити тепловіддачу за рахунок використання матеріалів із низькою теплопровідністю. У разі відсутності постійного опалення дуже актуальним є питання теплоізоляції циліндричних труб. Питання підбору ізоляції та розробки конструктиву потребує комплексного та професійного підходу. Для розрахунку задачі теплоізоляції фахівцю необхідно мати інструментарій для проведення розрахунку температурного поля із збереженням отриманих результатів для подальшого аналізу.

Зростання вартості енергії та нові вимоги сьогодення до теплового захисту існуючих будівель та теплотрас визначає необхідність використання інформаційних технологій розрахунку ізоляції багатошарових конструкцій.

В дипломній роботі досліджується теплоізоляція багатошарових циліндричних труб.

Зв'язок роботи з науковими програмами, планами, темами. Тематика кваліфікаційної роботи включена до плану науково-дослідних робіт кафедри механіки, автоматизації та інформаційних технологій Одеського національного університету імені І.І. Мечникова на 2023–2027 роки «Теорія і практика застосування математичного моделювання та інформаційних технологій в технологічному та організаційному керуванні» (номер держ. реєстрації 0123U103362).

Ціль та задачі дослідження. Метою роботи є знаходження температурного поля багат шарової конструкції та діаметру теплоізоляції циліндричної труби.

Для досягнення поставленою мети необхідно розв'язати наступні задачі:

1. Проаналізувати існуючі математичні моделі заходження температурного поля.
2. Створити розрахункову модель багат шарової циліндричної труби для граничних умов III-го роду.
3. Розробити технологію розрахунку зазначеної моделі.
4. Розробити інформаційну систему, яка реалізує модель розрахунку температурного поля та діаметру теплоізоляції.
5. Провести аналіз отриманих результатів.

Об'єкт дослідження дипломної роботи – теплопровідність багат шарової поверхні.

Предмет дослідження – критичний діаметр ізоляції багат шарової циліндричної труби.

1 МЕТОДИ ДОСЛІДЖЕННЯ ТЕПЛОПРОВІДНОСТІ ПОВЕРХНОСТЕЙ

1.1 Огляд літератури по темі дипломної роботи

Процеси передачі теплоти (теплообміну) широко поширені в природі і багато в чому визначають практичну діяльність людини. Поняття теплообміну охоплює весь комплекс явищ перенесення теплоти у просторі, зумовлених різницею температур окремих елементів аналізованого середовища. У загальному випадку перенесення теплоти це складний процес, пов'язаний з різноманітними фізичними явищами, проте можна виділити три основні види (механізму) теплообміну: теплопровідність, конвективний теплообмін, теплообмін випромінюванням.

Теплопровідність - молекулярне перенесення теплоти в тілах (або між ними), обумовлений неоднорідністю температури в розглянутому просторі.

Теплопровідність не пов'язана з макрорухом тіл і здійснюється передачею енергії від одних мікрочастинок тіла до інших при їх взаємодії. Для газів, наприклад, такими частинками є молекули. Відомо, що молекули більш нагрітої частини будь-якого газу мають більшу середню кінетичну енергію в порівнянні з молекулами холодної його частини. При зіткненні молекул відбувається обмін кінетичною енергією, у результаті здійснюється передача теплоти від нагрітої частини газу холодної.

Процес теплопровідності в металах аналогічний до процесу електропровідності і пов'язаний з рухом вільних електронів. У найпростішому випадку вважатимуться, що вільні електрони поведуться як молекули газу, тобто переміщаються між атомами та здійснюють передачу теплоти. Таким чином, у газах та металах процес передачі теплоти здійснюється дифузією молекул та вільних електронів відповідно.

У рідинах і твердих тілах-діелектриках перенесення теплової енергії здійснюється пружними хвилями, що виникають в результаті коливань кристалічних ґрат. Зазначимо, що у металах деяка частина енергії також

передається пружними хвилями, проте вона незначна проти дифузійним перенесенням теплоти вільними електронами.

Конвекція – перенесення теплоти при переміщенні обсягів рідини (газу) у просторі. Конвекція можлива лише у поточному середовищі, у своїй перенесення теплоти нерозривно пов'язані з перенесенням самого середовища. У рідинах конвекція завжди супроводжується теплопровідністю.

Конвективний теплообмін - спільне перенесення теплоти конвекцією та теплопровідністю. Теплообмін між рідиною (газом) і поверхнею твердого тіла називають конвективною тепловіддачею або тепловіддачею. Теплопередача – це процес теплообміну між двома теплоносіями, розділеними твердою стінкою.

Теплове випромінювання – процес поширення теплоти електромагнітними хвилями. При такому вигляді передачі теплоти відбувається перетворення внутрішньої енергії речовини на енергію випромінювання, перенесення випромінювання та її поглинання речовиною.

Теплове випромінювання обумовлено лише температурою і оптичними властивостями випромінюючого тіла. Воно підпорядковується основним законам поширення світла, тобто законам відбиття, заломлення та поглинання. У чистому вигляді (тобто без інших видів теплообміну) променистий теплообмін має місце лише в умовах глибокого вакууму, наприклад, між космічним літальним апаратом і навколишнім безповітряним простором.

У процесах, з якими доводиться нам стикатися у природі та техніці, теплообмін випромінюванням супроводжується конвекцією та теплопровідністю. У цьому випадку теплообмін називають складним чи радіаційно-конвективним. Однак комплексне математичне вивчення складного теплообміну утруднене, тому часто попередньо вивчають кожен вид теплообміну окремо, а потім переходять до розрахунку складного теплообміну. Крім того, при вирішенні конкретних завдань один з видів теплопередачі, як правило, переважає, так що кількість теплоти, що

передається іншими видами теплообміну, незначно і можна знехтувати. У цьому інтерес цікавить вивчення кожного з видів теплообміну окремо.

У процесах теплопровідності основною фізичною величиною, характерною для цих процесів, є температура [1-4]. Завдання теорії теплопровідності полягає у знаходженні поля температур об'єкта, що розглядається, тобто у визначенні залежності

$$T = f(\tau, x, y, z), \quad (1.1)$$

де T - температура; τ - час; x, y, z - просторові координати в декартовій системі.

Температура є величина скалярна, тому температурне поле - також скалярна величина. Зазначимо, що наведене вище визначення поля є справедливим і для векторних величин, що показують і величину, і напрямок (сила, швидкість, прискорення тощо). Такі поля називають векторними полями фізичних величин.

Розрізняють стаціонарне та нестаціонарне температурні поля.

Нестаціонарним температурним полем називають поле, температура якого змінюється і просторі, і з часом. У цьому випадку говорять, що температура є функцією простору та часу. Прикладом математичного запису нестаціонарного температурного поля є рівняння (1.1).

Стаціонарним температурним полем називають поле, температура якого у його точці не змінюється з часом, тобто є функцією лише координат

$$T = f_1(x, y, z), \quad \frac{\partial T}{\partial \tau} = 0. \quad (1.2)$$

Теплові режими, що характеризуються нестаціонарними температурними полями, називаються невстановленими [1-4]. У разі коли температурні поля стаціонарні, теплові режими називають встановленими.

Відповідно до числа просторових координат, від яких залежить температура, температурне поле може бути тривимірним (формули (1.1), (1.2)), двовірним

$$T = f_2(x, y, \tau), \quad \frac{\partial T}{\partial z} = 0$$

та одновимірними

$$T = f_3(x, \tau), \quad \frac{\partial T}{\partial y} = 0, \quad \frac{\partial T}{\partial z} = 0.$$

Розглянемо багат шарове тіло простої форми, що складається з однорідних шарів, між якими має місце ідеальний тепловий контакт. Будемо вважати, що температура тіла, що розглядається, змінюється вздовж однієї координати, а теплопровідність λ_i ($i = 1, 2, \dots, m$) кожного із шарів постійна. В цьому випадку тепловий потік, що проходить через будь-яку ізотермічну поверхню тіла, однаковий: $Q = Q_i = \text{const}$.

Нехай на поверхнях, що обмежують тіло, задані граничні умови III-го роду (випадок граничних умов I-го роду буде отриманий з кінцевих формул граничним переходом при $\alpha_1 \rightarrow \infty$ та $\alpha_2 \rightarrow \infty$), а температури зовнішніх середовищ T_{C_1} та T_{C_2} підпорядковуються нерівності $T_{C_1} > T_{C_2}$.

Позначимо температури граничних шарів T_{S_1} . Розміри кожного шару вважаємо заданими. Вочевидь, температури зовнішніх поверхонь тіла мають індекси «1» і «m+1».

Через постійність теплових потоків на ізотермічних поверхнях для зовнішніх меж тіла можна записати

$$Q = \alpha_1 S_1 (T_{C_1} - T_{S_1}), \quad Q = \alpha_2 S_{m+1} (T_{S_{m+1}} - T_{C_2}). \quad (1.2)$$

Такий же тепловий потік передається через кожен із шарів багат шарового тіла. Тоді відповідно до закону Фур'є отримаємо

$$Q = \frac{\lambda_i (T_{S_i} - T_{S_{i+1}})}{\int_{n_i}^{n_{i+1}} \frac{dn}{S(n)}}, \quad i = 1, 2, \dots, m. \quad (1.3)$$

Позначимо

$$\int_{n_i}^{n_{i+1}} \frac{dn}{S(n)} = N_{n_i}^{n_{i+1}} = F_i \quad (1.4)$$

та об'єднаємо рівності (1.2) та (1.3) у систему рівнянь

$$\left\{ \begin{array}{l} Q = \alpha_1 S_1 (T_{C_1} - T_{S_1}), \\ Q = \lambda_1 \frac{(T_{S_1} - T_{S_2})}{F_1}, \\ Q = \lambda_2 \frac{(T_{S_2} - T_{S_3})}{F_2}, \\ \dots \\ Q = \lambda_m \frac{T_{S_m} - T_{S_{m+1}}}{F_m}, \\ Q = \alpha_2 S_{m+1} (T_{S_{m+1}} - T_{C_2}), \end{array} \right. \quad \text{або} \quad \left\{ \begin{array}{l} T_{C_1} - T_{S_1} = \frac{Q}{\alpha_1 S_1}, \\ T_{S_1} - T_{S_2} = \frac{F_1}{\lambda_1} Q, \\ T_{S_2} - T_{S_3} = \frac{F_2}{\lambda_2} Q, \\ \dots \\ T_{S_m} - T_{S_{m+1}} = \frac{F_m}{\lambda_m} Q, \\ T_{S_{m+1}} - T_{C_2} = \frac{Q}{\alpha_2 S_{m+1}}. \end{array} \right. \quad (1.5)$$

Складемо почленно праві і ліві частини рівнянь (1.5) і отримаємо наступний вираз для теплового потоку:

$$Q = \eta_* \cdot (T_{C_1} - T_{C_2}), \quad \eta_* = \left(\frac{1}{\alpha_1 S_1} + \sum_{i=1}^m \frac{F_i}{\lambda_i} + \frac{1}{\alpha_2 S_{m+1}} \right)^{-1}. \quad (1.6)$$

Величину η_* називають повним коефіцієнтом теплопередачі багат шарової стінки, а величину $R_* = 1/\eta_*$ - відповідно повним термічним опором теплопередачі багат шарової стінки. Вираз (1.6) для теплового потоку Q через багат шарову стінку подібно до виразу для одно шарової стінки. Для зручності порівняння процесів теплопровідності в багат шаровій та однорідній стінках вводять еквівалентну теплопровідність багат шарової стінки

$$\lambda_{\text{эк}} = \sum_{i=1}^m F_i / \sum_{i=1}^m \frac{F_i}{\lambda_i}. \quad (1.7)$$

Використовуючи це позначення, рівність (1.6) можна записати у вигляді, що формально збігається з виразом для однорідної стінки:

$$Q = (T_{C_1} - T_{C_2}) / \left(\frac{1}{\alpha_1 S_1} + \frac{N_{n_1}^{n_{m+1}}}{\lambda_{\text{эк}}} + \frac{1}{\alpha_2 S_{m+1}} \right).$$

Однак коефіцієнт $\lambda_{\text{эк}}$ тут вже не є константою, а залежить, як це впливає з (1.7), від числа шарів, їх теплопровідностей та розмірів, а також від геометричної форми багат шарової стінки. Вирішуючи систему рівнянь (1.5), отримуємо, що температури на межах шарів S_i та на зовнішніх поверхнях стінки S_1 і S_{m+1} визначаються виразами:

$$\begin{aligned} T_{S_1} &= T_{C_1} - Q \frac{1}{\alpha_1 S_1}, \\ T_{S_2} &= T_{C_1} - Q \left[\frac{1}{\alpha_1 S_1} + \frac{F_1}{\lambda_1} \right], \\ &\dots \end{aligned}$$

$$\begin{aligned}
T_{S_{i+1}} &= T_{C_1} - Q \left[\frac{1}{\alpha_1 S_1} + \sum_{k=1}^i \frac{F_k}{\lambda_k} \right], \\
&\dots \\
T_{S_m} &= T_{C_1} - Q \left[\frac{1}{\alpha_1 S_1} + \sum_{k=1}^{m-1} \frac{F_k}{\lambda_k} \right], \\
T_{S_{m+1}} &= T_{C_2} + Q \frac{1}{\alpha_2 S_{m+1}}.
\end{aligned} \tag{1.8}$$

Таким чином, вихідне завдання вирішено.

Розглянемо тепер випадок, коли на зовнішніх поверхнях тіла задані умови першого роду [1-4]. Для цього достатньо у всіх отриманих вище формулах здійснити граничний перехід при $\alpha_1 \rightarrow \infty$ та $\alpha_2 \rightarrow \infty$. Тоді з першого та останнього рівнянь системи (1.8) отримуємо

$$T_{S_1} = T_{C_1}, \quad T_{S_2} = T_{C_2},$$

тобто температури поверхонь S_1 та S_{m+1} відомі та рівні відповідно температурам зовнішніх середовищ. Рівняння для теплового потоку (1.6) набуде вигляду

$$Q = (T_{C_1} - T_{C_2}) / \sum_{i=1}^m \frac{F_i}{\lambda_i}, \tag{1.9}$$

а температури меж шарів можна визначити з рівності

$$T_{S_{i+1}} = T_{C_1} - Q \cdot \sum_{k=1}^i \frac{F_k}{\lambda_k}, \quad i = 1, 2, \dots, m. \tag{1.10}$$

З рівності (1.9) видно, що у разі граничних умов першого роду повний термічний опір $\sum_{i=1}^m (F_i/\lambda_i)$ багат шарова стінка складається з термічних опорів F_i/λ_i кожного i -го шару.

Розглянемо інший випадок завдання граничних умов багат шарової стінки. Нехай на одній поверхні стінки, наприклад внутрішньої, задані граничні умови II-го роду у вигляді $q|_{S_1} = q_0$ (постійна щільність теплового потоку). На іншій - задані граничні умови III-го роду: коефіцієнт тепловіддачі α_2 та температура середовища T_{C_2} (конвективний теплообмін із зовнішнім рухомим середовищем). Тоді вирази для теплового потоку через стінку і температури її шарів можуть бути отримані аналогічним чином. Наведемо остаточні формули:

$$\begin{aligned} Q &= q_0 S_1, \quad T_{S_{m+1}} = T_{C_2} + \frac{q_0}{\alpha_2} \frac{S_1}{S_{m+1}}, \\ T_{S_1} &= T_{C_2} + q_0 S_1 \left(\sum_{i=1}^m \frac{F_i}{\lambda_i} + \frac{1}{\alpha_2 S_{m+1}} \right), \\ T_{S_i} &= T_{C_2} + q_0 S_1 \left(\sum_{k=i}^m \frac{F_k}{\lambda_k} + \frac{1}{\alpha_2 S_{m+1}} \right). \end{aligned}$$

Досліджуємо нижче окремі випадки отриманих рішень для плоскої, циліндричної та кульової багат шарових стінок відповідно.

Нехай задані товщини δ_i та теплопровідності λ_i шарів плоскої стінки, на зовнішніх поверхнях якої виконуються умови III-го роду. Тому що для плоскої стінки $S(n) = S = \text{const}$, то урахуванням (1.4) маємо $F_i = \delta_i/S$ і вирази (1.6) - (1.8) набудуть вигляду

$$Q = S \frac{T_{C_1} - T_{C_2}}{\frac{1}{\alpha_1} + \sum_i \frac{\delta_i}{\lambda_i} + \frac{1}{\alpha_2}} = S \frac{T_{C_1} - T_{C_2}}{\frac{1}{\alpha_1} + \frac{\delta}{\lambda_{\text{эк}}} + \frac{1}{\alpha_2}} = S \frac{T_{C_1} - T_{C_2}}{R},$$

$$T_{S_{i+1}} = T_{C_1} - \frac{Q}{S} \left(\frac{1}{\alpha_1} + \sum_{k=1}^i \frac{\delta_k}{\lambda_k} \right), \quad \lambda_{\text{эк}} = \frac{\delta}{\sum_{i=1}^m \frac{\delta_i}{\lambda_i}},$$

де $\delta = \sum_{i=1}^m \delta_i$ - товщина стінки, $R = \frac{1}{\alpha_1} + \frac{\delta}{\lambda_{\text{эк}}} + \frac{1}{\alpha_2}$ - повний термічний опір плоскої багат шарової стінки. На прикладі плоскої стінки видно, що введення еквівалентної теплопровідності $\lambda_{\text{эк}}$ рівносильно заміні багат шарової стінки деякою фіктивною одно шарової стінкою тієї ж товщини, причому різниці температур $T_{C_1} - T_{C_2}$ на межах одно шарової та багат шарової стінок, а також кількість теплоти Q , яке проходить через них у одиницю часу, збігаються.

Для циліндричної стінки маємо:

$$F_i = \frac{1}{2\pi L} \ln \left(\frac{r_{i+1}}{r_i} \right), \quad S_i = 2\pi r_i L = \pi L d_i^2.$$

Підставляючи ці значення до рівності (1.6)-(1.8), отримаємо такі розрахункові формули:

- для потоку теплоти, що проходить через одиницю довжини циліндричної стінки:

$$q_L = \frac{Q}{L} = \pi \tilde{\eta}_z (T_{C_1} - T_{C_2}) = \frac{1}{\tilde{R}_z} \pi (T_{C_1} - T_{C_2}), \quad (1.11)$$

де $\tilde{\eta}_z$ - коефіцієнт теплопередачі багат шарової циліндричної стінки, а

$$\tilde{R}_z = \frac{1}{\tilde{\eta}_z} = \frac{1}{\alpha_1 d_1} + \sum_{i=1}^m \frac{1}{2\lambda_i} \ln \left(\frac{d_{i+1}}{d_i} \right) + \frac{1}{\alpha_2 d_{m+1}} \quad (1.12)$$

повний термічний опір теплопередачі багатошарової циліндричної стінки;

- для температур на границях шарів:

$$T_{S_{i+1}} = T_{C_1} - \frac{q_L}{\pi} \left(\frac{1}{\alpha_1 d_1} + \sum_{k=1}^i \frac{1}{2\lambda_k} \ln \left(\frac{d_{k+1}}{d_k} \right) \right), \quad i = 1, 2, \dots, m; \quad (1.13)$$

- для температури на зовнішній границі S_{m+1} циліндричної стіни

$$T_{S_{m+1}} = T_{C_1} - \frac{q_L}{\pi} \left(\frac{1}{\alpha_1 d_1} + \frac{1}{2\lambda_{ек}} \ln \left(\frac{d_{m+1}}{d_1} \right) \right), \quad (1.14)$$

де $\lambda_{ек} = \ln \left(\frac{d_{m+1}}{d_1} \right) / \sum_{i=1}^m \frac{1}{\lambda_i} \ln \left(\frac{d_{i+1}}{d_i} \right)$ - еквівалентна теплопровідність

багатошарової циліндричної стінки.

Кульова стінка. Для кульової стінки знаходимо:

$$F_i = \frac{1}{4\pi} \left(\frac{1}{r_i} - \frac{1}{r_{i+1}} \right), \quad S_i = 4\pi r_i^2 = \pi d_i^2.$$

Відповідно до рівності (1.6) тепловий потік Q через довільну ізотермічну поверхню (сферу) може бути визначено за формулою

$$Q = \frac{\pi(T_{C_1} - T_{C_2})}{\frac{1}{\alpha_1 d_1^2} + \frac{1}{2} \sum_{i=1}^m \frac{1}{\lambda_i} \left(\frac{1}{d_i} - \frac{1}{d_{i+1}} \right) + \frac{1}{\alpha_2 d_{m+1}^2}} = \tilde{\eta}_b \pi (T_{C_1} - T_{C_2}),$$

де $\tilde{\eta}_b = \left[\frac{1}{\alpha_1 d_1^2} + \frac{1}{2} \sum_{i=1}^m \frac{1}{\lambda_i} \left(\frac{1}{d_i} - \frac{1}{d_{i+1}} \right) + \frac{1}{\alpha_2 d_{m+1}^2} \right]^{-1}$ - повний коефіцієнт

теплопередачі багатошарової кульової стінки, а $\tilde{R}_b = 1/\tilde{\eta}_b$ - повний термічний

опір теплопередачі багатошарової кульової стінки. Температура на межі i -го та $(i+1)$ -го шарів визначається з виразу

$$T_{S_{i+1}} = T_{C_1} - \frac{Q}{\pi} \left(\frac{1}{\alpha_1 d_1^2} + \frac{1}{2} \sum_{k=1}^i \frac{1}{\lambda_k} \left(\frac{1}{d_k} - \frac{1}{d_{k+1}} \right) \right).$$

1.2 Критичний діаметр ізоляції

Вивчимо, як впливає зміна зовнішнього діаметра d_2 однорідної циліндричної стінки на її термічний опір $\tilde{R}_z$.

Згідно з формулою (1.11) маємо

$$\tilde{R}_z = \frac{1}{\alpha_1 d_1} + \frac{1}{2\lambda} \ln \left(\frac{d_2}{d_1} \right) + \frac{1}{\alpha_2 d_2}.$$

Якщо α_1 , d_1 , λ , α_2 зафіксовані, то $\tilde{R}_z$ стає функцією лише зовнішнього діаметра d_2 , т.е. $\tilde{R}_z = \tilde{R}_z(d_2)$. Така постановка завдання є цілком закономірною, оскільки значення α_1 , α_2 , λ повинні бути задані при розрахунку будь-якого теплообмінного апарату, а величина d_1 визначається вимогам до пропускної здатності трубопроводу.

Диференціюючи функцію $\tilde{R}_z = \tilde{R}_z(d_2)$ і прирівнюючи результат до нуля, отримуємо

$$\frac{d(\tilde{R}_z)}{d(d_2)} = \frac{1}{2\lambda d_2} - \frac{1}{\alpha_2 d_2^2} = 0,$$

що має місце при $d_2 = \frac{2\lambda}{\alpha_2} = d_*$.

Величину d_* називають критичним діаметром. Обчислимо другу похідну від функції $\tilde{R}_z = \tilde{R}_z(d_2)$ в точці d_* :

$$\left. \frac{d^2(\tilde{R}_z)}{d(d_2)^2} \right|_{d_2=d_*} = \frac{\alpha_2^2}{8\lambda^3} > 0.$$

Оскільки друга похідна позитивна, то критичний діаметр d_* відповідає мінімальному термічному опору $\tilde{R}_z$. Отримана формула може бути використана для найпростішого розрахунку теплоізоляційного товщини покриття. Покажемо це.

Теплова ізоляція являє собою шар спеціального матеріалу, що наноситься на поверхню тіла з метою збільшення його повного термічного опору. Теплоізоляційні покриття застосовуються у двох випадках:

- коли необхідно захистити конструкцію від надто високих теплових навантажень;
- для зменшення теплових втрат у довкілля.

Як тепла ізоляція застосовують матеріали, що мають малий коефіцієнт теплопровідності. З матеріалів органічного походження використовуються шерсть, бавовна, деревина. З неорганічних – азбест, шлак, глина, пісок. У суднобудуванні широко використовуються пробка, склоповсть, скляна та шлакова вата, пінопласти, пінополіуретани.

Досліджуємо вплив матеріалу та товщини зовнішнього діаметра ізоляції на повний лінійний термічний опір та теплові втрати теплоізованого трубопроводу. Повний термічний опір циліндричної труби, покритої шаром теплоізоляції, визначається за формулою

$$\tilde{R}_z = \frac{1}{\alpha_1 d_1} + \frac{1}{2\lambda} \ln\left(\frac{d_2}{d_1}\right) + \frac{1}{2\lambda_h} \ln\left(\frac{d_h}{d_2}\right) + \frac{1}{\alpha_2 d_h}.$$

Вважатимемо заданими внутрішній d_1 та зовнішній d_2 діаметри труби, коефіцієнти теплопровідності матеріалів труби λ та ізоляція λ_h . Критичний діаметр ізоляції d_{h*} визначаємо з умови:

$$\frac{d(\tilde{R}_z)}{d(d_h)} = 0$$

і після очевидних перетворень отримаємо

$$d_{h*} = \frac{2\lambda_h}{\alpha_2}.$$

Таким чином, d_{h*} , а отже, і товщина шару ізоляції $[d_{h*} - d_2]$ залежать лише від теплопровідності ізоляції λ_h та коефіцієнта тепловіддачі α_2 від зовнішньої поверхні ізоляції до довкілля.

З'ясуємо тепер, як залежить теплові втрати трубопроводу від товщини ізоляції. $[d_{h*} - d_2]$ та її теплопровідності λ_h . Якщо $d_2 < d_h < d_{h*}$, то зі збільшенням товщини ізоляції до значення d_{h*} теплові втрати зростають, перевищуючи втрати неізольованої труби. При подальшому зростанні товщини ізоляції ($d_h > d_{h*}$) теплові втрати зменшуватимуться.

Якщо $d_2 > d_{h*}$, то увесь час $d_h > d_{h*}$. І тут ізоляційне покриття будь-якої товщини забезпечить зменшення теплових втрат. Досягти виконання останньої рівності можна шляхом варіювання різних теплоізоляційних матеріалів, тобто за рахунок підбору λ_h .

Визначимо d_{h*} для трубопроводу, покритого різними теплоізоляційними матеріалами при $\alpha_2 = 10$ (Вт/(м²·К)):

- для бетону ($\lambda_h = 1,28$ (Вт/(м·К))) $d_{h*} = 2\lambda_h/\alpha_2 = 2 \cdot 1,28/10 = 0,256$ м,
- для азбесту ($\lambda_h = 0,11$ (Вт/(м·К))) $d_{h*} = 2\lambda_h/\alpha_2 = 2 \cdot 0,11/10 = 0,022$ м,

- для совелиту ($\lambda_h = 0,098 \text{ (Вт / (м} \cdot \text{К))}$) $d_{h*} = 2\lambda_h / \alpha_2 = 2 \cdot 0,098 / 10 = 0,0196 \text{ м}$.

Таким чином, бетонна ізоляція ефективна для ізольованих труб із зовнішнім діаметром, що перевищує 256 мм, при азбестовому покритті зовнішній діаметр повинен перевищувати 22 мм, для солітової ізоляції - 19,6 мм.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Архітектура проекту

В роботі побудовано інформаційну технологію [5-6] розрахунку температурного поля циліндричної багатошарової стінки з архітектурним рішенням виду:

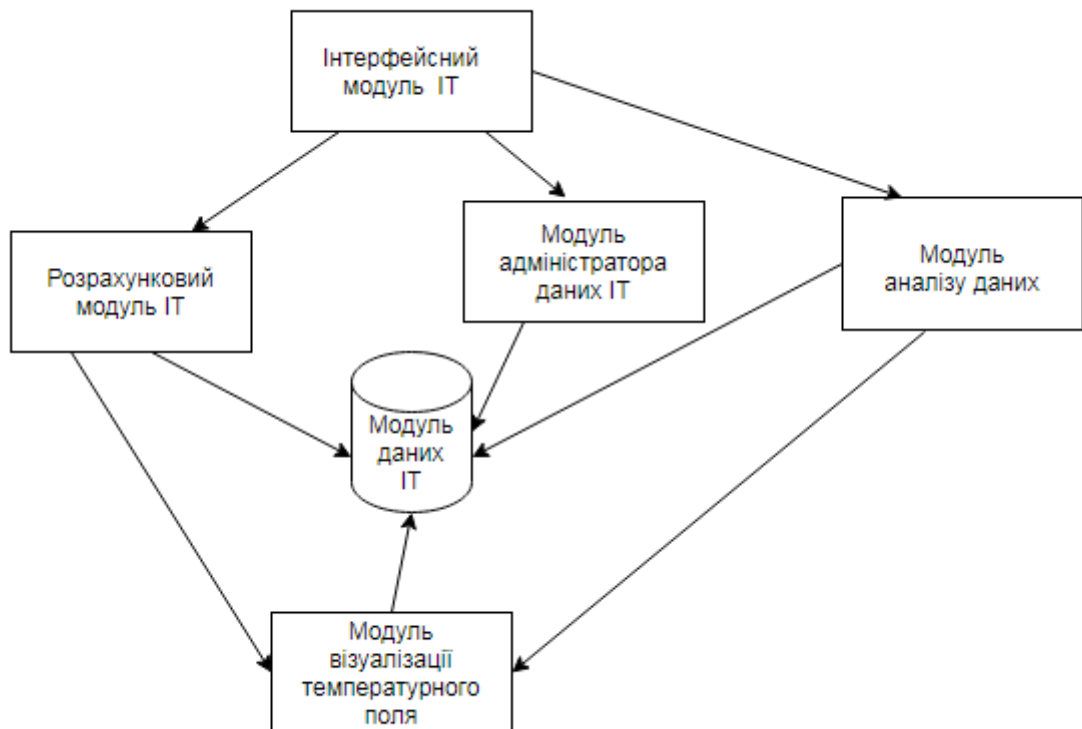


Рисунок 2.1. Архітектура інформаційної технології розрахунку

Інтерфейсний модуль забезпечую взаємодію з користувачем за допомогою Winows Forms Application [7]. Розрахунковий модуль відповідає за чисельний розрахунок математичної моделі згідно математичної моделі (1.11) - (1.14). Модуль даних – це БД, яка спроектована за допомогою об'єктно-реляційної системи управління базами даних – PostgreSQL [8-9]. Модуль адміністратора БД - Winows Forms Application на об'єктно-орієнтованій мові програмування C# за допомогою інтегрованого середовища розробки Microsoft Visual Studio [7-9]. Модуль візуалізації

температурного поля - візуальна наочність розрахунків числової моделі з використанням пакету засобів розробки тривимірної графіки DirectX. Модуль аналізу даних має функціонал для аналізу числових моделей, що зберігаються в БД.

Мовою розробки ПЗ системи обрано - C#, так як ця мова була розроблена Microsoft і є основною мовою для платформи .NET. Завдяки своїй простоті і зрозумілості мова C# дозволяє швидко розробляти програми. В мові є потужна та ефективна збірка сміття. C# автоматично звільняє виділену пам'ять для об'єктів після того, як вони не використовуються. C# - це повністю сучасне об'єктно-орієнтоване програмування.

Для створення БД обрано PostgreSQL, бо не просто реляційна, а об'єктно-реляційна СУБД. Фундаментальна характеристика об'єктно-реляційної бази даних — це підтримка об'єктів користувача та їх поведінки, включаючи типи даних, функції, операції, домени та індекси. Це робить PostgreSQL неймовірно гнучким та надійним. Серед іншого, він вміє створювати, зберігати та отримувати складні структури даних.

У роботі використовується пакет DirectX. Це набір компонентів у ОС Windows, який дозволяє програмному забезпеченню безпосередньо взаємодіяти з відеобладнанням.

2.2 Інформаційне моделювання

Перший етап проектування полягає у створенні схеми БД, яка включає визначення сутностей та існуючих між ними зв'язків та атрибутів. Результатом даного проектування є схема БД, представлена у вигляді ER-діаграми (Entity-Relationship), на якій відображаються основні сутності предметної області, їх атрибути та зв'язки. Схема БД створюється з урахуванням функціональних вимог користувачів і не залежить від будь-яких особливостей фізичної реалізації інформаційної системи, як-от тип обраної СУБД чи мову програмування.

Послідовність етапів створення схеми бази даних:

- 1) визначення сутностей;
- 2) визначення атрибутів сутностей;
- 3) завдання первинних та зовнішніх ключів.

З аналізу предметної області, для функціонування інформаційної системи необхідні такі сутності (табл.1):

Таблиця 1- Опис сутностей

Ім'я атрибуту	Опис	Тип	Обмеження
Матеріал - material			
id	Id матеріалу	цілий, унікальний	первинний ключ
name	назва матеріалу	символьний (30)	за замовченням пустий рядок
koefteploizol	Коефіцієнт теплопровідності матеріалу	десятькове число	за замовченням -1
Модель - model			
id	Id моделі	цілий, унікальний	первинний ключ
tvn	Температура внутрішнього шару	десятькове число	не пусте
tzov	Температура зовнішнього шару	десятькове число	не пусте
alfavn	Коефіцієнт тепловіддачі внутрішнього шару з середовищем	десятькове число	не пусте
alfazov	Коефіцієнт тепловіддачі зовнішнього шару з середовищем	десятькове число	не пусте
n	Кількість шарів моделі	ціле число	не пусте
Шар - shar			
id	Id шару	цілий, унікальний	первинний ключ
d	діаметр шару	ціле число	не пусте
idmaterial	матеріал	цілий, унікальний	зовнішній ключ
Розрахункова модель - modelsharsculc			
id	Id розрахунку	цілий, унікальний	первинний ключ
idmodel	модель	цілий, унікальний	зовнішній ключ
idshar	шар	цілий, унікальний	зовнішній ключ
t	температура	десятькове число	не пусте

Схема побудованої бази даних представлена на рис. 2.1

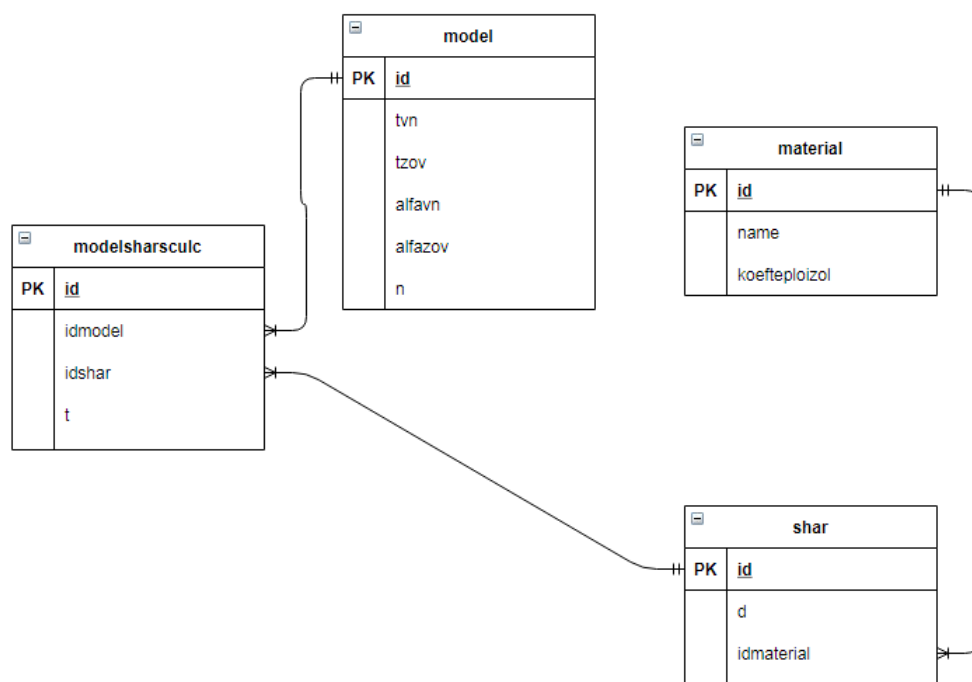


Рисунок 2.1 – Схема бази даних

Ключовою є таблиця **modelsharsculc**, яка містить в собі розрахункові моделі кваліфікаційної роботи, що складаються з математичної моделі таблиці **model** та шарів різного матеріалу з таблиці **shar**.

2.3 Визначення користувачів інструментарію системи

Інструментарій системи, що проектується призначений для користувачів трьох ролей: адміністратора баз даних, інженер побудови математичних моделей та фахівець аналізу числових моделей (рис. 2.2).

Адміністратор баз даних несе відповідальність за внесення даних про матеріали, які можуть бути використаними організацією для проведення інженерних робіт по теплоізоляції циліндричних труб. Основна характеристика матеріалу – це коефіцієнт теплопровідності. Доступ адміністратору БД додається тільки після введення паролю.

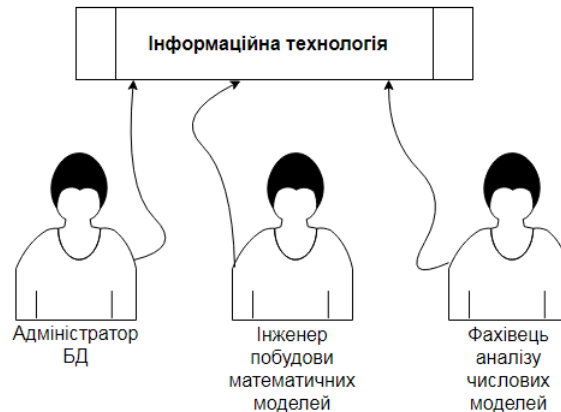


Рисунок 2.2 – Користувачі ІТ

Інженер побудови математичних моделей відповідає за введення коректних параметрів математичної моделі розрахунку: температуру та коефіцієнт тепловіддачі зовнішнього шару, температуру та коефіцієнт тепловіддачі внутрішнього шару, кількість шарів теплоізоляції, діаметр та матеріал кожного шару.

Фахівець аналізу числових моделей може аналізувати моделі, що зберігаються в БД під час розрахунків. Він має функціонал вибору необхідної моделі, візуалізації температурного поля числової моделі та інструмент для аналізу даних по шарам багатошарової стінки.

2.4 Проектування візуальної моделі температурного поля

Для побудови візуальної моделі температурного поля обирається пакет DirectX. Основним об'єктом, що використовується в тривимірній графіці, є трикутник, він є елементарним полігоном при побудові зображень. З достатньою кількістю трикутників можна уявити і відобразити будь-що, навіть гладкі вигнуті поверхні.

З точки зору математики багатошарова циліндрична конструкція у поперечному перетині – це кільце, де внутрішній розмір визначається внутрішнім шаром, а зовнішній розмір – розміром останнього шару. Температурне поле – це частина простору, що належить кільцю. Якщо для

найвищої температури поля вибирається – червоний колір, а найнижчої температури – синій, то необхідно зафарбувати кільце відповідними кольорами.

Кожний шар конструкції будемо розбивати на трикутники, дві вершини якого лежать на одній границі, а третя на іншій. Щоб спростити процедуру, малюватимемо трикутник, використовуючи екранні координати. Для цього нам необхідно, по-перше, сформувати дані, що включають інформацію про трикутник, по-друге, дати команду пристрою для виконання процедури малювання, тобто викликати метод малювання. DirectX має вбудовану конструкцію для зберігання та обробки інформації про зазначені трикутники. У складі імен Direct3D існує клас вершин CustomVertex, який розміщує різні зборки вершин у форматі CustomVertex, що використовуються в Direct3D. Структура вершинного формату містить дані у зрозумілому для Direct3D вигляді.

Будемо використовувати для наших трикутників структуру TransformedColored — перетворення кольорів. Ця структура повідомляє Direct3D runtime, що наші трикутники не вимагають будь-яких перетворень, наприклад, обертання або переміщення, оскільки ми визначатимемо або встановлюватимемо їх положення за допомогою екранних координат. Ця структура включає параметри кольору для кожної вершини трикутників. Далі ми встановлюємо колір кожної вершини. Слід звернути увагу на те, що ми повинні викликати метод завдання стандартних кольорів ToArgb(), що використовується для цього. У додатку Direct3D мається на увазі, що кольори будуть визначатися 32-розрядними цілими числами, і цей алгоритм перетворить базовий колір на цей формат.

Таким чином, знаючи температуру на кожному шару конструкції ми за допомогою TransformedColored будемо мати трикутники з плавним переходом від червоного кольору до синього.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДОСЛІДЖЕННЯ

3.1 Реалізація функціоналу модулів системи

Інтерфейсний модуль складається з одного класу FMain, який має п'ять полів та шість методів (див. Додаток А). Інтерфейсний модуль забезпечує доступ до системи користувачем трьох ролей. Функціонал кнопки bAdmin - адміністратору баз даних, кнопки bCulc - інженеру побудови математичних моделей та button1 - фахівцю аналізу числових моделей.

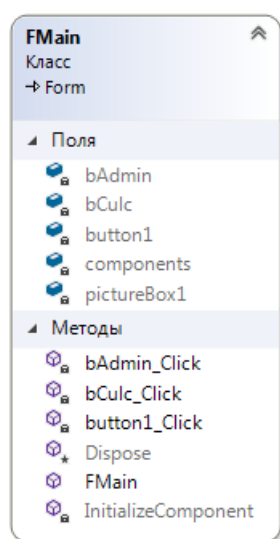


Рисунок 3.1 – Діаграма інтерфейсного модуля

Розрахунковий модуль має вигляд, що представлений на рис. 3.2. Він складається з п'яти класів та одного інтерфейсу. Інтерфейс I3D має дві властивості, що реалізовані в класі FParam (див. Додаток Б).

Властивість myRezT – колекція екземплярів власного класу TField, яке містить в собі одну властивість RezT. Поле RezT – це колекція екземплярів власного класу PointTField.

Клас PointTField має дві властивості розрахункового характеру: діаметр шару та його температура.

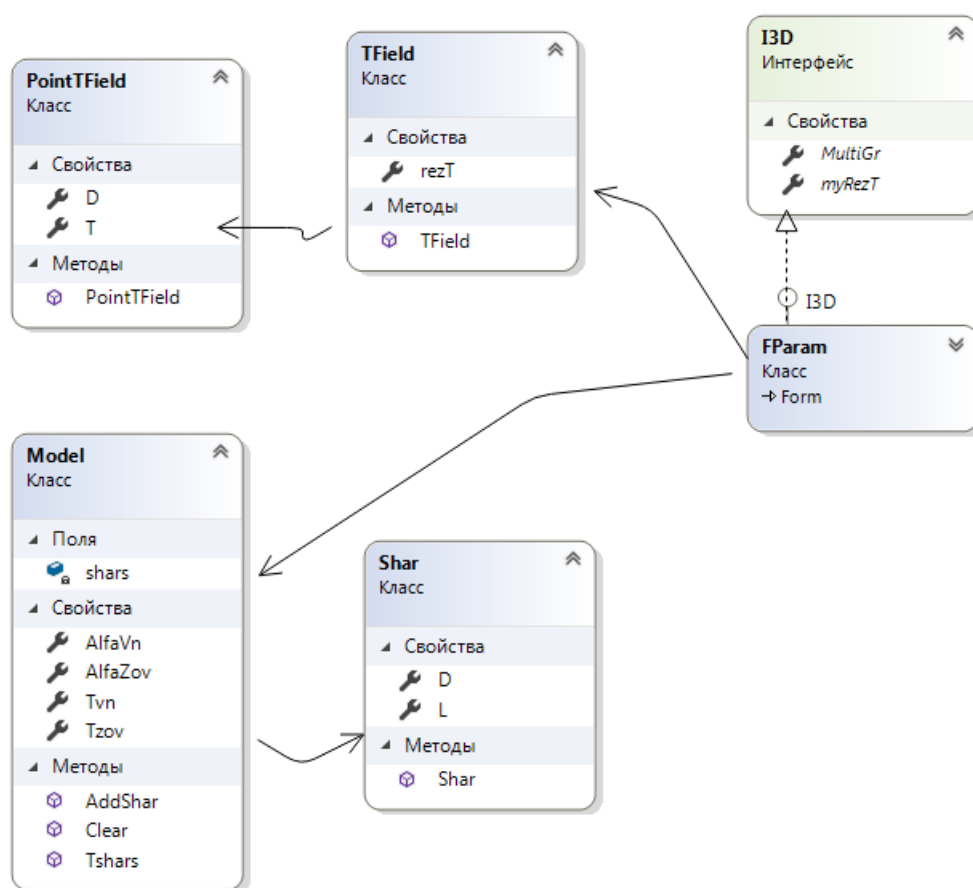


Рисунок 3.2 – Діаграма розрахункового модуля

Клас `FParam` містить екземпляр власного класу `Model`, який представлено на рис. 3.2. Цей клас інкапсулює в собі розрахункову модель задачі, тому містить колекцію шарів конструкції, як екземплярів класу `Shar`. Клас `Shar` містить геометричні характеристики шару конструкції: діаметр та коефіцієнт теплопровідності матеріалу.

В класі моделі `Model` містяться всі необхідні параметри розрахункової моделі за допомогою використання властивостей `Tvn`, `Tzov`, `AlfaVn`, `AlfaZov`. В класі є метод додавання нових шарів до конструкції та метод розрахунку температурного поля конструкції.

На рис.3.3 представлено діаграму класу `FParam`. Клас містить екземпляри класів `DataSet` та `DataTable` для роботи з БД (див. Додаток Б). Для роботи з БД необхідно використовувати бібліотеку `Npgsql`.

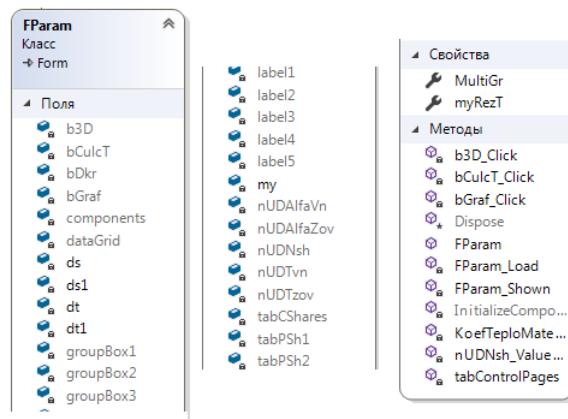


Рисунок 3.3 – Діаграма класу FParam

Клас містить чотири контейнери для загальних параметрів моделі, колекція вкладок для кожного шару багатошарової стіни (рис. 3.4). За замовченням конструкція містить два шари, тому колекція складається тільки з двох вкладок. Вікно модуля розрахунку дозволяє провести розрахунок побудованої математичної моделі, для отриманої числової моделі візуалізувати температурне поле конструкції та провести розрахунок критичного діаметру ізоляції.

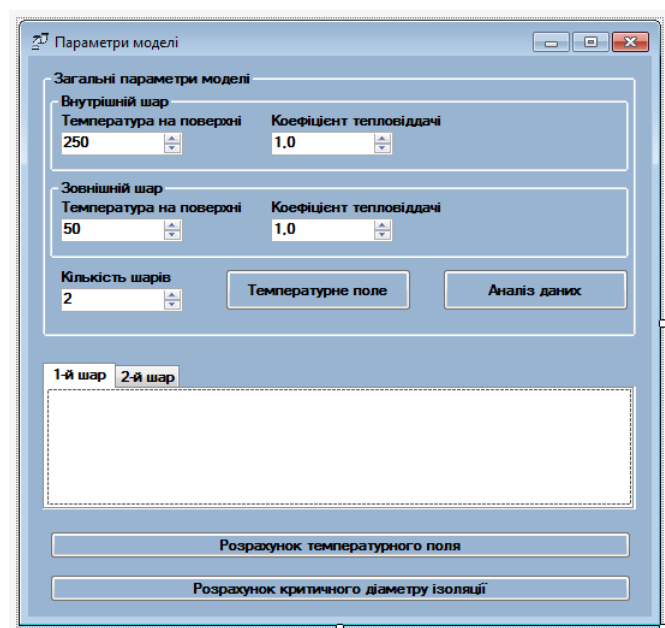


Рисунок 3.4 – Зовнішній вид вікна модуля розрахунку на етапі візуального програмування

Роботу модуля адміністратора баз даних забезпечує клас FBD, що складається з одинадцяти полів та шести методів (див. Додаток В). Діаграму класу представлено на рис. 3.5.

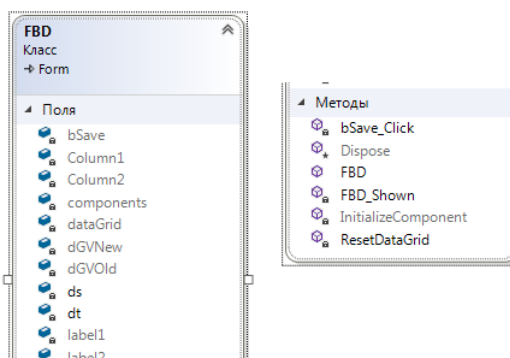


Рисунок 3.5 – Діаграма модуль адміністратора БД

Класі вікна модуля містить дві таблиці з існуючими матеріалами та новими матеріалами. Функціонал модуля адміністратора БД дозволяє завантажувати існуючі матеріали з БД та додавати нові матеріали до БД.

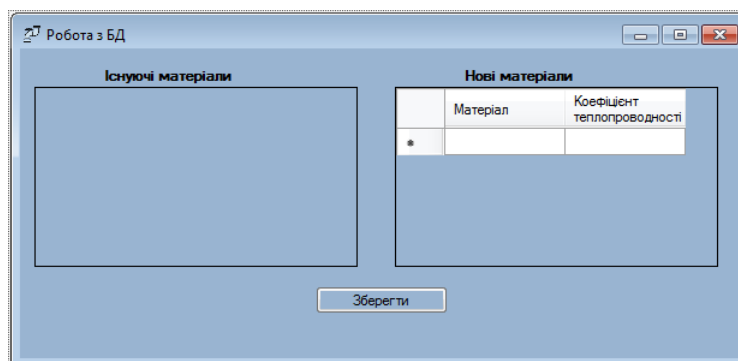


Рисунок 3.6 – Зовнішній вид вікна модуля адміністратора БД на етапі візуального програмування

Доступ до вікна модуля адміністратора БД здійснюється тільки за паролем, який задається за допомогою вікна виду, що представлено на рис. 3.7. Під час вводу паролю його символи закриваються символом маски. А функціонал класу FPassword забезпечує перевірку паролю, що введено (див. Додаток В).

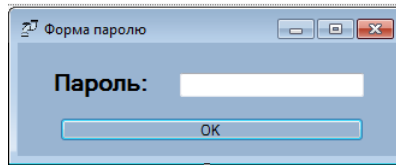


Рисунок 3.7 – Зовнішній вид вікна паролю модуля адміністратора БД на етапі візуального програмування

Модуль візуалізації розрахованого температурного поля багатосарової конструкції складається з одного класу F3D, діаграма якого представлено на рис. 3.8.

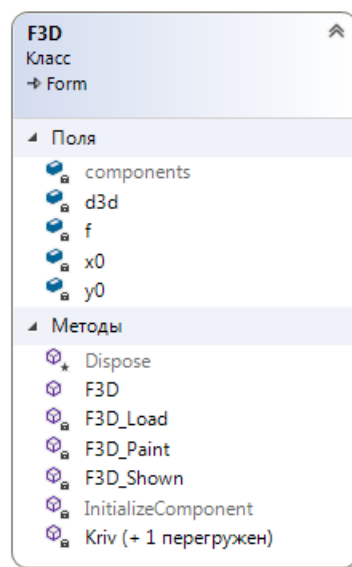


Рисунок 3.8 – Діаграма класу F3D

Клас містить п'ять полів та сім методів (див. Додаток Г). Поле d3d типу Device використовується для зберігання посилання на пристрій Direct3D. Виконання всіх графічних операцій на Direct3D визначається класом пристрою. Пристрій Direct3D зручно створювати під час першого показу вікна на екрані у функції події Load. Для обробки можливих помилок код ініціалізації пристрою помістимо в блок try/catch.

Особливістю реалізації даного класу є те, що в конструктор класу передається інтерфейсне посилання I3D f, так як цей модуль може використовуватися, як в розрахунковому блоці, так і в модулі аналізу даних. На рис. 3.9 представлено схему ієрархії класів, що реалізують інтерфейс I3D,

тому можуть бути використаними в конструкторі класу F3D за рахунок принципу об'єктно - орієнтованого програмування – поліморфізму.

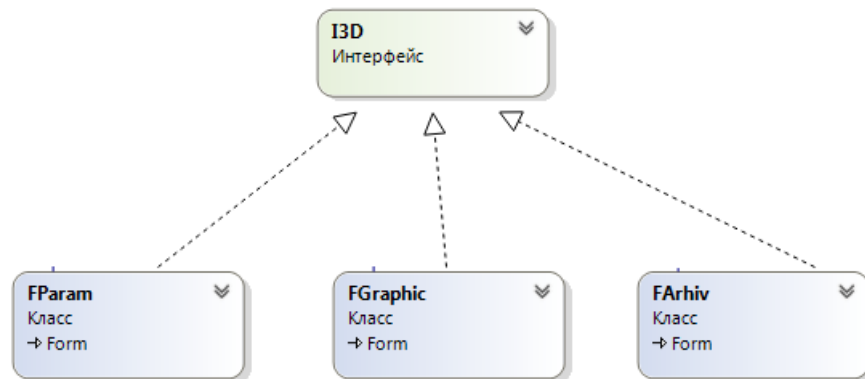


Рисунок 3.9 – Схема реалізації загального інтерфейсу I3D

Модуль аналізу даних включає в себе клас форми FArhiv, що на етапі візуального програмування має вигляд, що представлено на рис. 3.10.

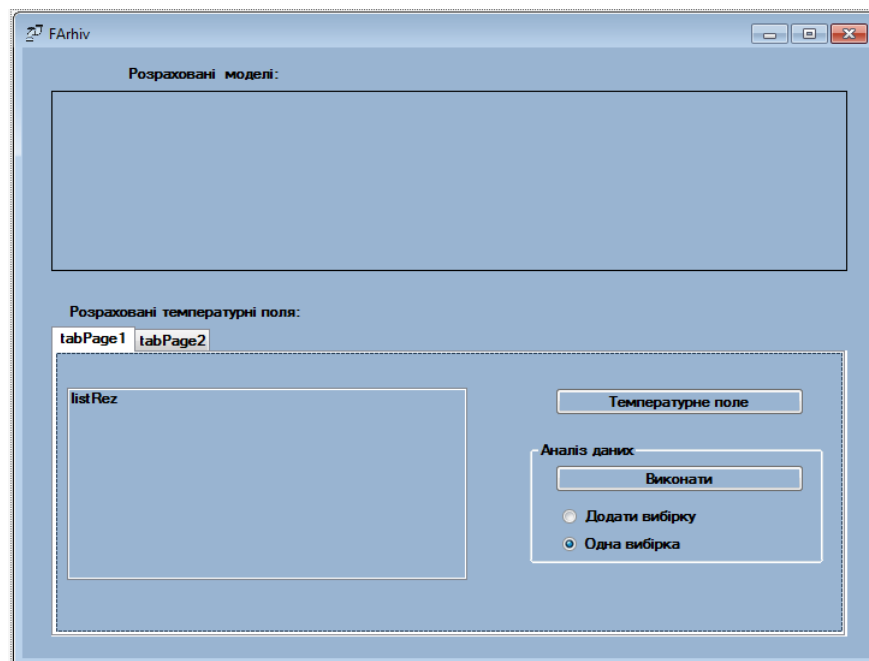


Рисунок 3.10 – Зовнішній вид вікна модуля аналізу даних на етапі візуального програмування

Вікно містить два функціональні контейнери: контейнер розрахованих моделей, що зберігаються в БД та контейнер температурних полів, що отримані для різних моделей, які також зберігаються в БД.

Користувач має функціонал побудови візуалізації температурного поля та аналіз обраних вибірок даних для даної моделі (див. Додаток Д).

3.2 Реалізація роботи з БД

Для роботи з БД використовується бібліотека Npgsql пакету NuGet. Спочатку треба виконати приєднання до БД за допомогою екземпляру класу `NpgsqlConnection`. Для цього треба створити екземпляр класу `NpgsqlConnection`. У якості вхідного параметру в конструктор цього класу передається рядок, що містить інформацію про тип серверу, ідентифікатор користувача, пароль доступу та назву БД. За допомогою методу `Open()` виділяється відкрите з'єднання з пулом. І тільки якщо в пулі вільних відкритих сполук не виявилось – створюється нова фізична сполука.

Після цього треба створити екземпляр класу `NpgsqlDataAdapter`, що є адаптером команд. Конструктор класу приймає два вхідних параметри: рядок запиту команди та екземпляр класу `NpgsqlConnection`, що отримав пул для роботи з БД.

Основним типом, що використовується для реалізації роботи з БД є `DataSet`. `DataSet` - є кеш даних у пам'яті, отриманих із джерела даних, є основним компонентом архітектури ADO.NET. Складається `DataSet` з колекції `DataTable` об'єктів, які можна зв'язати один з одним за допомогою об'єктів `DataRelation`.

Для екземпляру класу `DataSet` викликається метод `Reset()`, який очищає всі таблиці та видаляє всі зв'язки, зовнішні обмеження та таблиці з `DataSet`. Таким чином ми відновлюємо `DataSet` у вихідний стан.

Далі необхідно для екземпляру класу `NpgsqlDataAdapter` вткнути метод `Fill()`, який приймає один вхідний параметр – екземпляр класу `DataSet`. Метод `Fill()` додає або оновлює рядки у `DataSet` для отримання відповідності

рядкам у джерелі даних. Нульова колекція DataSet передається в екземпляр класу DataTable.

Для роботи з колекцією DataTable в роботі використовується елемент управління DataGridView. Властивість DataSource елемента DataGridView прив'язується до джерела даних – екземпляра класу DataTable.

Викликається NpgsqlConnection.Close(), що забезпечує повернення в пул.

В класі FArhiv розроблено власну функцію ResetDataGrid() (див. Додаток Д) для завантаження даних з БД про розраховані моделі за допомогою запиту:

```
string sql = ("SELECT * FROM Model ");
```

В функції події кліку на елементі управління таблиці розрахованих моделей (див. рис. 3.10) dGVModel_RowHeaderMouseClick() виконується запит про завантаження всіх розрахованих температурних полів для обраної моделі для їх завантаження до таблиці температурних полів:

```
string sql = "Select modelSharsCulc.id,d,T,Name from modelSharsCulc inner join " +
            "(shar INNER JOIN material on shar.idmaterial = material.id)" +
            "on modelSharsCulc.idshar = shar.id where modelsharsCulc.idmodel" +
            "= " + Convert.ToString(dGVModel.Rows[e.RowIndex].Cells[0].Value) + "order by" +
            "modelSharsCulc.id";
```

В класі FBD розроблено власну функцію ResetDataGrid() (див. Додаток В), в якій виконується запит всіх матеріалів, яка можуть використовуватися для теплоізоляції:

```
string sql = ("SELECT name, koefteploizol FROM Material where name != 'повітря'");
```

В функції події кліку на елементі управління збереження нових матеріалів виконується два запити:

```
sql = "select count(*) from material where name= '"+a+"'";
sql = "INSERT INTO Material(name, koefteploizol) VALUES( ' " + a + "', " + l + " )";
```

В класі FParam в функції події завантаження вікна FParam_Load() виконується запит по матеріалам ізоляції

```
string sql = ("SELECT name, koefteploizol FROM Material where name !='повітря');
```

Головною в функціоналі класу FParam є функція події розрахунку bCulcT_Click() (див. Додаток Б). Під час розрахунку необхідно використати збереження всіх моделей в БД, якщо вони там не існують. Для реалізації такої задачі використовуються наступні запити:

```
sql = ("SELECT name, koefteploizol FROM Material where name !='повітря');

sql = "select count(*) from model where (tvn=" +
Convert.ToString(nUDTVn.Value).Replace(',', '.') + " and tzov=" +
Convert.ToString(nUDTZov.Value).Replace(',', '.') + " and n=" +
Convert.ToString(nUDNsh.Value) +
" and alfavn=" + Convert.ToString(nUDAlfaVn.Value).Replace(',', '.') + " and
alfazov=" + Convert.ToString(nUDAlfaZov.Value).Replace(',', '.') + ")";
sql = "INSERT INTO Shar (d, idmaterial) Values (" + s1 + ",
(select id from material where name = '";

sql = "INSERT INTO ModelSharsCulc (t, idmodel, idshar) Values(" + (String.Format("{0:F2}",
myRezT[myRezT.Count - 1].rezT[i].T)).Replace(',', '.') + ", (select model.id from model
where (tvn = " + Convert.ToString(nUDTVn.Value).Replace(',', '.') +
" and tzov = " + Convert.ToString(nUDTZov.Value).Replace(',', '.') +
" and n=" + Convert.ToString(nUDNsh.Value) + " and alfavn = " +
Convert.ToString(nUDAlfaVn.Value).Replace(',', '.') +
" and alfazov = " + Convert.ToString(nUDAlfaZov.Value).Replace(',',
'.') +
"))), (select shar.id from shar INNER JOIN material on shar.idmaterial
= material.id where(d = " + Convert.ToString((tabCShares.TabPages[i].Controls[1] as
NumericUpDown).Value).Replace(',', '.') +
" and name = '";
```

3.3 Візуалізація температурного поля багат шарової конструкції

Для візуалізації розрахованого температурного поля використовується клас F3D (див. рис. 3.8). Для проведення візуалізації підключені бібліотеки

```
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;
```

В класі F3D (див. Додаток Г) створено екземпляр класу Device за допомогою конструктору з п'ятьма вхідними параметрами. Перший параметр

adapter (адаптер) відноситься до того фізичного пристрою, який хочемо класифікувати. Кожен пристрій на комп'ютері має ідентифікатор адаптера (від 0 до числа, на одиницю меншого, ніж кількість наявних пристроїв). Нульове значення для параметра adapter передбачає встановлення пристрою за замовчанням.

Наступний параметр DeviceType (тип пристрою) повідомляє Direct3D, який тип пристрою ми хочемо створити. Зазвичай значенням цього параметра є опція DeviceType.Hardware, що передбачає використання апаратного пристрою.

Параметр renderWindow (вікно рендерингу) пов'язує вікно із створеним пристроєм. Оскільки клас управління вікнами містить дескриптор вікна, це полегшить використання відповідного класу як вікно рендерингу. Для цього параметра можна використовувати форму, панель або будь-який інший похідний клас. Ми будемо використовувати лише форми.

Наступний параметр behaviorFlags (прапорці режимів) використовується для керування режимами пристрою після його створення. Ми будемо використовувати прапорець SoftwareVertexProcessing, який передає всю обробку вершин центральному процесору. Незважаючи на те, що таким чином операції виконуються повільніше, ніж при використанні графічного процесора, буде надійніше використовувати саме такий підхід, так як не завжди відомо, напевно, чи підтримує наша графічна плата цей режим.

Останній параметр presentationParameters управляє поданням даних на екрані. За допомогою цього класу можна керувати будь-якими параметрами даних, що відображаються.

Фон зображення температурного поля робиться білим, а зображення температурного поля змінює колір від червоного до блакитного. Для зображення використовуються рендерінг вершини типу CustomVertex.TransformedColored.Format.

В класі розроблено власну функцію `Kriv()` з двома перевантаженнями. Перший варіант функції має чотири вхідних параметра: радіус шару, максимальне та мінімальне значення температури в конструкції та температура в шарі. Розраховується крок з яким треба змінювати відтінок кольору в залежності від мінімального та максимального значень температури в конструкції, а потім визначається колір поточного шару. Для розрахунку кольору використовується функція `Color.FromArgb()` з трьома вхідними параметрами для змішування двох відтінків червоного та блакитного кольорів.

Другий варіант функції `Kriv()` приймає два вхідних параметра: радіус шару та колір яким треба зобразити межу шару. Ця функція призначення для побудови меж між шарами конструкції.

Результатом роботи перевантажених функцій `Kriv()` є колекція елементів типу `CustomVertex.TransformedColored.Format`.

На сцену спочатку завантажується колекція вершин температурного поля, а потім колекція меж шарів.

4 РОБОТА СИСТЕМИ ДОСЛІДЖЕННЯ

4.1 Тестування роботи

В момент запуску додатку головне вікно інтерфейсу системи має вигляд, що представлений на рис. 4.1.

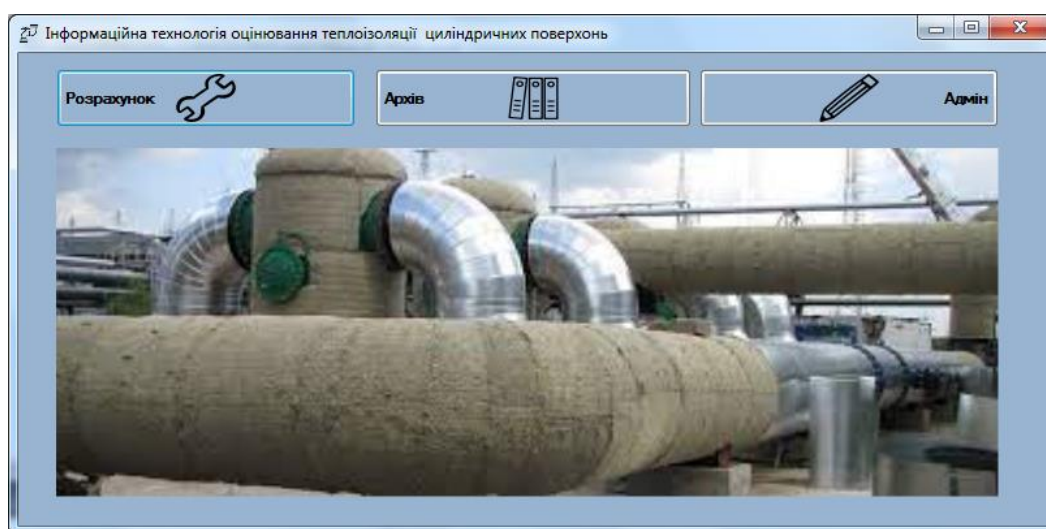


Рисунок 4.1 – Головне вікно інтерфейсного модуля

На формі розташовано три функціональні кнопки. Після натискання на кнопку Адмін, відкривається вікно пароля, що має вигляд представлений на рис. 3.7. Якщо буде введений правильний пароль доступу, то відкривається вікно модуля адміністратору БД (див. рис. 4.2).

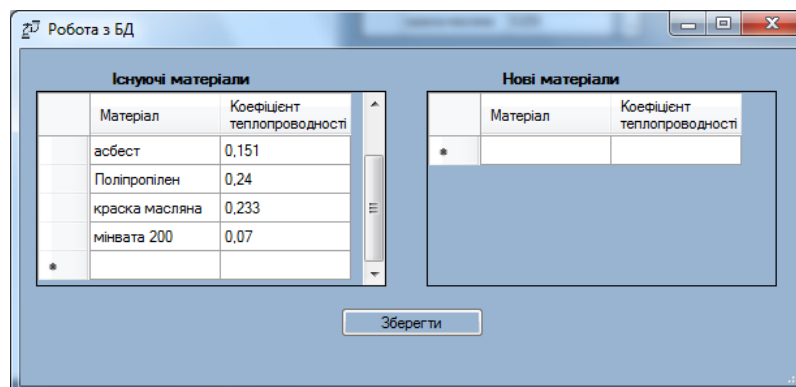


Рисунок 4.2 – Зовнішній вид вікна модуля адміністратору БД

В лівій частині вікна розташовано таблиці з вертикальною прокруткою з рядками матеріалів, що зберігаються в БД. Присутні назва матеріалу та його коефіцієнт теплопровідності.

В правій частині міститься таблиця, в якій можна додавати нові матеріали до існуючих. Наприклад, додамо два матеріали, як показано на рис. 4.3.

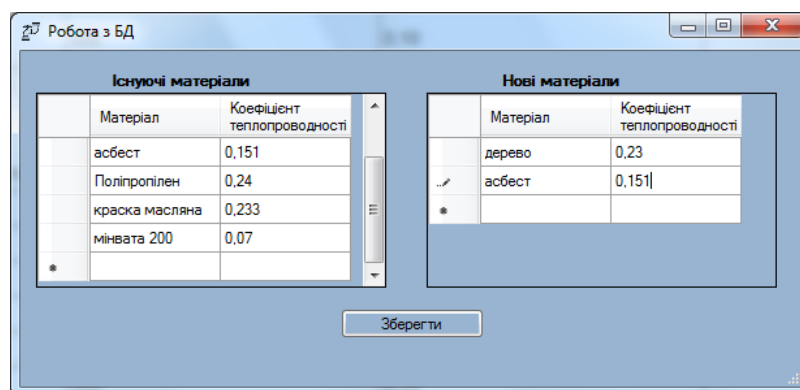


Рисунок 4.3 – Функціонал додавання нових матеріалів

Слід підкресли, що ми спеціально ввели матеріал «азбест», що вже існує в БД. Матеріалу «дерево» в БД не було. Під час збереження було виведено повідомлення, як показано на рис. 4.4.

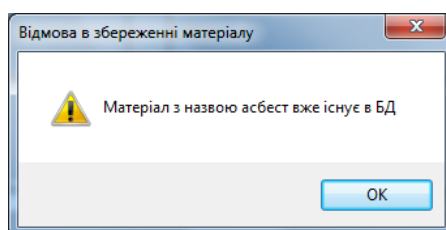


Рисунок 4.4 – Повідомлення у разі існування матеріалу в БД

Після закриття вікна повідомлення оновлюється інформація в лівій таблиці матеріалів, що представлено на рис. 4.5. Згідно отриманого результату видно, що до БД додався тільки один матеріал – дерево. Після закриття вікна модуля адміністратора БД, додаток відкриває вікно інтерфейсного модуля (рис. 4.1).

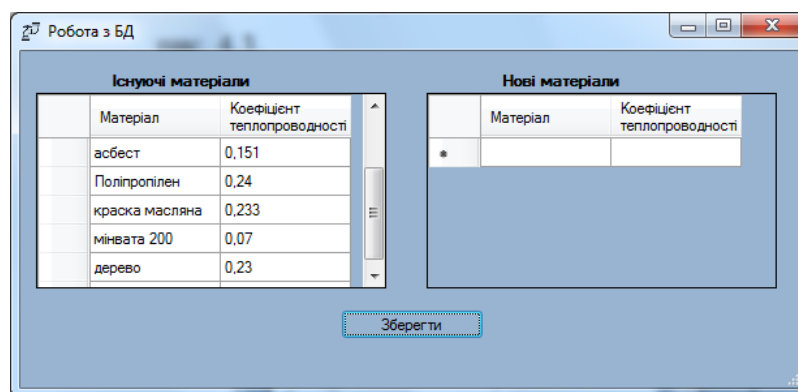


Рисунок 4.5 – Вікно модуля адміністратора БД з новим матеріалом

В разі вибору функціональної кнопки Розрахунок відкривається вікно модуля розрахунків, як представлено на рис. 4.6. За замовченням конструкція має два шари, тому в нижній частині вікна є три закладки: для двох шарів конструкції та зовнішнього шару – повітря.

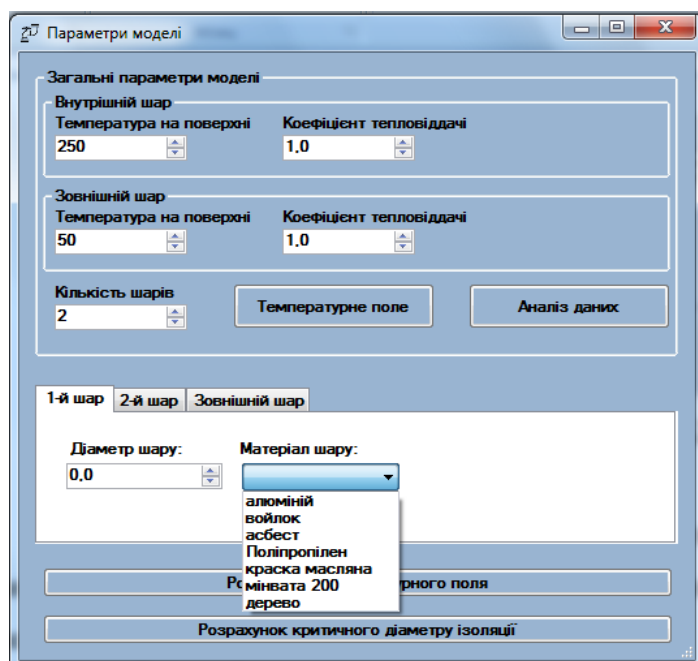


Рисунок 4.6 – Зовнішній вид вікна модуля розрахунку

З рис. 4.6 видно, що у списку матеріалів, який завантажується з БД, присутній новий матеріал «дерево», який ми додали під час тестування роботи додатку.

Для зовнішнього шару не вибирається матеріал, тому закладка цього шару має вигляд, що представлено на рис. 4.7.

Параметри моделі

Загальні параметри моделі

Внутрішній шар

Температура на поверхні: 250

Коефіцієнт теплопровідності: 1,0

Зовнішній шар

Температура на поверхні: 50

Коефіцієнт теплопровідності: 1,0

Кількість шарів: 2

Температурне поле

Аналіз даних

1-й шар 2-й шар Зовнішній шар

Діаметр шару:

0,0

Розрахунок температурного поля

Розрахунок критичного діаметру ізоляції

Рисунок 4.7 – Інтерфейс закладки «Зовнішній шар»

Користувач може змінювати кількість шарів конструкції, тоді кількість закладок буде динамічно змінюватися, як показано на рис. 4.8.

Параметри моделі

Загальні параметри моделі

Внутрішній шар

Температура на поверхні: 250

Коефіцієнт теплопровідності: 1,0

Зовнішній шар

Температура на поверхні: 50

Коефіцієнт теплопровідності: 1,0

Кількість шарів: 4

Температурне поле

Аналіз даних

1-й шар 2-й шар 3-й шар 4-й шар Зовнішній шар

Діаметр шару:

0,0

Матеріал шару:

Розрахунок температурного поля

Розрахунок критичного діаметру ізоляції

Рисунок 4.8– Інтерфейс вікна модуля розрахунку для чотирьох шарів конструкції

Після побудови розрахункової моделі користувач може клацнути на функціональній кнопці Розрахунок температурного поля. Після завершення розрахунку користувач отримає повідомлення, як представлено на рис. 4.9.

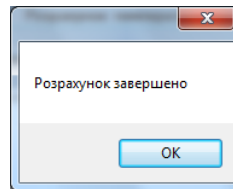


Рисунок 4.9 – Повідомлення про завершення розрахунку

Для отриманої числової моделі користувач може провести візуалізацію температурного поля за допомогою функціональної кнопки Температурне поле, як показано на рис. 4.10. Видно, що числова модель має чотири шари, як було задано на рис.4.8.

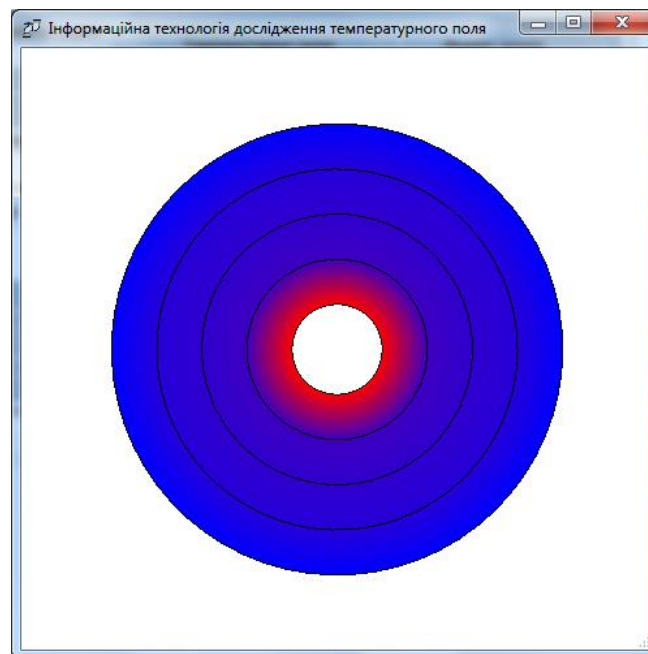


Рисунок 4.10 – Зовнішній вид вікна модуля візуалізації температурного поля

Крім того, можна провести аналіз отриманої вибірки температурного поля за допомогою функціонала кнопки Аналіз даних. Відкриється вікно модуля аналізу даних, як показано на рис. 4.11.

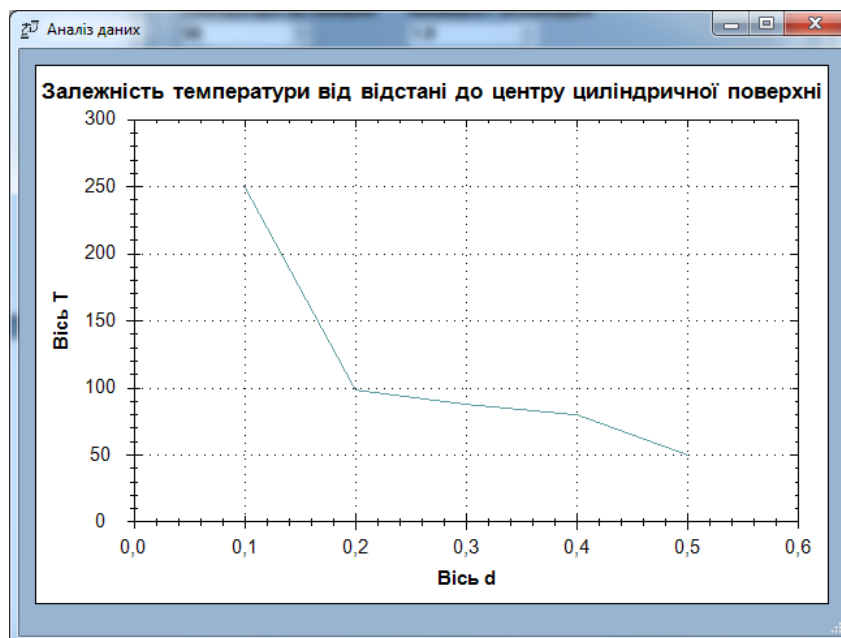


Рисунок 4.11 – Зовнішній вид вікна модуля аналізу даних для поточної числової моделі.

Крім того, користувач може провести розрахунок критичного діаметру ізоляції для заданої конструкції за допомогою функціональної кнопки Розрахунок критичного діаметру ізоляції (див. рис.4.12).

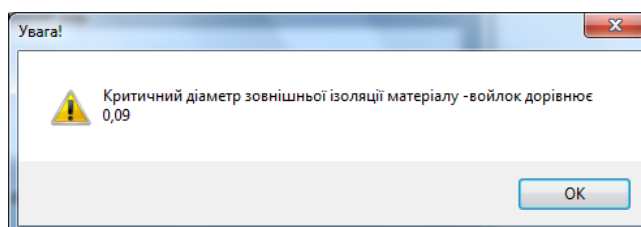
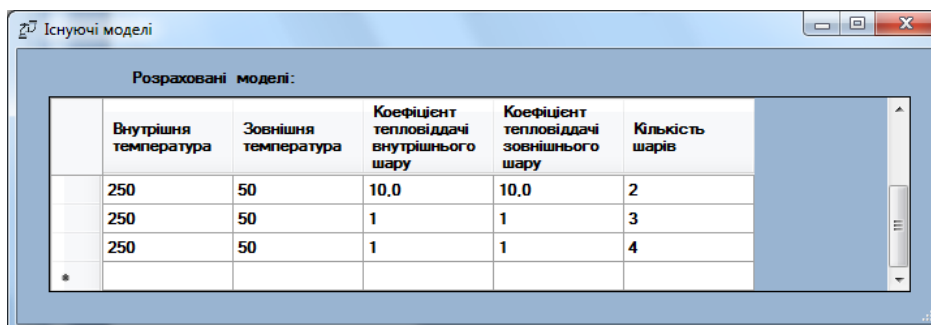


Рисунок 4.12 – Повідомлення про розрахований критичний діаметр зовнішньої ізоляції конструкції

Після закриття вікна розрахункового модуля, додаток відкриває вікно інтерфейсного модуля (рис. 4.1).

В разі вибору функціональної кнопки **Архів** відкривається вікно модуля аналізу даних, як представлено на рис. 4.13.



Існуючі моделі

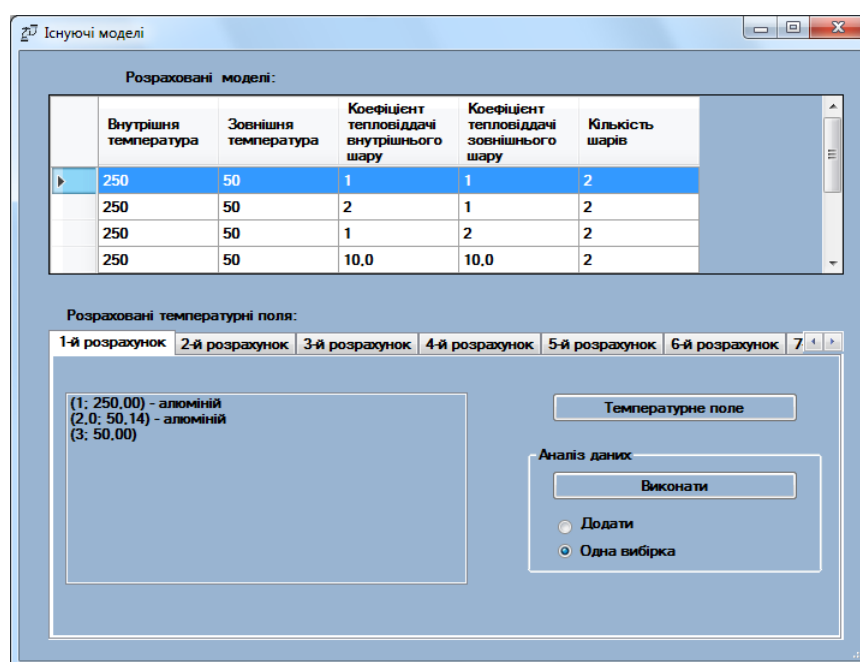
Розраховані моделі:

	Внутрішня температура	Зовнішня температура	Коефіцієнт тепловіддачі внутрішнього шару	Коефіцієнт тепловіддачі зовнішнього шару	Кількість шарів
	250	50	10,0	10,0	2
	250	50	1	1	3
	250	50	1	1	4
*					

Рисунок 4.13 – Зовнішній вид вікна модуля аналізу даних

За замовченням в таблицю завантажується колекція розрахованих моделей. На рис. 4.13 можна побачити, що останньою моделлю є наша чотирьохшарова модель.

Якщо користувач обирає якусь конкретну модель, то вікно приймає вигляд, що показано на рис.4.14.



Існуючі моделі

Розраховані моделі:

	Внутрішня температура	Зовнішня температура	Коефіцієнт тепловіддачі внутрішнього шару	Коефіцієнт тепловіддачі зовнішнього шару	Кількість шарів
▶	250	50	1	1	2
	250	50	2	1	2
	250	50	1	2	2
	250	50	10,0	10,0	2

Розраховані температурні поля:

1-й розрахунок 2-й розрахунок 3-й розрахунок 4-й розрахунок 5-й розрахунок 6-й розрахунок 7-й розрахунок

(1: 250,00) - алюміній
(2: 50,14) - алюміній
(3: 50,00)

Температурне поле

Аналіз даних:

Виконати

☐ Додати
☒ Одна вибірка

Рисунок 4.14 – Зовнішній вид вікна модуля аналізу даних для обраної моделі

В нижній частині вікна з'являється елемент управління з такою кількістю вкладок скільки в БД збережено числових моделей температурного поля. Для кожного збереженого температурного поля можна зробити візуалізацію за допомогою вікна модуля візуалізації (див. рис. 4.10).

Особливістю функціоналу даного вікна є можливість побудови графіку не однієї вибірки. Наприклад оберемо три числові моделі, тоді отримуємо результат, що представлено на рис. 4.15.

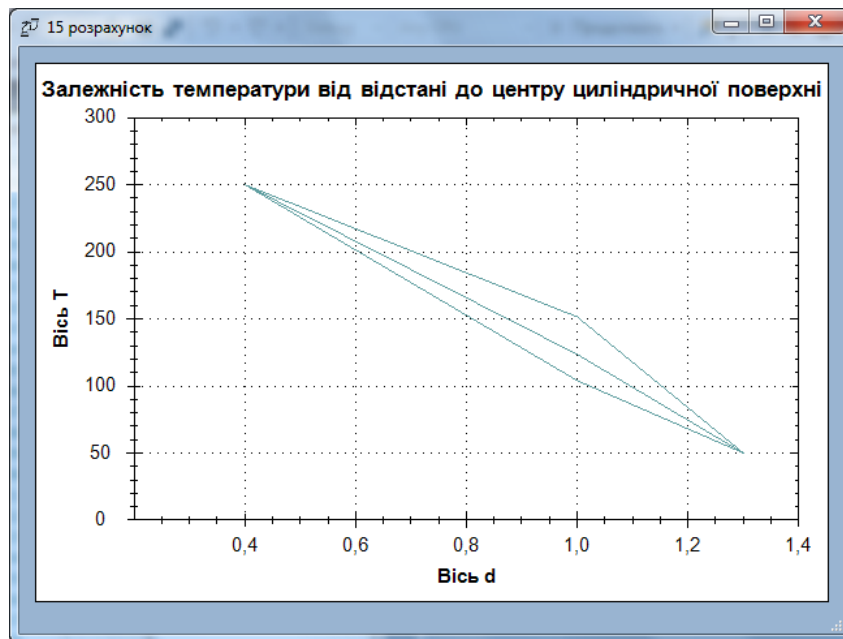


Рисунок 4.15 – Аналіз даних для декількох числових моделей з однаковою геометрією

4.2 Аналіз результатів

Проведемо аналіз отриманих результатів, що дає побудована інформаційна технологія. На рис. 4.16 представлено візуалізацію температурного поля для двошарової циліндричної труби з матеріалів: алюміній – повсть. А на рис. 4.17 представлено візуалізацію температурного поля для двошарової циліндричної труби з матеріалів: поліпропілен – азбест.

На рис. 4.18 представлено графік зміни температури ($^{\circ}\text{C}$) двошаровій стіні для різних матеріалів. Крива 1 відповідає температурному полю рис. 4.16, в крива 2 – температурному полю рис. 4.17. Для проведення обчислення

моделі задавалися наступні параметри: температура внутрішнього середовища $T_{C_1} = 250^\circ\text{C}$, температура зовнішнього середовища $T_{C_2} = 50^\circ\text{C}$, геометрія конструкції – $d_1 = 0.4\text{ м}$, $d_2 = 0.6\text{ м}$, $d_3 = 1.1\text{ м}$, коефіцієнти тепловіддачі $\alpha_1 = \alpha_2 = 1$.

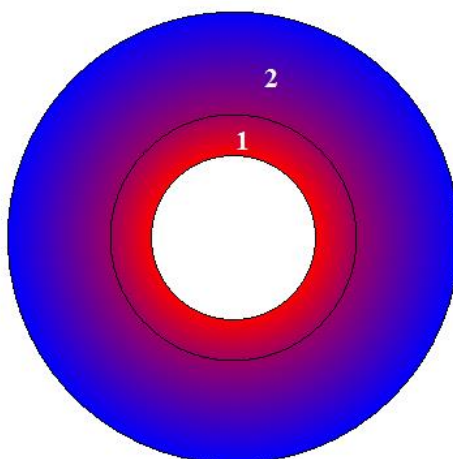


Рисунок 4.16 – Температурне поле двошарової труби: алюміній – повсть

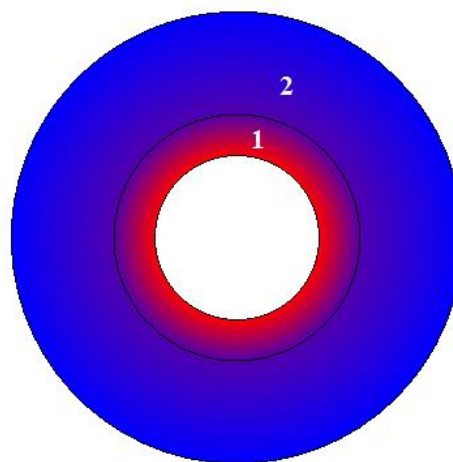


Рисунок 4.17 – Температурне поле двошарової труби: поліпропілен – азбест

Проведено чисельний експеримент для тришарової циліндричної труби для тих самих параметрів зовнішнього та внутрішнього середовищ. Геометрія конструкції: $d_1 = 0.8\text{ м}$, $d_2 = 1.0\text{ м}$, $d_3 = 1.2\text{ м}$, $d_4 = 1.3\text{ м}$. На рис. 4.21 представлено графік зміни температури ($^\circ\text{C}$) тришаровій стіні для різних

матеріалів. Крива 1 відповідає температурному полю рис. 4.19, в крива 2 – температурному полю рис. 4.20.

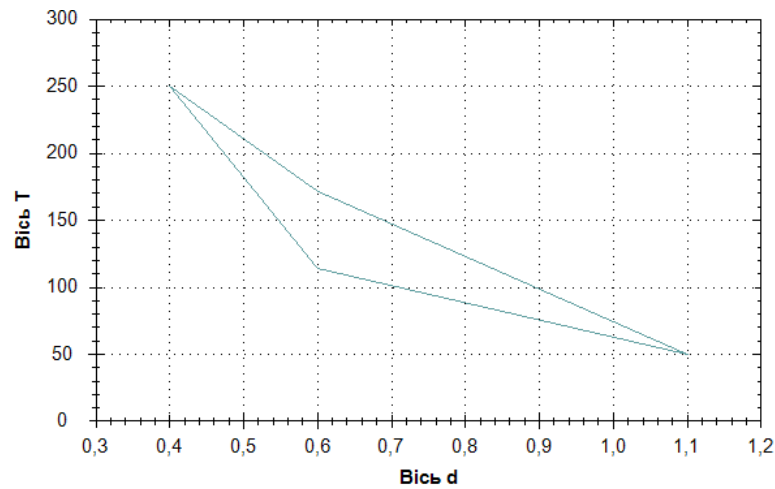


Рисунок 4.18 – Графік зміни температури в залежності від відстані до центру двошарової труби

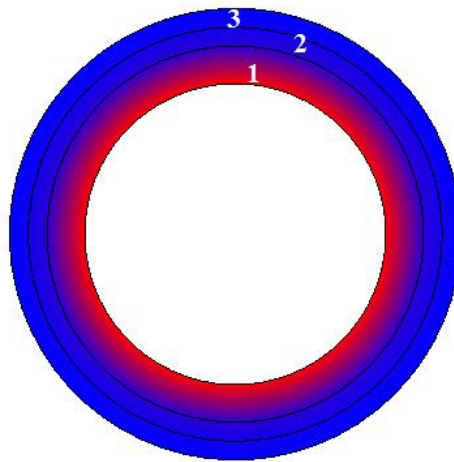


Рисунок 4.19 – Температурне поле тришарової труби: поліпропілен – азбест-масляна краска

З проведеного аналізу видно, що при однаковій геометрії конструкції для більш сучасного матеріалу циліндричних труб – поліпропілену, теплоізоляція конструкції досягається краще, навіть при використанні азбесту.

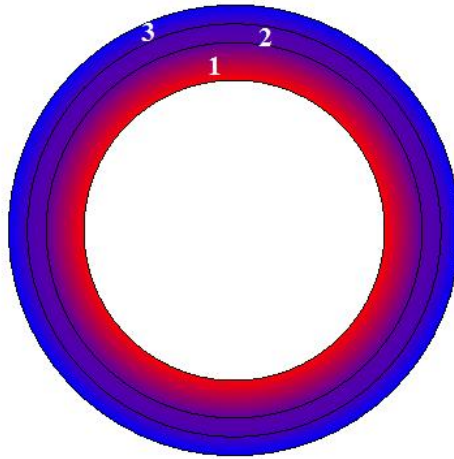


Рисунок 4.20 – Температурне поле тришарової труби: алюміній –
масляна краска – повсть

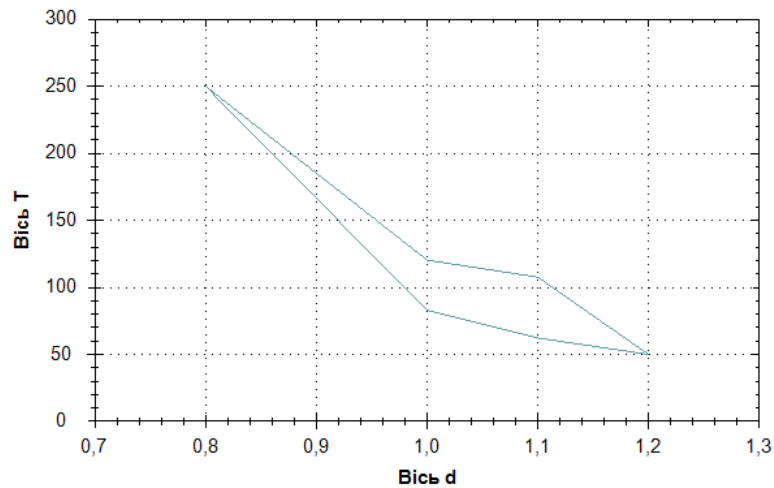


Рисунок 4.21 – Графік зміни температури в залежності від відстані до
центру тришарової труби

ВИСНОВОК

Більшість технологічних процесів здійснюються з виділенням або поглинанням теплоти. Інженер у своїй діяльності має справу з різними тепловими процесами і їх конструктивним оформленням у вигляді теплотехнічного обладнання, яке вбудоване в технологічні процеси. Тому він повинен вміти грамотно і ефективно використовувати таке обладнання, керувати теплоенергетичними схемами у виробництві за допомогою інформаційних технологій. Фахівець повинен добре знати теплові процеси та проводити розрахунок технічних конструкцій, що використовуються на практиці.

В дипломній роботі розроблено інструментарій для обчислення-візуалізації температурного поля багатошарової конструкції та діаметру теплоізоляції циліндричної труби.

Для досягнення поставленою мети розв'язано наступні задачі:

1. Проаналізовано існуючі математичні моделі заходження температурного поля. Проблемі розрахунку процесів теплопровідності присвячено багато робіт сучасних інженерних досліджень в машинобудівній та атомній промисловості, в технологічних процесах теплоенергетичних галузей господарської діяльності.
2. Розглянуто проблему теплопровідності багатошарових стін різної форми: плоскої, циліндричної та сферичної стін. Створено розрахункову модель багатошарової циліндричної труби для граничних умов III-го роду.
3. Розроблено інформаційну технологію дослідження багатошарової циліндричної стіни, яка може бути використана фахівцями для розрахунку та дослідження температурного поля циліндричних труб. Розроблено алгоритм побудови кольорового температурного поля згідно з проведеними розрахунками.
4. Розроблено інформаційну систему, яка реалізує модель розрахунку температурного поля та діаметру теплоізоляції. Слід зазначити, що при

проведенні розрахунків немає необхідності задавати характеристики матеріалів труб та ізоляції, оскільки всі дані зберігаються у БД. Після проведення розрахунків усі побудовані розрахункові моделі та отримані числові моделі результатів так само зберігаються у БД. Це дозволяє в разі потреби не проводити повторний розрахунок, а використовувати дані, що вже є.

5. Проведено аналіз отриманих результатів. Для аналізу результатів розроблений інструментарій дозволяє будувати графіки зміни температури багатошарової стінки. При цьому вибір необхідних розрахункових моделей користувач здійснює самостійно з даних БД ІТ. Для візуальної наочності розрахунків числової моделі використовують пакет засобів розробки тривимірної графіки DirectX.

Особливістю запропонованої інформаційної технології є те, що її функціонал дозволяє працювати з інструментарієм трьом користувачам із різним рівнем доступу: адміністратору БД, інженеру побудови математичних моделей та фахівцеві аналізу числових моделей.

Робота має довідку про впровадження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лабай В. Й. Тепломасообмін / Львів: Тріада Плюс, 2004. - 258 с.
2. Погорєлов А. І. Тепломасообмін / Львів: Новий Світ, 2006. - 144 с.
3. Лыков В. А. Теория теплопроводности / М.: Высшая школа, 1967. - 600 с.
4. Беляев Н. М. Основы теплопередачи / К.: Выща шк. Головное изд-во, 1989. - 343 с.
5. Peter Bernus, Handbook on Architectures of Information Systems / Peter Bernus, Kai Mertins, Gunter Schmidt. // Berlin: Springer, 2016. - 886 p.
6. Авраменко В.С., Проектування інформаційних систем / Авраменко В.С., Авраменко А.С. // Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. - 434 с.
7. Daniel Solis. Illustrated C# 2012 / Berkeley: APress, 2012. - 732 p.
8. Elmasri R., Fundamentals of Database Systems / Elmasri R., Navathe S. B. // Pearson, 2015. - 1280 p.
9. Малахов Є.В., Проектування баз даних та їх реалізація засобами стандартного SQL та PostgreSQL / Малахов Є.В., Блажко О.А., Глава М.Г. // О: БМВ, 2012. – 248 с.
10. Маценко В. Г. Комп'ютерна графіка. / Чернівці: Рута, 2009. 343 с.
11. Пічугін М. Ф., Комп'ютерна графіка. / Пічугін М. Ф., Канкін І. О., Воротніков В. В. // Київ: Центр учбової літ., 2013. 346 с.
12. Рачинська А.Л. Комп'ютерна графіка / Одеса: Одес. нац. ун-т імені І.І. Мечникова, 2019. – 119 с.
https://drive.google.com/file/d/1KASCVo3Sqd0aVv5jP5YrhBD9l-LESvP4/view?usp=drive_link

ДОДАТОК А. ІНТЕРФЕЙСНИЙ МОДУЛЬ

```
public partial class FMain : Form
{
    public FMain()
    {
        InitializeComponent();
    }

    private void bCulc_Click(object sender, EventArgs e)
    {
        FParam f = new FParam();
        f.Owner = this;
        Visible = false;
        f.ShowDialog();
        Visible = true;
    }

    private void bAdmin_Click(object sender, EventArgs e)
    {
        FPassword f = new FPassword();
        f.Owner = this;
        Visible = false;
        f.ShowDialog();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        FARhiv f = new FARhiv();
        f.Owner = this;
        Visible = false;
        f.ShowDialog();
        Visible = true;
    }
}
```

ДОДАТОК Б. РОЗРАХУНКОВИЙ МОДУЛЬ

```

public interface I3D
{
    List<TField> myRezT { get; set; }
    bool MultiGr { get; set; }
}

public partial class FParam : Form, I3D
{
    Model my = new Model();
    private DataSet ds = new DataSet();
    private DataTable dt = new DataTable();
    private DataSet ds1 = new DataSet();
    private DataTable dt1 = new DataTable();
    public List<TField> myRezT { get; set; }
    public bool MultiGr { get; set; }
    public FParam()
    {
        InitializeComponent();
        myRezT = new List<TField>();
    }
    private void nUDNsh_ValueChanged(object sender, EventArgs e)
    {
        tabControlPages();
    }
    private void tabControlPages()
    {
        TabPage tabPage;
        Label label;
        NumericUpDown numeric;
        Label label1;
        ComboBox combo;
        tabCShares.TabPages.Clear();
        for (int i = 0; i < nUDNsh.Value+1; i++)
        {
            tabPage = new TabPage();
            tabPage.Text = Convert.ToString(i + 1) + "-й шар";
            tabPage.BackColor = Color.White;

            label = new Label();
            label.Text = "Діаметр шару:";
            label.Location = new Point(20, 20);
            label.AutoSize = true;
            tabPage.Controls.Add(label);

            numeric = new NumericUpDown();
            numeric.Location = new Point(label.Location.X, label.Location.Y +
label.Height + 5);
            numeric.DecimalPlaces = 1;
            tabPage.Controls.Add(numeric);

            label1 = new Label();
            label1.Text = "Матеріал шару:";
            label1.Location = new Point(20 + label.Width + 50, 20);
            label1.AutoSize = true;
            tabPage.Controls.Add(label1);

            combo = new ComboBox();
            combo.Location = new Point(label1.Location.X + 5, label1.Location.Y +
label1.Height + 5);
            combo.DropDownStyle = ComboBoxStyle.DropDownList;
            for (int ik = 0; ik < dataGrid.Rows.Count-1; ik++)
                combo.Items.Add(dataGrid.Rows[ik].Cells[0].Value);
            tabPage.Controls.Add(combo);
        }
    }
}

```

```

        tabCShares.TabPages.Add(tabPage);
    }

    tabCShares.TabPages[tabCShares.TabPages.Count-1].Text = "Зовнішній шар";
    tabCShares.TabPages[tabCShares.TabPages.Count - 1].Controls[2].Visible =
false;
    tabCShares.TabPages[tabCShares.TabPages.Count - 1].Controls[3].Visible =
false;
    tabCShares.TabPages[tabCShares.TabPages.Count - 1].Controls[3].Text =
"Повітря";
    }

    private void FParam_Load(object sender, EventArgs e)
    {
        NpgsqlConnection con = new NpgsqlConnection("Server=localhost;Port=5432; User
Id=postgres;Password=123456789; Database=it_teplo;");
        con.Open();
        string sql = ("SELECT name, koefteploizol FROM Material where name
!='повітря'");
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(sql, con);
        ds.Reset();
        da.Fill(ds);
        dt = ds.Tables[0];
        dataGrid.DataSource = dt;
        con.Close();
    }

    private void FParam_Shown(object sender, EventArgs e)
    {
        tabControlPages();
    }

    private void bCulcT_Click(object sender, EventArgs e)
    {
        NpgsqlConnection con = new NpgsqlConnection("Server=localhost;Port=5432; User
Id=postgres;Password=123456789; Database=it_teplo;");
        con.Open();
        string sql = "";
        NpgsqlDataAdapter da;
        my.Clear();
        my.Tvn = Convert.ToDouble(nUDTvn.Value);
        my.Tzov = Convert.ToDouble(nUDTzov.Value);
        my.AlfaVn = Convert.ToDouble(nUDAlfaVn.Value);
        my.AlfaZov = Convert.ToDouble(nUDAlfaZov.Value);
        sql = ("SELECT name, koefteploizol FROM Material where name !='повітря'");
        da = new NpgsqlDataAdapter(sql, con);
        ds.Reset();
        da.Fill(ds);
        dt = ds.Tables[0];
        dataGrid.DataSource = dt;

        for (int i = 0; i < tabCShares.TabPages.Count; i++)
        {
            double d = Convert.ToDouble((tabCShares.TabPages[i].Controls[1] as
NumericUpDown).Value);

            double l = KoefTeploMaterial((tabCShares.TabPages[i].Controls[3] as
ComboBox).Text);
            my.AddShar(new Shar(d, l));
        }

        myRezT.Add(new TField());
        myRezT[myRezT.Count-1].rezT = my.Tshars();
    }

```

```

        for (int i = 0; i < tabCShares.TabPages.Count; i++)
        {
            sql = "select count(*) from model where (tvn=" +
Convert.ToString(nUDTVn.Value).Replace(',', '.') + " and tzov=" +
Convert.ToString(nUDTZov.Value).Replace(',', '.') + " and n=" +
Convert.ToString(nUDNsh.Value) +
" and alfavn=" + Convert.ToString(nUDAlfaVn.Value).Replace(',', '.') + " and
alfazov=" + Convert.ToString(nUDAlfaZov.Value).Replace(',', '.') + ")";
            da = new NpgsqlDataAdapter(sql, con);
            ds.Reset();
            da.Fill(ds);
            dt = ds.Tables[0];
            dataGrid.DataSource = dt;
            string s1 = "";

            //додаємо нову модель
            if (Convert.ToInt32(dataGrid.Rows[0].Cells[0].Value) == 0)
            {
                s1 = Convert.ToString(nUDTVn.Value).Replace(',', '.');
                sql = "INSERT INTO Model (tvn, tzov, alfavn, alfazov, n) VALUES( " +
s1 + ", ";

                s1 = Convert.ToString(nUDTZov.Value).Replace(',', '.');
                sql += s1 + ", ";
                s1 = Convert.ToString(nUDAlfaVn.Value).Replace(',', '.');
                sql += s1 + ", ";
                s1 = Convert.ToString(nUDAlfaZov.Value).Replace(',', '.');
                sql += s1 + ", " + Convert.ToString(nUDNsh.Value) + ")";
                da = new NpgsqlDataAdapter(sql, con);
                ds.Reset();
                da.Fill(ds);
            }

            sql = "select count(*) from shar INNER JOIN material on shar.idmaterial =
material.id where( d = " +
Convert.ToString((tabCShares.TabPages[i].Controls[1] as
NumericUpDown).Value).Replace(',', '.') + " and name = ";
            if (i != tabCShares.TabPages.Count - 1)
                sql += (tabCShares.TabPages[i].Controls[3] as ComboBox).Text + "')";
            else
                sql += "новітря')";
            da = new NpgsqlDataAdapter(sql, con);
            ds.Reset();
            da.Fill(ds);
            dt = ds.Tables[0];
            dataGrid.DataSource = dt;

            //додаємо новий шар
            if (Convert.ToInt32(dataGrid.Rows[0].Cells[0].Value) == 0)
            {
                s1 = Convert.ToString((tabCShares.TabPages[i].Controls[1] as
NumericUpDown).Value).Replace(',', '.');
                sql = "INSERT INTO Shar (d, idmaterial) Values (" + s1 + ", (select
id from material where name = ";
                if (i != tabCShares.TabPages.Count - 1)
                    sql += (tabCShares.TabPages[i].Controls[3] as ComboBox).Text +
"')";
                else
                    sql += "новітря')";

                da = new NpgsqlDataAdapter(sql, con);
                ds.Reset();
                da.Fill(ds);
            }
        }
    }

```

```

{
    sql = "INSERT INTO ModelSharsCulc (t, idmodel, idshar) Values(" +
    (String.Format("{0:F2}", myRezT[myRezT.Count - 1].rezT[i].T)).Replace(',', '.') + ",
    (select model.id from model where (tvn = " + Convert.ToString(nUDTvn.Value).Replace(',',
    '.')) +
        " and tzov = " + Convert.ToString(nUDTzov.Value).Replace(',', '.') +
    " and n=" + Convert.ToString(nUDNsh.Value) + " and alfavn = " +
    Convert.ToString(nUDAlfaVn.Value).Replace(',', '.') +
        " and alfazov = " + Convert.ToString(nUDAlfaZov.Value).Replace(',',
    '.')) +
        ")), (select shar.id from shar INNER JOIN material on shar.idmaterial
    = material.id where(d = " + Convert.ToString((tabCShares.TabPages[i].Controls[1] as
    NumericUpDown).Value).Replace(',', '.') +
        " and name = ";
    if (i != tabCShares.TabPages.Count - 1)
        sql += (tabCShares.TabPages[i].Controls[3] as ComboBox).Text +
    "''))";

    else
        sql += "повітря'))";

    da = new NpgsqlDataAdapter(sql, con);
    ds.Reset();
    da.Fill(ds);
}

sql = ("SELECT name, koefteploizol FROM Material where name
!='повітря'");
da = new NpgsqlDataAdapter(sql, con);
ds.Reset();
da.Fill(ds);
dt = ds.Tables[0];
dataGrid.DataSource = dt;

}

con.Close();

MessageBox.Show("Розрахунок завершено");
}

double KoefTeploMaterial(string name)
{
    double rez=0;
    for (int ik = 0; ik < dataGrid.Rows.Count - 1; ik++)
        if (Convert.ToString(dataGrid.Rows[ik].Cells[0].Value) == name)
        {
            rez = Convert.ToDouble(dataGrid.Rows[ik].Cells[1].Value);
            break;
        }
    return rez;
}

private void b3D_Click(object sender, EventArgs e)
{
    F3D f = new F3D(this);
    f.Owner = this;
    f.ShowDialog();
}

private void bGraf_Click(object sender, EventArgs e)
{
    FGraphic f = new FGraphic(this);
    f.Owner = this;
    f.ShowDialog();
}
}

```

```

public class TField
{
    public List<PointTField> rezT { get; set; }

    public TField()
    {
        rezT = new List<PointTField>();
    }
}

public class PointTField
{
    public double D { get; set; }
    public double T { get; set; }

    public PointTField(double x, double y)
    {
        D = x;
        T = y;
    }
}

public class Model
{
    public double Tvn { get; set; }
    public double Tzov { get; set; }
    public double AlfaVn { get; set; }
    public double AlfaZov { get; set; }

    List<Shar> shars = new List<Shar>();

    public void AddShar(Shar s)
    {
        shars.Add(s);
    }
    public void Clear()
    {
        shars.Clear();
    }
    public List<PointTField> Tshars()
    {
        List<PointTField> T = new List<PointTField>();
        T.Add(new PointTField(shars[0].D, Tvn));
        double sum = 0;
        for (int i = 0; i < shars.Count-1; i++)
        {
            sum += 0.5 / shars[i].L * Math.Log(shars[i + 1].D / shars[i].D);
        }
        double KL1 = 1.0 / AlfaVn / shars[0].D + sum + 1.0 / AlfaZov /
shars[shars.Count - 1].D;
        double KL = 1.0 / KL1;
        double qL = KL * (Tvn - Tzov);
        sum = 0;
        for (int i = 0; i < shars.Count-1; i++)
        {
            sum += 0.5 / shars[i].L * Math.Log(shars[i + 1].D / shars[i].D);

            T.Add(new PointTField(shars[i + 1].D, Tvn - qL * (1.0 / AlfaVn /
shars[0].D + sum + 1.0 / AlfaZov / shars[shars.Count - 1].D)));
        }
        return T;
    }
}

```

ДОДАТОК В. МОДУЛЬ АДМІНІСТРАТОРА БД

```

public partial class FBD : Form
{
    private DataSet ds = new DataSet();
    private DataTable dt = new DataTable();

    public FBD()
    {
        InitializeComponent();
    }

    private void FBD_Shown(object sender, EventArgs e)
    {
        ResetDataGrid();
        dGVOld.Columns[0].HeaderText = "Матеріал";
        dGVOld.Columns[1].HeaderText = "Коефіцієнт теплопровідності";
    }

    private void bSave_Click(object sender, EventArgs e)
    {
        if (dGVNew.RowCount == 0)
            return;
        NpgsqlConnection con = new NpgsqlConnection("Server=localhost;Port=5432; User
Id=postgres;Password=123456789; Database=it_teplo;");
        con.Open();
        string sql = "";
        DataSet ds1 = new DataSet();
        DataTable dt1 = new DataTable();

        for (int i = 0; i < dGVNew.RowCount-1; i++)
        {
            string a = Convert.ToString(dGVNew.Rows[i].Cells[0].Value);
            string l =
Convert.ToString(dGVNew.Rows[i].Cells[1].Value).Replace(',', '.');

            sql = "select count(*) from material where name= '"+a+"'";
            NpgsqlDataAdapter da = new NpgsqlDataAdapter(sql, con);
            ds1.Reset();
            da.Fill(ds1);
            dt1 = ds1.Tables[0];
            dataGrid.DataSource = dt1;

            if (Convert.ToInt32(dataGrid.Rows[0].Cells[0].Value) == 0)
            {
                sql = "INSERT INTO Material(name, koefteploizol) VALUES( '" + a + "',
" + l + "')";

                da = new NpgsqlDataAdapter(sql, con);
                ds.Reset();
                da.Fill(ds);
            }
            else
            {
                MessageBox.Show("Матеріал з назвою " + a + " вже існує в БД",
"Відмова в збереженні матеріалу", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }

        con.Close();
    }
}

```

```

        ResetDataGrid();
        dGVNew.Rows.Clear();
    }

    void ResetDataGrid()
    {
        NpgsqlConnection con = new NpgsqlConnection("Server=localhost;Port=5432; User
Id=postgres;Password=123456789; Database=it_teplo;");
        con.Open();
        string sql = ("SELECT name, koefteploizol FROM Material where name
!='новітря'");
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(sql, con);
        ds.Reset();
        da.Fill(ds);
        dt = ds.Tables[0];
        dGVOld.DataSource = dt;
        con.Close();

        for (int i = 0; i < dGVOld.RowCount; i++)
            dGVOld.Rows[i].ReadOnly = true;
    }
}

public partial class FPassword : Form
{
    string Password;
    public FPassword()
    {
        InitializeComponent();
    }

    private void tBPassword_KeyUp(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
            bOK_Click(sender, new EventArgs());
    }

    private void bOK_Click(object sender, EventArgs e)
    {
        StreamReader sr = new
StreamReader(Application.StartupPath+"..\..\..\..\TextFilePassword.txt");
        Password = sr.ReadLine();
        sr.Close();

        if (tBPassword.Text == "1")
        {
            FBD f = new FBD();
            f.Owner = this;
            Visible = false;
            f.ShowDialog();
            (this.Owner as FMain).Visible = true;
        }
        else
        {
            Owner.Visible = true;
        }
    }
}

```

ДОДАТОК Г. МОДУЛЬ ВІЗУАЛІЗАЦІЇ ТЕМПЕРАТУРНОГО ПОЛЯ

```

public partial class F3D : Form
{
    Device d3d;
    I3D f;
    int x0;
    int y0;
    public F3D(I3D f)
    {
        InitializeComponent();
        x0 = this.ClientSize.Width / 2;
        y0 = this.ClientSize.Height / 2;
        this.f = f;
    }
    private void F3D_Load(object sender, EventArgs e)
    {
        try
        {
            PresentParameters d3dpp = new PresentParameters();
            d3dpp.BackBufferCount = 1;
            d3dpp.SwapEffect = SwapEffect.Discard;
            d3dpp.Windowed = true;
            d3dpp.MultiSample = MultiSampleType.None;
            d3dpp.EnableAutoDepthStencil = true;
            d3dpp.AutoDepthStencilFormat = DepthFormat.D16;
            d3d = new Device(0,
                DeviceType.Hardware,
                this,
                CreateFlags.SoftwareVertexProcessing,
                d3dpp);
        }
        catch (Exception exc)
        {
            MessageBox.Show(this, exc.Message, "Ошибка инициализации");
            Close();
        }
    }
    private void F3D_Paint(object sender, PaintEventArgs e)
    {
        d3d.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.White, 1.0f, 0);
        d3d.BeginScene();
        d3d.VertexFormat = CustomVertex.TransformedColored.Format;
        double k = (this.ClientSize.Height / 2) /
(f.myRezT[0].rezT[f.myRezT[0].rezT.Count - 1].D * 100)*75;
        for (int i = 0; i < f.myRezT[0].rezT.Count - 1; i++)
        {
            List<CustomVertex.TransformedColored> L1 = Kriv(f.myRezT[0].rezT[i+1].D *
k, f.myRezT[0].rezT[0].T, f.myRezT[0].rezT[f.myRezT[0].rezT.Count - 1].T,
f.myRezT[0].rezT[i+1].T);
            List<CustomVertex.TransformedColored> L2 = Kriv(f.myRezT[0].rezT[i].D *
k, f.myRezT[0].rezT[0].T, f.myRezT[0].rezT[f.myRezT[0].rezT.Count - 1].T,
f.myRezT[0].rezT[i].T);
            List<CustomVertex.TransformedColored> rez = new
List<CustomVertex.TransformedColored>();
            for (int j = 0; j < L1.Count - 1; j++)
            {
                rez.Clear();
                rez.Add(L1[j]);
                rez.Add(L2[j + 1]);
                rez.Add(L2[j]);
                CustomVertex.TransformedColored[] verts = rez.ToArray();
                d3d.DrawUserPrimitives(PrimitiveType.TriangleList, 1, verts);
                rez.Clear();
            }
        }
    }
}

```

```

        rez.Add(L1[j]);
        rez.Add(L1[j + 1]);
        rez.Add(L2[j + 1]);
        verts = rez.ToArray();
        d3d.DrawUserPrimitives(PrimitiveType.TriangleList, 1, verts);
    }
}
for (int i = 0; i < f.myRezT[0].rezT.Count; i++)
{
    List<CustomVertex.TransformedColored> L3 = Kriv(f.myRezT[0].rezT[i].D *
k, Color.Black);
    CustomVertex.TransformedColored[] verts = L3.ToArray();
    d3d.DrawUserPrimitives(PrimitiveType.LineStrip, verts.Length - 1, verts);
}
d3d.EndScene();
d3d.Present();
}
List<CustomVertex.TransformedColored> Kriv(double R, Color C)
{
    List<CustomVertex.TransformedColored> myList = new
List<CustomVertex.TransformedColored>();
    for (double teta = 0; teta <= 2 * Math.PI + 0.1; teta += 0.1)
    {
        float x = Convert.ToSingle(x0 + R * Math.Cos(teta));
        float y = Convert.ToSingle(y0 + R * Math.Sin(teta));
        CustomVertex.TransformedColored one = new
CustomVertex.TransformedColored();
        one.Position = new Vector4(x, y, 0.5f, 1.0f);
        one.Color = C.ToArgb();
        myList.Add(one);
    }
    return myList;
}
List<CustomVertex.TransformedColored> Kriv(double R, double Tmax, double Tmin,
double T)
{
    List<CustomVertex.TransformedColored> myList = new
List<CustomVertex.TransformedColored>();
    for (double teta = 0; teta <= 2 * Math.PI + 0.1; teta += 0.1)
    {
        float x = Convert.ToSingle(x0 + R * Math.Cos(teta));
        float y = Convert.ToSingle(y0 + R * Math.Sin(teta));
        CustomVertex.TransformedColored one = new
CustomVertex.TransformedColored();
        one.Position = new Vector4(x, y, 0.5f, 1.0f);
        one.Color = Color.FromArgb(Convert.ToInt32(255/(Tmin-Tmax)*(Tmin-T)),0,
Convert.ToInt32(255 / (Tmax-Tmin) *(Tmax- T)).ToArgb());
        myList.Add(one);
    }
    return myList;
}
private void F3D_Shown(object sender, EventArgs e)
{
    Invalidate();
}
}

```

ДОДАТОК Д. МОДУЛЬ АНАЛІЗУ ДАНИХ

```

public partial class FARhiv : Form, I3D
{
    private DataSet ds = new DataSet();
    private DataTable dt = new DataTable();
    private DataSet ds1 = new DataSet();
    private DataTable dt1= new DataTable();
    public List<TField> myRezT { get; set; }
    public bool MultiGr { get; set; }
    F3D f3d;
    FGraphic fgr;
    int N,index;
    public FARhiv()
    {
        InitializeComponent();
        myRezT = new List<TField>();
    }

    private void FARhiv_Shown(object sender, EventArgs e)
    {
        ResetDataGrid();
        dGVModel.Columns[0].HeaderText = "Номер моделі";
        dGVModel.Columns[1].HeaderText = "Внутрішня температура";
        dGVModel.Columns[2].HeaderText = "Зовнішня температура";
        dGVModel.Columns[3].HeaderText = "Коефіцієнт тепловіддачі внутрішнього шару";
        dGVModel.Columns[4].HeaderText = "Коефіцієнт тепловіддачі зовнішнього шару";
        dGVModel.Columns[5].HeaderText = "Кількість шарів";
        tabControlPages(0);
    }
    void ResetDataGrid()
    {
        NpgsqlConnection con = new NpgsqlConnection("Server=localhost;Port=5432; User
Id=postgres;Password=123456789; Database=it_teplo;");
        con.Open();
        string sql = ("SELECT * FROM Model ");
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(sql, con);
        ds.Reset();
        da.Fill(ds);
        dt = ds.Tables[0];
        dGVModel.DataSource = dt;
        con.Close();
        dGVModel.Columns[0].Visible = false;
    }

    private void dGVModel_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
    {
        N = Convert.ToInt32(dGVModel.Rows[e.RowIndex].Cells[5].Value) + 1;
        NpgsqlConnection con = new NpgsqlConnection("Server=localhost;Port=5432; User
Id=postgres;Password=123456789; Database=it_teplo;");
        con.Open();
        string sql = "Select modelSharsCulc.id,d,T,Name from modelSharsCulc inner
join " +
            "(shar INNER JOIN material on shar.idmaterial = material.id)" +
            "on modelSharsCulc.idshar = shar.id where modelsharsCulc.idmodel ="
+Convert.ToString(dGVModel.Rows[e.RowIndex].Cells[0].Value) + "order by
modelSharsCulc.id";
        NpgsqlDataAdapter da = new NpgsqlDataAdapter(sql, con);
        ds1.Reset();
        da.Fill(ds1);
        dt1 = ds1.Tables[0];
        dGVFieldT.DataSource = dt1;
    }
}

```

```

        con.Close();
        int kol=(dGVFieldT.Rows.Count - 1) /
(Convert.ToInt32(dGVModel.Rows[e.RowIndex].Cells[5].Value) + 1);
        tabControlPages(kol);
        tabAddT();
    }
    private void tabControlPages(int k)
    {
        TabPage tabPage;
        ListBox listRez;
        Button bT;
        Button bRun;
        RadioButton rbAdd;
        RadioButton rbOne;
        GroupBox groupT;
        tbCulcRez.TabPages.Clear();
        tbCulcRez.Visible = true;
        for (int i = 0; i < k; i++)
        {
            tabPage = new TabPage();
            tabPage.Text = Convert.ToString(i + 1) + "-й розрахунок";
            tabPage.BackColor = SystemColors.ActiveCaption;
            tabPage.Size = new Size(655, 230);
            listRez = new ListBox();
            listRez.Location = new Point(10, 30);
            listRez.Size = new Size(335, 160);
            listRez.BackColor = SystemColors.ActiveCaption;
            tabPage.Controls.Add(listRez);
            bT = new Button();
            bT.Location = new Point(415, 30);
            bT.Text = "Температурне поле";
            bT.Size = new Size(205, 25);
            bT.BackColor = SystemColors.ActiveCaption;
            bT.Click += b3D_Click;
            tabPage.Controls.Add(bT);
            groupT = new GroupBox();
            groupT.Text = "Аналіз даних";
            groupT.Location = new Point(395, 75);
            groupT.Size = new Size(245, 105);
            groupT.BackColor = SystemColors.ActiveCaption;
            tabPage.Controls.Add(groupT);
            bRun = new Button();
            bRun.Location = new Point(20, 20);
            bRun.Text = "Виконати";
            bRun.Size = new Size(205, 25);
            bRun.BackColor = SystemColors.ActiveCaption;
            bRun.Click += bGraf_Click;
            groupT.Controls.Add(bRun);
            rbAdd = new RadioButton();
            rbAdd.Location = new Point(25, 55);
            rbAdd.Text = "Додати вибірку";
            rbAdd.BackColor = SystemColors.ActiveCaption;
            groupT.Controls.Add(rbAdd);
            rbOne = new RadioButton();
            rbOne.Location = new Point(25, 75);
            rbOne.Text = "Одна вибірка";
            rbOne.Checked = true;
            rbOne.BackColor = SystemColors.ActiveCaption;
            groupT.Controls.Add(rbOne);
            tabPage.Controls.Add(groupT);
            tbCulcRez.TabPages.Add(tabPage);
        }
    }

    if (tbCulcRez.TabPages.Count != 0) this.Size = new Size(720, 545);

```

```

    }

    void tabAddT()
    {
        for (int i = 0; i < tbCulcRez.TabPages.Count; i++)
        {
            for (int j = 0; j < N; j++)
            {
                string s = "(" + Convert.ToString(dGVFieldT.Rows[N * i +
j].Cells[1].Value) + "; " + Convert.ToString(dGVFieldT.Rows[N * i + j].Cells[2].Value) +
                ")";
                if(Convert.ToString(dGVFieldT.Rows[N * i +
j].Cells[3].Value)!="новітра")
                s+=" - " + dGVFieldT.Rows[N * i + j].Cells[3].Value;
                (tbCulcRez.TabPages[i].Controls[0] as ListBox).Items.Add(s);
            }
        }
    }

    private void b3D_Click(object sender, EventArgs e)
    {
        CulcT();
        f3d = new F3D(this);
        f3d.Owner = this;
        f3d.Text = Convert.ToString(index+1) + " розрахунок";
        f3d.ShowDialog();
    }

    private void bGraf_Click(object sender, EventArgs e)
    {
        CulcT();
        fgr = new FGraphic(this);
        fgr.Owner = this;
        fgr.Text = Convert.ToString(index+1) + " розрахунок";
        fgr.ShowDialog();
    }

    void CulcT()
    {
        index = tbCulcRez.SelectedIndex;

        if ((tbCulcRez.SelectedTab.Controls[2].Controls[2] as RadioButton).Checked)
            myRezT = new List<TField>();

        myRezT.Add(new TField());
        for (int j = 0; j < N; j++)
        {
            myRezT[myRezT.Count-1].rezT.Add(new
PointTField(Convert.ToDouble(dGVFieldT.Rows[N * index + j].Cells[1].Value),
Convert.ToDouble(dGVFieldT.Rows[N * index + j].Cells[2].Value)));
        }

        for(int k=0;k< myRezT.Count;k++)
        for (int i = 0; i < myRezT[k].rezT.Count - 1; i++)
            for (int j = 0; j < myRezT[k].rezT.Count - 1; j++)
                if (myRezT[k].rezT[j].D > myRezT[k].rezT[j + 1].D)
                {
                    PointTField mm = myRezT[k].rezT[j];
                    myRezT[k].rezT[j] = myRezT[k].rezT[j+1];
                    myRezT[k].rezT[j + 1] = mm;
                }
    }
}

```

ДОДАТОК Ж. ДОВІДКА ПРО ВПРОВАДЖЕННЯ