

УДК 539.3

Є.В. ДУДНИК, Г. І. ЛЬВОВ

СПЕЦІАЛІЗОВАНА СИСТЕМА АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ ТА АНАЛІЗУ МІЦНОСТІ РЕМОНТНИХ З'ЄДНАНЬ ТРУБОПРОВОДУ

Створена програма керування процесом побудови моделі та проведення розрахунків на міцність. Додаток дає можливість вводити дані для розрахунку, включаючи геометричні характеристики, навантаження, властивості матеріалів. Побудовано геометричну модель труби з корозійним пошкодженням та приєднанням композитного бандажу. Інтерактивна взаємодія в моделюванні відбувається за допомогою графічної бібліотеки OpenGL. Розрахунок відбувається з використанням системи ANSYS. Додаток дозволяє перегляд і аналіз отриманих результатів.

Ключові слова: композитний бандаж, труба, напруження, МСЕ, САПР.

Создана программа управления процессом построения модели и проведения расчетов на прочность. Приложение позволяет вводить данные для расчета, включая геометрические характеристики, нагрузки, свойства материалов. Построена геометрическая модель трубы с коррозионным повреждением на поверхности и присоединением композитного бандажа. Интерактивное взаимодействие с моделью производилось с помощью графической библиотеки OpenGL. Расчет проведен с использованием системы ANSYS.

Ключевые слова: композитный бандаж, труба, напряжения, МКЭ, САПР.

The program has been created to control of model building and carrying out calculations for strength. An application allows users to input data for calculations, including geometrical parameters, pressure, materials. A construction of a geometrical model of pipe with both a corrosion deformation and a joining of composite bandage is developed. Interactive interaction is made by using the OpenGL graphics library. The ANSYS CAD system is used for FEM calculations.

Keywords: composite bandage, pipe, stress, FEM, CAD/CAE.

Вступ. Бандажовані труби знаходять широке застосування при спорудженні трубопроводів. Бандажування вигідно відрізняється від відомих конструкцій, оскільки дозволяє суттєво підвищити продуктивність діючих трубопроводів. Ефективним є також використання бандажів для підсилення несучої здатності технологічних трубопроводів, об'язки компресорних і насосних станцій, а також можуть бути застосовані для зменшення рівня вібрацій за рахунок збільшення моменту опору поперечного перерізу трубопроводу, підсиленого бандажем [1].

В даній роботі розглядається побудова геометричної моделі труби з корозійним пошкодженням та приєднанням композитного бандажу. Модель реалізована за допомогою системи ANSYS. Створена програма керування процесом побудови моделі та проведення розрахунків на міцність.

Існує велика кількість різноманітних трубопроводів і часто виникає проблема у тому, щоб мати інструмент для аналізу міцності елементів трубопроводів за різних геометричних параметрів конструкції та різних умов експлуатації. [2]

Важливою задачею є створення спеціалізованої системи автоматизованого проектування (САПР) для аналізу міцності труб як з бандажем, так і без нього. Це можливо завдяки вбудованій в САПР ANSYS мові параметризованого проектування APDL.

1 Постановка задачі. Мета роботи полягає у створенні автоматизованої системи визначення напружено-деформованого стану в трубі з корозійним пошкодженням, проведенням аналізу впливу композитного бандажу, що приєднується до труби в місці корозії на міцність конструкції. Автоматизувати про-

цес створення скінчено-елементної моделі в ANSYS.

Об'єкт дослідження – фрагмент магістрального трубопроводу, що має посередині корозійний дефект із випадковою геометрією його поверхні (рис. 1).

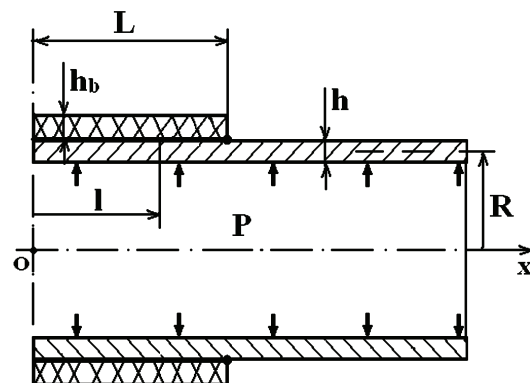


Рисунок 1 – Розрахункова схема: R – радіус внутрішньої поверхні оболонки; h – товщина оболонки; h_b – товщина бандажа

Для побудови сітки використано 4 елемента: *Solid92* і *Solid95* – елементи для твердотільних об'єктів, *Conta170* і *Conta174* – елементи для моделювання контактів.

Елемент *Solid92* підходить для моделювання нерегулярних сіток. Елемент визначається десятьма вузлами, що мають по три ступені свободи в кожному вузлі. Може переміщатися в вузлових напрямках X , Y , і Z . Елемент також має пластичність, повзучість, великі відхилення, а також великі можливості деформації.

Елемент *SOLID95* це 3D версія 8-вузла елемента

SOLID45 більш високого порядку. Він може витримувати нерівні форми без значних втрат точності. *SOLID95* добре підходить для моделювання криволінійних границь.

Елемент визначається 20 вузлами, що мають три ступені свободи на вузол. Переміщення в вузлових напрямках X , Y , і Z . *SOLID95* має пластичність, повзучість, великі відхилення, а також великі можливості деформації.

Опис характеристик матеріалів, використаних при побудові моделі представлений в табл. 1.

Таблиця 1 – Характеристики матеріалів

Труба (Сталь 09Г2С)		Бандаж (склопластик)					
Е, ГПа	200	Е1, ГПа	18,6	μ 12	0,15	Г12, ГПа	4
		Е2, ГПа	24,6	μ 13	0,42	Г13, ГПа	3
μ	0,3	Е3, ГПа	6	μ 23	0,18	Г23, ГПа	3

2 Теоретичні відомості зі створення програми.

Створення програми – це процес, що складається з наступних етапів: постановка задачі, визначення технічних характеристик, програмування, підтримка.

Задача полягає у створенні клієнта для роботи з САПР *ANSYS*. Програма повинна виконувати наступні функції:

1. Програмно вказувати розміри моделі, навантаження, матеріали.
2. Запускати розрахунок попередньо створеного макросу в окремому процесі.
3. Створення моделі за допомогою відкритої графічної бібліотеки *OpenGL* з можливістю змінювати розміри окремих частин графічної моделі.
4. Перегляд отриманих при розрахунку результатів за допомогою вбудованого журналу та перегляд отриманих зображень.
5. Можливість самостійно налаштовувати параметри програми та макросу(шляхи, назви).



Рисунок 2 – Діаграма прецедентів

Діаграма прецедентів демонструє взаємодію користувача з програмою (рис. 2). [3]

Для створення програми використовувалися наступні програми та бібліотеки:

Qt – крос-платформовий інструментарій розробки програмного забезпечення (ПЗ) мовою програмування $C++$. Дозволяє запускати написане за його допомогою ПЗ на більшості сучасних операційних сис-

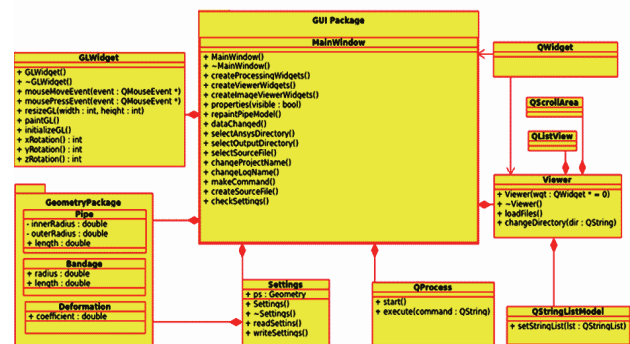
тем (ОС), як для ПК, так і для мобільних пристроїв, просто компілюючи текст програми для кожної операційної системи без зміни сирцевого коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного програмного забезпечення, починаючи з елементів графічного інтерфейсу й закінчуючи класами для роботи з мережею, базами даних, *OpenGL*, *SVG* і *XML*. Бібліотека дозволяє керувати нитями, працювати з мережею та забезпечує крос-платформовий доступ до файлів.

Qt Creator – інтегроване середовище розробки, призначене для створення крос-платформових застосунків з використанням бібліотеки *Qt*. Підтримується розробка як класичних програм мовою $C++$, так і використання мови *QML*, для визначення сценаріїв, в якій використовується *JavaScript*, а структура і параметри елементів інтерфейсу задаються *CSS*-подібними блоками. *QtCreator* може використовувати *GCC* або *MicrosoftVC++* в якості компілятора і *GDB* як дебагер. Для *Windows* версій бібліотека комплектується компілятором, заголовними і об'єктними файлами *MinGW*.

OpenGL – (англ. *OpenGraphics Library* – відкрита графічна бібліотека) – специфікація, що визначає незалежний від мови програмування крос-платформовий програмний інтерфейс (API) для написання застосунків, що використовують 2D та 3D комп'ютерну графіку. Цей інтерфейс містить понад 250 функцій, які можуть використовуватися для малювання складних тривимірних сцен з простих примітивів. Широко застосовується індустрією комп'ютерних ігор і віртуальної реальності, у графічних інтерфейсах (*Comviz*, *Clutter*), при візуалізації наукових даних, в системах автоматизованого проектування тощо.

InstallCreator – зручна програма для створення файлів встановлення в ОС *Windows*.

Структура проекту зображена у вигляді дерева елементів проекту (рис.3):

Рисунок 3 – Файлова структура проекту в *QtCreator*

Діаграма класів демонструє особливості проектування програми (рис. 3) [4]. Можна помітити, що деяких класів із діаграми класів немає в структурі проекту. Це пов'язано з тим, що багато класів бібліотеки *Qt* було використано під час створення програми. Без включення їх до діаграми класів опис був би не повним. Для опису структури потрібно зрозуміти призначення файлів на рис. 4, а потім описати докладно кожен клас в структурі програми.

Структура проекту представляє собою дерево, в

системі представлена звичайним XML-файлом. Корнем є назва проекту (Process). Файл *Process.pro* – файл проекту в QtCreator. В ньому зберігається список усіх файлів із заголовками та класами, файли форм, ресурсів, тип проекту(програма, бібліотека, плагін та інші). За допомогою ключового слова LIBS можна додати бібліотеки сторонніх розробників, а за допомогою слова QT додаються модулі бібліотеки Qt, наприклад, в даному проектному файлу є рядок QT += opengl, що говорить про необхідність додавання модуля OpenGL, котрий є адаптацією графічної бібліотеки OpenGL в Qt. При необхідності в Qt можна додати модулі для роботи з мережею, мультимедіа, формами, webkit, sql, xml та інших. За замовчуванням в QtCreator додано модулі core і gui, в версії 5 бібліотеки Qt модулі core та gui замінив модуль widgets. Якщо необхідно працювати в консольному режимі і жодні віджети не потрібні, можна виключити модуль графічного інтерфейсу з проекту шляхом додавання рядка Q -= gui. HEADERS і SOURCES в структурі дерева представляють собою заголовки класів відповідно. Ключове слово RESOURCES включає список файлів з ресурсами. Файл ресурсів може включати різноманітні сторонні файли: зображення, іконки, звукові файли, файли баз даних чи будь-яких інших, що можуть використовуватися в проекті. Файл ресурсів з розширенням .qrc представляє собою XML-документ.

3 Опис класової структури програми. Головний клас в проекті представлений центральним віджетом – QTabWidget. Даний клас є аналогом всім відомим вкладень в веб-браузерах. Він був вибраний у якості основного для компактності і легкості переключення між вікнами, при створенні проекту вкладки замінили компоненти меню. QTabWidget має метод для створення підвікон – addTabWidget (QWidget *, constQString&). Цей метод перевантажений, в конкретному випадку з його структури видно, що метод дозволяє використовувати будь-який успадкований від QWidget клас як елемент вкладки, другий параметр відповідає за назву вкладки.

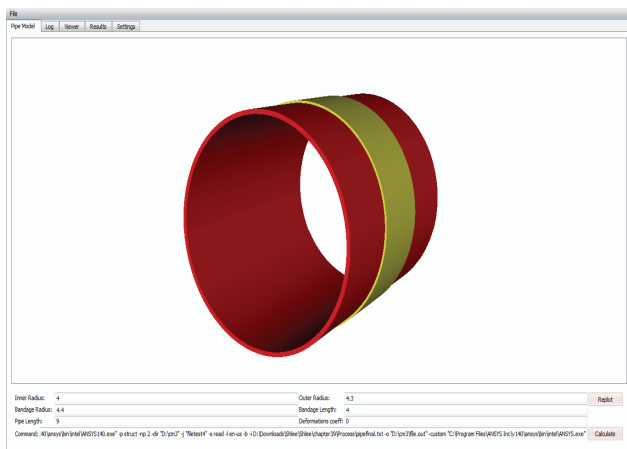


Рисунок 4– Модель труби, реалізована за допомогою OpenGL

Клас головного вікна програми представлений на рисунку 7.5. В ньому головним об'єктом є вікно, а

центральним віджетом – вкладки, кожна з яких має свій власний віджет, що реалізують визначений функціонал.

Модель труби(рис. 4) реалізована за допомогою OpenGL модуля бібліотеки Qt, котрий є нічим іншим як адаптацією графічної бібліотеки OpenGL до бібліотеки Qt, використовуючи переваги об'єктно-орієнтованого стилю програмування і популярною бібліотекою для роботи з комп'ютерною графікою. Ці дві бібліотеки крос-платформові, що дозволяє використовувати їх поєднання на великій кількості операційних систем. Різним може бути контекст рисування, що дозволяє виводити інформацію не тільки на монітор ПК, а й у графічний-файл, на принтер, у текстові файли(ODF, PDF), буфер обміну.

3.1 Опис класу GLWidget. Клас *GLWidget* успадкований від *QWidget*. Конструктор цього класу має такий вигляд:

```
GLWidget(PipeStruct pipeStruct, QWidget *parent = 0);
```

Структура *PipeStruct* представляє собою опис характеристик труби та бандажу. Вона включає наступні дані:

1. *innerRadius*– внутрішній радіус труби
2. *outerRadius*– зовнішній радіус труби
3. *bandageRadius* – радіус бандажної частини із зовнішнім радіусом
4. *pipeLength*– довжина труби
5. *bandageLength*– довжина бандажу

Другий параметр класу *GLWidget* говорить про те, що вінне лише успадкований від *QWidget*, а й використовує його у якості базового. Стандартне значення об'єкту класу *QWidget*, що дорівнює нулю говорить про те, що цей віджет є кореневим у структурі класів. Клас використовує механізм сигналів/слотів, а також подій, тому написаний макрос *Q_OBJECT* одразу після визначення класу.

Методи класу *GLWidget* дозволяють визначити положення труби у просторі, її зміщення відносно початкових координат, задати значення геометричних характеристик труби, а також отримати ці значення. Крім того за допомогою захищених методів класу *mousePressEvent* і *mouseMoveEvent* можна маніпулювати геометричною фігурою за допомогою мишки. Методи *initializeGL*, *paintGL*, *resizeGL* потрібні для відображення графічної інформації за допомогою бібліотеки OpenGL. Закриті методи включають методи створення графічної моделі труби – *makeGear*, *drawGear*, *normalizeAngle*. [5]

В методі *initializeGL* відбувається встановлення освітлення та створюються два елемента типу *GLuint* за допомогою *makeGear*, що представляють собою трубу та бандаж. В методі *paintGL* відбувається перерисовування у відповідності до поточних значень повороту за кожною з осей. В методі *resizeGL* створюється сам елемент на якому відображена модель, його початкові координати і геометричні розміри. Метод *makeGear* реалізує графічну модуль допомогою графічних примітивів бібліотеки OpenGL, а метод *drawGear* відображає модель.

Окрім об'єкту класу `GLWidget` у вкладці `PipeModel` розміщені елементи для вводу значень параметрів моделі. А також розміщена елемент-кнопка для перерисовування моделі у відповідності до визначених. Найдовший елемент вводу представляє собою строку в якій записана команда для подальшого запуску через консоль використовуючи сторонній процес. Команду можна створити самостійно або згенерувати у вкладці `Settings`.

Для запуску команд, що обробляють макрос в `ANSYS`, додаток використовує клас `QProcess` із бібліотеки `Qt`. Коли спрацьовує сигнал `returnPressed`, тобто користувач після редагування тексту натиснув `Enter`, починається запуск команди системою `ANSYS`, якщо команда і макрос правильні користувач отримує результат розрахунку. Процес запускається методом `start` або `execute`, перший запускає процес асинхронно, не блокуючи основну програму, `execute` блокує. Якщо логіка програми потребує блокування, крім запуску через `execute` можна також використати методи `waitForStarted` чи `waitForFinished` для очікування початку запуску процесу чи закінчення операції відповідно. Після закінчення виконання операції про це надсилає відповідні сигнали: `started` та `finished`. Можна також отримати статус виходу з процесу. Статуси можуть бути `NormalExit` (нормальний вихід) і `CrashExit` (аварійний вихід). [6]

Досить поширена операція – читання запущеного процесу. Багато програм спочатку створюються у якості консольних, а потім для них створюється графічний інтерфейс. Одне з можливих рішень при проектуванні полягає у створенні кількох процесів. Значна частина програм проектується саме таким чином. Прикладами таких програм написаних на `Qt` є мультимедіа плеєр `SMPlayer` та `bittorrent` клієнт `qtorrent`. Для читання процесу клас `QProcess` дає два розподілені канали: канал стандартного виводу (`stdout`) і канал помилок (`stderr`). Канали можна переключати за допомогою методу `setReadChannel`. Якщо канал може надати дані у встановленому каналі, то надсилається сигнал `readyRead`.

Клас `QIODevice` є базовим для будь-якого типу операцій вводу/виводу, включно з `QProcess`. Даний клас працює з усіма типами файлів, розглядаючи їх як бінарні, сонетами в мережах, буфером обміну, а також процесами. Це дає змогу працювати з даними процесу як зі звичайним файлом, зчитуючи чи записуючи інформацію. Таким чином, для обробки інформації можна використовувати методи `read`, `write`, `readLine`, `writeLine`, `getChar`. Під час оброблення великих об'ємів інформації важливо критично важлива відсутність так званих «зависань» [7]. Для того, щоб процес обробив до кінця подібну задачу в даному додатку використано статичний метод `qApp->processEvents()`;

Метод `processEvents` є методом класу `QApplication` – головного класу для роботи з додатками в `Qt`. `qApp` – глобальний вказівник, що вказує на унікальний об'єкт класу `QCoreApplication`. Аналогічний об'єкт можна отримати використавши метод `QCoreApplication::instance()`. Метод реалізований з допомогою відомого шаблону проектування одинак (`Singleton`) [8]. Клас

`QCoreApplication` запускає на виконання програму з графічним інтерфейсом, `QCoreApplication` потрібен для запуску кожної консольної програми.

3.2 Система налаштувань. Клас `Settings`. Клас `Settings` створено для збереження інформації про роботу даної програми. До такої інформації відносяться: геометричні розміри вікна, позиція вікна, стандартні розміри труби та бандажу, маршрути до вихідних файлів, директорія з файлами результатів, назва проекту для `ANSYS`, лог-файл, проте для кожного конкретного видадку роботи програми створюються свої власні налаштування. Команда також є у системі налаштувань, хоча вона і складається з окремих даних з цієї системи. Для створення класу агрегується клас бібліотеки `QtSettings`. Він дозволяє зберігати дані в реєстрі чи спеціальних файлах налаштувань (*.ini). Для створення подібного класу часто використовується шаблон проектування `singleton`, щоб мати доступ з будь якої частини програми, або під час використання пишеться ключове слово мови `C++` `extern`, за умови, що є дані в різних класах, які потрібно зберігати, а використання декількох класів налаштувань не бажане. [9] В даному проекті всю необхідну інформацію можна отримати безпосередньо з класу `MainWindow` і тому потрібно лише об'явити об'єкт класу `Settings` і використовувати його у методах `readSettings` і `writeSettings`. Останній запускається у деструкторі і саме під час виходу з додатку зберігає всю необхідну інформацію. Дані методи демонструють використання об'єктів класу `QSettings`. У конструкторі вказується назва компанії розробника та проекту. Метод `void QSettings::setValue` застосовується для запису налаштувань, а метод `QVariant QSettings::value` для зчитування. Початок та закінчення читання виділяють методами `beginGroup` та `endGroup`. Клас `Settings` реалізований як структура, тобто навмисне ігнорувалася інкапсуляція де це було необхідно. Всі дані зберігаються в цьому класі і тому для запису чи зчитування не потрібно створювати окремі методи.

Для редагування налаштувань створено п'ять слотів (методи, які можна обробляти за допомогою механізму сигналів/слотів `Qt`):

`selectAnsysDirectory` – директорія з файлом `ANSYS[version].exe`

`selectOutputDirectory` – директорія з файлами результатів

`selectSourceFile` – вибір макросу `ANSYS`

`changeProjectName` – зміна назви проекту

`changeLogName` – вибір файлу до якого відбуватиметься запис результатів розрахунку.

Після завершення налаштувань необхідно натиснути на кнопку `MakeCommand`. За допомогою сигналу `clicked` буде визвано слот `makeCommand`, що генерує команду для обробки макросу в `ANSYS`. Ця команда змінює поточну команду, спочатку очищаючи її, заповнюючи необхідними значеннями з системи налаштувань і наприкінці змінює значення директорії в котрій система відображення результатів розрахунку шукає зображення та змінює рядок для запуску через консоль на актуальний (згенерований).

3.3 Опис класів Viewer і ModelViewer. Viewer – система відображення результатів. Клас Viewer (рис. 5) представляє собою головний клас за допомогою якого додаток відображає результати. Жодних особливих функцій відображення немає, система представляє собою звичайний переглядач зображень зі списком усіх файлів-зображень у лівій частині. Система має за предка клас QWidget. На ньому розташовано основні об'єкти. Відображення списку відбувається за рахунок класу QListView.

Перша функція говорить про те, що через даний об'єкт редагування елементів моделі даних заборонено. Друга вказує режим перегляду моделі даних, у випадку даної програми використовується спискова модель. Цих режимів всього два, інший – режим піктограм, що добре виглядає, наприклад, у файлової системи.

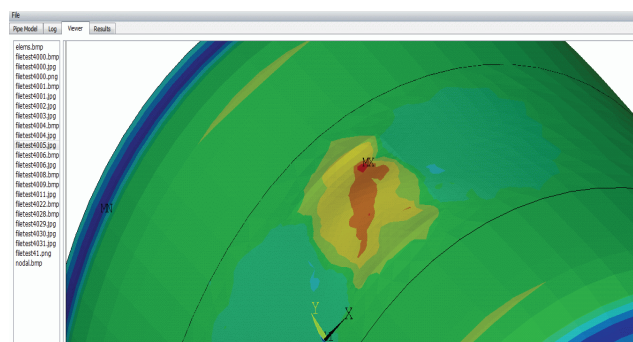


Рисунок 5 – Система відображення результатів

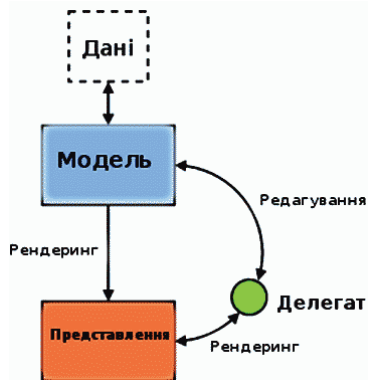


Рисунок 6 – Qt-версія шаблону проектування MVC

Реалізація системи відображення побудована за шаблоном проектування інтерв'ю або як ще його називають модель/представлення (MVC, ModelViewController) [10]. В Qt широко застосовується цей шаблон проектування (рис. 6). Його суть полягає у відмежуванні представлення від моделі даних, що дозволяють використовувати одну і ту ж саму модель з різними формами представлення та не змінювати представлення при редагуванні даних, якщо звісно редагування не веде до зміни структури даних. Інформація про зміни в моделі даних (додавання, видалення, редагування) автоматично змінює представлення. За кожен окремий елемент моделі відповідає так званий делегат, котрий можна змінити, надавши системі відображення вигляду, що необхідний програмісту. Модель даних у випадку цього додатку використовує

спискову модель (клас QListModel), що повністю відповідає шаблону проектування інтерв'ю.

При натисненні на якийсь елемент у списку відбувається зміна попереднього зображення на зображення, що відповідає натиснутому елементу. Перехід відбувається за допомогою системи сигналів/слотів. При зміні директорії у об'єкті класу MainWindow методом makeCommand відбувається виклик слоту об'єкту Viewer і цим самим змінює директорію пошуку зоб

Оцінка отриманих результатів. Програма дозволила отримати наступні результати розрахунків:

Під час розрахунків отримали напруження в місці корозійного пошкодження (Рис. 7) в 30 МПа.

Отримані результати свідчать про зменшення напружень, що виникають в трубі при приєднанні бандажа (рис. 8).

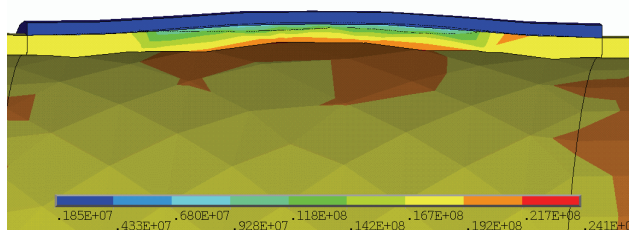


Рисунок 7 – Контурний графік місця деформації у розрізі

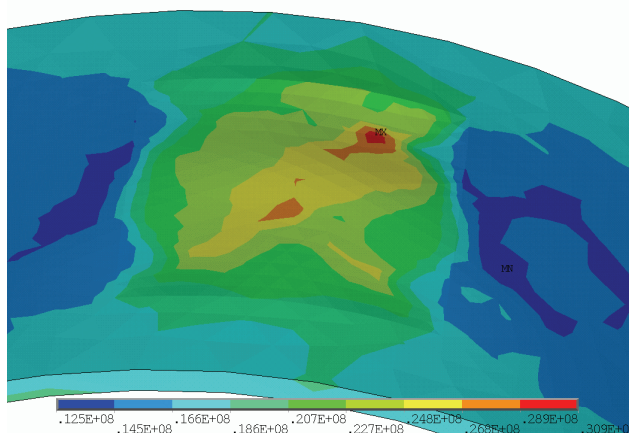


Рисунок 8 – Контурний графік деформованої частини

Без бандажа та філера напруження становлять 30,9 МПа (рис. 7), при приєднанні бандажа вони складають 24,1 МПа (рис. 8).

Напруження, що діють на філер складають 8,45 МПа, а на бандажну частину 3,1 МПа. Це демонструє зменшення напруження у напрямку від місця деформації.

Через випадковий характер поверхні, створюваної під час роботи макросу, отримані результати мають різні значення. Для тестування макросу була погіршена сітка SE. Результатом стало значне зменшення часу розрахунку (час зменшився в 50 разів), хоча отримані результати близькі до тих, що отримуємо з добре розбитою сіткою.

Висновки. Отримані результати розрахунків свідчать про зменшення напружень, що виникають в трубі при приєднанні бандажа.

Розроблено керуючу програму, що дозволяє у зручному для користувача вигляді застосовувати створені геометричну та розрахункову моделі для аналізу міцності фрагменту трубопроводу із корозійним пошкодженням, в тому числі посиленого композитним бандажем. Для розрахунку на міцність використовується САПР ANSYS в автоматизованому режимі. Дана програма дозволяє проводити велику кількість розрахунків автоматизовано, що дозволяє стверджувати, що розроблена керуюча програма, макроси та ANSYS утворюють інтегровану комп'ютерну систему.

Додаток дозволяє проводити розрахунки як з композитним бандажем, так і без нього. Є можливість оцінювання отриманих результатів.

Список литературы:

1. *Остсемин А.А.* К расчету предельного состояния бандажированных труб / *А.А. Остсемин, В.Ю. Заварухин* // Проблемы прочности. – 1990. – № 1. – С. 76-82.
2. *Львов Г.И.* О влиянии кинематических гипотез на характер контактного взаимодействия цилиндрической оболочки с бандажом / *Г.И. Львов, А.Н. Ткачук* // Вестник НТУ «ХПИ». – 2006. – Вып. 32. – С. 98-104.
3. *Буч Г.* UML 2.0 / *Г. Буч, А. Якобсон, Дж. Рамбо.* – СПб.: Питер, 2006. – 735 с.
4. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / *Г. Буч.* – М.: Бином, СПб.: Невский диалект, 1998. – 560 с.
5. *Бланишет Ж.* Qt 4. Программирование GUI на С++ (+ CD-ROM) / *Ж. Бланишет, М. Саммерфилд.* – М.: КУДИЦ-Пресс, 2008. – 718 с.
6. *Саммерфилд М.* Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на С++ / *М. Саммерфилд.* – М.: Символ-плюс, 2011. – 719 с.
7. *Шлее М.* Qt 4.8. Профессиональное программирование на С++ / *М. Шлее.* – М.: БХВ-Петербург, 2012. – 1473 с.

8. *Вандевурд Д.* Шаблоны С++: справочник разработчика: Пер. с англ. / *Д. Вандевурд, Н. М. Джосаттис.* – М.: Вильямс, 2003. – 544 с.

9. *Саттер Г.* Стандарты программирования на С++: Пер. с англ. / *Г. Саттер, А. Александреску.* – М.: Вильямс, 2005. – 224 с.

10. *Гамма Э.* Приемы объектно-ориентированного проектирования. Паттерны проектирования: Пер. с англ. / *Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес.* – СПб.: Питер, 2001. – 368 с.

Bibliography (transliterated):

1. Ostsemin A.A., Zavaruhin V.Yu. K raschetu predel'nogo sostoyaniya bandazhirovannyh trub. Problemy prochnosti. 1990. No 1. P. 76-82.
2. L'vov G.I., Tkachuk A.N. O vliyaniy kinematcheskikh gipotez na harakter kontaktного vzaimodejstviya cilindricheskoj obolochki s bandazhom. Vestnik NTU "KhPI". 2006. Vol. 32. PP. 98-104.
3. Buch G., Yakobson A., Rambo Dzh. UML 2.0. Sankt Petersburg: Piter, 2006. 735 p.
4. Buch G. Ob#ektno-orientirovannyj analiz i proektirovanie s primerami prilozhenij na S++. Moscow: Binom, Sankt Petersburg: Nevskij dialekt, 1998. 560 p.
5. Blanshet Zh., Sammerfild M. Qt 4. Programmirovaniye GUI na C++ (+ CD-ROM). Moscow: KUDIC-Press, 2008. 718 p.
6. Sammerfild M. Qt. Professional'noe programmirovaniye. Razrabotka krossplatformennyh prilozhenij na S++. Moscow: Simvol-plyus, 2011. 719 p.
7. Shlee M. Qt 4.8. Professional'noe programmirovaniye na C++. Moscow: BHV-Peterburg, 2012. 1473 p.
8. Vandevurd D., Dzhosattis N.M. Shablony C++: spravochnik razrabotchika: Per. s angl. Moscow: Vil'yams, 2003. 544 p.
9. Satter G., Aleksandresku A. Standarty programmirovaniya na S++: Per. s angl. Moscow: Vil'yams, 2005. 224 p.
10. Gamma E., Helm R., Dzhonson R., Vliissides Dzh. Priemy ob#ektno-orientirovannogo proektirovaniya. Patterny proektirovaniya: Per. s angl. Sankt Petersburg: Piter, 2001. 368 p.

Надійшла (received) 19.10.2017

Бібліографічні описи / Библиографические описания / Bibliographic descriptions

Спеціалізована система автоматизованого проектування та аналізу міцності ремонтних з'єднань трубопроводу / Є.В. Дудник, Г. І. Львов // Вісник НТУ «ХПІ». Серія: Динаміка і міцність машин. – Х.: НТУ «ХПІ», 2017. – № 40 (1262). – С. 26-31. – Бібліогр.: 10 назв. – ISSN 2078-9130.

Спеціалізована система автоматизованого проектування та аналізу міцності ремонтних з'єднань трубопроводу / Є.В. Дудник, Г. І. Львов // Вісник НТУ «ХПІ». Серія: Динаміка і міцність машин. – Х.: НТУ «ХПІ», 2017. – № 40 (1262). – С. 26-31. – Бібліогр.: 10 назв. – ISSN 2078-9130.

Specialized system of automated designing and strength analysis of pipeline repair connections / E.V.Dudnik, G.I.L'vov // Bulletin of NTU "KhPI". Series: Dynamics and strength of machines. – Kharkiv: NTU "KhPI", 2017. – № 40 (1262). – P. 26-31. – Bibliogr.: 10. – ISSN 2078-9130.

Відомості про авторів / Сведения об авторах / About the Authors

Дудник Євгеній Володимирович – аспірант; кафедра Динаміки та міцності машин; НТУ «ХПІ»; тел.: (099) 624-72-45; e-mail: martynenko.volodymyr@gmail.com.

Дудник Євгеній Володимирович – аспірант; кафедра Динаміки та міцності машин; НТУ «ХПІ»; тел.: (099) 624-72-45; e-mail: martynenko.volodymyr@gmail.com.

Dudnik Yevhenii Volodymyrovych – Postgraduate student; Dynamic and Strength of Machines Department; NTU "KhPI"; тел.: (099) 624-72-45; e-mail: martynenko.volodymyr@gmail.com.

Львов Геннадій Іванович – доктор технічних наук, професор, завідувач кафедри Динаміки та міцності машин; НТУ «ХПІ»; тел.: (057) 707-63-43; e-mail: lvovdpm@ukr.net.

Львов Геннадий Иванович – доктор технических наук, профессор, заведующий кафедры Динамики и прочности машин; НТУ «ХПІ»; тел.: (057) 707-63-43; e-mail: lvovdpm@ukr.net.

L'vov Gennadii Ivanovych – Doctor of Technical Sciences, Full Professor, Head of the Dynamic and Strength of Machines Department, NTU "KhPI", тел.: (057) 707-63-43; e-mail: lvovdpm@ukr.net.