

Міністерство освіти і науки України  
Державний заклад  
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

**Крижановський Максим Ігорович**

**РОЗРОБКА ІНСТРУМЕНТУ ДЛЯ ГЕНЕРАЦІЇ ВХІДНОГО ФАЙЛУ  
ДЛЯ CAD-СИСТЕМИ З ВИКОРИСТАННЯМ МОВИ APDL**

**кваліфікаційна робота  
здобувача вищої освіти другого (магістерського) рівня  
освітньої програми «Мультимедійні системи»  
за спеціальністю 121 Інженерія програмного забезпечення**

Особистий підпис \_\_\_\_\_ Максим КРИЖАНОВСЬКИЙ

Науковий керівник \_\_\_\_\_ Світлана ДОНЧЕНКО,  
асистент кафедри інформаційних  
технологій та систем

В.о. завідувача кафедри \_\_\_\_\_ Микола СЕМЕНОВ,  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

## **АНОТАЦІЯ**

**Крижановський М. І.**

**Тема:** Розробка інструменту для генерації вхідного файлу для CAD-системи з використанням мови APDL.

**Спеціальність:** 121 «Інженерія програмного забезпечення».

**Установа:** ЛНУ імені Тараса Шевченка, 2024р.

**Магістерська робота містить:** 64 с., 3 таб., 12 рис., 52 джерел.

**Об'єкт дослідження** – CAD-системи з використанням мови APDL.

**Предмет дослідження** – програмний засіб для генерації вхідного файлу для CAD-системи з використанням мови APDL.

**Мета роботи** – розробити програмний засіб, що генерує коректну геометрію майбутнього намотаного виробу в середовищі ANSYS та забезпечує можливості імпорту геометрії з поширених графічних форматів та її власноручну правку.

**Результати роботи** – проведено вивчення специфіки технологічного процесу намотування, наведено класифікацію виробів та схем намотування. Проаналізовано можливості пакету інженерного аналізу ANSYS щодо створення геометрії виробів, внутрішньої мови APDL та особливостей прямого імпорту стосовно наступного аналізу намотаного тіла.

У практичній частині розроблено програму – генератор вхідного файлу для програми ANSYS на її мові APDL. Запуск цього файлу в ANSYS в автоматичному режимі будує геометрію намотаного виробу, яка є коректною стосовно подальшого аналізу фізико-механічних властивостей. Програма-генератор створена засобами програмного середовища Delphi.

**Ключові слова:** КОМПОЗИЦІЙНИЙ МАТЕРІАЛ, ПРОЦЕС НАМОТУВАННЯ, МОВА APDL, ПРОГРАМА ANSYS, ІНТЕГРОВАНІ СИСТЕМИ, КІНЦЕВИЙ ЕЛЕМЕНТ.

## ANNOTATION

**Kryzhanovskyi Maksym**

**Theme:** Development of a tool for generating an input file for a CAD system using the APDL language.

**Speciality:** 121 "Software Engineering".

**Institution:** Luhansk Taras Shevchenko National University (LTSNU), 2024 year.

**Master's work of:** 64 p., 12 im, 52 sources.

**The object of research** is CAD systems using the APDL language.

**The subject** of the study is a software tool for generating an input file for a CAD system using the APDL language.

**An aim of research is** - to develop a software tool that generates the correct geometry of the future coiled product in the ANSYS environment and provides the ability to import geometry from common graphic formats and edit it manually.

**Job performanes** - The specificity of the winding process is studied, the classification of winding products and schemes is given. The ANSYS engineering analysis package has been analyzed to create product geometry, APDL internal language, and direct import features for subsequent winding analysis.

The practical part is an application program - an input file generator for ANSYS in its APDL language. Running this file in ANSYS automatically builds the geometry of the wound product, which is correct for further analysis of the physical and mechanical properties. The generator program is created by means of Delphi software environment.

**Keywords:** COMPOSITION MATERIAL, WINDING PROCESS, APDL LANGUAGE, ANSYS PROGRAM, INTEGRATED SYSTEMS, FINAL ELEMENT.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>6</b>
<b>РОЗДІЛ 1. ПРОБЛЕМИ ВИГОТОВЛЕННЯ ВИРОБІВ З ПОЛІМЕРНИХ КОМПОЗИЦІЙНИХ МАТЕРІАЛІВ .....</b>	<b>11</b>
1.1. Історичні відомості .....	11
1.2. Застосування виробів із КМ в різних галузях науки й техніки .....	12
1.3. Проблеми створення виробів із ПКМ .....	17
1.4. Шляхи розв'язання проблеми підвищення ефективності застосування ПКМ .....	22
1.5. Загальна характеристика й класифікація методу намотування .....	23
1.6. Етапи технологічного процесу виготовлення виробів методом намотування .....	25
1.7. Класифікація видів намотування й намотуваних виробів .....	27
1.8. Висновки до розділу .....	30
<b>РОЗДІЛ 2. ЗАСОБИ МОДЕЛЮВАННЯ ВИРОБІВ, ОТРИМАНИХ МЕТОДОМ НАМОТУВАННЯ.....</b>	<b>32</b>
2.1. Огляд засобів моделювання .....	32
2.2. Введення в ANSYS.....	34
2.2.1. Призначення ANSYS.....	34
2.2.2. Інтерфейс користувача.....	34
2.2.3. Графічні можливості .....	36
2.2.4. Процесори .....	38
2.2.5. База даних.....	39
2.2.6. Формат файлів .....	40
2.2.7. Внутрішня мова APDL.....	41
2.3. Огляд мови APDL.....	41
2.3.1. Умовчання .....	42
2.3.2. Попередження й помилки.....	42
2.3.3. Інтерактивний режим у порівнянні з пакетним режимом.....	42
2.3.4. Робота LOG файлу.....	43
2.4. Типові види аналізів в ANSYS.....	44
2.4.1. Створення геометрії моделі.....	45
2.4.2. Додавання навантажень і одержання рішень .....	45
2.4.3. Перегляд результатів.....	45

2.5. Аналіз геометрії, що враховується при аналізі намотаних тіл .....	46
2.6. Висновки до розділу .....	48
<b>РОЗДІЛ 3. СТВОРЕННЯ ГЕНЕРАТОРУ ВХІДНОГО ФАЙЛУ ДЛЯ</b>	
<b>CADСИСТЕМИ ANSYS НА МОБІ APDL .....</b>	<b>49</b>
3.1. Аналіз кінцевого результату програмного засобу .....	49
3.1.1. Перший етап.....	49
3.1.2. Другий крок.....	51
3.1.3. Третій етап .....	51
3.2. Розробка програмного засобу .....	52
3.2.1. Організація даних .....	52
3.2.2. Організація інтерфейсу .....	53
3.3. Висновки до розділу .....	56
<b>ВИСНОВКИ .....</b>	<b>57</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>59</b>
<b>Додаток А. ....</b>	<b>64</b>
Зміст модуля uMainForm .....	64
Зміст модуля uForm2f .....	66
Зміст модуля uForm2s .....	69
Зміст модуля uForm3 .....	73
Зміст модуля uGraph .....	78
Зміст модуля uSource .....	79

## ВСТУП

В останні десятиліття людство все ширше й різнобічніше використовує у своїй діяльності композиційні матеріали (часто їх називають просто композитами), що утворюються шляхом об'ємного сполучення хімічно різнорідних речовин (компонентів). З них за допомогою різних процесів одержують нові матеріали складної структури з особливими фізико-хімічними властивостями.

Як вважають учені, одержання нових полімерних композитів із заданими властивостями є в цей час однією з головних завдань сучасної науки. Створювані за допомогою сучасної технології нові матеріали й речовини в країнах НАТО використовуються в конструкціях зразків військової техніки і зброї. Приклад тому – використання матеріалів, одержуваних на основі ароматичних поліамідів, синтетичних волокон типів номекс та кевлар [1].

Композиційні матеріали (КМ) знаходять більш широке застосування в тих галузях науки й техніки, де від виробів потрібні високі міцносні, корозійні або хімічно інертні властивості. Ще перед другою світовою війною, в 40-х рр. був досягнутий істотний прогрес в області КМ, і особливо армованих пластиків (АП) [2]. Тоді були створені перші високоміцні композити, які на перших етапах застосовувалися для модернізації літальних апаратів військового призначення. Саме впровадження АП в авіаційну техніку породило нову технологію – технологію композиційних матеріалів. Подальший розвиток промисловості, що випускає АП, привело до того, що зараз не існує області техніки, де не застосовувалися б ці матеріали. В 1990 р. загальний світовий обсяг споживання полімерних композиційних матеріалів (ПКМ) склав приблизно 3 млн.т.

Композиційні матеріали (армовані пластики, зокрема) – це матеріали, що складаються, як правило, із двох компонентів (армуючого матеріалу та сполучної матриці) та мають специфічні властивості, відмінні від сумарних властивостей складових їхніх компонентів.

Найпоширенішими армуючими компонентами при створенні композиційних матеріалів є скляні, поліамідні, азбестові волокна, папір (целюлозні волокна), бавовна, сизаль, джут й інші натуральні волокна. Все більше місце в технології виробництва композитів посідають такі матеріали, як вуглецеві, графітові, борні, сталеві волокна й вуси (дуже короткі армуючі волокна, як зазвичай кристалічні). Вибір того або іншого армуючого наповнювача визначається ціною, складом і технологічними вимогами, пропонованими до властивостей АП.

Головним елементом будь-яких структур армованих пластиків є односпрямований шар, що складається з паралельно орієнтованих армуючих елементів волокон, ниток, джгутів, стрічок або полотен.

Одним з основних напрямків використання таких КМ є виробництво ракет, літаків, надводних і підводних обшивань кораблів. Скловолокно є основним видом армуючого матеріалу для зміцнення різних пластиків, тому що його вартість невисока. Вуглецеві, борні й арамідні волокна значно дорожче скловолокна, але перевершують його по фізико-механічних властивостях. Останні застосовуються переважно у військовій й авіакосмічній промисловості [41;47].

У короткому огляді [6] наведено висновок, що перспективними будуть композити не стільки з високими властивостями, скільки з досить низькою вартістю виготовлення. Для багатьох областей техніки високі властивості матеріалів бажані, але це не є головним критерієм їхньої застосованості. Як правило, матеріалів з необхідним рівнем властивостей досить у розпорядженні, а проблема полягає в тім, щоб виготовити виріб досить швидко й з низькою вартістю.

В [7] зазначено п'ять найбільш перспективних технологічних процесів для переробки полімерних композитів: лиття, лиття під тиском, намотування ниткою, термоформування та пултрузія.

З перспективних методів метод намотування займає провідну позицію у виробництві високоміцних і розміростабільних виробів у військовій й аерокосмічній промисловості по наступних причинах [7].

По-перше, метод намотування – один з методів, що дозволяють створювати високоорієнтований АП. По-друге, він дозволяє створювати вироби, що складаються з декількох односпрямованих шарів різними напрямками АМ у мікросшарах. Різні способи армування в шарах дають можливість створювати вироби з високоміцними властивостями в заданих умовах експлуатації. По-третє, просочення АМ сполучиною відбувається або завчасно, або в процесі формування стрічки при укладанні її на оправлення. При цьому ступінь просочення АМ контролюється до формування КМ, що також веде до підвищення фізико-механічних властивостей готового виробу. На відміну від пултрузії вироби можуть бути армовані не тільки уздовж головної осі (тіла обертання), а на відміну від методів формування на матриці є помітна перевага високої орієнтації волокон, що істотно підвищує як міцність КМ, так і стабільність властивостей готових виробів.

Однак цей метод несе в собі цілий ряд труднощів, пов'язаних з високою технологічністю процесу [23]. Однією з перших стоїть проблема визначення параметрів укладання АМ, що забезпечили б необхідні фізико-механічні властивості виробу в певних умовах експлуатації.

Для розв'язання цієї проблеми необхідне залучення цілого ряду наукових дисциплін, що включають фізичні науки й технологічні розробки. Оскільки процеси створення нового КМ і процес формування виробу з'єднані, необхідна спільна робота багатьох фахівців: технологів, експертів по міцності, матеріалознавців [25]. Розробка нового виробу – це нова розробка матеріалу, нова розробка технології, нова розробка допоміжних пристроїв тощо. Існує два основних шляхи:

- проведення великої кількості натурних експериментів, що дозволило б виявити закономірності між властивостями готового виробу, властивостями КМ і параметрами процесу виготовлення;



- розробка алгоритмів і моделей, що враховують технологічні й конструктивні особливості конкретного виробу.

Перший шлях є досить праце- та матеріалоемним. Потрібне залучення різних фахівців. Більше того, такий підхід може бути як недоступний самому підприємству, що займається виробництвом, так і недоцільний у плані строків виконання замовлення.

Другий шлях – розробка алгоритмів і моделей – дозволить як уникнути більших матеріальних і тимчасових витрат, так і залучення великого ряду фахівців. Таким чином, існує необхідність автоматизації етапу підготовки виробництва, ціль якої відшукування технологічних параметрів процесу намотування виробу із заданими конструктивними й фізико-математичними властивостями.

Математичні моделі, реалізовані за допомогою сучасних обчислювальних засобів, дозволяють досить просто враховувати такі фактори, як наявність дискретних і безперервних елементів, нелінійні характеристики елементів системи, численні випадкові впливи й ін., які часто створюють труднощі при аналітичних дослідженнях.

Завдання створення математичної моделі щодо методу намотування є складною й обширною. За деякими дослідженнями існуючі програмні засоби інженерного аналізу не відповідають вимогам процесу намотування. Тому актуальною є задача створення програмних оболонок, що поєднували б можливості сучасних систем інженерного аналізу та твердотілого проектування з вимогами та особливостями методу намотування та виробів, отриманих в його процесі.

У даній роботі пропонується вирішення першого кроку названого завдання – питання коректного створення/імпорту геометрії майбутнього виробу, для подальшого аналізу. За основу взято систему інженерного аналізу ANSYS [8] як програмний комплекс не залежний від програмно-апаратної платформи, який налічує багату бібліотеку можливостей фізико-механічного аналізу, в якого є засоби моделювання багатопарових КМ.

**Об’єкт дослідження** – CAD-системи з використанням мови APDL.

**Предмет дослідження** – програмний засіб для генерації вхідного файлу для CAD-системи з використанням мови APDL.

**Мета роботи** – розробити програмний засіб, що генерує коректну геометрію майбутнього намотаного виробу в середовищі ANSYS та забезпечує можливості імпорту геометрії з поширених графічних форматів та її власноручну правку.

**Для досягнення поставленої мети необхідно вирішити наступні завдання:**

- 1) проаналізувати проблеми виробів з КМ методом намотування;
- 2) зробити аналіз можливостей системи ANSYS на предмет створення/імпорту геометрії;
- 3) створити програмний засіб, що генерує коректну геометрію майбутнього намотаного виробу в середовищі ANSYS та забезпечує можливості імпорту геометрії з поширених графічних форматів та її власноручну правку.

У першому та другому розділі магістерської роботи надано аналіз специфіки технологічного процесу намотування, класифікацію виробів та схем намотування. Проаналізовано можливості пакету інженерного аналізу ANSYS, внутрішньої мови APDL щодо можливості імпорту геометрії з поширених графічних форматів.

У третьому розділі розглядається розробка програми – генератора вхідного файлу інженерної системи ANSYS для коректної побудови геометрії виробу, отриманого методом намотування. Програма-генератор створена засобами програмного середовища Delphi.

# РОЗДІЛ 1. ПРОБЛЕМИ ВИГОТОВЛЕННЯ ВИРОБІВ З ПОЛІМЕРНИХ КОМПОЗИЦІЙНИХ МАТЕРІАЛІВ

## 1.1. Історичні відомості

Перші армовані скляними волокнами пластики для авіаконструкцій були задумані й зроблені в лабораторії конструкцій і матеріалів Райт-патерсоновської військово-повітряної бази (м. Дейтон, шт. Огайо) в 1943 р. [4]. Після аналізу результатів випробувань армованих матеріалів теоретичні розрахунки показали, що при проектуванні й створенні конструкцій можуть бути використані високоміцні композиції на основі поліефірних смол, армованих скловолокном, для складних структур із серцевиною з легкого матеріалу. У військовому літакобудуванні такі матеріали добре зарекомендували себе в різних сендвічевих конструкціях. Задня частина фюзеляжу двомісного навчального літака «Вальті 5М-15» була посилена склопластиком.

Статичні випробування першого виготовленого фюзеляжу показали більш високі характеристики, ніж ті, які були передбачені. Використовуючи переваги в міцності й масових характеристиках склопластів, вдалося на 50 % підвищити властивості міцності сендвічевих конструкцій у порівнянні із властивостями металів або дерева. Крім гарних конструкційних властивостей сендвічеві конструкції з АП показали відсутність значних зрушень шарів склопластику при більших навантаженнях, що скручують. Якщо сендвічева конструкція з облицюванням з алюмінію може витримати 100 % розрахункового навантаження, то візуальні й приладові спостереження за аналогічною конструкцією з композиційних волокнистих матеріалів показали, що цей матеріал витримує до 180 % розрахункового навантаження. Через недостатньо гарні вихідні дані для проектування, якими володіє дерево, ВПС США завжди вважали дерево небажаним матеріалом у конструкціях. Було виготовлено з використанням склопластиків зі стільниковим заповненням три різних типи фюзеляжів. Склопластики, використовувані для цієї мети, мали наступні властивості: межа міцності при розтяганні 276 МПа, при стиску

234 МПа, при вигині 393 МПа, при зрушенні 131 МПа; модуль пружності при вигині 18,7 ГПа; щільність 1800 кг/м<sup>3</sup>. Теоретичне значення питомої міцності (відношення міцності до щільності матеріалу) було істотно вище, ніж для алюмінієвих сплавів або спеціальних сталей, використовуваних у конструкціях. На жаль, конструкційні можливості СПКМ виявляються обмеженими внаслідок порівняно низького модуля пружності (магнієві сплави з такою ж щільністю мають модуль пружності рівний 44,2 ГПа).

Природним розв'язанням цієї проблеми було б використання АП у покриттях для основної, стабільної структури матеріалу, що мав би й гарну міцність при стиску. Для створення такої структури були розроблені стільникові й сендвічеві конструкції. Дві високоміцних пластини розділяються дуже легкою ( $1700 - 2800 \cdot 10^3$  кг/м<sup>3</sup>) серцевиною-заповнювачем. Зв'язок кожної пластини із заповнювачем дуже міцний та суцільний. Заповнювач працює на розтягання, стиск і зрушення, підтримуючи зовнішні пластини.

## **1.2. Застосування виробів із КМ в різних галузях науки й техніки**

З того часу КМ знайшли й продовжують знаходити широке застосування в різних галузях науки й техніки. У таблиці 1.1 приводяться галузі застосування КМ (армованих волокнами, зокрема) і характерні вироби з них.

В останні десятиліття виявляється значний інтерес до впровадження КМ у військову й авіакосмічну промисловість як перспективних матеріалів. Однак разом з тим до КМ в авіації й космонавтиці пред'являються підвищені вимоги до характеристик конструкцій із КМ. До них ставляться:

- високі міцносні характеристики при малій вазі;
- високостабільність і надійність властивостей конструкції;
- довговічність.

**Характерні вироби й вимоги до композиційних матеріалів у різних галузях діяльності [20;28;36]**

Галузь	Характерні вироби	Вимоги до властивостей КМ
Суднобудування	корпуси яхт, катерів, гребних човнів, гідроциклів, малогабаритних судів, рятувальних шлюпок, корпуси, щогли, надбудови, резервуари, обтічники, буї	<ul style="list-style-type: none"> <li>- високе співвідношення міцносних характеристик до маси;</li> <li>- довговічність і стійкість до водного середовища;</li> <li>- відносна простота експлуатації й ремонту;</li> <li>- низька (у порівнянні з металами) теплопровідність;</li> <li>- низький тепловий коефіцієнт лінійного розширення;</li> <li>- широкий діапазон робочих температур;</li> <li>- гарні електроізоляційні властивості.</li> </ul>

Галузь	Характерні вироби	Вимоги до властивостей КМ
Автомобілебудування	підкрилки, бампери, накладки, антикрила, фургони, у тому числі ізотермічні. А також передні й задні панелі кузова автобусів, тролейбусів, трамваїв, елементи внутрішнього інтер'єра тощо.	<ul style="list-style-type: none"> <li>- зниження загальної маси автотранспортного засобу;</li> <li>- зменшення товщини деталей;</li> <li>- заміна сполучення деталей однією деталлю;</li> <li>- зниження кількості операцій при виготовленні різних конструкцій;</li> <li>- термостійкість;</li> <li>- зносостійкість;</li> <li>- неохильністю корозії й гниттю.</li> </ul>

Галузь	Характерні вироби	Вимоги до властивостей КМ
Будівництво, комунальне господарство	підвіконня, плити оздоблювальні, двері, віконні плетіння, сходи, поруччя, огороження балконів, водостічні жолоби, кіоски, зупинки суспільного транспорту. Контейнери для сміття, ящики для піску, ящики для води, мобільні санвузли, телефонні будки, басейни, ємності для рідких відходів, дитячі майданчики, атракціони, аквапарки. Резервуари для корму худоби, напівпрозорі покрівельні аркуші для оранжерей, теплиць, промислових будинків, плафони для вуличного освітлення, ремонт трубопроводів, рекламні тумби й щити	<ul style="list-style-type: none"> <li>- забезпечення необхідних деформаційно-міцносних властивостей;</li> <li>- гігієнічність;</li> <li>- термомеханічна стабільність;</li> <li>- зносостійкість;</li> <li>- волого- та світлостійкість;</li> <li>- широкий інтервал робочих температур.</li> </ul>

Галузь	Характерні вироби	Вимоги до властивостей КМ
Залізничний транспорт і метро	<p>внутрішнє й зовнішнє облицювання вагонів, сидіння, ціпки багажні, санвузол, віконні лиштви, поручні, коробки для електропроводів, кожух контактної рейки, шафи для апаратур, двері, корпуси товарних вагонів</p> <p>лиття – практично все;</p> <p>намотування – відповідальні частини осесиметричної форми</p>	<p>- руйнування конструкції повинне відбуватися передбачуваним образом у режимі пластичної деформації з поглинанням великої кількості енергії при заздалегідь певнім значенні прикладеного зусилля</p> <p>- величина цього зусилля повинна залежати від місця знаходження конструкції;</p> <p>- бажано використовувати в конструкціях відносно недорогі композиційні матеріали.</p>
Сантехніка й меблі	<p>ванни, душові піддони, раковини зі склопластику. Посилення склопластиком акрилових ванн, піддонів, душових кабін. Ванни, піддони, раковини, унітази з імітацією під камінь (малахит, граніт, мармур і т.д.), стільниці для журнальних, обідніх столів, кухонні стільниці з мийками й т.д.</p>	<p>- влаго- та світлостійкість;</p> <p>- гігієнічність;</p> <p>- неохильністю корозії й гниттю.</p>



Галузь	Характерні вироби	Вимоги до властивостей КМ
Хімічні виробництва	ємності для зберігання й транспортування хімічно активних речовин, трубопроводи, трапи	<ul style="list-style-type: none"> <li>- високою міцністю при малій вазі;</li> <li>- підвищеною стійкістю до впливу хімічно активних середовищ;</li> <li>- високої атмосферостійкістю, неохильністю корозії й гниттю;</li> <li>- відносно невеликою вагою виробів (дозволяє одержувати значну економію на транспортованих, вантажно-розвантажувальних і монтажних роботах).</li> </ul>

### 1.3. Проблеми створення виробів із ПКМ

Загальноспецифічними особливостями, властивим всім виробам з армованих пластиків є [14;15]: з'єднання в одному технологічному процесі створення матеріалу й конструкції, наявність різноманітних механізмів руйнування [23], які не завжди приводять до повного вичерпання несучої здатності виробу, велика розмаїтність компонентів [25;26;27] тощо. Однак, для малогабаритних конструкцій варто виділити додаткові особливості, які й визначають необхідність їхнього вивчення як самостійного класу. Це сумірність розмірів вихідних армуючих елементів з товщиною стінки виробу, підвищена чутливість характеристик виробу до відхилення технологічних параметрів від оптимальних, вплив конструктивних особливостей устаткування на параметри виробу, конкурентноздатність із аналогічними металевими, тільки при оптимальному співвідношенні вихідних компонентів

КМ, оптимальній структурі й оптимальному виготовленню [22, 23, 24, 25]. Всі ці особливості приводять до того, що створення виробів з армованих пластиків приймає форму безперервного процесу.

Наукові дослідження, проектування, виготовлення зливаються в єдину систему. Як відзначено в роботі [38] очевидна необхідність системного підходу, як до процесу розробок у цілому, так і до окремих його фрагментів.

Визначення шляхів підвищення параметрів якості виробів з АП багато в чому залежить від обраної схеми процесу їхнього створення й принципів, які визначають протікання самого процесу створення.

Методи системного аналізу застосовуються до завдань проектування окремих типів виробів порівняно недавно [39]. У той же час роль системного підходу залишається актуальною при створенні виробів із КМ. Це обумовлено тим, що сполучення в єдиному технологічному процесі виготовлення матеріалу й конструкції значно ускладнює етап проектування, що пов'язаний з необхідністю обліку технологічних обмежень, впливу параметрів виготовлення на властивості матеріалу, а отже й на конструкцію в цілому [40].

Деякі найбільш прості схеми процесу конструювання деталей із КМ наведені в [41;42]. У роботі [43] конструкції із КМ рекомендується розробляти в чотири етапи: попереднє вивчення умов роботи деталі, конструювання й розрахунок, виготовлення експериментального зразка і його оптимізація, кожний з яких розбивається на безліч підетапів. Досить різнобічно процес проектування КМ у конструкції розглянутий у роботах [44;45], де автори роблять висновок про необхідність співдружності розраховувачів, конструкторів, технологів і матеріалознавців на всіх етапах проектування.

Системність підходу в перерахованих вище роботах полягає в тому, що проектування розглядається, як система взаємозалежних елементів, спрямованих на досягнення глобальної мети. У той же час, такі основні відмінності системного підходу, як ітеративність процесів, взаємозв'язок проектування з іншими етапами створення, у них висвітлені недостатньо.

З позицій системного аналізу в роботі [48] викладений підхід до вирішення завдання конструктивно-технологічної оптимізації виробів із КМ. Розглядається два ієрархічних рівні: верхній – конструкторський і нижній – технологічний, що використовують свої критерії оцінки якості, що досягають локальних екстремумів і взаємодіють між собою для підвищення показників якості. Однак, не викладені методологічні й організаційні принципи, які лежать в основі взаємодії між різними підрозділами, часто визначають економічність застосування КМ, фактичні строки розробки нових виробів й ефективність різних форм організації виробництва виробів із КМ.

У роботі [49] найбільш повно сформульована сутність системного підходу. У ній визначений ряд взаємозалежних аспектів, що становлять у сукупності і єдності системний підхід. Це такі аспекти як:

- системно-елементний, що показує з яких компонентів утворена система;
- системно-структурний, що відбиває внутрішню організацію системи;
- системно-функціональний, що відповідає на питання, які функції виконує система і її компоненти;
- системно-комунікаційний, що розкриває зв'язок системи з іншими системами;
- системно-інтегративний, що виявляє механізми й фактори збереження системи, її вдосконалювання й розвиток;
- системно-історичний, що відповідає на питання як виникла система, як вона розвивалася і яка її перспектива.

Якщо в процесі розв'язання прикладних завдань будуть розглянуті не всі з перерахованих аспектів, то навіть сам факт знаходження реальної системи, визначення її як конкретного взаємозалежного цілого створює певну логіку й методологію її якісного дослідження. Уже це одне виправдовує застосування системного підходу, як самостійного методологічного засобу.

Узагальнений опис процесу створення виробів із КМ, як системи і її формалізація на рівні блок-схеми дано в роботах [48;49]. Автор, на основі сформульованих понять «створення виробів із КМ» й «функціональні блоки» (ФБ) і критерію вибору останніх, виділяє поза залежністю від конкретної мети чотири ФБ «Системи створення виробів із КМ»:

- матеріалознавство вихідних компонентів КМ;
- механіка матеріалів і конструкцій із КМ;
- технологія виробництва КМ і виробів з них;
- машини й устаткування для виробництва виробів із КМ.

Це впливає з основного принципу створення конструкцій із КМ, взаємозумовленості процесів розвитку матеріалів, конструкцій і технології [38], які є наслідком ряду специфічних властивостей КМ, і накопиченого досвіду по створенню виробів [28;33;36;37;41].

Конструкція системи, крім елементів (функціональних блоків), які вона містить, повинна визначати й спосіб взаємного з'єднання цих елементів. На рис.1.1 представлена принципова схема системи створення виробів із КМ. Як видно, всі функціональні блоки взаємозалежні між собою. Ці зв'язки здійснюються за допомогою інформаційних (І), матеріальних (S) і енергетичних ( $E_n$ ) потоків. Показано, що матеріальні й енергетичні потоки існують постійно, а частина інформаційних потоків ( $I_{3-1}$ ,  $I_{3-2}$ ,  $I_{3-4}$ ) при існуючих підходах до створення найчастіше відсутня, або є мало ефективною й приводить до подовження строків розробки виробів із КМ, зниженню ефективності їхнього застосування. Аналіз і регулювання потоками інформації, з позицій цілей і завдань, поставлених перед системою дозволить звільнитися від одержання надлишкової інформації на рівні ФБ й, навпаки, розвивати дослідження, що сприяють розкриттю саме системних властивостей ФБ. Більші досягнення в одній з областей знань не можуть принципово вплинути на рівень якості виробів із КМ. Тільки комплексне використання й погоджений розвиток всіх ФБ може забезпечити успіх.

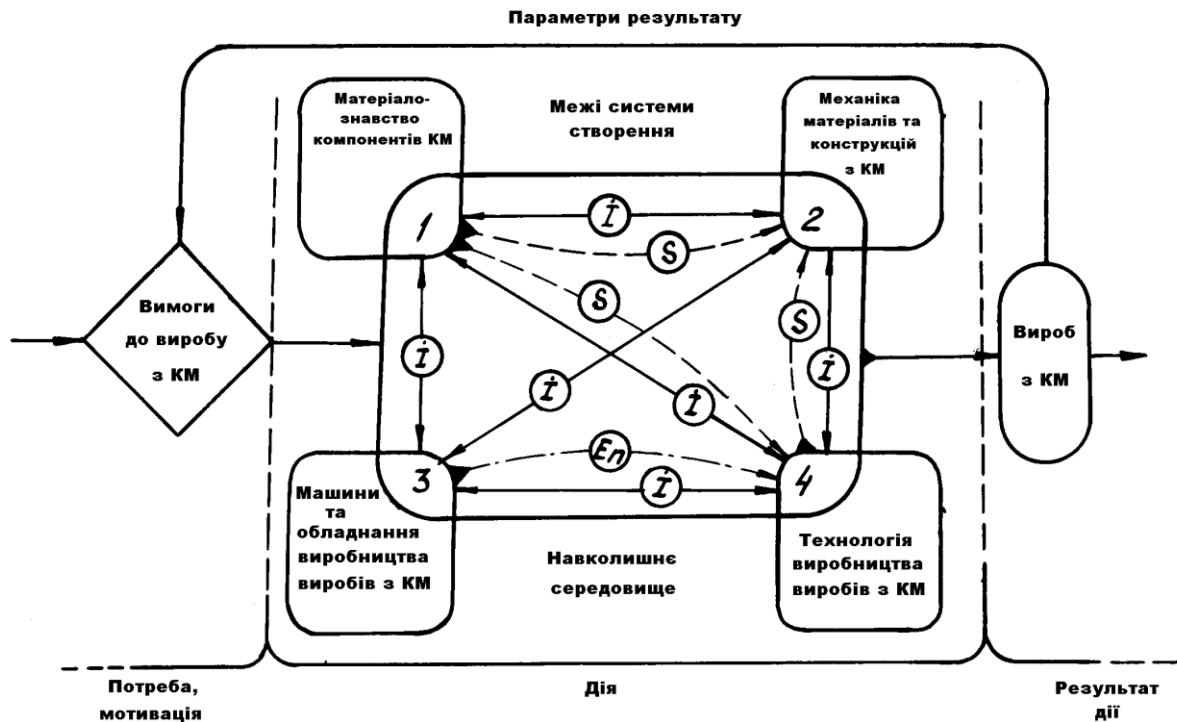


Рис. 1.1. Схема системи створення виробів із КМ

Через те, що більшість ФБ погано формалізуються (за винятком механіки матеріалів і конструкцій із КМ), практичного значення набуває побудова концептуальних і математичних моделей системи. Це дозволить на перших етапах не стільки точно описати роботу системи, скільки виявити можливі альтернативні шляхи досягнення поставлених перед нею цілей.

Універсальність схеми (рис.1.1) полягає в тому, що взявши за базу основні принципи створення виробів із КМ, створювану конструкцію й запропоновані до останньої вимоги, можливо наповнити новим конкретним змістом всієї ФБ системи «Створення виробів із КМ».

Що стосується властивостей виробів із КМ, то до існуючих невирішених проблем відносять одержання виробів (або конструкцій) з високою стабільністю властивостей. У першу чергу це відноситься до розміростабільності конструкцій. Це завдання вирішується більше повною

автоматизацією процесу виготовлення деталей із КМ, модифікацією конструктивних і технологічних особливостей процесу виготовлення.

У процесах виготовлення виробів із КМ, що найбільш автоматизуються, технологічні й конструктивні фактори впливають на характеристики готового виробу. Тому виникає наступна проблема – максимально повний облік цих факторів (технологічних і конструктивних) у ході автоматизації процесу виготовлення виробів із КМ.

#### **1.4. Шляхи розв’язання проблеми підвищення ефективності застосування ПКМ**

Застосовувані АП у різних областях промисловості не завжди відповідають тим вимогам, які пред’являються до матеріалів відповідної галузі. Сюди відносять вимоги забезпечення необхідних деформаційно-міцносних властивостей, термомеханічної стабільності, зносостійкості, волого- і світлостійкості, а також тепло- і морозостійкості. Однак їхні потенційні можливості й експлуатаційні ресурси можуть бути підвищені. Це досягається цілеспрямованим модифікуванням полімерних матеріалів.

Наповнювачі для полімерів у значній мірі впливають на властивості композицій. Властивості полімерних КМ залежать від природи, змісту, дисперсності й активності наповнювачів. При цьому високонаповнені полімерні композиції являють собою дисипативні системи, здатні до самоорганізації, що забезпечує оптимальні властивості матеріалів.

Істотним підвищенням властивостей КМ є армування пластику в заданому напрямку (наприклад, методом намотування). При цьому КМ здобуває анізотропні властивості, що найбільш повно проявляють себе в напрямку волокон [42].

Відповідно до системного підходу (рис. 1.1) до процесу створення виробів із КМ можна виділити чотири шляхи розв’язання проблеми підвищення ефективності виробів із КМ:

- пошук сполучення компонентів КМ або створення нових, здатних задовольнити вимогам, запропонованим до готового виробу;
- оптимізація КМ, наприклад, шляхом пошуку схеми армування пластику;
- удосконалювання технології виробництва КМ і виробів з них;
- удосконалювання встаткування для виробництва виробів із КМ.

Однак, як слідує із системного підходу й факту взаємозв'язку всіх факторів процесу створення виробів із КМ, розв'язання кожного із завдань окремо неможливо. Необхідне розв'язання одного комплексного завдання.

У силу складності й аналітичної нерозв'язності поставленого завдання актуальною стає проблема пошуку алгоритмів і моделей створення виробів із КМ із їхньою реалізацією на сучасних обчислювальних засобах, які в найбільш повній мірі враховували б технологічні фактори процесу виробництва.

Таким чином, вивчення створення виробів із КМ є важливим й актуальним у сфері одержання нових матеріалів, що володіють унікальними властивостями. Алгоритми й моделі, що дозволяють системно врахувати фактори, що впливають на характеристики готового виробу, дозволять вирішити проблему одержання виробів із заданими властивостями й підвищити їх стабільність.

### **1.5. Загальна характеристика й класифікація методу намотування**

Намотуванням називається процес формоутворення елементів або конструкцій, при якому заготівлі одержують шляхом автоматизованого укладання безперервного композиційного волокнистого напівфабрикату (КВН) на форму (або оправлення) по заданих траєкторіях. КВН являє собою армуючий матеріал (АМ) просочений сполучиною (СП). Оправлення звичайно має конфігурацію й розміри, що відповідають внутрішнім розмірам елемента, що виготовляється. Після намотування потрібна операція термообробки (отвердження), що фіксує форму виробу й утворює остаточну структуру

композиційного матеріалу (КМ). Оправлення після отвердження, звичайно, віддаляється. У деяких випадках воно (або їх елементи) може залишатися й виконувати функцію конструктивних елементів [14]

Намотування один із самих продуктивних способів виготовлення. Одна з особливостей методу є більше висока трудомісткість технологічної підготовки виробництва, неможливість самотійного розгляду питань вибору встаткування, оснащення, технологічних режимів і вихідних компонентів КМ [15].

При намотуванні вирішуються одночасно два завдання:

- формування КМ із заданими структурними параметрами;
- формування основних і вільних поверхонь виробу необхідної геометрії, точності й чистоти.

Основною перевагою цього методу є можливість орієнтувати АМ у заданому напрямку відповідно до розподілу напруг в елементі конструкцій, тобто одержати КМ із заданою анізотропією міцності й твердості. Метод намотування розвивається в продовженні не набагато більше п'ятдесяти років.

Він застосовується для виготовлення труб різного діаметра, ємностей, посудин, трансмісійних валів, елементів конструкцій, що є тілами обертання, виробів коробчастого перетину, деталей геометричної форми типу лопат і більше складного виду, і ін. Технологічне встаткування дозволяє одержувати вироби з характерними розмірами від декількох десятків міліметрів до 10 м і більше. Волокна пропускаються через ванну зі смолою, потім через натяжні валики, що служать для натягу волокна й видалення надлишків смоли. Волокна намотуються на сердечник з необхідним перетином, кут намотування контролюється відношенням швидкості руху візка до швидкості обертання.

#### **Застосовувані матеріали:**

Смоли: будь-які.

Волокна: будь-які, волокна подаються прямо від рами для катушок без додаткового зшивання в тканину.

Наповнювачі: будь-які.



### **Основні переваги:**

- Це може бути дуже швидкий і тому економічно вигідний метод укладання матеріалу.
- Регульоване співвідношення смола/скло.
- Висока міцність при малій власній вазі.
- Несхильність корозії й гниттю.
- Недорогі матеріали.
- Гарні структурні властивості ламінатів, тому що профілі мають спрямовані волокна й високий зміст скломатеріалу.

### **Основні недоліки:**

- Обмежена номенклатура виробів.
- Дороге встаткування.
- Волокно важко точно покласти по довжині сердечника.
- Високі витрати на сердечник для більших виробів.
- Рельєфна лицьова поверхня.

У середньому волокна мають наступні характеристики: механічна щільність  $1500 - 2500 \text{ кг/м}^3$ , межа міцності при розтяганні – до 3 ГПа, модуль пружності при розтяганні – до 800 ГПа [15].

## **1.6. Етапи технологічного процесу виготовлення виробів методом намотування**

Технологічний процес виготовлення виробів методом намотування складається з наступних операцій [16]:

1. Підготовка й налагодження верстата (розробка й налагодження керуючої програми або розрахунок кінематики руху розкладника й оправлення; установка АМ або підготовленого КВН залежно від обраного способу намотування «мокрого» або «сухого»).

2. Підготовка оправлення (зборка, закріплення заставних елементів, нанесення антиадгезійного шару, при необхідності, герметизуючих шарів й установка на верстаті).

3. Заправлення й протягання АМ або КВН по стрічкоформуєчому тракті намотувального верстата, настроювання вузлів на підтримку заданих основних технологічних параметрів і перевірка малюнка намотування шляхом пробного укладання декількох витків.

4. Намотування заданої кількості КВН для створення необхідної товщини й структури стінки виробу. При цьому ведеться контроль основних технологічних параметрів.

5. Закінчення процесу намотування, зняття оправлення із заготівлею деталі з верстата й переміщення їх у термопіч або автоклав.

6. Отвердження по заданому режимі, обумовленому типом СП, розмірами виробу й т.д.

7. Зняття виробу з оправлення.

8. Термічна й механічна обробка виробу (останнє при необхідності).

9. Контроль міцності, геометричних розмірів й інших параметрів (герметичності, теплофізичних тощо).

З перерахованих вище етапів використовуються в моделюванні пп. 1, 4, 6 та 9. Етапи 6 та 9 стосуються вже намотаного виробу. Вони стосуються роботи з готовою моделлю. Етапи 1 та 4 є важливими на початковому етапі, особливо етап 4. Для завдання моделювання він є досить складним, оскільки вимагає розв'язання наступного циклу завдань:

- вибір оптимальних параметрів моделювання процесу намотування;
- оптимальне моделювання цього процесу (згідно параметрів п. 1);
- одержання імітаційної моделі намотаного виробу з можливістю її переносу в програмне середовище фізичного моделювання.

Подальша робота з моделлю стосується обробки віртуального виробу термічно та/або механічно, а також до проведення комплексу експериментальних впливів з метою вивчення властивостей віртуально намотаного виробу й оптимізації параметрів процесу намотування.

Кінцевою метою моделювання є набір технологічних параметрів процесу намотування для одержання виробу із заданими конструктивними й фізико-механічними властивостями, що володіють високою стабільністю стосовно часу й зовнішніх факторів.

### **1.7. Класифікація видів намотування й намотуваних виробів**

Існує кілька видів класифікацій методу намотування залежно від обраних ознак. За способом одержання КВН методи намотування діляться на «сухий» й «мокрый» [19].

Суть сухого методу намотування полягає в тім, що АМ перед установкою на намотувальний верстат попередньо просочується сполучиною на спеціальних просочувальних машинах. Це дозволяє забезпечити не тільки якісне просочення, але й рівномірний зміст СП за рахунок застосування різних розчинників. Останні застосовуються для регулювання в'язкості СП. Отримані таким способом КВН називають препрегом. Після нанесення сполучного шару препреги проходять через камери попереднього «підсушування», у яких відбувається видалення розчинника й збільшення в'язкості СП.

Спосіб «мокрого» намотування відрізняється тим, що АМ просочується СП безпосередньо перед укладанням на оправлення. Необхідна в'язкість сполучини забезпечується вибором відповідного СП і застосуванням його підігріву в просочувальному пристрої. Цей спосіб вимагає більше низького контактного тиску формування, а, отже, і встаткування з меншою потужністю привода. Переваги цього способу найбільш повно проявляються при виготовленні великогабаритних виробів складної конфігурації [4].

За видом використовуваного АМ намотування класифікується на нитяну, джгутову, стрічкову або тканинну [5]. Намотування нитками, як правило, застосовується для виготовлення високонавантажених елементів невеликих розмірів [6].

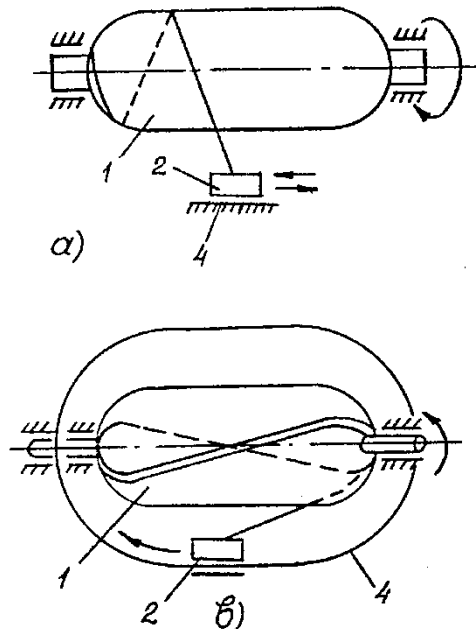
За способом укладання КВН виділяють тангенціальне (окружне, поперечне), поздовжнє, спіральне намотування, їхню комбінацію або

сполучення [7]. Деякі кінематичні схеми спірального намотування оболонок подвійної кривизни наведені на рис. 1.3 [2]. Сполучене намотування дозволяє одержати плетену структуру стінки виробу на відміну від шаруватої, здійснюваної при послідовній реалізації способів (рис. 1.4). Можливі варіанти повздошно-поперечного намотування наведені на рис. 1.5 [14].

Найбільше доцільно використати класифікатор, заснований на конфігурації виробів, що виготовляють, і умові забезпечення рівноваги КВН на поверхні оправлення (табл. 1.2).

За умовою забезпечення рівноваги КВН на поверхні оправлення розрізняють наступні види намотування: геодезичне; негеодезичне; кусочно-безперервне.

Геодезичне намотування передбачає укладання КВН по траєкторіях без обліку сил тертя, тобто КВН перебуває в положенні рівноваги на всій траєкторії укладання.



(1 – оправлення, 2 – НР, 4 – напрямна).

Рухи НР: а) поступально; в) по замкнутій криволінійній напрямній.

Рис. 1.2. Кінематичні схеми спірального намотування оболонок подвійної кривизни

Таблиця 1.2

**Класифікатор намотуваних виробів. «+», «X» позначена наявність або відсутність методів розрахунку УНП**

Конфігурація виробу	Траскторія укладання КВН на оправлення		
	Геодезична	негеодезична	Кусочно-безперервна
Тіла обертання	<b>+</b>	<b>+</b>	<b>+</b>
Тіла опуклого перетину із прямою утворюючою	<b>+</b>	<b>×</b>	<b>×</b>
Тіла довільної форми	<b>×</b>	<b>×</b>	<b>×</b>

Негеодезичне намотування (укладання КВН у межах ліній граничного відхилення) передбачає укладання КВН по траєкторіях з урахуванням сил тертя.

Кусочно-безперервне намотування складається з геодезичних (негеодезичних) траєкторій укладання КВН на обмежених ділянках поверхні оправлення із крапками переходу з ділянки на ділянку, у яких КВН утримується механічним способом (наприклад, за допомогою штирів).

Намотування може також застосовуватися як метод попереднього укладання КВН із наступним формоутворенням іншим способом.

Аналіз закордонних джерел в області переробки КМ показав, що за останні 5 – 7 років скоротилося число публікацій, пов'язаних з технологією намотування. У той же час розвиток методу намотування триває в напрямку виготовлення неосиметричних тіл і тіл більш складної форми [17] та створення виробів з новими структурними схемами армування [20].

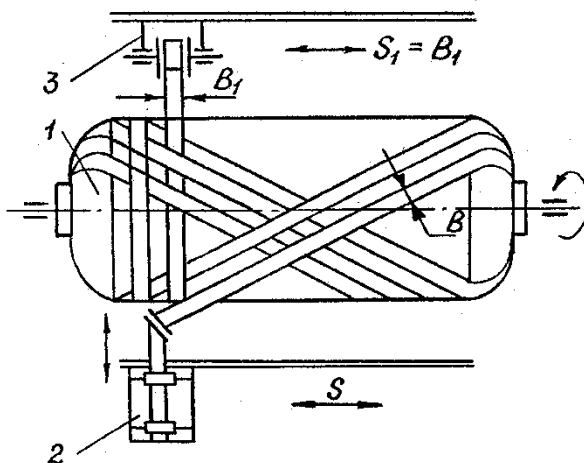


Рис. 1.3. Схема сполученого спірального-кільцевого намотування: 1 – оправлення; 2 – НР спіральні намотування; 3 – НР кільцеві намотування.

Таким чином, по-перше, всі види процесів намотування поєднує те, що їхню основу становить безперервний (кусочно-безперервний) процес, що має найчастіше циклічність (періодичність). Отже, для їхнього моделювання знадобляться безперервно-детерміновані (для обліку фізичних закономірностей) і безперервно-стохастичні моделі (облік випадкових факторів).

По-друге, моделювання процесу намотування дозволить обійти стороною відсутність загальних методик укладання КВН (див. табл. 1.2) і вирішувати це завдання для конкретного випадку.

### 1.8. Висновки до розділу

Композиційні матеріали посіли значне місце серед сучасних рентабельних матеріалів. Властивості композитів, що наповнені армуючими наповнювачами використовуються навіть в авіакосмічній промисловості. Особливо це стосується тих методів виготовлення виробів з КМ, в яких армуюча складова є неперервною, що забезпечує властивості матеріалів, які переважають властивості класичних металів, дерева та ін. Серед цих методів – метод намотування – дозволяє одержати високу стабільність властивостей як виробів, так і самого композиційного матеріалу, що отримується.

Технічна підготовка виробництва вимагає попереднього аналізу, який в сучасних умовах виконується за допомогою обчислювальних засобів (персональних комп'ютерів, зокрема) в альянсі з програмними засобами інженерного аналізу. Метод намотування потребує системного підходу, в якому б враховувався зв'язок геометрії майбутнього виробу з фізико-механічними параметрами композиційного матеріалу.

## **РОЗДІЛ 2. ЗАСОБИ МОДЕЛЮВАННЯ ВИРОБІВ, ОТРИМАНИХ МЕТОДОМ НАМОТУВАННЯ**

### **2.1. Огляд засобів моделювання**

Як було показано в [8] щодо засобів моделювання на одному полюсі перебувають мови програмування типу Pascal, що дозволяють при достатній кваліфікації в програмуванні, створювати моделі практично будь-якого призначення й різної складності, але ціною, як правило, значних витрат часу й праці. А на іншому – сучасні інтегровані системи широкого призначення (наприклад, ANSYS).

Проте не завжди загальнопризначені засоби відповідають усім вимогам конкретних завдань. Остання обставина привела до появи надзвичайно великої кількості різних моделюючих засобів і мов, орієнтованих на конкретні області застосування. Понад десять років тому налічувалося біля ста мов і систем, пропонованих різними фірмами й організаціями. До теперішнього часу розмір списку збільшився в кілька разів і має стійку тенденцію до подальшого росту.

Для моделювання виробів, виготовлених методом намотування, самим суттєвим є зв'язок технологічних параметрів процесу виготовлення із властивостями готового виробу (і навіть із умовами його експлуатації). Зміна траєкторії укладання армуючого матеріалу (АМ) на поверхню оправлення приводить до зміни як властивостей КМ, так й умов навантаження (експлуатації) [23].

Аналіз можливостей CAD/CAM/CAE-систем, що себе зарекомендували, на предмет реалізації цього зв'язку показує, що жодна з них не може враховувати повною мірою вищезгаданий зв'язок. При цьому існують CAD/CAM-системи, що мають можливість реалізувати досить складну геометрію (що необхідно для графічної реалізації намотаного виробу). До таких систем відносяться Solid Works, AutoCAD, Solid Edge, Intermech CadMech, Pro/ENGINEER. Крім того, AutoCAD 2002 має потужну мову програмування – AutoLISP, що дає можливість згенерувати вихідний код



вхідних даних для аналізу за допомогою якої-небудь CAE-системи. Інші CAD/CAM-системи, в основному, націлені на геометричне моделювання твердих тіл і вузлів, що складаються із твердих тіл.

До відомих CAE-систем, що дозволяють виконати комплексний аналіз поведінки виробу в різних режимах навантаження й фізичних умов, належать такі пакети, засновані на методі кінцевих елементів, як ANSYS (лідуюча позиція), NASTRAN, ADAMS, LS-DYNA. Однак пакет LS-DYNA призначений для аналізу нелінійних динамічних процесів (наприклад, поведінка транспортного засобу при зіткненні з перешкодою), а така система як ADAMS націлена на аналіз кінематики рухомих вузлів механізмів.

З останніх – ANSYS та NASTRAN – ANSYS має потужну мову програмування, а також можливості модифікації геометричної моделі будь-яким чином, оскільки при імпорті здійснюється перетрансляція даних у геометричний формат ANSYS і деталь не підмінюється «недоторканою» кінцево-елементною сіткою. Користувач може видаляти несуттєві дрібні деталі, добудовувати певні деталі, проводити згущення-розрідження сітки й інші найважливіші операції, без яких, часом, подальше розв'язування буде зовсім некоректно або взагалі не зможе бути досягнуто. Побудова поверхневої, твердотільної та каркасної геометрії та внесення змін може здійснюватися засобами власного геометричного моделера. ANSYS дозволяє вирішувати завдання міцності, теплофізики, гідрогазодинаміки, електромагнетизму разом з розрахунком характеристик втоми і процедурами оптимізації. Єдиний набір команд та єдина база даних повністю усувають труднощі інтеграції та взаємного обміну інформацією між вказаними сферами. Більше того, шляхом використання в програмі спеціалізованих кінцевих елементів, що мають, крім переміщень і поворотів у вузлах, також і ступені свободи по температурі, напрузі й ін. і перемикання типу елемента (наприклад, з електромагнітного на міцносний) реалізовані більші можливості проведення зв'язаного аналізу. Також ANSYS містить велику бібліотеку елементів, серед яких 16- та 100-шаровий полімерний композит, гнучка нитка та інші. Серед

них існує чотири типи кінцевих елементів, призначених для моделювання багатошарових полімерних і сендвічевих композицій [11].

## **2.2. Введення в ANSYS**

### **2.2.1. Призначення ANSYS**

Програма ANSYS є гнучким інструментом для проєктування та аналізу, сумісним з операційними системами на різних комп'ютерах - від особистих комп'ютерів до робочих станцій і суперкомп'ютерів. Важливою особливістю є файлова сумісність всіх версій ANSYS для всіх платформ. Можливість вирішення різноманітних задач, таких як міцність під тепловим навантаженням, вплив магнітних полів на міцність конструкцій, теплоперенесення у електромагнітному полі, робить цю програму багатоцільовою. Модель, створена на особистому комп'ютері, може легко використовуватися на суперкомп'ютері. Це забезпечує всім користувачам програми можливості розв'язування широкого кола інженерних завдань [13].

Програма ANSYS розширює свій набір розрахункових інструментів, які [12]:

- можуть враховувати різноманітні конструктивні нелінійності;
- надають можливість вирішити навіть найзагальніше завдання контакту між поверхнями;
- дозволяють обробляти великі (кінцеві) деформації та кути повороту;
- забезпечують інтерактивну оптимізацію та аналіз впливу електромагнітних полів, розв'язання гідроаеродинамічних завдань
- і багато іншого - включаючи параметричне моделювання, адаптивне перебудування сітки, використання р-елементів та обширні можливості створення макроскриптів за допомогою мови параметричного проєктування програми ANSYS (APDL).

### **2.2.2. Інтерфейс користувача**

Використовується інтерфейс (рис. 2.1), що відповідає сучасним вимогам, створений на основі розробки Motif Standard. За допомогою цього інтерфейсу забезпечується інтерактивний доступ до функцій, команд,

документації й довідковим матеріалам програми. Вихідні дані можна вводити за допомогою маніпулятора “миша”, клавіатури або сполучаючи ці два варіанти.

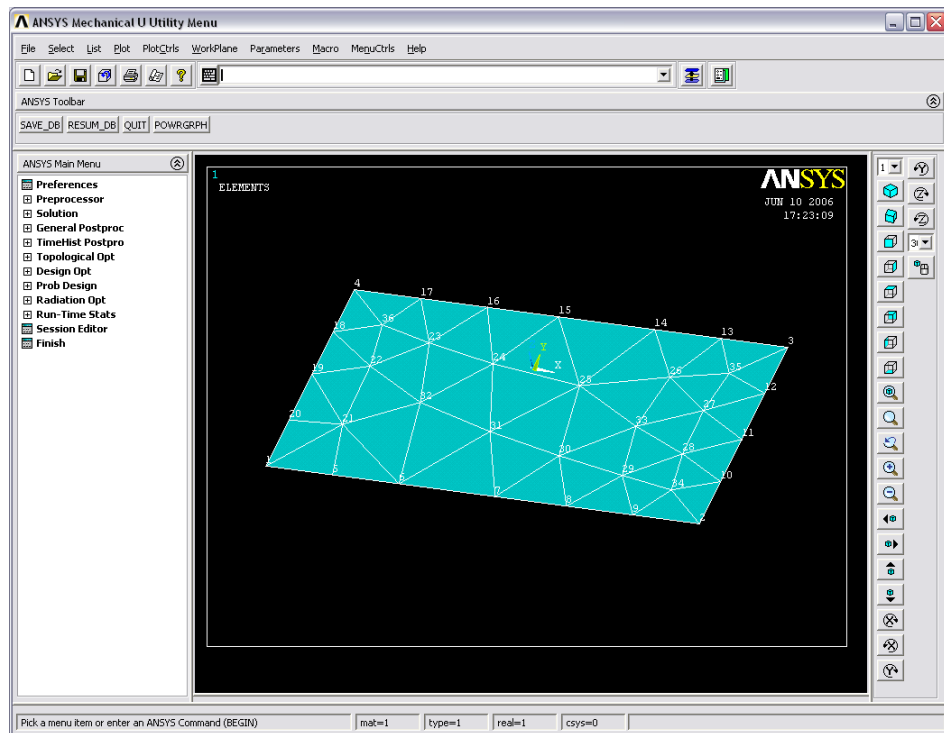


Рис. 2.1. Інтерфейс Motif Standard

Існують чотири загальних методи введення інструкцій для програми, коли використовуються наступні можливості інтерфейсу: меню, панелі діалогу, лінійка інструментів, безпосереднє введення команд [9].

Користувачі, знайомі з командами програми, можуть вводити їх безпосередньо з клавіатури.

Будь-яка один раз виконана команда записується у файл реєстрації сеансу роботи. Доступ до цього файлу можливий з вікна виводу, що дозволяє звернутися до списку команд у випадку помилки або записати його у вигляді файлу для виконання в пакетному режимі.

Система меню (на основі розробки фірми Motif) забезпечує введення даних і вибір дій програми за допомогою панелей діалогу, що випадають з меню і вікон списку, допомагаючи користувачеві управляти програмою.

### 2.2.3. Графічні можливості

Засоби твердотільного моделювання містять у собі подання геометрії, засноване на використанні сплайнової технології NURBS, геометричних примітивів й операцій булевої алгебри (виконуваних модулем SHAPES™ фірми XOX Corp., що вбудований у програму ANSYS) [13].

Програма ANSYS підтримує велику кількість графічних опцій для роботи із графічними системами X Windows, OPEN GL і деякими іншими системами відображення тривимірної графіки.

Повністю інтерактивна графіка (тобто засоби й системи введення, відображення й редагування зображень) є складовою частиною програми ANSYS. Графіка важлива для перевірки вихідних даних і перегляду результатів розв'язання на етапі постпроцесорної обробки.

Модуль PowerGraphics має значну швидкість побудови геометричних об'єктів і графіків результатів. Висока швидкість досягається за рахунок збереження геометрії як деякого «готового» об'єкта в пам'яті машини, а не побудовою її щораз заново. Засоби візуалізації цього модуля придатні для зображення елементів сітки й областей рівних значень напруг як при використанні *p*-елементів, так й *h*-елементів.

Можливості модуля PowerGraphics дозволяють швидко видавати зображення: ізоповерхонь (тобто поверхонь рівного значення якої-небудь величини); графічних об'єктів, розділених на складові частини, у вигляді однієї зборки або сукупності так званих Q-розрізів (у вигляді тонких “скибочок”); Q-розрізів з топологічними деталями. Графічні засоби програми ANSYS включають наступне:

- відображення граничних умов на твердотільних і кінцево-елементних моделях;
- подання результатів кольоровими областями рівних значень;
- графіки залежностей отриманих результатів від часу або від деякої відстані в межах розрахункової моделі;

- перетворення зображень загального характеру (зміна напрямку погляду, розорювання вікна до максимального розміру, укрупнення плану, обертання);
- розтягування твердотільних примітивів;
- багатовіконний режим роботи;
- показ невидимих ліній, перетинів і перспективних зображень;
- програмний засіб для Z-буферування (плавне затінення й швидка “відрисовка” об’єкта);
- зображення тіней на об’єкті від джерела світла;
- підвищення ясності зображення (видалення внутрішніх ліній, поділ суміжних ліній елемента й вибір незалежного масштабу зображення по вертикалі й горизонталі);
- створення композиції з декількох об’єктів (наприклад, доповнення твердотільної моделі);
- наявність палітри до 256 кольорів;
- тривимірна візуалізація, що включає зображення градієнтів, ізоповерхонь, траєкторій часток потоку й розрізів обсягів;
- апроксимація графіків X-Y за допомогою широкого набору кривих, їх двовимірне й об’ємне подання; вибір колірної гами графіків, тла й ліній сітки, вибір товщини ліній;
- графічне відображення тривалості процедур побудови сітки, підготовки листингу й процесу розв’язування завдання;
- засоби доповнення графічних зображень текстом, розмірними лініями, фігурами, символами, круговими діаграмами й т.п.;
- засоби анімації для відображення змін деформованої форми, результатів рахунку залежно від часу, “пожвавлення” Q-розрізів й ізоповерхонь;

- колірна індикація більшості графічних об'єктів (елементів сітки, ліній, областей, обсягів, граничних умов, фарбування екрана, контурних ліній й індексів) залежно від їхнього рангу або типу;
- напівпрозорі зображення для елементів, твердотільних об'єктів, складових частин групи об'єктів й ізоповерхонь;
- показ дійсної форми й поперечного переріза трубопроводів, патрубків, балок і магнітів;
- показ окремих шарів композитних матеріалів та їхньої просторової орієнтації;
- поділ вікон по кольорах тла;
- збереження у файлі специфікацій виводу на екран для повторного звертання до них;
- засоби одержання твердих копій графіки, включаючи системи Postscript, HPGL й TIFF.

#### 2.2.4. Процесори

Всі функції, виконувані програмою ANSYS, об'єднані в групи, які називаються процесорами [13]. Програма має один препроцесор, один процесор обчислень, два постпроцесори й кілька допоміжних процесорів, включаючи оптимізатора. Препроцесор використовується для створення кінцево-елементної моделі й вибору опцій для виконання процесу обчислень. Процесор обчислень використовується для додавання навантажень і граничних умов, а потім для визначення відгуку моделі. За допомогою постпроцесору користувач звертається до результатів обчислень для оцінки поведінки розрахункової моделі, а також для проведення додаткових обчислень, що представляють інтерес.

Таблиця 2.1

#### Процесори (підпрограми), доступні в програмі ANSYS

Процесор	Функції	Команда
PREP7	основний препроцесор – побудова моделі (геометрія, матеріали, зв'язки, рівняння обмеження й т.д.)	/PREP7

Процесор	Функції	Команда
SOLUTION	навантаження та запис – застосовувані навантаження й одержання рішення методом кінцевих елементів	/SOLU
POST1	основний постпроцесор – огляд результатів у певних точках моделі протягом часу	/POST1
POST26	часовий постпроцесор – огляд результатів у певних точках моделі протягом часу	/POST26
OPT	оптимізація проектування	/OPT
AUX2	утиліти бінарного файлу – печатка бінарних файлів у читаємій формі	/AUX2
AUX12	генерація матриці випромінювання – обчислення проєктованих факторів випромінювання й створення матриці випромінювання	/AUX12
AUX15	трансляція файлу – транслюються файли з CAD або FEA програм	/AUX15
RUN STAT	статистика під час виконання – для припущення часу CPU, вимог до фронту хвилі й т.д.	/RUNSTAT

#### 2.2.5. База даних

У програмі ANSYS використовується одна, центральна, база даних для всього набору відомостей, що стосуються моделі й результатів обчислень. Відомості про моделі (включаючи дані про геометрію твердотіЛЬНОЇ і кінцево-елементної моделей, властивостях матеріалів і т.д.) записуються в базу даних на стадії препроцесорної підготовки. Навантаження й результати обчислень записуються процесором обчислень. Дані, отримані на основі результатів обчислень при їх постпроцесорній обробці, записуються постпроцесором. Відомості, внесені одним із процесорів, доступні, при необхідності, для інших процесорів. Наприклад, загальний постпроцесор може зчитувати дані, що стосуються обчислень й моделі, а потім використати їх для постпроцесорних обчислень.

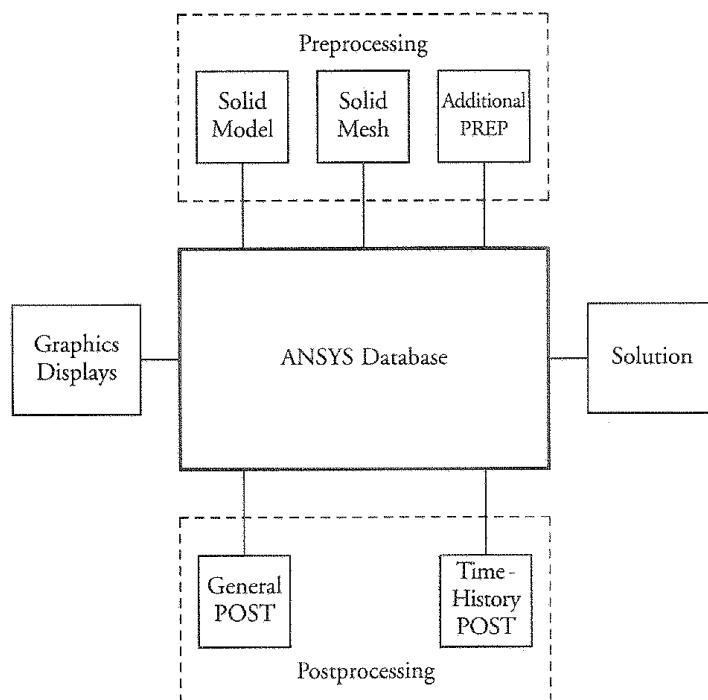


Рис. 2.2. Діаграма зв'язків центральної бази даних програми ANSYS

Важливим фактом є те, що для більшості процесорів база даних повинна містити принаймні геометрію моделі. Наприклад, щоб зробити постпроцесування, використовуючи POST1, перед тим, як прочитається результуючий файл геометрія моделі повинна вже існувати в базі даних.

#### 2.2.6. Формат файлів

Файли використовуються для передачі даних між різними частинами програми, для створення бази даних і для збереження результатів роботи. Ці файли можуть включати файли баз даних, результатів, графічних об'єктів тощо. Створювані програмою файли мають формат ASCII, що дозволяє їм легко читатися та редагуватися, або двійковий формат. За замовчуванням програма ANSYS створює бінарні файли з використанням зовнішнього формату (згідно зі стандартом IEEE), що забезпечує сумісність обробки даних різними апаратними засобами. Наприклад, дані про геометрію моделі, створені користувачем на одній комп'ютерній системі, можуть легко використовуватися іншим користувачем програми, який працює на іншій платформі, без будь-яких труднощів.



### 2.2.7. Внутрішня мова APDL

Мова APDL – надає користувачеві ANSYS широкі можливості автоматизації процесу роботи в програмі. APDL – ANSYS Parametric Design Language – параметрична мова, створена на базі мови Fortran 77. APDL – мова сценаріїв. ANSYS підтримує безліч функцій мови Fortran 77, таких як організація розгалужень і циклів, операції зі скалярними, векторними й матричними величинами, робота із зовнішніми файлами й т.д.

Мова APDL може бути застосована на будь-якому етапі – від створення геометрії до виводу результатів на екран або у файл.

### 2.3. Огляд мови APDL

APDL містить понад 800 команд, кожна виконує певну функцію – для завдання даних (наприклад, властивостей матеріалів), або для виконання якої-небудь дії (наприклад, вивід моделі на екран) [9;10]. Типові команди містять ім'я команди й кілька *аргументів* після коми. Кожен аргумент відокремлюється комою.

Імена команд можуть містити від 1 до 8 символів. Імена, що мають довжину більше 5 символів, можуть бути скорочені до перших 4-х (крім спеціальних випадків, таких як **/PREP7**, **/POST26**, і **\*ENDDO**). Деякі команди (наприклад, **ADAPT** й **ETEXP**) у дійсності є *макрокомандами*. Імена макрокоманд повинні бути введені повністю. Команди, які починаються зі «/» (слешу), відомі як *слеш-команди* або утиліти команд, звичайно призначені для контролю за основною програмою, такий як вхід у контроль за підпрограмами, диспетчером файлів і графікою. Дані вводяться спеціальними командами (утилітами), що зберігають їхні поточні установки під час зміни програми. Команди, які починаються «\*», відомі як *star-команди*, є частинами мови параметричного проектування ANSYS.

Команди вводяться, використовуючи вільні формати. Фіксований формат не використовується. Наступні один за одним коми можуть бути використані для пропуску поля. У цьому випадку значення за замовчуванням

приймається для цього аргументу. Максимальне число символів, включаючи коми, пропуски й інші спеціальні символи не повинне перевищувати 80.

### **2.3.1. Умовчання**

Для мінімізації кількості даних, що вводять, у більшості команд передбачені значення за замовчуванням. Існує 2 типи за замовчуванням:

- команда за замовчуванням;
- поле за замовчуванням.

*Команда за замовчуванням* передбачається, коли команда не запускається.

*Поле за замовчуванням* – значення передбачається аргументом команди, якщо він не визначений. Чисельний аргумент за замовчуванням завжди дорівнює нулю (за винятком деяких спеціальних випадків).

### **2.3.2. Попередження й помилки**

При необхідності програма видає повідомлення про помилки, попередження, та інші інформаційні повідомлення. Із включеним меню повідомлення виводяться на екран у кольорових текстових рамках:

- сірі – інформаційні повідомлення,
- жовті – попередження,
- червоні – помилки.

Число повідомлень про помилки й попередження, виведених на екран для команди обмежено:

- 5 – при включеному меню,
- 20 – з виключеним,
- 200 – у пакетному режимі.

### **2.3.3. Інтерактивний режим у порівнянні з пакетним режимом**

В ANSYS передбачені два режими – *інтерактивний* і *пакетний*. *Інтерактивний режим* припускає взаємодію користувача з комп'ютером: запуск команд, спостереження її результатів і контроль виконання; потім запуск іншої команди й т.д. Інтерактивний режим дозволяє використати найбільш підходящі властивості графічного режиму, порядкової підказки,

системного меню й графічного вибору (pick). За замовчуванням установлюється інтерактивний режим.

*Пакетний режим* – режим коли програма підлегла командному файлу. Перший рядок у файлі повинен бути **/BATCH**, що позначає пакетний режим. Залежно від операційної системи комп'ютера, можна покласти пакет із завданням в основу, що дозволяє виконати іншу роботу на комп'ютері. Пакетний режим особливо використовується для рішення частини аналізу, під час якого не потрібен зв'язок з комп'ютером.

#### **2.3.4. Робота LOG файлу**

Кожний раз при роботі в ANSYS створюється log-файл із ім'ям *jobname.LOG*. (*Jobname* установлюється ім'ям завдання, певною системною командою, використовуваною для запуску програми ANSYS). LOG-файл відкривається, коли здійснюється перший вхід у програму, і закривається, при виході.

Кожна команда копіюється в LOG-файл, створюючи завершену копію поточної роботи в ANSYS. Команди, що втримуються в log-файлі можна використати як засіб відновлення при системному фатальному збої або катастрофічній помилці користувача. Просто при прочитанні (**/INPUT**) перейменованої копії вашого log-файлу (або підпорядкуванням його пакетному файлу), кожна команда, що втримується в log-файлі буде перезапускатися, перетворюючи вашу базу даних у тому виді, як вона записана. Log-файл також досить корисний як засіб налагодження, що може допомогти виявити помилки, можливі в сеансі із програмою. Якщо при налагодженні сеансу буде потрібна допомога, майже безсумнівно виникне необхідність перегляду копії вашого log-файлу.

При кожному новому сеансі нові команди будуть дописуватися у вже існуючий log-файл. Під час інтерактивного режиму можливий перегляд log-файлу за допомогою команди LIST, або команди LIST,LOG.

Оскільки log-файл існує в кодовому форматі, він може легко читатися й редагуватися при використанні зовнішнього текстового редактора.

Перша команда, записана в LOG-файл у будь-якому сеансі ANSYS – **/BATCH**. Ця команда пишеться навіть у тому випадку, якщо під час сеансу ANSYS вона не запускалася. Якщо передбачається використати перейменований файл для введення в інтерактивному режимі, під час редагування необхідно видалити цю команду з LOG-файлу.

Для читання перейменованого й редагованого LOG-файлу в інтерактивному режимі використовується команда **/INPUT**. У пакетному режимі можна використати перейменований і редагований LOG-файл як пакетний файл для уведення.

## **2.4. Типові види аналізів в ANSYS**

Типовим алгоритмом аналізу в ANSYS є послідовність наступних дій [9;12]:

Побудова моделі.

Визначення імені завдання й заголовка аналізу.

Визначення типів елементів.

Визначення реальних констант елементів.

Визначення властивостей матеріалу.

Лінійні властивості матеріалу.

Нелінійні властивості матеріалу.

Анізотропні пружні властивості матеріалів.

Створення геометрії моделі.

Навантаження, що прикладають, і одержання рішень.

Визначення типу аналізу й опцій аналізу.

Навантаження, що прикладають.

Визначення опцій кроків навантаження.

Ініціалізація обчислень [SOLVE].

Перегляд результатів.

Всі роботи в типовому ANSYS-аналізі можна поділити на три різних етапи:

- побудова моделі;

- додавання навантажень і одержання рішень;
- перегляд результатів.

Побудова моделі – це та частина аналізу, що вимагає найбільших витрат часу користувача. На цьому етапі визначаються ім'я завдання й заголовки аналізу й потім використовується препроцесор PREP7 [/PREP7] для визначення типу елементів, реальних констант елементів, властивостей матеріалів і геометрії моделі.

#### **2.4.1. Створення геометрії моделі**

Ціль цього етапу – створення кінцево-елементної моделі (вузлів й елементів), що адекватно описує геометрію моделі. Існує два методи створення кінцево-елементної моделі:

- моделювання суцільних тіл;
- пряма генерація.

При моделюванні суцільних тіл описуються геометричні границі моделі й потім віддається наказ ANSYS для автоматичної побудови сітки (геометрія з вузлами й елементами). Можливо контролювати розмір і форму елементів, які створює програма.

При прямій генерації положення кожного вузла й пов'язаних з вузлами елементів визначається «вручну». Доступні кілька зручних операцій, таких як копіювання моделі існуючих вузлів й елементів, симетричне відбиття тощо.

#### **2.4.2. Додавання навантажень і одержання рішень**

На цьому етапі використовується команда SOLUTION (SOLU) для визначення типу аналізу та опцій навантажень, у т.ч. покрокового навантаження, і потім запускається програма на обчислення.

#### **2.4.3. Перегляд результатів**

Коли результати обчислені, для їхнього перегляду ANSYS використовує постпроцесор. Доступні два постпроцесори POST1 й POST26.

POST1 – основний постпроцесор, використовується для перегляду результатів на першому кроці збільшення (кроці часу) всієї моделі. Команда для введення – /POST1 – має силу тільки на початковому рівні. Можливо

одержати контур екрана, деформовані форми й табличний список для перегляду й інтерпретації результатів аналізу. В POST1 доступні багато інших можливостей, такі як обчислення помилки, комбінація випадків навантаження, обчислення результуючих даних й операції доступних шляхів.

POST26 – основний постпроцесор, використовується для перегляду результатів певних точок моделі для всіх тимчасових кроків. Команда для введення – **/POST26**, має силу тільки на початковому рівні. Можливо одержати графіки результатів даних залежно від часу (або частоти) і табличні лістинги. Інші можливості **POST26** включають арифметичні обчислення й комплексну алгебру.

## **2.5. Аналіз геометрії, що враховується при аналізі намотаних тіл**

Під час аналізу геометричних об'єктів, що моделюють вироби, отримані методом намотування, враховуються бокові поверхні. Для виконання аналізу ці поверхні (де передбачається проходження нитки зі сполучиною) повинні бути розбиті сіткою кінцевих елементів. ANSYS програма має багат шарові кінцеві елементи, які накладаються на плоскі поверхні. Таким чином, первинна геометрія намотаного виробу має представляти собою поверхню без товщини, що моделює, наприклад, поверхню оправки майбутнього процесу намотування.

Для створення такої геометрії є два основних шляхи, але кожен з них має свої перешкоди.

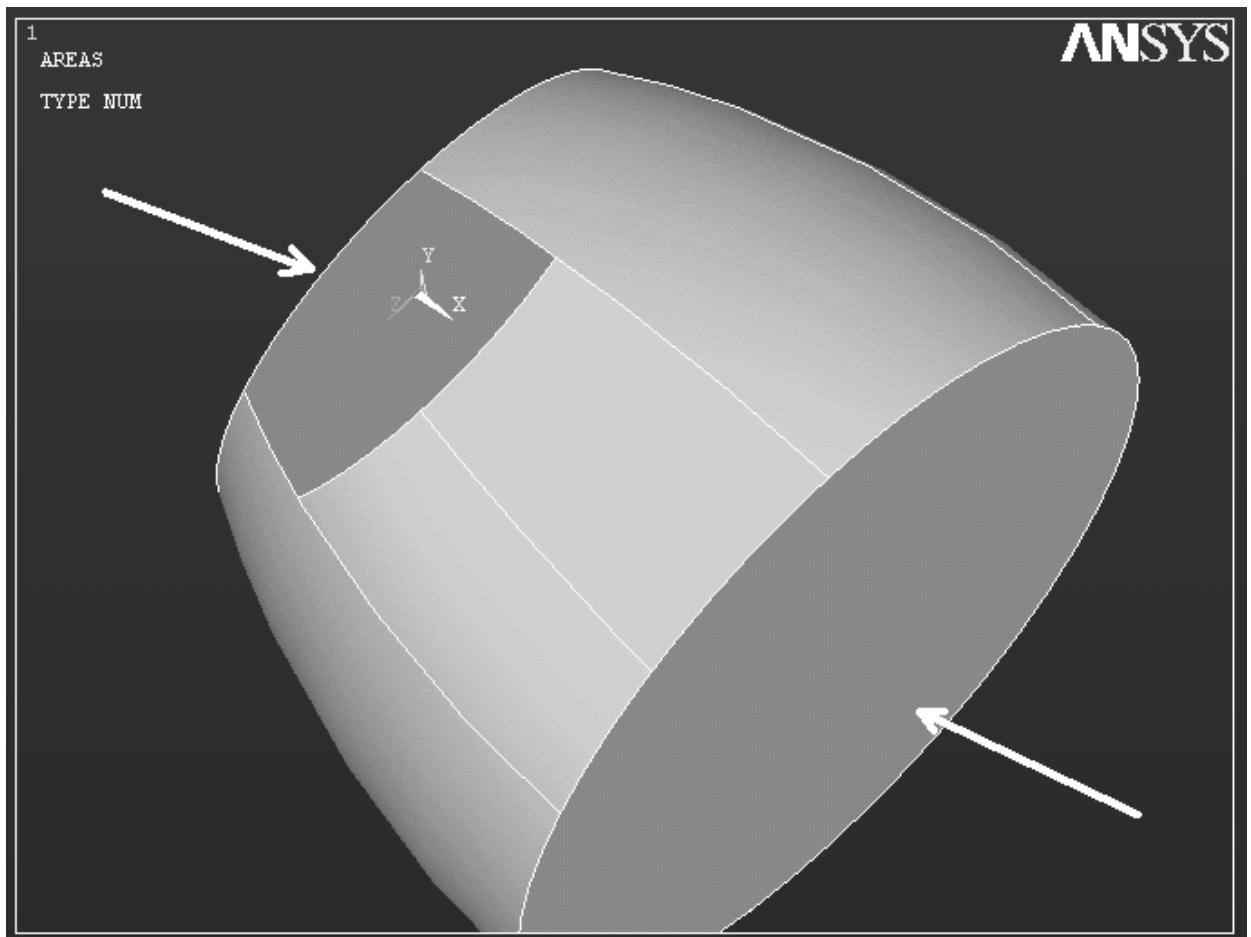
По-перше, геометрія може бути створена власними CAD-засобами ANSYS. Але цей шлях припускає попереднє знайомство інженера з геометричними можливостями програми ANSYS або з деталями її внутрішньої мови APDL.

По-друге, геометрія може бути імпортована в одному з графічних форматів, що підтримує ANSYS. Система ANSYS має засоби імпорту наступних графічних форматів [13]:

- IGES;
- CATIA;

- PRO/E;
- UG;
- SAT;
- PARA.

Під час імпорту поверхонь, які потрібні для моделювання намотаних тіл, імпортуються лінії, що їх обмежують, а ANSYS вже автоматично добудовує поверхні чи об'єми або не виконує добудову (за параметрами команди імпорту). Таким чином, виявляється проблема аналізу моделі, що імпортується з позицій *потрібна/непотрібна поверхня* (див. рис.2.3). Такий шлях з погляду автоматизації не є бажаним, тим більше якщо врахувати графічні можливості програми ANSYS та її мови APDL.



Стрілками показані поверхні, що мають бути видалені з моделі

Рис. 2.3. Поверхні, що отримані при імпорті

Таким чином, сенсу набирає перший шлях – створення геометрії власне в ANSYS, але за допомогою деяких додаткових програмних засобів,

наприклад, оболонкою, що генерує вхідний код на мові APDL для побудови необхідних елементів моделі.

Для такої побудови доцільно виконувати три етапи:

- побудова ключових точок внутрішнього профілю тіла, що намотується;
- побудова сплайнової огинаючої (профілю виробу);
- поворот профільної кривої навколо осі обертання.

Такий алгоритм обмежує множину намотаних тіл осесиметричними тілами, але дозволяє виконати імпорт геометрії, використовуючи додаткову оболонку як буфер між зовнішнім графічним файлом та системою ANSYS.

## **2.6. Висновки до розділу**

Таким чином, з аналізу наведеного вище матеріалу можна зробити наступні висновки.

Існуючі програмні засоби інженерного аналізу представлені широким колом CAD/CAE-пакетів, але жоден з них без додаткових засобів не придатний для моделювання та аналізу виробів, що отримані з КМ методом намотування.

Згідно результатам досліджень програмних моделюючих засобів найбільш прийнятним шляхом є використання CAD/CAE-системи ANSYS, яка має потужну базу для проведення комплексного аналізу та наявність потужної внутрішньої мови (APDL).

Першим кроком аналізу є побудова геометрії тіла, що намотується. Найприємнішою з точки зору особливостей наступного аналізу є побудова геометрії власними засобами ANSYS. Для його реалізації є потреба створення деякої програмної оболонки, що генерувала б вхідний файл на мові APDL, виконання якого в ANSYS автоматично буде необхідну в подальшому геометрію. Однією з вимог до оболонки є те, що вона повинна мати можливість імпорту геометрії (принаймні, точок) з поширених графічних форматів.



## РОЗДІЛ 3. СТВОРЕННЯ ГЕНЕРАТОРУ ВХІДНОГО ФАЙЛУ ДЛЯ CADСИСТЕМИ ANSYS НА MOBI APDL

### 3.1. Аналіз кінцевого результату програмного засобу

Вхідний файл для CAD/CAE-системи ANSYS повинен реалізувати три етапи, наведені в розділі 2.

#### 3.1.1. Перший етап

Для реалізації першого кроку – побудови точок профілю – доцільно використовувати спеціальний список програми ANSYS, так званий P51X. Слід відзначити, що порядок роботи з цим списком вкрай мало документовано. Наповнення списку виконується двома командами: **FLST** та **FITEM**. [10]

Перша команда – **FLST** ініціалізує список та має наступний синтаксис:

**FLST, *NFIELD*, *NARG*, *TYPE*, *Otype*, *LENG***

***NFIELD***   Номер поля команди ANSYS, в яке підставлятимуться значення зі списку. (Назва команди є першим полем, перший аргумент – 2 й т.д.) Вказане поле в наступному повинно містити мітку P51X.

***NARG***       Кількість об'єктів у списку.

***TYPE***       Тип об'єктів:

- 1 --**   Номери вузлів
- 2 --**   Номери елементів
- 3 --**   Номери точок
- 4 --**   Номери ліній
- 5 --**   Номери поверхонь
- 6 --**   Номери об'ємів
- 7 --**   Номери точок маршрутів
- 8 --**   Локалізація координат
- 9 --**   Екранні пікселі

***Otype***       Дозвіл додавання:

**NOOR --**   Новий список (за замовченням).

**ORDER --** Дані добавлятимуться до існуючого списку.

**LENG** Загальна довжина списку об'єктів (за замовчуванням дорівнює *NARG* якщо *Otype* = NOOR).

Друга команда – **FITEM** – ідентифікує одиницю списку:

FITEM, NFIELD, ITEM, ITEMU, ITEMZ

**NFIELD** Номер поля команди ANSYS, в яке підставлятимуться значення зі списку. (Назва команди є першим полем, перший аргумент – 2 й т.д.) Вказане поле в наступному повинно містити мітку P51X.

**ITEM** X-координата, або параметр команди.

**ITEMU,** Y та Z координати.

**ITEMZ**

Наприклад, код:

FLST,3,6,8

FITEM,3,101,153,0

FITEM,3,192,288,0

FITEM,3,423,349,0

FITEM,3,646,358,0

FITEM,3,872,297,0

FITEM,3,1004,157,0

K, ,P51X

побудує шість точок в площині XY з координатами:

	№	X	Y	Z
п/п				
	1	101	153	0
	2	192	288	0
	3	423	349	0
	4	646	358	0
	5	872	297	0
	6	1004	157	0

Тут команда **K, ,P51X** виконує побудову точок, а друге поле, що пусте, автоматично заповниться загальним номером точки в базі даних ANSYS.

### 3.1.2. Другий крок

Генерація сплайну виконується командою BSPLINE [10]:

BSPLIN, *P1, P2, P3, P4, P5, P6, XV1, YV1, ZV1, XV6, YV6, ZV6*  
***P1, P2, P3, P4, P5,***      Список точок, по яким будуватиметься сплайн.  
***P6***

***XV1, YV1, ZV1***      Орієнтація кінця сплайну в точці  
P1.

***XV6, YV6, ZV6***      Орієнтація кінця сплайну в точці  
P2.

При відсутності останніх параметрів закручення сплайну на кінцях враховується нульовим.

Для коректної роботи з точками бази даних треба знати їх абсолютні номери. Для цього можна використати команду **\*GET**, яка в наступній стрічці присвоює змінній **\_MaxKP** максимальний номер точки в поточній базі даних.

**\*GET,\_MaxKP,KP,Cnt,NUM,MAX**

Тепер достатньо навести, наприклад, такий код:

FLST,3,6,3

FITEM,3,\_MaxKP-5

FITEM,3,\_MaxKP-4

FITEM,3,\_MaxKP-3

FITEM,3,\_MaxKP-2

FITEM,3,\_MaxKP-1

FITEM,3,\_MaxKP-0

BSPLINE, ,P51X

### 3.1.3. Третій етап

Для створення необхідної поверхні можна використати команду AROTAT, що обертає лінію навколо деякої осі, створюючи поверхню обертання [10]:

AROTAT, NL1, NL2, NL3, NL4, NL5, NL6, PAX1, PAX2, ARC, NSEG

<i>NL1, NL2, NL3, NL4, NL5, NL6</i>	Список ліній, що обертатимуться навколо осі.
<i>PAX1, PAX2</i>	Точки, що відзначають положення осі обертання.
<i>ARC</i>	Величина обертання в градусах.
<i>NSEG</i>	Кількість окремих площин, на які буде роздроблено повну поверхню (максимум 8).

Якщо попередньо побудувати дві точки, що належать осі обертання виробу та визначити максимальні значення номерів точок та ліній, остання з котрих – сплайн, то можна адекватно використати команду AROTAT:

```

K,,101,0,0
K,,1004,0,0
*GET,_MaxKP,KP,Cnt,NUM,MAX
*GET,_MaxLN,LINE,0,NUM,MAX
AROTAT,_MaxLN,,,,,_MaxKP-1,_MaxKP,90,4
Останні два параметри можуть мати інші значення.

```

### 3.2. Розробка програмного засобу

Для розробки оболонки, що служить генератором вхідного файлу ANSYS автор дипломного проекту обрав мову Object Pascal (середовище Delphi 7). Вибір обґрунтований, по-перше, суб'єктивними факторами автора; по-друге, розповсюдженістю операційних систем серії Windows, під керівництвом яких програма ANSYS успішно працює.

#### 3.2.1. Організація даних

Для збереження ключових даних під час виконання програми-оболонки використано тип даних **TList** для роботи зі списком координат точок та тип даних **TStringList** роботи зі змістом файлу, що генерується.

Об'єкт класу **TList** складає список покажчиків на запис типа:

```

TXYRecord = record
    X, Y : Extended;
end; //TXYRecord

```

$TPointRec = ^TXYRecord;$

який зберігає пари координат точок в площині ХУ.

Список **TList** після кожної зміни сортується та перевіряється на співпадіння записів.

Всі загальні типи даних, константи, та змінні організовані в окремий модуль (див. додаток А).

### 3.2.2. Організація інтерфейсу

Віконний інтерфейс організовано покроковим методом. На першому кроці (рис. 3.1) виконується вибір між ручним вводом координат чи імпортом точок із зовнішнього файлу формату DXF (Рис. 3.2).

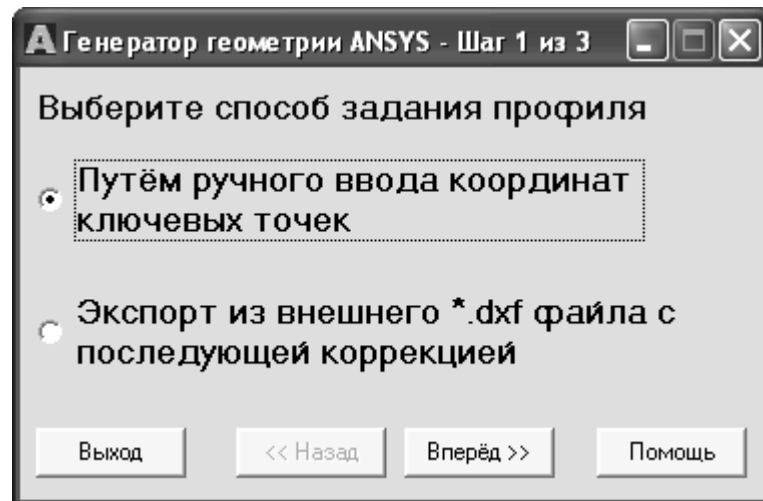


Рис. 3.1. Вікно першого кроку

Формат DXF було обрано з погляду двох причин. По-перше, він підтримується всіма розповсюдженими CAD-системами. По-друге, він організований в вигляді текстового ASCII-файлу, що забезпечує простоту в роботі з ним. Достатньо, щоб в DXF-файлі були збережені точки. Він їх зберігає за директивою POINT:

за директивами « 10» – координата Х та« 20» – координата У.

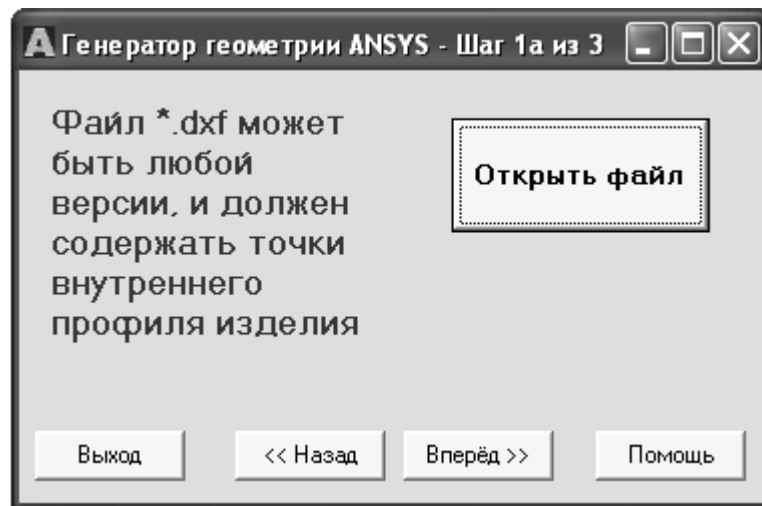


Рис. 3.2. Вікно імпорту

На другому кроці виконується огляд та правка списку точок профілю (рис. 3.3).

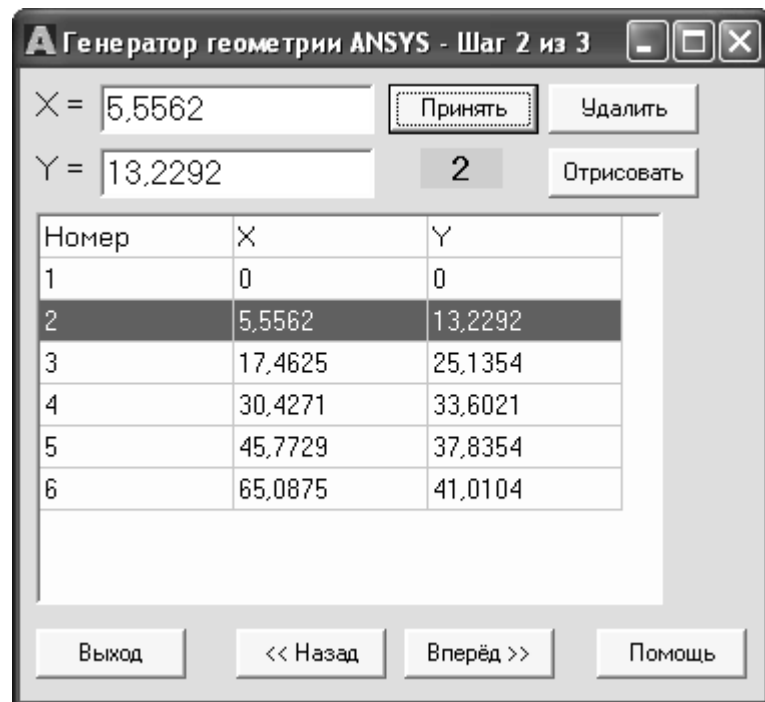


Рис. 3.3. Вікно другого кроку

Кнопка «Отрисовать» відображає профіль у вигляді відрізків (рис. 3.4).

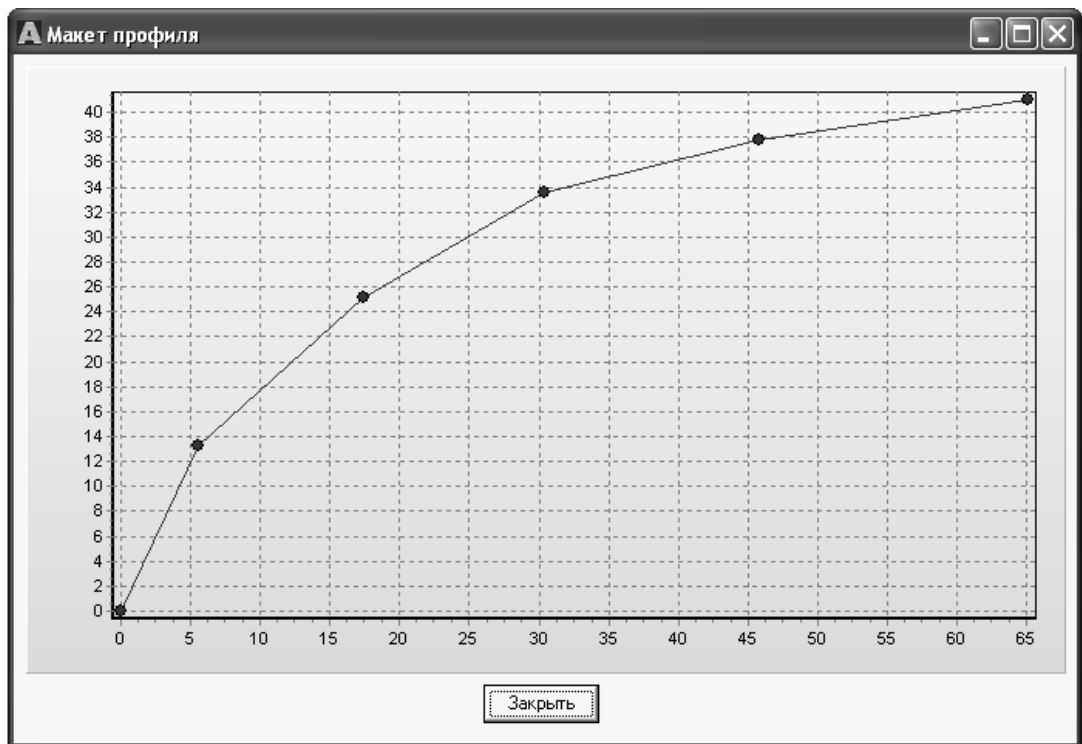


Рис. 3.4. Вікно візуалізації.

Це вікно може залишатися на екрані під час корекції списку точок.

На останньому кроці виконуються підбір останніх параметрів та генерація файлу (рис. 3.5).

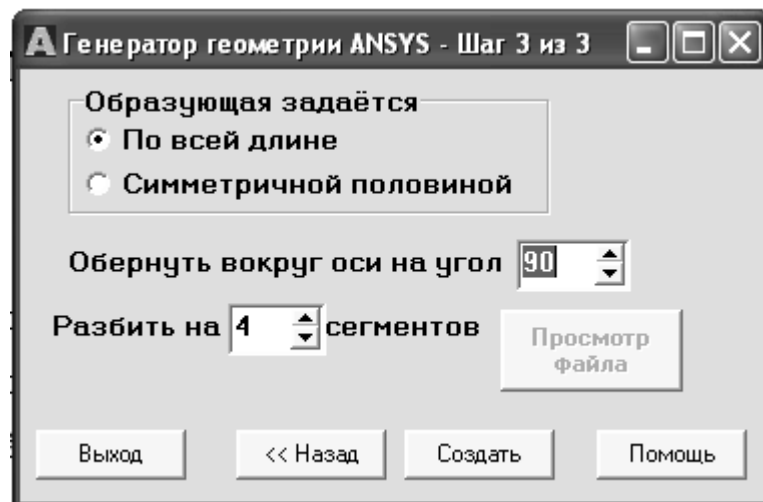


Рис. 3.5. Останній крок генератора

Можна також задати профіль однією половиною по довжині, якщо майбутній виріб є симетричним відносно серединного поперечного розрізу.

Після генерації файлу є можливість його продивитися в Блокноті та повернутись на будь який крок програми з метою виправлення недоліків.

Код програми-генератора наведений у додатку А.

### **3.3. Висновки до розділу**

Таким чином, в процесі виконання роботи було створено програму – генератор вхідного файлу інженерної системи ANSYS для коректної побудови геометрії виробу, отриманого методом намотування. Це – перший етап створення моделюючого середовища, на базі котрого будуть створюватись засоби побудови кінцево-елементної сітки згенерованої поверхні, надання їм матеріальних властивостей згідно особливостям процесу намотування та виконання аналізу в середовищі ANSYS.



## ВИСНОВКИ

В границях завдання створення математичної моделі щодо методу намотування дана робота запропонувала вирішення першого кроку – коректного створення/імпорту геометрії майбутнього виробу, для подальшого аналізу. За основу взято систему інженерного аналізу ANSYS [8] як програмний комплекс не залежний від програмно-апаратної платформи, який налічує багату бібліотеку можливостей фізико-механічного аналізу, в якого є засоби моделювання багатошарових КМ.

В роботі:

1) проаналізовано метод намотування з позицій системного підходу та виявлено проблему потреби попереднього аналізу технічної підготовки виробництва, який в сучасних умовах виконується за допомогою обчислювальних засобів (персональних комп'ютерів, зокрема) в альянсі з програмними засобами інженерного аналізу.

2) зроблено аналіз можливостей системи ANSYS на предмет створення/імпорту геометрії. Найприємнішою з точки зору особливостей подальшого аналізу є побудова геометрії власними засобами ANSYS. Для його реалізації є потреба створення деякої програмної оболонки, що генерувала б вхідний файл на мові APDL, виконання якого в ANSYS автоматично буде необхідну в подальшому геометрію. Однією з вимог до оболонки є те, що вона повинна мати можливість імпорту геометрії (принаймні, точок) з поширених графічних форматів.

3) Створено програму – генератор вхідного файлу інженерної системи ANSYS для коректної побудови геометрії виробу, отриманого методом намотування. Це – перший етап створення моделюючого середовища, на базі якого будуть створюватись засоби побудови кінцево-елементної сітки згенерованої поверхні, надання їм матеріальних властивостей згідно особливостям процесу намотування та виконання аналізу в середовищі ANSYS.

Мета роботи – розробити програмний засіб, що генерує коректну геометрію майбутнього намотаного виробу в середовищі ANSYS та забезпечує можливості імпорту геометрії з поширених графічних форматів та її власноручну правку – виконана.

Програма-генератор займає один EXE-файл розміром 653 KB, та потребує операційної системи WINDOWS.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ANSYS. Basic Analysis Procedures Guide. Rel. 5.7. / ANSYS Inc. Houston, 2000.
2. ANSYS. Commands Reference. Rel. 5.7. / ANSYS Inc. Houston, 2000.
3. ANSYS. Elements Reference. Rel. 5.7. / ANSYS Inc. Houston, 2000.
4. ANSYS. Theory Reference. Rel. 5.7. Ed. P. Kothnke / ANSYS Inc. Houston, 2000.
5. ANSYS. Verification Manual. Rel. 5.7. / ANSYS Inc. Houston, 2000.
6. Ashok K., Munjal. Use of fiber reinforced composites in rocket motor industry // 17 th National SAMPE technical conference, october 22-24.- 1985.- p. 371-382.
7. Ashok K. Munjal. Optimization of design allowables for composite structures // SAMPE Quart.,1987. – 18. – No2. – p.18-27.
8. Donald L., Hunston M. In: Composites update (The Newsletter of the University of Delaver Center for Composite Materials), Fall, 1992.
9. Donald L., Hunston M. In: Composites update (The Newsletter of the University of Delaver Center for Composite Materials), Fall, 1992.
10. Employment of database management systems for structural engineering applications / M.Klein. C.Stavrinidis, R.Valent, J.Dubois // AIAA/ASME/ASCE/AHS 28th Struct., Struct. Dyn. and Mater. Conf., Monterey, Calif., Apr. 6-8. 1987. Collect. Techn. Pap. Pt 1.- New York, 1987.- P. 579-586.
11. Hayashi R. Future of composite material asstssment technigues // Frans. JSCM.- V. 11.- № 2.- P. 41-44.
12. Hollingsworth R.D., Osment D.R. Filament wound thermoplastic matrix pressure vessels // International SAMPE Symposium. – 1987, April 6-9. – p.662-669.
13. Morris E.E., Segimoto M., Lynn V. Structural Composites Industries. AIAA-86-1504 Lighter Weight Fiber/ Metal Pressure Vessels Using Carbon Overwrap. – 1986. – p.1-9.

14. Rach V.A. Composite materials of non-continuous structure // Advanced materials and processes / Third Russian-Chinese Symposium.-Kaluga, Russia.-October 9-12, 1995. – p. 43.
15. Андреевская Г.Д. Высокопрочные ориентированные стеклопластики. М.: Химия, 1966.
16. Брайан Р. Нотон. О некоторых трудностях при использовании новых материалов // Композиционные материалы. Т. 3. Применение композиционных материалов в технике. - М., 1978.-С. 491-495.
17. Буров А.К., Андреевская Г.Д. Синтетические волокнистые анизотропные структуры. М.: Изд-во АН СССР, 1952.
18. Васильев В.В., Протасов В.Д., Болотин В.В. и др. Композиционные материалы: Справочник. – Под общ. ред. Васильева В.В., Тарнопольского Ю.М. – М.: Машиностроение, 1990. – 512 с.
19. Влияние ширины ленты на несущую способность емкостей высокого давления, изготовленных намоткой / В.А.Гречишкин, Г.Р.Борох, Н.Н.Белякова, В.А.Калинин // Конструкции из композиционных материалов. - 1990.- № 2.- С. 41-45. (ДСП).
20. Гайдачук В.Е., Паршин В.М., Цыбульник И.М. Влияние технологии изготовления на механические свойства материала конструкций из композитов // Прочность конструкций летательных аппаратов.- Харьков, 1974. - Вып. 2. - С. 103-111.
21. Гайдачук В.Е., Кобрин В.Н., Молодцов Г.А. Технология производства летательных аппаратов из композиционных материалов: Учеб. пособие. – Харьков: Харьк. авиац. ин-т, 1989. – 332 с.
22. Гуняев Г.М. Проектирование высококомодульных полимерных композитов с заданными свойствами // Композиционные материалы.- М., 1981.- С. 24-28.
23. Гуняев Г.М. Проектирование высококомодульных полимерных композитов с заданными свойствами // Композиционные материалы – М.: Машиностроение, 1981. – С. 24-28.

- 24.Дубровин Б.А., Новиков С.П., Фоменко А.Т. Современная геометрия. - М.: Наука, 1979.- 719с.
- 25.Егоров Л.А., Крылов М.С., Якунин С.П. Оптимизация физико-механических характеристик органопластиков // Применение полимерных композиционных материалов в машиностроении: Тез. докл. 1 Всесоюз. науч.-техн. семинара.- Ворошиловград, 1987. – С. 166.
- 26.Забашта В.Ф., Кривов Г.О., Бондарь В.Г. Полимерные композиционные материалы конструкционного назначения. – К.: Техника, 1993 – 160 с.
- 27.Зеленский Э.С. Взаимосвязь структуры и свойств однонаправленных армированных пластиков, получаемых методом намотки: Дис. ... д-ра техн. наук (в форме науч. докл.). – М.,1990. – 60 с. (ДСП).
- 28.Зеленский Э.С., Кульков А.А., Куперман А.М. и др. Технология намоточных пластиков: Журн. Всесоюз. хим. о-ва им. Д.И. Менделеева.– 1989. – N 5. – с. 515-520.
- 29.Калинин В.А. Теоретические основы геометрического моделирования процессов намотки и выкладки конструкций из волокнистых композиционных материалов. // Автореферат на соиск. уч. ст. доктора техн. наук. - Москва, 1997.- 50с.
- 30.Калинин В.А., Якунин В.И. Геометрическое моделирование технологического процесса намотки в производстве ЛА. - М.: изд-во МАИ, 1995.- 68с.
- 31.Калинчев В.А., Макаров М.С. Намоточные стеклопластики. – М.: Химия, 1986. – 272 с.
- 32.Келлерер Х., Геркерт С.М. Композиционные материалы в авиационно-космической промышленности: современное применение и развитие в будущем // Достижения в области композиционных материалов. - М., 1982.- С. 242-266.
- 33.Корб Л. Дж. Космические летательные аппараты // Композиционные материалы. Т. 3. Применение композиционных материалов в технике. - М., 1978.- С. 78-129.

- 34.Кривелли-Висконти И. Конструирование деталей из композиционных материалов // Достижения в области композиционных материалов.- М., 1982.- С. 58-69.
- 35.Любутин О.С. Автоматизация производства стеклопластиков. – М.: Химия, 1969. – 256 с.
- 36.Малков И.В. Научные основы технологии формообразования намоткой технологии углепластиковых элементов ферменных конструкций космических аппаратов. – Луганск: Изд-во Восточноукр. нац. ун-та, 2001. – 30 с.
- 37.Могильний Г.А., Текучев О.Ю.. Особливості моделювання виробів з композиційних матеріалів, що випускаються методом намотування // Вісник СНУ – 2003– №8 – С. 96 – 99.
- 38.Образцов И.Ф. Проблемы создания эффективных моделей и методов для расчета сложных пространственных конструкций // Механика и научно-технический прогресс. Т. 3. Механика деформируемого тела.- М., 1988.- С. 7-22.
- 39.Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ.- М.: Высш. шк., 1989.- 367 с.
- 40.Поляков В.П., Зеленский Э.С. Некоторые особенности технологии намотки изделий из композиционных материалов // Журн. Всесоюз. хим. о-ва им. Д.И. Менделеева. – 1978.- N 3. – с. 293 – 297.
- 41.Рач В.А. Инновационная деятельность: системные аспекты// Вісник Східноукраїнського державного університету.-1997.- №2(6).- С.120-127.
- 42.Рач В.А. Создание корпусов малогабаритных РДТТ одноразового и кратковременного действия из армированных пластиков: Автореф. дис....д-ра техн. наук.- Харьков,1992.- 42с.
- 43.Рач В.А. Опыт и проблемы применения композиционных материалов в железнодорожном транспорте // Проблемы развития локомотивостроения. Тез.докл.IV междун. научн.-техн. конф., 19-24 апреля 1993г., Луганск, ЛМСИ, с.72.

44. Рач В.А., Малков И.В. Классификация многолучевых фитингов ферменных конструкций из композиционных материалов // Вестник Восточноукр. гос. ун-та. Сер. Машиностроение. – Луганск: Изд-во ВУГУ. – 1996. – С. 168-172.
45. Рашевский П.К. Курс дифференциальной геометрии. - М.: Гостехиздат, 1946.- 420с.
46. Росато Д.В., Грове К.С. Намотка стеклонитью: Пер. с англ. – М.: Машиностроение, 1969. – 310 с.
47. Рыбников С.И. Автоматическое управление намоткой. – М.: Энергия, 1972. – 112 с.
48. Справочник по композиционным материалам: В 2-х кн. Кн. 1 / Под ред. Дж. Люблина; Пер. с англ. А.Б. Геллера, М.М. Гельмонта; Под ред. Б.Э. Геллера. – М.: Машиностроение, 1988. – 448 с.: ил.
49. Томашевский В.Т., Яковлев В.С. Основы теории и задачи оптимизации технологических проектов изделий из композитных материалов // Механика композитных материалов. - 1984.- № 5.- С. 888-899.
50. Чамис К.К. Проектирование элементов конструкций из композитов // Композиционные материалы. Т. 8. Анализ и проектирование конструкций / Под ред. Л. Браутмана и Р. Крока.- М., 1978.- С. 214-254.
51. Шукшунов В.Е., Жуковский В.Г., Евченко А.И. и др. Автоматизированные системы управления намоточными станками. – М.: Машиностроение, 1985. – 208 с.
52. Э.С. Зеленский, А.М. Куперман, Ю.А. Горбаткина, В.Г. Иванова-Мумжиева, А.А. Берлин. Армированные пластики, современные конструкционные материалы // Рос. хим. ж. (Ж. Рос. хим. об-ва им. Д.И. Менделеева), 2001, т. XLV, № 2. – С. 56-74.

## Додаток А.

### Програмний код

#### Зміст файлу GraphGen.dpr

```
program GraphGen;

uses
  Forms, uSource, uMainForm, uForm2s, uForm2f,
  uGraph, uForm3;

{$R *.res}

begin
  with Application do
    begin
      Initialize;
      CreateForm(TForm1, Form1);
      CreateForm(TForm2s, Form2s);
      CreateForm(TForm2f, Form2f);
      CreateForm(TForm3, Form3);
      Run;
    end;
  end.
end.
```

#### Зміст модуля uMainForm

```
unit uMainForm;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, ExtCtrls, uSource;

type
  TForm1 = class(TForm)
    lbStep1: TLabel;
    rbtnSpline: TRadioButton;
    rbtnFunc: TRadioButton;
    procedure FormCreate(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure btnHelpClick(Sender: TObject);
    procedure btnGoClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormHide(Sender: TObject);
  protected
    procedure WMGetSysCommand(var Message : TMessage);
      message WM_SYSCOMMAND;
  end;

var
  Form1 : TForm1;
  btnExit : TButton;
  btnHelp : TButton;
  btnGo : TButton;
  btnBack : TButton;
  procedure CloseAll;

implementation
{$R *.dfm}
uses uForm2s, uForm2f;

procedure TForm1.FormCreate(Sender: TObject);
```



```

begin
Height:= fmHeight;
Width := fmWidth;
Left := fmLeft;
Top := fmTop;
btnExit := TButton.Create(self);
btnExit.Parent := Form1;
btnExit.Caption := btnExitText;
btnExit.Top := ClientHeight-btnTop;
btnExit.Left := btnExitLeft;
btnExit.OnClick := btnExitClick;
btnExit.Anchors := [akBottom];
btnHelp := TButton.Create(self);
btnHelp.Parent := Form1;
btnHelp.Caption := btnHelpText;
btnHelp.Top := ClientHeight-btnTop;
btnHelp.Left := btnHelpLeft;
btnHelp.OnClick := btnHelpClick;
btnHelp.Anchors := [akBottom];
btnGo := TButton.Create(self);
btnGo.Parent := Form1;
btnGo.Caption := btnGoText;
btnGo.Top := ClientHeight-btnTop;
btnGo.Left := btnGoLeft;
btnGo.OnClick := btnGoClick;
btnGo.Anchors := [akBottom];
btnBack := TButton.Create(self);
btnBack.Parent := Form1;
btnBack.Caption := btnBackText;
btnBack.Top := ClientHeight-btnTop;
btnBack.Left := btnBackLeft;
btnBack.Anchors := [akBottom];
end;

procedure CloseAll;
begin
Form1.Close;
end;

procedure TForm1.btnExitClick(Sender: TObject);
begin
if CloseOrNot then CloseAll;
end;

procedure TForm1.FormShow(Sender: TObject);
begin
if Form2s.Showing then Form2s.Hide;
if Form2f.Showing then Form2f.Hide;
Caption:=CapMainForm+CapStep1;
Left := fmLeft;
Top := fmTop;
btnBack.Enabled:=False;
Step:=1;
end;

procedure TForm1.btnHelpClick(Sender: TObject);
begin
IWantHelp;
end;

procedure TForm1.btnGoClick(Sender: TObject);
begin
fmLeft := Left;

```

```

fmTop := Top;
Step := 1;
if rbtnSpline.Checked then
begin
  Form1.Hide;
  Form2s.Show;
end
else
begin
  Form1.Hide;
  Form2f.Show;
end;//if
end;

procedure TForm1.FormHide(Sender: TObject);
begin
  fmLeft := Left;
  fmTop := Top;
end;

procedure TForm1.WMGetSysCommand(var Message : TMessage) ;
begin
  if (Message.wParam = SC_CLOSE) then
  begin
    if CloseOrNot then CloseAll
    end
  else Inherited;
end;

end.

```

## Зміст модуля uForm2f

```

unit uForm2f;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, ExtCtrls, uSource;

type
  TForm2f = class(TForm)
    lbTyNePonyal: TLabel;
    odDXF: TOpenDialog;
    btnOpenDXF: TButton;
    procedure Form2fCreate(Sender: TObject);
    procedure btnExit2fClick(Sender: TObject);
    procedure btnHelp2fClick(Sender: TObject);
    procedure btnBack2fClick(Sender: TObject);
    procedure btnGo2fClick(Sender: TObject);
    procedure Form2fShow(Sender: TObject);
    procedure Form2fHide(Sender: TObject);
    procedure btnOpenDXFClick(Sender: TObject);
    procedure odDXFCanClose(Sender: TObject; var CanClose: Boolean);
  protected
    procedure WMGetSysCommand(var Message : TMessage);
      message WM_SYSCOMMAND;
  end;

var
  Form2f : TForm2f;
  btnExit2f : TButton;
  btnHelp2f : TButton;

```

```

btnGo2f : TButton;
btnBack2f : TButton;
Menu2f : TMainMenu;

```

implementation

```
{ $R *.dfm }
```

```
uses uMainForm, uForm2s;
```

```
procedure TForm2f.Form2fCreate(Sender: TObject);
```

```
begin
```

```
with Form2f do
```

```
begin
```

```
Height := fmHeight;
```

```
Width := fmWidth;
```

```
Color := fmColor;
```

```
Caption:=CapMainForm+CapStep3;
```

```
btnExit2f := TButton.Create(Form2f);
```

```
btnExit2f.Parent := Form2f;
```

```
btnExit2f.Caption := btnExitText;
```

```
btnExit2f.Top := ClientHeight-btnTop;
```

```
btnExit2f.Left := btnExitLeft;
```

```
btnExit2f.OnClick := Form2f.btnExit2fClick;
```

```
btnHelp2f := TButton.Create(Form2f);
```

```
btnHelp2f.Parent := Form2f;
```

```
btnHelp2f.Caption := btnHelpText;
```

```
btnHelp2f.Top := ClientHeight-btnTop;
```

```
btnHelp2f.Left := btnHelpLeft;
```

```
btnHelp2f.OnClick := Form2f.btnHelp2fClick;
```

```
btnGo2f := TButton.Create(Form2f);
```

```
btnGo2f.Parent := Form2f;
```

```
btnGo2f.Caption := btnGoText;
```

```
btnGo2f.Top := ClientHeight-btnTop;
```

```
btnGo2f.Left := btnGoLeft;
```

```
btnGo2f.OnClick := Form2f.btnGo2fClick;
```

```
btnBack2f := TButton.Create(Form2f);
```

```
btnBack2f.Parent := Form2f;
```

```
btnBack2f.Caption := btnBackText;
```

```
btnBack2f.Top := ClientHeight-btnTop;
```

```
btnBack2f.Left := btnBackLeft;
```

```
btnBack2f.OnClick := Form2f.btnBack2fClick;
```

```
end;//with
```

```
end;
```

```
procedure TForm2f.btnExit2fClick(Sender: TObject);
```

```
begin
```

```
if CloseOrNot then CloseAll;
```

```
end;
```

```
procedure TForm2f.btnHelp2fClick(Sender: TObject);
```

```
begin
```

```
IWantHelp;
```

```
end;
```

```
procedure TForm2f.btnBack2fClick(Sender: TObject);
```

```
begin
```

```
fmLeft := Left;
```

```
fmTop := Top;
```

```
Form2f.Hide;
```

```
Form1.Show;
```

```
end;
```

```
procedure TForm2f.btnGo2fClick(Sender: TObject);
```

```
begin
```

```

Step := 21;
Form2f.Hide;
Form2s.Show;
end;

procedure TForm2f.Form2fShow(Sender: TObject);
begin
  Left := fmLeft;
  Top := fmTop;
  Color := fmColor;
  Step := 2;
end;

procedure TForm2f.Form2fHide(Sender: TObject);
begin
  fmLeft := Left;
  fmTop := Top;
end;

procedure TForm2f.WMGetSysCommand(var Message : TMessage) ;
begin
  if (Message.wParam = SC_CLOSE) then
    begin
      if CloseOrNot then CloseAll
      end
    else Inherited;
  end;
end;

procedure TForm2f.btnOpenDXFClick(Sender: TObject);
begin
  btnOpenDXF.Enabled := False;
  odDXF.Execute;
end;

procedure TForm2f.odDXFCanClose(Sender: TObject; var CanClose: Boolean);
const
  KeyWordP = 'POINT';
  KeyWordA = 'AcDbPoint';
  KeyWordX = ' 10';
  KeyWordY = ' 10';
var DXF : TextFile;
    Buf : String;
begin
  CanClose := True;
  try
    Buf := odDXF.FileName;
    AssignFile(DXF,Buf);
    Reset(DXF);
    repeat
      Readln(DXF,Buf);
      if Buf = KeyWordP then
        begin
          repeat
            Readln(DXF,Buf);
          until (Buf = KeyWordX) or (Buf = KeyWordY);
          New(XYRecord);
          Readln(DXF,Buf);
          XYRecord^.X := StrToFloat(PointToComma(Buf));
          Readln(DXF,Buf);
          Readln(DXF,Buf);
          XYRecord^.Y := StrToFloat(PointToComma(Buf));
          XYList.Add(XYRecord);
        end;
      //if Buf

```

```

until EoF(DXF);
CloseFile(DXF);
except
on EReadError do
begin
Application.MessageBox('Ошибка чтения','Ошибка',0);
end;//on EReadError
end;//try
btnOpenDXF.Enabled := True;
if XYList.Count > 1 then
XYList.Sort(@CompItems);
DuplDel;
end;

end.

```

## Зміст модуля uForm2s

```

unit uForm2s;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, StdCtrls, ExtCtrls, uSource, Mask, ComCtrls, Grids;

type
TForm2s = class(TForm)
btnAccept: TButton;
edX: TEdit;
lbInfoX: TLabel;
lbInfoY: TLabel;
edY: TEdit;
TableXY: TStringGrid;
btnGraph: TButton;
btnDelete: TButton;
lbItem: TLabel;
procedure FormCreate(Sender: TObject);
procedure btnExit2Click(Sender: TObject);
procedure btnHelp2Click(Sender: TObject);
procedure btnBack2Click(Sender: TObject);
procedure btnGo2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormHide(Sender: TObject);
procedure btnAcceptClick(Sender: TObject);
procedure btnGraphClick(Sender: TObject);
procedure TableXYSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
procedure btnDeleteClick(Sender: TObject);
protected
procedure WMGetSysCommand(var Message : TMessage);
message WM_SYSCOMMAND;
end;

var
Form2s : TForm2s;
btnExit2 : TButton;
btnHelp2 : TButton;
btnGo2 : TButton;
btnBack2 : TButton;

function CompItems(Item1, Item2: Pointer):Integer;

implementation

```

```

uses uMainForm, uGraph, Math, uForm2f, uForm3;
{$R *.dfm}

procedure UpdateMemo;
var i : Integer;
    Rec : TPointRec;
begin
Form2s.TableXY.RowCount := XYList.Count+1;
for i:=1 to XYList.Count do
begin
Form2s.TableXY.Cells[0,i] := IntToStr(i);
Rec := XYList.Items[i-1];
Form2s.TableXY.Cells[1,i] := FloatToStr(Rec^.X);
Form2s.TableXY.Cells[2,i] := FloatToStr(Rec^.Y);
end;
end;//procedure UpdateMemo

function CompItems(Item1, Item2: Pointer):Integer;
type PExtended = ^Extended;
begin
if PExtended(Item1)^ < PExtended(Item2)^ then
Result := -1
else
if PExtended(Item1)^ > PExtended(Item2)^ then
Result := 1
else
Result := 0;
end;

function ScanInput(InputText : String):String;
var FBuffer : String;
    FNum : Extended;
begin
FBuffer := InputText;
if FBuffer = " " then
FBuffer := '0';
try
FNum := StrToFloat(FBuffer);
except
on EConvertError do
begin
FBuffer := PointToComma(FBuffer);
try
FNum := StrToFloat(FBuffer);
except
on EConvertError do
FBuffer := NeTo;
end;
end;//on
end;//try
Result := FBuffer;
end;//function ScanInput

procedure UpdateMaxMin;
begin
if XYRecord^.X > MaxX then MaxX := XYRecord^.X;
if XYRecord^.X < MinX then MinX := XYRecord^.X;
if XYRecord^.Y > MaxY then MaxY := XYRecord^.Y;
if XYRecord^.Y < MinY then MinY := XYRecord^.Y;
end;//UpdateMaxMin

procedure TForm2s.FormCreate(Sender: TObject);
begin

```

```

Height := fmHeight+100;
Width := fmWidth;
Color := fmColor;
Caption := CapMainForm+CapStep2;
btnExit2 := TButton.Create(Form2s);
btnExit2.Parent := Form2s;
btnExit2.Caption := btnExitText;
btnExit2.Top := ClientHeight-btnTop;
btnExit2.Left := btnExitLeft;
btnExit2.OnClick := Form2s.btnExit2Click;
btnExit2.Anchors := [akBottom];
btnHelp2 := TButton.Create(Form2s);
btnHelp2.Parent := Form2s;
btnHelp2.Caption := btnHelpText;
btnHelp2.Top := ClientHeight-btnTop;
btnHelp2.Left := btnHelpLeft;
btnHelp2.OnClick := Form2s.btnHelp2Click;
btnHelp2.Anchors := [akBottom];
btnGo2 := TButton.Create(Form2s);
btnGo2.Parent := Form2s;
btnGo2.Caption := btnGoText;
btnGo2.Top := ClientHeight-btnTop;
btnGo2.Left := btnGoLeft;
btnGo2.OnClick := Form2s.btnGo2Click;
btnGo2.Anchors := [akBottom];
btnBack2 := TButton.Create(Form2s);
btnBack2.Parent := Form2s;
btnBack2.Caption := btnBackText;
btnBack2.Top := ClientHeight-btnTop;
btnBack2.Left := btnBackLeft;
btnBack2.OnClick := Form2s.btnBack2Click;
btnBack2.Anchors := [akBottom];
TableXY.Cells[0,0] := 'Home';
TableXY.Cells[1,0] := 'X';
TableXY.Cells[2,0] := 'Y';
TableXY.Height := ClientHeight-TableXY.Top-btnTop-12;
end;

```

```

procedure TForm2s.btnExit2Click(Sender: TObject);
begin
  if CloseOrNot then Form1.Close;
end;

```

```

procedure TForm2s.btnHelp2Click(Sender: TObject);
begin
  IWantHelp;
end;

```

```

procedure TForm2s.btnBack2Click(Sender: TObject);
begin
  fmLeft := Left;
  fmTop := Top;
  Form2s.Hide;
  if (Step = 1) then
  begin
    Form1.Show;
  end
  else
  begin
    Form2f.Show;
  end;
end;
end;

```

```

procedure TForm2s.btnGo2Click(Sender: TObject);
begin
  Form2s.Hide;
  Form3.Show;
end;

procedure TForm2s.FormShow(Sender: TObject);
begin
  Left := fmLeft;
  Top := fmTop;
  if XYList.Count > 1 then
    XYList.Sort(@CompItems);
  UpdateMemo;
  UpdateMaxMin;
end;

procedure TForm2s.FormHide(Sender: TObject);
begin
  fmLeft := Left;
  fmTop := Top;
end;

procedure TForm2s.btnAcceptClick(Sender: TObject);
var SelClls : TGridRect;
    Buffer : String;
begin
  Buffer := ScanInput(edX.Text);
  edX.Text := Buffer;
  if Buffer <> NeTo then
    begin
      Buffer := ScanInput(edY.Text);
      edY.Text := Buffer;
      if Buffer <> NeTo then
        begin
          New(XYRecord);
          XYRecord^.X := StrToFloat(edX.Text);
          XYRecord^.Y := StrToFloat(edY.Text);
          if XYList.Add(XYRecord) > 0 then
            XYList.Sort(@CompItems);
          DuplDel;
          UpdateMaxMin;
          UpdateMemo;
          lbItem.Caption := IntToStr(XYList.IndexOf(XYRecord)+1);
          SelClls.Top := XYList.IndexOf(XYRecord)+1;
          SelClls.Bottom:= XYList.IndexOf(XYRecord)+1;
          SelClls.Left := 0;
          SelClls.Right := 2;
          TableXY.Selection := SelClls;
          fmGraph.FormPaint(fmGraph);
          Application.ProcessMessages;
          btnAccept.SetFocus;
        end;//if ScanInput(edY.Text)
      end;//if ScanInput(edX.Text)
    end;
end;

procedure TForm2s.WMGetSysCommand(var Message : TMessage) ;
begin
  if (Message.wParam = SC_CLOSE) then
    begin
      if CloseOrNot then CloseAll
      end
    else Inherited;
  end;
end;

```



```

procedure TForm2s.btnGraphClick(Sender: TObject);
begin
  fmGraph := TfmGraph.Create(fmGraph);
  btnGraph.Enabled := False;
  fmGraph.Show;
end;

procedure TForm2s.TableXYSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
var SelRow : TGridRect;
begin
  if (XYList.Count > 0)and(ARow > 0) then
  begin
    lbItem.Caption := IntToStr(ARow);
    XYRecord := XYList.Items[ARow-1];
    edX.Text := FloatToStr(XYRecord.X);
    edY.Text := FloatToStr(XYRecord.Y);
    with SelRow do
    begin
      Left := 0;
      Right := 2;
      Top := ARow;
      Bottom := ARow;
    end;
    TableXY.Selection := SelRow;
    CanSelect := False;
  end
  else CanSelect := False;
  btnAccept.SetFocus;
end;

procedure TForm2s.btnDeleteClick(Sender: TObject);
begin
  if XYList.Count > 0 then
    XYList.Delete(TableXY.Selection.Top - 1);
  UpdateMemo;
  fmGraph.FormPaint(fmGraph);
  Application.ProcessMessages;
  btnAccept.SetFocus;
end;

end.

```

### **Зміст модуля uForm3**

```

unit uForm3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, uSource, Spin, ExtCtrls;

type
  TForm3 = class(TForm)
    svdlgCreateFile: TSaveDialog;
    btnView: TButton;
    lbDegr: TLabel;
    seDegr: TSpinEdit;
    rgLine: TRadioGroup;
    Label1: TLabel;
    seSegm: TSpinEdit;

```

```

Label2: TLabel;
procedure FormCreate(Sender: TObject);
procedure btnExit3Click(Sender: TObject);
procedure btnHelp3Click(Sender: TObject);
procedure btnBack3Click(Sender: TObject);
procedure btnGo3Click(Sender: TObject);
function AnsysExe(Sender: TObject) : boolean;
procedure svdlgCreateFileCanClose(Sender: TObject;
    var CanClose: Boolean);
procedure btnViewClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormHide(Sender: TObject);
protected
    procedure WMGetSysCommand(var Message : TMessage);
        message WM_SYSCOMMAND;
end;

var
    Form3 : TForm3;
    btnExit3 : TButton;
    btnHelp3 : TButton;
    btnBack3 : TButton;
    btnGo3 : TButton;

implementation
{$R *.dfm}
uses uMainForm, uForm2s, uForm2f;

procedure CreateProFile;
begin
    //
end;

procedure TForm3.FormCreate(Sender: TObject);
begin
    Height := fmHeight;
    Width := fmWidth;
    Color := fmColor;
    Caption := CapMainForm+CapStep3;
    btnExit3 := TButton.Create(Form3);
    btnExit3.Parent := Form3;
    btnExit3.Caption := btnExitText;
    btnExit3.Top := ClientHeight-btnTop;
    btnExit3.Left := btnExitLeft;
    btnExit3.OnClick := Form3.btnExit3Click;
    btnExit3.Anchors := [akBottom];
    btnHelp3 := TButton.Create(Form3);
    btnHelp3.Parent := Form3;
    btnHelp3.Caption := btnHelpText;
    btnHelp3.Top := ClientHeight-btnTop;
    btnHelp3.Left := btnHelpLeft;
    btnHelp3.OnClick := Form3.btnHelp3Click;
    btnHelp3.Anchors := [akBottom];
    btnGo3 := TButton.Create(Form3);
    btnGo3.Parent := Form3;
    btnGo3.Caption := btnGoText;
    btnGo3.Top := ClientHeight-btnTop;
    btnGo3.Left := btnGoLeft;
    btnGo3.OnClick := Form3.btnGo3Click;
    btnGo3.Anchors := [akBottom];
    btnGo3.Caption := 'Создать';
    btnBack3 := TButton.Create(Form3);
    btnBack3.Parent := Form3;

```

```

    btnBack3.Caption := btnBackText;
    btnBack3.Top := ClientHeight-btnTop;
    btnBack3.Left := btnBackLeft;
    btnBack3.OnClick := Form3.btnBack3Click;
    btnBack3.Anchors := [akBottom];
end;

procedure TForm3.WMGetSysCommand(var Message : TMessage);
begin
    if (Message.wParam = SC_CLOSE) then
        begin
            if CloseOrNot then CloseAll
            end
            else Inherited;
        end;
end;

procedure TForm3.btnExit3Click(Sender: TObject);
begin
    if CloseOrNot then Form1.Close;
end;

procedure TForm3.btnHelp3Click(Sender: TObject);
begin
    IWantHelp;
end;

procedure TForm3.btnBack3Click(Sender: TObject);
begin
    Form3.Hide;
    Form2s.Show;
end;

function TForm3.AnsysExe(Sender: TObject) : boolean;
var Wnd : HWND;
    buff : array [0..127] of Char;
    Name : String[5];
    i : Integer;
begin
    Result := False;
    Wnd := GetWindow(Handle, gw_HWndFirst);
    while Wnd <> 0 do
        begin {Не показываем;}
            if (Wnd <> Application.Handle) and {-Собственное окно}
                IsWindowVisible(Wnd) and {-Невидимые окна}
                (GetWindow(Wnd, gw_Owner) = 0) and {-Дочерний окна}
                (GetWindowText(Wnd, buff, sizeof(buff))
                    <> 0) then
                begin
                    GetWindowText(Wnd, buff, sizeof(buff));
                    Name := "";
                    for i:=1 to 5 do
                        Name := Name + buff[i-1];
                    if Name = PriznakAnsys then Result := True;
                end;
                Wnd := GetWindow(Wnd, gw_hWndNext);
            end;
        end;
end;

function GetBatchForPics : String;
var Buffer : String;
    i,n : Integer;
    Coords : TPointRec;
    CoordC : TPointRec;

```

```

begin
n := XYList.Count;
Buffer := "";
if n > 1 then
begin
if Form3.rgLine.ItemIndex = 0 then
Buffer := '!* Постановка точек профиля'+cEOLN+
'FLST,3,'+IntToStr(n)+'8'+cEOLN
else
Buffer := '!* Постановка точек профиля'+cEOLN+
'FLST,3,'+IntToStr(2*n-1)+'8'+cEOLN;
CoordC := XYList.Items[n-1];
for i:=1 to n do
begin
Coords := XYList.Items[i-1];
Buffer := Buffer + 'FITEM,3,'+
CommaToPoint(FloatToStr(Coords^.X))+'+'+
CommaToPoint(FloatToStr(Coords^.Y))+'0'+
cEOLN;
end; //for i:=1 to n
if Form3.rgLine.ItemIndex = 1 then
for i:=n-1 downto 1 do
begin
Coords := XYList.Items[i-1];
Buffer := Buffer + 'FITEM,3,'+
CommaToPoint(FloatToStr(2*CoordC^.X-
Coords^.X))+'+'+
CommaToPoint(FloatToStr(Coords^.Y))+'+'+
'0'+cEOLN;
end; //for i:=n-1 downto 1
Result := Buffer;
end //n > 1
else
begin
Result := "";
end; //else (if n > 1)
end; //function GetBatchForPics : TStringList

```

```

function GetBatchForSpl : String;
var Buffer : String;
i,n : Integer;
begin
n := XYList.Count;
Buffer := "";
if n > 1 then
begin
if Form3.rgLine.ItemIndex = 0 then
Buffer := '!* Отрисовка сплайна'+cEOLN+
'FLST,3,'+IntToStr(n)+'3'+cEOLN
else
Buffer := '!* Отрисовка сплайна'+cEOLN+
'FLST,3,'+IntToStr(2*n-1)+'3'+cEOLN;
if Form3.rgLine.ItemIndex = 0 then
for i:=1 to n do
Buffer := Buffer+'FITEM,3,_MaxKP-'+
IntToStr(n-i)+cEOLN
else
for i:=1 to 2*n-1 do
Buffer := Buffer+'FITEM,3,_MaxKP-'+
IntToStr(2*n-i-1)+cEOLN; //if Form3.rgLine
Result := Buffer;
end //n > 1
else

```

```

begin
  Result := '';
end; //else (if n > 1)
end;

procedure TForm3.btnGo3Click(Sender: TObject);
begin
  svdlgCreateFile.Execute;
  btnView.Enabled := True;
end;

procedure TForm3.svdlgCreateFileCanClose(Sender: TObject;
  var CanClose: Boolean);
var CoordB : TPointRec;
    CoordE : TPointRec;
begin
  CoordB := XYList.Items[0];
  CoordE := XYList.Items[XYList.Count-1];
  try
    with ProFileText do
      begin
        Clear;
        Append(BeginAnsFile);
        Append(GetBatchForPics);
        Append(AnsDrawPic);
        Append('*GET,_MaxKP,KP,Cnt,NUM,MAX');
        Append(GetBatchForSpl);
        Append(AnsDrawSpline);
        if rgLine.ItemIndex=0 then
          Append('K,,'+CommaToPoint(FloatToStr(CoordB.X))+',0,0'+cEOLN+
            'K,,'+CommaToPoint(FloatToStr(CoordE.X))+',0,0')
        else
          Append('K,,'+CommaToPoint(FloatToStr(CoordB.X))+',0,0'+cEOLN+
            'K,,'+CommaToPoint(FloatToStr(2*CoordE.X))+',0,0');
        Append('*GET,_MaxKP,KP,Cnt,NUM,MAX');
        Append('*GET,_MaxLN,LINE,0,NUM,MAX');
        Append('AROTAT,_MaxLN,,,,,_MaxKP-1,_MaxKP, '+
          IntToStr(seDegr.Value)+' , '+
          IntToStr(seSegm.Value));
      end; //with ProFileText
    FullDirFile := svdlgCreateFile.FileName +
      svdlgCreateFile.DefaultExt;
    AssignFile(FileAns,FullDirFile);
    Rewrite(FileAns);
    Writeln(FileAns,ProFileText.Text);
    CloseFile(FileAns);
  except
    on EExternal do CanClose := False;
  end; //try
end;

procedure TForm3.btnViewClick(Sender: TObject);
var
  si : TStartupInfo;
  p : TProcessInformation;
  CmdLn: PAnsiChar;
begin
  FillChar( si, SizeOf( si ) , 0 );
  with si do
    begin
      cb := SizeOf( si );
      dwFlags := startf_UseShowWindow;
      wShowWindow := 4;
    end;
  CreateProcess(nil, CmdLn, nil, nil, false, 0, nil, nil, si, p);
end;

```

```

end;
CmdLn := PAnsiChar('notepad.exe '+FullDirFile);
CreateProcess(nil,CmdLn,nil,nil,false,
    Create_default_error_mode,nil,nil,si,p);
WaitForSingleObject(p.hProcess,INFINITE);
end;

procedure TForm3.FormShow(Sender: TObject);
begin
    Left := fmLeft;
    Top := fmTop;
end;

procedure TForm3.FormHide(Sender: TObject);
begin
    fmLeft := Left;
    fmTop := Top;
end;

end.

```

## Зміст модуля uGraph

```

unit uGraph;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ExtCtrls, uSource, TeEngine, Series, TeeProcs, Chart;

type
    TfmGraph = class(TForm)
        btnClose: TButton;
        Chart1: TChart;
        Series1: TLineSeries;
        procedure btnCloseClick(Sender: TObject);
        procedure FormPaint(Sender: TObject);
        procedure FormActivate(Sender: TObject);
    end;

var
    fmGraph: TfmGraph;

implementation
{$R *.dfm}
uses uForm2s;

const space = 10;

procedure TfmGraph.btnCloseClick(Sender: TObject);
begin
    Form2s.btnGraph.Enabled := True;
    fmGraph.Close;
end;

procedure TfmGraph.FormPaint(Sender: TObject);
var i : Integer;
    Coord : TPointRec;
begin
    if Chart1.SeriesCount>0 then
        Chart1.SeriesList[0].Clear;
    for i:=0 to XYList.Count-1 do

```

```

begin
  Coord := XYList.Items[i];
  Chart1.SeriesList[0].AddXY(Coord^.X,Coord^.Y);
end; //for
end;

procedure TfmGraph.FormActivate(Sender: TObject);
begin
  FormPaint(self);
end;

end.

```

## Зміст модуля uSource

```

unit uSource;
interface
uses Graphics, Forms, Classes;

{Надписи на кнопках}
const btnExitText  = 'Выход';
      btnHelpText  = 'Помощь';
      btnGoText    = 'Вперёд >>';
      btnBackText  = '<< Назад';
      btnFinishText = 'Готово';

{Расположение кнопок}
const btnExitLeft  = 8;
      btnHelpLeft  = 288;
      btnGoLeft    = 192;
      btnBackLeft  = 108;
      btnTop       = 36;
      {}
      fmHeigt = 250;
      fmWidth = 380;
      {}
      btnHeith = 24;
      btnWidth = 96;

const CapMainForm = 'Компилятор ANSYS';
      CapStep1    = ' - Шаг 1 из 3';
      CapStep2    = ' - Шаг 2 из 3';
      CapStep3    = ' - Шаг 3 из 3';

var
  fmColor : TColor = clMoneyGreen;
  fmLeft  : Integer = 200;
  fmTop   : Integer = 200;
  Step    : Integer = 1;

function CloseOrNot: boolean;
procedure IWantHelp;
function PointToComma(FSource : String):String;
function CommaToPoint(Comma: String):String;
procedure DuplDel;

const NeTo      = 'Ошибка !';
      cEOLN     = #13+#10;
      PriznakAnsys = 'ANSYS';
      BeginAnsFile = '/PREP7'+cEOLN;
      AnsDrawPic   = 'K, ,P51X'+cEOLN;
      AnsDrawSpline= 'BSPLINE, ,P51X'+cEOLN;
      TooFewXY     = 'Слишком мало координат точек профиля. Вернитесь на 2-й шаг и пополните список.';

```

```

type
  TXYList = TList; //Список координат точек
  TXYRecord = record //Шаблон записи координат точки
    X, Y : Extended;
  end; //TXYRecord
  TPointRec = ^TXYRecord;

var
  XYRecord : TPointRec;
  XYList : TXYList;
  MaxX,MinX : Extended;
  MaxY,MinY : Extended;

var ProFileText : TStringList;
  FullDirFile : String;
  FileAns : Text;

implementation

function CloseOrNot: boolean;
begin
  Result := (Application.MessageBox('Вы действительно хотите выйти'+#13+
    'из программы ?', 'Внимание !', 1)
    = 1);
end;

procedure IWantHelp;
begin
  Application.MessageBox('А помощи-то и нету !', 'Ой !', 0);
end;

function PointToComma(FSource : string) : string;
var Commas : Integer;
begin
  Commas := Pos('.', FSource);
  if Commas <> 0 then
    begin
      Delete(FSource, Commas, 1);
      Insert(',', FSource, Commas);
    end;
  Result := FSource;//;
end;

function CommaToPoint(Comma: String):String;
var Buffer : String;
  ComPos : Integer;
begin
  Buffer := Comma;
  ComPos := Pos('.', Buffer);
  if ComPos<>0 then
    begin
      Delete(Buffer, ComPos, 1);
      Insert('.', Buffer, ComPos);
    end;
  Result := Buffer;
end;

procedure DuplDel;
const errxy = 0;
var i : integer;
  key,oi : Boolean;

```



```

    RX    : Extended;
begin
repeat
key := False;
i := 0;
while (i <= XYList.Count-2) and
    (key = False) do
begin
    New(XYRecord);
    XYRecord := XYList.Items[i];
    RX := XYRecord^.X;
    XYRecord := XYList.Items[i+1];
    oi := (RX <= (XYRecord^.X + errxy))and
        (RX >= (XYRecord^.X - errxy));
    if oi then
        begin
            XYList.Delete(i+1);
            key := True;
        end;//if
        inc(i);
    end;//while
until key = False;
end;

initialization
XYList := TList.Create;
New(XYRecord);
ProFileText := TStringList.Create;

finalization
XYList.Free;
Dispose(XYRecord);
ProFileText.Free;
end.

```