

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

Навчально-науковий інститут фізики, математики та інформаційних
технологій

Кафедра інформаційних технологій та систем

Донченко Володимир Юрійович

**ДОСЛІДЖЕННЯ ТА РОЗРОБКА МОДЕЛІ ТРАЄКТОРІЇ
ГЕОДЕЗИЧНОЇ УКЛАДКИ АРМУЮЧОГО МАТЕРІАЛУ НА
ОПРАВКУ ВИРОБУ ЗА ДАНИМИ КІНЦЕВО-ЕЛЕМЕНТНОЇ МОДЕЛІ
ТА ПАРАМЕТРАМИ ПРОЦЕСУ НАМОТКИ В СЕРЕДОВИЩІ ANSYS**

кваліфікаційна робота

**здобувача вищої освіти другого (магістерського) рівня
освітньої програми «Комп'ютерні мережі»
за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис _____ Володимир ДОНЧЕНКО

Науковий керівник _____ Юрій ТИХОНОВ,
кандидат технічних наук, доцент
кафедри інформаційних технологій
та систем

Завідувач кафедри _____ Микола СЕМЕНОВ,
кандидат педагогічних наук, доцент
кафедри інформаційних технологій
та систем

Полтава – 2023

АНОТАЦІЯ

Донченко В.Ю.

Тема: Дослідження та розробка моделі траєкторії геодезичної укладки армуючого матеріалу на оправку виробу за даними кінцево-елементної моделі та параметрами процесу намотки в середовищі ANSYS.

Спеціальність: 123 «Комп'ютерна інженерія».

Установа: ЛНУ імені Тараса Шевченка, 2023р.

Магістерська робота містить: 78 с., 26 рис., 1 табл., 52 джерел.

Об'єкт дослідження – властивості геодезичних ліній на поверхнях різного способу утворення.

Предмет дослідження – використання властивостей геодезичних ліній конструювання поверхонь за заданою геодезичною лінією методом кінцево-елементної моделі.

Мета роботи - аналіз моделювання геодезичної лінії методом кінцево-елементної моделі та розробити програмний засіб, що генерує траєкторію геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі в середовищі ANSYS та забезпечує можливості вибору початку траєкторії з довільного кінцевого елемента.

Результати роботи – в роботі виконано огляд практичного застосування властивостей геодезичних ліній в прикладній геометрії поверхонь, способи їх відшукування і побудови та конструювання поверхонь за заданою геодезичною лінією. Проаналізовано можливості пакету інженерного аналізу ANSYS, внутрішньої мови APDL щодо побудови траєкторії геодезичної укладки по даним кінцево – елементної моделі. Розроблена математична модель розрахунку траєкторії укладання армуючого матеріалу та програму для розрахунку траєкторії геодезичної укладки по даним кінцево – елементної моделі.

Ключові слова: МЕТОД СКІНЧЕНИХ ЕЛЕМЕНТІВ, ГЕОДЕЗИЧНА ЛІНІЯ, ПРОЦЕС НАМОТУВАННЯ, МОВА APDL, ПРОГРАМА ANSYS, ІНТЕГРОВАНІ СИСТЕМИ, КІНЦЕВИЙ ЕЛЕМЕНТ, ТРАЄКТОРІЯ НАМОТУВАННЯ.

ANNOTATION

Donchenko Volodymyr

Theme: Research and development of a model of the trajectory of the geodesic laying of the reinforcing material on the mandrel of the product based on the data of the finite element model and the parameters of the winding process in the ANSYS environment.

Speciality: 123 "Computer Engineering".

Institution: Luhansk Taras Shevchenko National University (LTSNU), 2023 year.

Master's work of: 78 p., 26 im, 52 sources.

A research object of: - properties of geodesic lines on surfaces of various ways of formation.

The article of research- using the properties of geodesic lines for constructing surfaces with a given geodesic line using the finite element model method.

An aim of research is - analysis of geodesic line modeling using a finite element model and developing a software tool that generates a geodetic laying trajectory of a reinforcing material on a product mandrel according to a finite element model in the ANSYS environment and provides the ability to select the beginning of a trajectory from an arbitrary finite element.

Job performanes.- an overview of the practical application of the properties of geodesic lines in the applied geometry of surfaces, the methods of their retrieval and construction and construction of surfaces according to a given geodetic line is performed in the work. The possibilities of the ANSYS engineering analysis package, the internal language of the APDL for the construction of the geodetic stretch path according to the finite element model, are analyzed. The mathematical model of calculating the trajectory of laying reinforcing material and the program for calculating the trajectory of geodetic stacking according to the data of the finite element model are developed.

Keywords: FINAL ELEMENT METHOD, GEODESIC LINE, WINDING PROCESS, APDL LANGUAGE, ANSYS PROGRAM, INTEGRATED SYSTEMS, END ELEMENT, WINDING TRAJECTORY.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ЗАСОБИ МОДЕЛЮВАННЯ ВИРОБІВ, ОТРИМАНИХ МЕТОДОМ НАМОТУВАННЯ.....	11
1.1. Композиційні матеріали та їх типи	11
1.2. Геодезична лінія – найкоротший шлях між двома точками на поверхні	15
1.3. Геодезична лінія – траєкторія намотки армуючих на поверхню ниток	23
1.4. Геодезична лінія – ймовірна траєкторія примусового руху частинки технологічного матеріалу по поверхні.....	28
1.5. Висновки	31
РОЗДІЛ 2. ЗАСОБИ МОДЕЛЮВАННЯ ВИРОБІВ, ОТРИМАНИХ МЕТОДОМ НАМОТУВАННЯ.....	32
2.1. Огляд засобів моделювання	32
2.2. Загальні можливості ANSYS.....	38
2.2.1. Загальна методика розрахунку на міцність композиційних конструкцій методом скінченних елементів в програмі ANSYS	38
2.2.2. Призначення ANSYS.....	40
2.2.3. Інтерфейс користувача.....	41
2.2.4. Графічні можливості	42
2.2.5. Процесори	44
2.2.6. База даних.....	45
2.2.7. Побудова сітки.....	46
2.3. Мова APDL у середовищі ANSYS	51
2.3.1. Замовчення	52
2.3.2. Попередження й помилки.....	52
2.3.3. Інтерактивний режим у порівнянні з пакетним режимом.....	53
2.3.4. Робота LOG файлу.....	53
2.4. Висновки	54
РОЗДІЛ 3. ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ НАМОТУВАННЯ ДЛЯ ВИРОБІВ З КОМПОЗИЦІЙНИХ МАТЕРІАЛІВ	56
3.1. Математичні моделі розрахунку траєкторії геодезичної укладки по даним кінцево – елементної моделі без застосування диференціалів	57
3.1.1. Перша математична модель визначення координат геодезичної лінії	59
3.1.2. Друга математична модель визначення координат геодезичної лінії	67
3.2. Розробка програмного засобу	73
3.3. Висновки	76
ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ДОДАТОК А.....	84

ВСТУП

Однією із найдавніших задач застосування геодезичних ліній поверхні була задача морської навігації при прокладанні курсу корабля. Для цього були створені різноманітні карти земної поверхні, які мали полегшити мореплавання, тобто спростити кораблеводіння між двома портами за найкоротшою відстанню. Не дивлячись на те, що геодезична лінія для земної кулі, прийнятої за сферу, є дуга великого кола і давно відома, все ж таки курс прокладали по компасу, при якому корабель перетинав всі меридіани під заданим кутом, тобто рухався по локсодромі не найкоротшим шляхом. Перепоною у відшуканні геодезичних ліній поверхні є проблеми обчислювального характеру, які зводяться до розв'язування диференціальних рівнянь. Тільки для обмеженого переліку відомих поверхонь (циліндр, конус, псевдосфера) можна знайти рівняння геодезичних ліній в кінцевому вигляді. Для всіх інших, включаючи сферу, їх пошук зводиться до інтегрування диференціальних рівнянь чисельними методами. Тільки для обмеженого класу поверхонь диференціальне рівняння другого порядку можна понизити до першого або ж звести до інтегралу, як, наприклад, для поверхонь обертання. Проте і в цьому випадку чисельне інтегрування відомого виразу на основі теореми Клеро не дозволяє побудувати геодезичну лінію повністю, а тільки окремі її фрагменти. Це пояснюється тим, що при зростанні змінної, за якою відбувається інтегрування, геодезична лінія будується тільки при зміні меридіана в одному напрямі. При зміні напрямку геодезичної лінії на поверхні обертання подальше інтегрування стає неможливим, оскільки незалежна змінна не може змінюватися в зворотну сторону.

Окрім найкоротшої відстані між точками поверхні, геодезична лінія характеризується тим, що її головна нормаль в кожній точці збігається із нормаллю до поверхні. Ця властивість використовується в задачах армування оболонок за допомогою ниток. Якщо нитка намотана по геодезичній лінії, то вона не сповзатиме зі свого місця при виникненні напружень в оболонці і працюватиме на розтяг. Ще одна властивість геодезичних ліній - кінематична.

Якщо частинка на певній швидкості зустрічається із поверхнею, то далі вона рухається по геодезичній лінії, тобто здійснює інерційний рух, і тільки сила ваги частинки згодом змушує відхилитися її від цього курсу, причому чим більша швидкість частинки, тим більш точно її траєкторія руху наближається до геодезичної лінії. Ця властивість може бути покладена в основу проектування поверхонь робочих органів за бажаною траєкторією руху частинок технологічного матеріалу по них. Сучасні можливості комп'ютерних засобів і програмних продуктів усувають перешкоду для точної побудови геодезичних ліній на поверхні і тим само дозволяють розв'язувати прикладні задачі на новому якісному рівні.

Названі властивості геодезичних ліній знайшли своє застосування в працях багатьох дослідників. Перш за все це задачі зміцнення оболонок шляхом армування їх нитками. В цьому напрямі активно велися дослідження в московській школі прикладної геометрії. В період її розквіту були захищені кандидатські дисертації Парняковим А.Ф., Орловим М.В., Завидським А.В. [12, 18]. Останнім часом захищено дві докторських дисертації Битюковим Ю.І. і Калініним В.А. [11, 17]. Поряд з іншими задачами, в них вивчається питання допустимих відхилень намотки армуючих ниток від геодезичних ліній поверхні при забезпеченні функціональності оболонок. В наших дослідженнях геодезичні лінії поверхні розшуковуються настільки точно, наскільки її забезпечують алгоритми чисельних обчислень. Дослідженнями по намотці ниток по геодезичних лініях в українській школі прикладної геометрії в останні роки займається професор Куценко Леонід Миколайович та його учні Руденко С. Ю. і Табакова І. С. В різні роки геодезичні лінії поверхонь також привертали увагу інших науковців цієї школи [3, 16, 33]. Стосовно оберненої задачі, тобто конструювання поверхні за заданою геодезичною лінією із врахуванням інерційного руху частинки по поверхні дослідження є обмеженими. Зважаючи на це і було обрано напрям досліджень.

Намотування виконується на багатокоординатних намотувальних верстатах з числовим програмним керуванням (ЧПК). Відомі системи

автоматизованого програмування намотувальних верстатів (САП НВ) орієнтовані винятково на підготовку програм намотування (ПН) для формоутворення оболонок обертання. Крім того, ці системи функціонують на ЕОМ старих поколінь, через що процес підготовки ПН забирає тривалий час (до декількох годин). Важливо також відзначити, що інтерфейс існуючих систем значно застарів.

Разом з тим, відбувається постійне розширення класу намотуваних виробів, підвищуються вимоги до їхньої якості, а також до швидкості розробки нових ПН. У той же час, з'явилися й нові інструментальні засоби: швидкодіюча обчислювальна техніка, здатна реалізувати ті алгоритми, які раніше вважалися нереалізованими; апаратні засоби для реалістичного графічного моделювання методу намотування на екрані комп'ютера, використання яких дозволить скоротити до мінімуму число випробувань нової ПН на реальному обладнанні; зручні елементи інтерфейсу, значно полегшуючі роботу із прикладними програмами; велика кількість різноманітних периферійних пристроїв, що поліпшують можливості документування проектних рішень.

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 експериментів, що дозволили 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 матеріалоємним. 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 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 та 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 твердотільного проектування з вимогами та особливостями методу намотування та виробів, отриманих в його процесі.

У даній роботі пропонується вирішення першого кроку названого завдання – питання створення траєкторії геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі для подальшого аналізу. За основу взято систему інженерного аналізу ANSYS [9] як програмний комплекс не залежний від програмно-апаратної платформи,

який налічує багату бібліотеку можливостей фізико-механічного аналізу, в якого є засоби моделювання багатоплощинних КМ.

Об’єкт дослідження – властивості геодезичних ліній на поверхнях різного способу утворення.

Предмет дослідження – використання властивостей геодезичних ліній конструювання поверхонь за заданою геодезичною лінією методом кінцево-елементної моделі.

Мета роботи - аналіз моделювання геодезичної лінії методом кінцево-елементної моделі та розробити програмний засіб, що генерує траєкторію геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі в середовищі ANSYS та забезпечує можливості вибору початку траєкторії з довільного кінцевого елемента.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- 1) проаналізувати практичне застосування властивостей геодезичних ліній в прикладній геометрії поверхонь, способи їх відшукування і побудови та конструювання поверхонь за заданою геодезичною лінією.
- 2) зробити аналіз можливостей системи ANSYS на предмет створення геометрії та побудови сітки високої якості для CAD-моделей.
- 3) розробити математичну модель траєкторії геодезичної укладки та програмний засіб який буде на поверхні опрацювання довільної форми криву лінію (траєкторію намотування) із заданим кутом геодезичного відхилення, по даним кінцево – елементної моделі в середовищі ANSYS.

Новизна отриманих результатів:

- у розробці математичної моделі в якій замість систем звичайних диференціальних рівнянь, що задають геодезичні лінії на поверхні, розглядаються лінії першого порядку на площині.

Методи дослідження. Поставлені теоретичні завдання вирішені методами аналітичної й диференціальної геометрії, математичного аналізу, лінійної алгебри.

У першому розділі розглянуто композиційні матеріали та їх типи, специфіки технологічного процесу намотування, класифікацію виробів та схем намотування. Виконано огляд літератури з аналізом використання властивостей геодезичних ліній в прямій (знаходження ліній на поверхнях) і оберненій (конструювання поверхні за заданою геодезичною лінією) задачах.

У другому розділі проаналізовано можливості пакету інженерного аналізу ANSYS, внутрішньої мови APDL щодо побудови траєкторії геодезичної укладки по даним кінцево – елементної моделі.

У третьому розділі розроблена математична модель розрахунку траєкторії укладання армуючого матеріалу та програму для розрахунку . Вона формує макрос для програми ANSYS на мові APDL. Запуск цього макросу в ANSYS будує траєкторії геодезичної укладки по даним кінцево – елементної моделі намотаного виробу. Програма-генератор створена засобами програмного середовища Delphi.

РОЗДІЛ 1. ЗАСОБИ МОДЕЛЮВАННЯ ВИРОБІВ, ОТРИМАНИХ МЕТОДОМ НАМОТУВАННЯ

1.1. Композиційні матеріали та їх типи

Композиційний матеріал – це дво- або багатоскладовий матеріал, в якому чітко відслідковуються межі між його компонентами. Зазвичай композит включає в себе матрицю (основний матеріал, характеристики якого необхідно покращити) та наповнювач (основний компонент, що підвищує характеристики матриці). А також в багатьох випадках включаються різноманітні модифікатори, завдання яких є покращення властивостей наповнювача, матриці чи їх взаємного впливу один на одного. Наука про композиційні матеріали є розділом матеріалознавства і зародилась порівняно недавно – в 60-х роках.

Проте композити існують і використовуються з давніх часів. Ще за часів Древнього Єгипту людство виготовляло цеглини із глини та соломи, де чітко прослідковується поєднання двох різнорідних матеріалів і утворення третього, характеристики якого перевищують характеристики компонентів. Неважко навести приклади композитів у природі. Найтипівішим матеріалом, який залишається актуальним і сьогодні, є бамбук. У ньому поєднуються целюлозні армуючі волокна, котрі зв'язуються між собою низько-модульною пластичною органічною матрицею. Що і послужило прикладом для створення сучасних композиційних матеріалів (КМ).

Щодо більш сучасних матеріалів, але вже давно відомих, то широко використовуються такі композити як бетон, зокрема залізобетон (поєднання залізних стержнів із цементною матрицею). Також загально відомим композитом є фанера – поєднання шарів шпону, просякнутих і склеєних між собою фенольним в'язучим. Простим прикладом є медична шина: марля поєднується із гіпсом. За даними прикладами можна прослідкувати, що композит є поєднанням кількох матеріалів із різними характеристиками, яке в результаті дає властивості вищі за суму вхідних компонентів. У наш час існує безліч різнорідних композитів, кожен з яких має своє застосування.

Класифікувати їх можна за багатьма критеріями, що пов'язано з їхньою різноманітністю. Власне композит можна назвати певною конструкцією з кількох вхідних матеріалів побудованій по заданому принципу, тому і число комбінацій видів композитів є надзвичайно великим. Щодо загальноприйнятої класифікації то всі композити можна розділити на 3 основні класи [1], що групуються по типу мікроструктури: дисперсно-зміцнені, зміцнені частинками та армовані волокном. Для дисперсно-зміцнених композитів характерним є рівномірне розподілення мілких частинок розміром 0,01-0,1 мкм в кількості 1-15% по об'єму.

Для композитів зміцнених частинками характерним є рівномірне розподілення частинок розміром від 1 мкм в кількості 20-25% по об'єму. Для армованих волокнами композитів характерною є наявність різнорідних волокон, діаметр яких може варіюватись від долі мікрметра до десятків мікрметрів, а довжина від кількох мікрон до неперервної на всю довжину деталі. При цьому вміст волокон сягає 70-80% по об'єму.

В останній час з'явився новий клас композитів – нанокомпозити, котрий характеризується включенням частинок, волокон, трубок, розміри яких складають кілька нанометрів, і вміст яких складає лише 0,01-0,1% по масі.

Для досягнення максимального зміцнювального ефекту найбільш міцний компонент композита повинен відігравати роль підсилюючої структури. Для цього необхідний хороший зв'язок з матрицею, що викликає необхідність збільшення площі контакту. Логічним в такому випадку є використання волокон та волоконних матеріалів для армування. Існує широкий ряд волокон, котрі виготовляють з полімерів, мінералів, металів: Е - скло, S - скло, SiO₂, вуглець, Al₂O₃, ZrO₂, BN, W, Be, Mo, Рене 41, Cr, Cu, Fe, Ni, BeO, SiC, B₄C, графіт, поліамід, поліефір [1]. Існує чималий також ряд різнотипних матриць, котрі можна розділити на органічні, металеві, полімерні, керамічні. З котрих найбільш поширеними є полімерні, що під розподіляються на термопластичні та реактопластичні. До реактопластичних відносять суміші, що при нагріванні проходять незворотні структурні та хімічні перетворення,

тобто полімеризуються і не можуть при повторному нагріванні перейти у пластичний стан. До них можна віднести: поліефіри, феноли, епоксиди, силікони, алкіди, меламіни. Тоді як термопластичні можуть багаторазово змінювати агрегатний стан. До котрих можна віднести поліаміди, полікарбонат, акрил, ацеталі, поліпропілен, АБС, полістирол, поліетилен. Проте у напрямках машинобудування, де необхідна жаростійкість використовуються керамічні і металеві матриці, зокрема: Al, Ni, Ag, Cu, SiC, SiO₂, Al₂O₃ [2], [3]. Ширшим за великий спектр композиційних матеріалів є лише сфера їх застосування, яка охоплює майже всі поля діяльності людини так чи інакше. Найбільш поширено композити використовуються у автомобілебудуванні, авіаційній та ракетно-космічній техніці, кораблебудуванні, будівництві, залізничному транспорті, оборонній промисловості, нафтовидобувній промисловості та сільському господарстві [4]. В автомобільній промисловості з композитів виготовляють баки для пального, кузови, цистерни, деталі внутрішнього інтер'єру. Широко використовуються композити при будівництві яхт, легких та міцних корпусів катерів, рятувальні човни для танкерів. Майже кожен будинок виготовлений із такого композиту як бетон, та і полімерні композитні панелі використовуються при будівництві. Завдяки антикорозійним властивостям із композитів виготовляють корпуси для сільськогосподарської техніки та контейнери для добрив. Також в залізничному транспорті використовуються композитні кузови чи частини кузовів, елементи інтер'єру. В оборонній промисловості із композитів виготовляються каски, бронежилети, бронепанелі. В видобувній промисловості із композитів виготовляють елементи бурових платформ, нафто- та газопроводи. Однак найбільш необхідні композити в авіаційній галузі, де критично важливим є зниження ваги при збереженні несучої здатності конструкції. Це дозволяє знизити затрати пального, підвищити масу корисного навантаження, тобто підвищити економічну ефективність [2]. Проте композити потребують більш творчого підходу для реалізації свої переваг, адже вони є анізотропним матеріалами і

реалізують ті характеристики, які закладені при проектуванні складу і типу композиту.

Полімерні композити є широкою групою матеріалів, з яких не всі знайшли своє застосування в авіації. Першими використовувались склопластики, з них композити почали свій хід у літакобудуванні, якщо не рахувати шпон і фанеру. Застосування скловолокна дозволило значно покращити конструкції планерів, а згодом воно знайшло своє використання і багатьох інших сферах. Скловолокно є дешевим і міцним матеріалом, проте має такі недоліки як порівняно висока густина та низький модуль пружності. Тому на заміну склопластиками прийшли вуглепластики. Вуглеволокно є дорогим, але має низьку густину, значно вищу міцність та жорсткість ніж у склопластику та практично нульовий коефіцієнт лінійного розширення. Тому деталі із вуглеволокон знайшли своє місце в авіації і широко використовуються і зараз, адже мають високу питому міцність. На ряду з вуглеволокном використовуються і полімерні волокна, такі як поліпарафенілентерефталамід, загально відомі як Кевлар. Такі волокна мають високу ударну та статичну міцність, ще вищий ніж у вуглеволокон модуль пружності і низьку густину. Проте вони також характеризуються низькою міцністю на стиск і надвисокою вартістю. Незважаючи на різнотипні характеристики усі 3 типи волокон знаходять своє застосування в різній мірі залежно від поставлених завдань [2]. Із вказаними типами волокон застосовуються і різні типи матриць. Від матриці залежить наскільки реалізується міцність та жорсткість волокон. Як зазначалося, вони діляться на термопластичні та реактопластичні. Найпоширенішою матрицею є епоксидні смоли, котрі мають високу адгезію до більшості наповнювачів та мале осадження, наявні в великій кількості модифікацій для різних випадків [3]. Наряду із епоксидною смолою також використовуються і поліефірсульфон, поліефіркетон та поліефірімід. Такі термопластичні в'язучі характеризуються більшою стійкістю до зовнішнього середовища, вищою міцністю, проте вони потребують і вищих температур термообробки [5]. Однак існують і інші

в'яжучі, котрі мають своє місце в авіаційних композитах [2], [3], [5]. Окремого згадування заслуговує тип композиту, який широко використовується в авіації та включає в себе попередньо згадані – сендвічний композит [6]. Зазвичай використовуються композитні панелі із склопластику чи вуглепластику між якими знаходиться пінопластова чи стільникова панель (металева або полімерна), товщина якої значно більша за товщину композитних панелей. Такий підхід дозволяє значно збільшити товщину композиту без закладання надлишкового запасу міцності і значного зростання маси панелі. Таким чином композити є надзвичайно різноманітними широко використовуються в багатьох сферах промисловості.

1.2. Геодезична лінія – найкоротший шлях між двома точками на поверхні

Властивостям геодезичних ліній поверхні присвячено багато робіт як наукового характеру [10, 13, 41], навчально-методичного [29], так і популярного [24, 19]. В праці [19] наведено приклад, в якому поставлено питання: якщо на протилежних гранях куба знаходиться павук і муха, то як павук має прокласти шлях, щоб він був найкоротшим до мухи? (рис. 1,а). Відомо, що найкоротшим шляхом між двома точками площини є пряма лінія. Якщо побудувати розгортку куба, то найкоротший шлях від павука до мухи буде відрізком прямої між місцем їхнього розташування на розгортці (рис. 1,б).

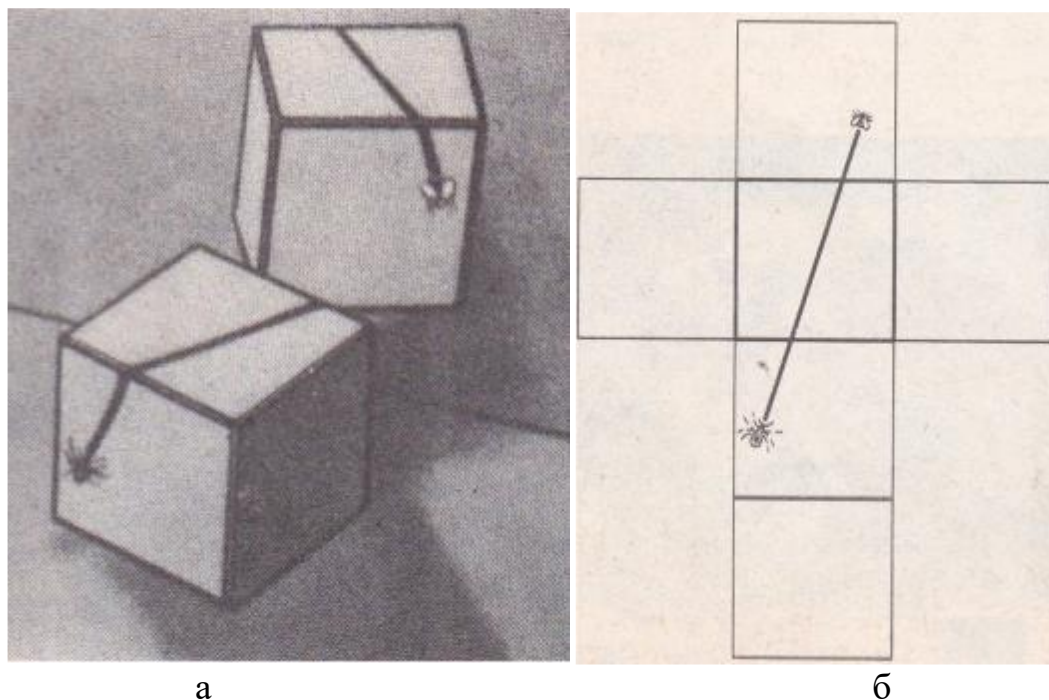
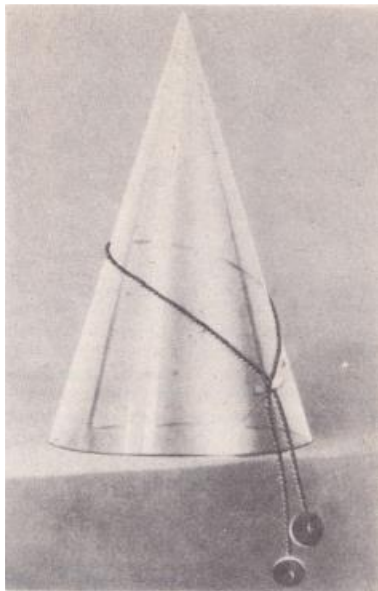
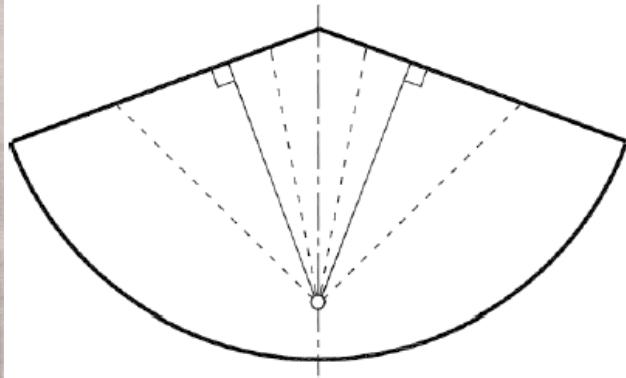


Рис. 1.1. Найкоротший шлях між двома точками грані поверхні:
а) на кубі; б) на його розгортці (пряма лінія)

Цим пояснюється те, що геодезична лінія розгортної поверхні перетворюється на пряму на її розгортці. Наприклад, якщо на конусі задана точка і стоїть задача знайти найкоротший шлях, щоб після обходу конуса знову повернутися в цю точку (рис. 1.2,а), то на розгортці бічної поверхні конуса, розрізаного вздовж протилежної твірної, цей шлях зобразиться двома прямолінійними відрізками, перпендикулярними до цієї твірної (рис. 1.2,б, суцільна лінія). Із точки на площині в усіх напрямках можна провести прямі лінії, які утворюють пучок геодезичних ліній. Так само на поверхні теж в усіх напрямках можна провести геодезичні лінії, проте вони мають деякі особливості в порівнянні із прямими на площині. Якщо із точки на конусі задати геодезичну лінію під іншим кутом, то вона теж повернеться в цю точку, проте це не буде найкоротший шлях обходу. На розгортці конуса ці геодезичні лінії зображені штриховими лініями (рис. 1.2,б).



а



б

Рис. 1.2. Найкоротший шлях обходу конуса по його бічній поверхні:

а) на конусі;

б) на його розгортці (прямі суцільні лінії, що йдуть перпендикулярно граничним радіусам сектора – спільній лінії на конусі).

Те ж саме стосується геодезичних ліній на круговому циліндрі. Як відомо, ними є гвинтові лінії. Найкоротшою відстанню між двома точками циліндра, які не розташовані на спільній прямолінійній твірній, є дуга гвинтової лінії, яка на розгортці циліндра перетворюється в пряму лінію. Однак між цими точками можна провести безліч гвинтових ліній, які є геодезичними, однак найкоротшою відстанню між цими точками буде тільки одна із них, зображена суцільною (рис. 1.3). Всі інші роблять більшу кількість витків.

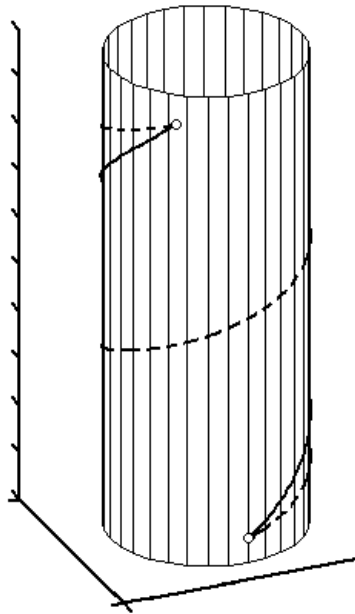
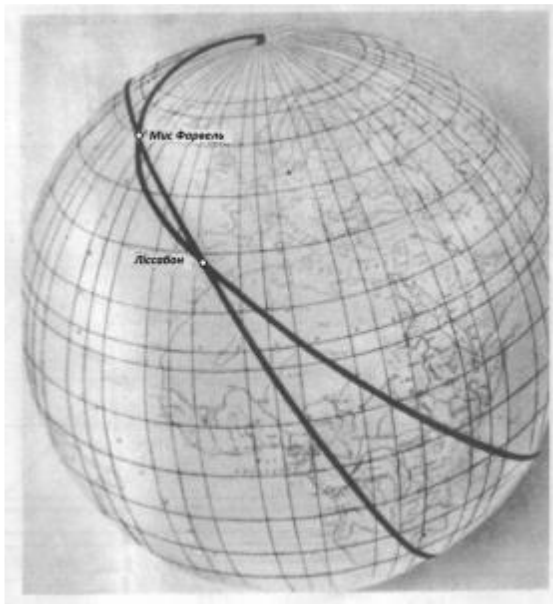


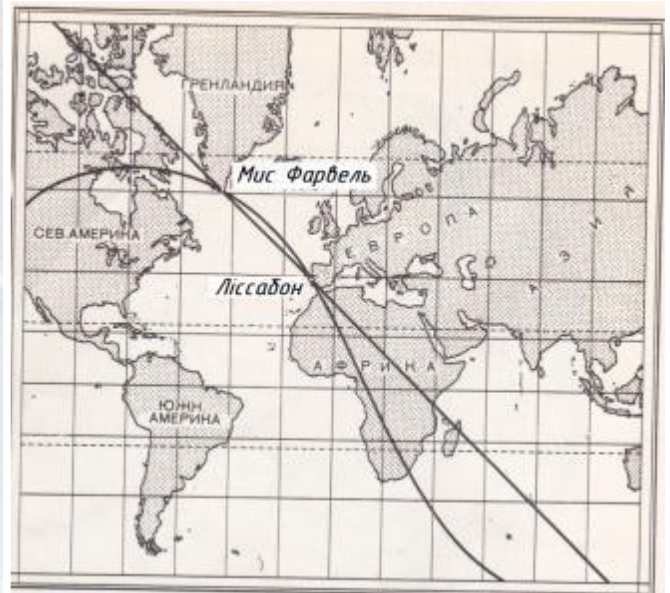
Рис. 1.3. Геодезичні лінії, що сполучають дві точки на бічній поверхні
циліндра

Якщо на аркуші паперу провести пряму лінію, а потім його згинати у розгортну поверхню, то пряма перетвориться у геодезичну лінію на поверхні.

На нерозгортних поверхнях теж є геодезичні лінії, однак для їх знаходження застосування розгорток неможливе. Проте є інший практичний спосіб знаходження геодезичної лінії між двома точками на поверхні: між ними потрібно натягнути розтяжну нитку. Сили розтягу нитки мінімізують її довжину і вона займає місце геодезичної лінії. Наприклад, на конусі (рис. 1.2,а) нитку натягують два вантажі, прикріплені до її кінців. Якщо за допомогою розтяжної нитки знайти геодезичну лінію між двома точками на поверхні кулі, то нею виявиться дуга великого кола. Отже, всі геодезичні лінії кулі замкнені. До них відносяться всі меридіани і одна паралель – екватор, а також всі інші кола – перерізи кулі площинами, що проходять через її центр. Для кулі ці криві прийнято називати ортодромами. При кораблеводінні вони є найкоротшим шляхом між двома портами. Однак в практиці мореплавання курс прокладали по іншій кривій – локсодромі, яка перетинає всі меридіани під постійним кутом. Наприклад, найкоротший шлях між портами Ліссабон і Мис Фарвель визначається ортодроною, однак кораблі водили по локсодромі, яка подовжує шлях (рис. 1.4,а).



а



б

Рис. 1.4. Шлях між двома портами, прокладений по ортодромі і локсодромі:

а) на земній поверхні у вигляді кулі;

б) на карті Меркатора (локсодрома зображується прямою лінією).

Це пояснюється тим, що в мореплаванні використовували карту на основі проєкції Меркатора, на якій меридіани і паралелі утворюють прямокутну сітку (рис. 1.4,б). Локсодрома зображується на ній прямою лінією, яка перетинає всі меридіани під однаковим кутом, а ортодрома зображується кривою. Капітан судна, який вибрав курс по компасу і зберігає його незмінним, рухається по локсодромі. Це спрощує керування кораблем, однак продовжує шлях. Провівши пряму лінію на карті Меркатора між потрібними портами, можна визначити курс корабля, тобто кут, під яким корабель перетинатиме меридіани, рухаючись по компасу. Як видно із рис. 1.5, при наближенні до полюса локсодрома закручується навколо нього подібно спіралі і стає зовсім непридатною для прокладання курсу в полярних широтах.

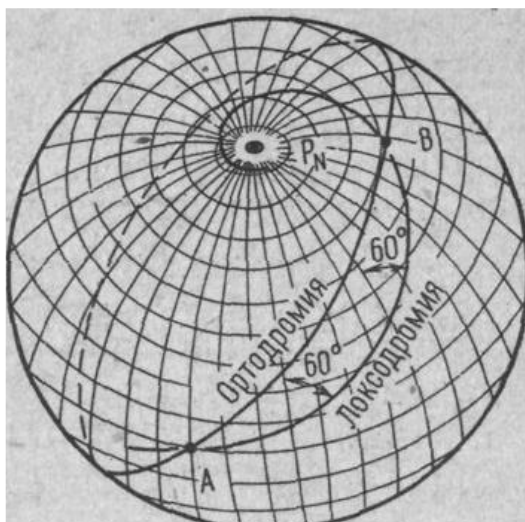


Рис. 1.5. Сполучення між точками A і B на кулі по ортодромі і локсодромі

Якщо із точки на площині в усі сторони відкласти відрізок однакової довжини, то їх кінці будуть лежати на колі. Аналогічну побудову можна здійснити на поверхні, де замість прямих будуть геодезичні лінії. Утворена кінцями дуг геодезичних ліній крива носить назву гаусового геодезичного круга. Довжина геодезичної лінії на поверхні, як і всякої іншої лінії, визначається за допомогою першої квадратичної форми.

Як відомо, поверхні можуть бути задані аналітично в неявному вигляді $F(X, Y, Z)=0$, явному вигляді $Z=F(X, Y)$, а також параметричними рівняннями, якими ми будемо в роботі оперувати:

$$\begin{aligned} X &= X(u, v); \\ Y &= Y(u, v); \\ Z &= Z(u, v), \end{aligned} \quad (1.1)$$

де u, v – незалежні змінні.

При наданні конкретних значень параметрам u і v на поверхні виділиться дві відповідні координатні лінії, які перетнуться в точці A (рис. 1.7,а). Таким чином, положення кожної точки на поверхні можна визначити двома числами – значеннями криволінійних координат u і v тих координатних ліній різних сімей, які проходять через дану точку. При їх підстановці в рівняння (1.1) можна знайти три координати точки A в декартовій системі координат $OXYZ$.

Сітка на поверхні в загальному випадку буде косокутною. Деякі поверхні можна описати параметричними рівняннями виду (1.1) таким чином, що вони будуть віднесені до прямокутної (ортогональної) сітки координатних ліній. Такий опис поверхні є більш прийнятним, оскільки спрощує аналітичні викладки і знаходження диференціальних характеристик поверхні, але не завжди можливий. Прикладом є поверхні обертання, віднесені до сімей координатних ліній із паралелей і меридіанів.

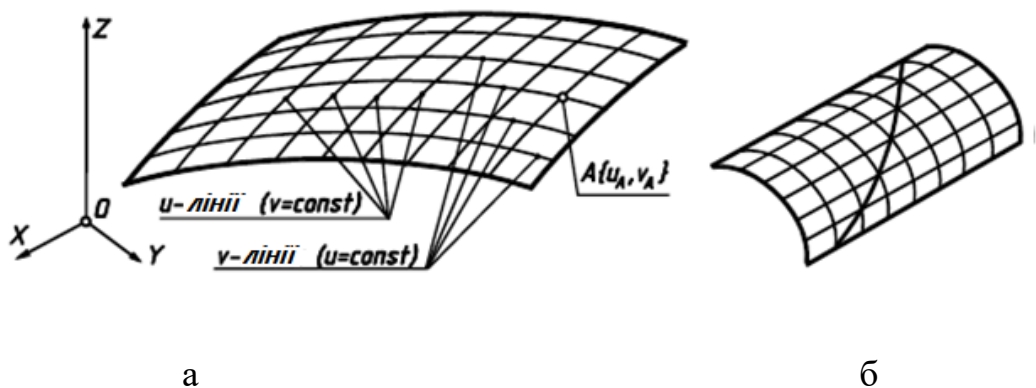


Рис. 1.6. Віднесення поверхонь до сімей координатних ліній:

- а) косокутна координатна сітка на поверхні загального виду;
- б) прямокутна координатна сітка на циліндричній поверхні.

На рис. 1.6, б показана циліндрична поверхня, віднесена до прямокутної системи координат. Однією сім'єю координатних ліній є паралельні прямі, другою – плоскі криві поперечного перерізу.

Якщо між незалежними змінними u і v поверхні встановити певний взаємозв'язок у вигляді $u=u(v)$ або $v=v(u)$ або ж через третю змінну $v=v(t)$; $u=u(t)$, то рівняння (1.1) стають рівняннями однієї змінної (v , u або ж t), тобто вони описують лінію на поверхні. Такий взаємозв'язок між змінними поверхні називається внутрішнім рівнянням лінії на ній. Наприклад, якщо внутрішнє рівняння лінії для циліндричної поверхні (рис. 1.6,б) задати у вигляді $u=v$, то відповідна лінія буде проходити через вершини криволінійних прямокутників по діагоналі аналогічно прямій, яка задана такою ж залежністю в плоскій декартовій системі координат і буде геодезичною для циліндра. Знаходження

внутрішніх рівнянь геодезичної лінії на довільній поверхні є непростю задачею і зводиться до розв’язування диференціальних рівнянь.

Перша квадратична форма поверхні (або лінійний елемент) має вигляд:

$$dS^2 = Edu^2 + 2Fdu dv + Gdv^2. \quad (1.2)$$

Три коефіцієнти E, F, G , які входять до виразу (1.2), визначаються через частинні похідні поверхні (1.1):

$$E = \left(\frac{\partial X}{\partial u} \right)^2 + \left(\frac{\partial Y}{\partial u} \right)^2 + \left(\frac{\partial Z}{\partial u} \right)^2; \quad F = \frac{\partial X}{\partial u} \cdot \frac{\partial X}{\partial v} + \frac{\partial Y}{\partial u} \cdot \frac{\partial Y}{\partial v} + \frac{\partial Z}{\partial u} \cdot \frac{\partial Z}{\partial v};$$

$$G = \left(\frac{\partial X}{\partial v} \right)^2 + \left(\frac{\partial Y}{\partial v} \right)^2 + \left(\frac{\partial Z}{\partial v} \right)^2. \quad (1.3)$$

Якщо сітка координатних ліній ортогональна, то середній член (коефіцієнт F дорівнює нулю).

Розглянемо приклад. Запишемо рівняння циліндра, всі прямолінійні твірні якого паралельні осі OZ . Його параметричні рівняння запишуться:

$$\begin{aligned} X &= R \cos v; \\ Y &= R \sin v; \\ Z &= u, \end{aligned} \quad (1.4)$$

де R – радіус основи циліндра;

u, v – незалежні змінні поверхні, причому вони мають фізичний зміст: u – довжина прямолінійної твірної циліндра; v – кут повороту точки поверхні навколо осі OZ .

Частинні похідні запишуться:

$$\begin{aligned} \frac{\partial X}{\partial u} &= 0; & \frac{\partial Y}{\partial u} &= 0; & \frac{\partial Z}{\partial u} &= 1; \\ \frac{\partial X}{\partial v} &= -R \sin v; & \frac{\partial Y}{\partial v} &= R \cos v; & \frac{\partial Z}{\partial v} &= 0. \end{aligned} \quad (1.5)$$

Згідно (1.3) коефіцієнти E, F, G першої квадратичної форми запишуться:

$$E = 1; \quad F = 0; \quad G = R^2. \quad (1.6)$$

Отже, перша квадратична форма для циліндра (1.4) набуває вигляду:

$$dS^2 = du^2 + R^2 dv^2. \quad (1.7)$$

Внутрішнє рівняння геодезичної лінії на поверхні циліндра відоме – воно має лінійний вигляд: $u=bv$, де від сталої b залежить кут підйому гвинтової лінії. При $b=1$ будемо мати частковий випадок, зображений на рис. 1.6,б. Отже ми від двох незалежних змінних перейшли до однієї – v . Розділимо квадратичну форму (1.7) на dv^2 :

$$\left(\frac{dS}{dv}\right)^2 = \left(\frac{du}{dv}\right)^2 + R^2. \quad (1.8)$$

Знаходимо похідну $du / dv = b$ і підставляємо в (1.8):

$$\frac{dS}{dv} = \sqrt{b^2 + R^2}. \quad (1.9)$$

Інтегруванням виразу (1.9) знаходимо довжину гвинтової лінії:

$$S = \int \sqrt{b^2 + R^2} dv = \sqrt{b^2 + R^2} v. \quad (1.10)$$

1.3. Геодезична лінія – траєкторія намотки армуючих на поверхню ниток

Виходячи із ручного розшукування геодезичної лінії між двома точками на поверхні за допомогою натягу між ними розтяжної нитки, можна зробити висновок, що це її найбільш стійке положення на поверхні. При виникненні напружень в оболонці, армованій нитками вздовж геодезичних ліній, нитки працюватимуть на розтяг і при цьому не виникатиме зусиль, що зміщують їх із свого початкового положення. Пояснюється це внутрішньою геометрією геодезичних ліній. Розглянемо паралель (коло) на поверхні циліндра і на поверхні конуса (рис. 1.7).

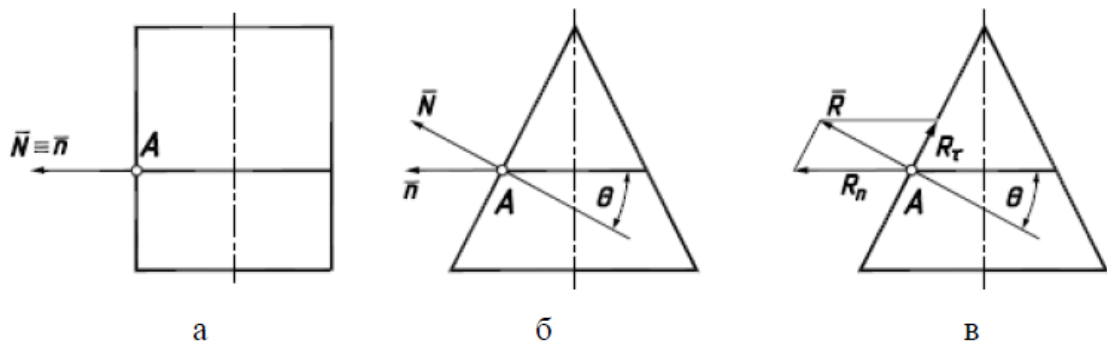


Рис. 1.7. До визначення зусиль, що діють на нитку, намотану на вздовж паралелі на поверхні обертання:

- а) нормаль поверхні і головна нормаль кривої на ній збігаються;
- б) нормаль поверхні і головна нормаль кривої утворюють між собою кут θ ;
- в) розкладання сили реакції поверхні на складові R

В обох випадках в точці A кола показано головну його нормаль n , яка знаходиться в площині кола, і нормаль до поверхні N . В першому випадку (рис. 1.7,а) вони збігаються, а в другому між ними існує кут θ (рис. 1.7,б). Як відомо, реакція R поверхні при взаємодії із тілом (в нашому випадку із елементом нитки) завжди спрямована по нормалі до поверхні. Її можна розкласти на складові (рис. 1.7,в), одна із яких R_n знаходиться в дотичній до поверхні площині. Саме ця складова спричинює зсув нитки із свого початкового положення при виникненні зусиль розтягу в нитці. Цього не відбувається, коли нормаль до поверхні і головна нормаль кривої збігаються (рис. 1.7,а). Геодезична лінія характеризується тим, що такий збіг відбувається в кожній точці кривої вздовж її дуги.

В диференціальній геометрії через кут θ розкладають кривину кривої в заданій точці на нормальну і геодезичну складові. Кривину k кривої в точці A зображують вектором, який розташований в стичній площині. Якщо крива плоска, то вектор лежить в площині кривої на її головній нормалі, як, наприклад, на рис. 1.8, де він розташований в площині еліпса – плоского перерізу циліндра. Кривину k можна розкласти на дві складові в нормальній площині кривої: одна складова в проекції на нормаль до поверхні носить назву нормальної кривини k_n , а друга складова в проекції на дотичну до поверхні

площину – геодезичної кривини k_g . Виходячи із позначень на рис. 1.8, можна записати:

$$k_g = k \sin \theta; \quad k_n = k \cos \theta. \quad (1.11)$$

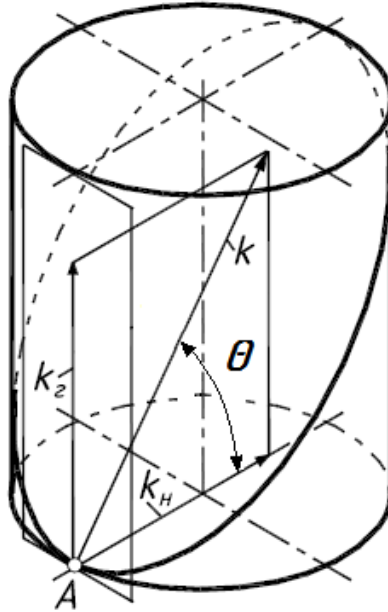


Рис. 1.8. Розкладання кривини кривої на нормальну і геодезичну складові

Якщо головна нормаль кривої збігається із нормаллю до поверхні, то кут $\theta=0$, отже геодезична кривина теж дорівнює нулю. Таким чином, геодезична лінія – це лінія на поверхні, в якій геодезична кривина всіх її точок дорівнює нулю. У прямої на площині кривина дорівнює нулю, однак при згинанні площини вона перетворюється у криву, у якій є кривина. Однак ця кривина з'являється за рахунок нормальної складової; геодезична кривина дорівнює нулю і не змінюється під час згинання поверхні, тобто при згинанні поверхні крива трансформується, однак залишається геодезичною.

Геодезичну кривину кривої можна визначити через її диференціальні характеристики і характеристики поверхні за формулою [14]:

$$k_g = \left(\frac{dv}{ds} \right)^3 \begin{vmatrix} N_x & N_y & N_z \\ x' & y' & z' \\ x'' & y'' & z'' \end{vmatrix}, \quad (1.12)$$

де dv/ds - похідна параметра v по довжині дуги геодезичної лінії s у випадку, коли внутрішнє рівняння геодезичної лінії задано у вигляді $u=u(v)$; N_x, N_y, N_z – координати одиничного вектора нормалі до поверхні;

$x', y', z', x'', y'', z''$ - перші і другі похідні рівнянь кривої по параметру v .

Проекції нормалі до поверхні знаходяться із векторного добутку частинних похідних рівнянь поверхні:

$$\vec{N} = \frac{\begin{vmatrix} X & Y & Z \\ \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & \frac{\partial Z}{\partial u} \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & \frac{\partial Z}{\partial v} \end{vmatrix}}{\begin{vmatrix} \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & \frac{\partial Z}{\partial u} \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & \frac{\partial Z}{\partial v} \end{vmatrix}}. \quad (1.13)$$

Розглянемо приклад. Нехай нам потрібно знайти геодезичну кривину еліпса – плоского перерізу циліндра площиною, нахиленою до його основи під кутом θ (рис. 1.8). Для цього нам потрібно знайти внутрішнє рівняння еліпса, тобто зв'язок між змінними u і v . Рівняння проекціовальної січної площини запишеться: $Z=X \operatorname{tg} \theta$. Оскільки точки січної площини і поверхні циліндра (1.4) мають бути спільними, то ми повинні рівняння площини і циліндра розв'язати як систему. Прирівняємо рівняння площини і останнє рівняння (1.4) і отримаємо: $u=X \operatorname{tg} \theta$. В одержаний вираз підставимо перше рівняння (1.4) і отримаємо внутрішнє рівняння еліпса у вигляді $u=u(v)$:

$u=R \operatorname{tg} \theta \cos v$. Отже, рівняння еліпса на поверхні циліндра після цього запишуться:

$$\begin{aligned} x &= R \cos v; \\ y &= R \sin v; \\ z &= R \operatorname{tg} \theta \cos v. \end{aligned} \quad (1.14)$$

Ми перейшли від великих літер X, Y, Z у рівняннях (1.4) до малих x, y, z у рівняннях (1.14). Так будемо робити і надалі: в рівняннях поверхні використовуємо прописні літери, а в рівняннях лінії – строчні. Знайдемо перші і другі похідні рівнянь (1.14):

$$\begin{aligned}
x' &= -R \sin v; & x'' &= -R \cos v; \\
y' &= R \cos v; & y'' &= -R \sin v; \\
z' &= -R \operatorname{tg} \theta \sin v; & z'' &= -R \operatorname{tg} \theta \cos v.
\end{aligned}
\tag{1.15}$$

Знайдемо похідну довжини дуги лінії (1.14) за відомою формулою:

$$\frac{ds}{dv} = \sqrt{x'^2 + y'^2 + z'^2} = R \sqrt{1 + \operatorname{tg}^2 \theta \sin^2 v}.
\tag{1.16}$$

Із (1.16) знаходимо вираз dv/ds , який входить до формули (1.12):

$$\frac{dv}{ds} = \frac{1}{R \sqrt{1 + \operatorname{tg}^2 \theta \sin^2 v}}.
\tag{1.17}$$

Нарешті, знайдемо проєкції одиничного вектора нормалі N_x , N_y , N_z , які входять до формули (1.12) із визначника (1.13), використовуючи частинні похідні (1.5):

$$N_x = -\cos v; \quad N_y = -\sin v; \quad N_z = 0.
\tag{1.18}$$

Підставивши (1.18), (1.17) і (1.15) у формулу (1.12), після спрощень остаточно одержимо:

$$k_z = \frac{\operatorname{tg} \theta \cos v}{R(1 + \operatorname{tg}^2 \theta \sin^2 v)^{3/2}}.
\tag{1.19}$$

Отже, геодезична кривина еліпса на поверхні циліндра є змінною. При $\theta=0$, тобто при ортогональному перерізі циліндра ми отримаємо $k_z=0$, що і слід було чекати, оскільки ортогональним перерізом циліндра є коло – геодезична лінія циліндра (рис. 1.7,а).

Зміцненню оболонок намотуванням нитки на її поверхні по геодезичних лініях присвячено багато робіт [1, 6, 7, 9, 21, 25, 26, 28, 30]. Це пояснюється тим, що таким способом можна отримати балони для зберігання стисненого газу, які не поступаються за міцністю металевим, однак набагато легші і безпечніші (при вибуху немає осколків). При цьому вдосконалюються технології намотки. Використовуються не тільки машини для серійного

виробництва армованих оболонок (рис. 1.9,а), а і пристрої для намотки ниток на поверхні обертання із меридіаном заданої форми (рис. 1.9,б). В останньому випадку траєкторія намотки по геодезичних лініях може бути задана в залежності від форми поверхні за допомогою комп'ютерного забезпечення переміщення напрямної рейки в поздовжньому напрямі і узгодженого переміщення каретки 2 по рейці в поперечному (рис. 1.9,б) [5].



а



б

Рис. 1.9. Пристрої для намотування ниток на поверхні обертання

1.4. Геодезична лінія – ймовірна траєкторія примусового руху частинки технологічного матеріалу по поверхні

Якщо частинка на певній швидкості зустрічається із поверхнею, то далі вона рухається по геодезичній лінії і тільки сила ваги частинки згодом змушує відхилитися її від цього курсу, причому чим більша швидкість частинки, тим більш точно її траєкторія руху наближається до геодезичної лінії. Такий рух називається інерційним [17]. Це можна продемонструвати на прикладі руху частинки по похилій площині μ (рис. 1.10,а). Чим більша початкова швидкість V частинки при вступі її на площину в заданому напрямі, тим більше траєкторія її руху наближається до прямої лінії, яка є геодезичною для площини. Складова сили ваги змушує частинку відхилитися від прямолінійного напрямку руху, однак при криволінійній траєкторії виникає відцентрова сила, яка змушує частинку наближатися до прямолінійного напрямку. Зважаючи на те, для частинки масою m ця сила F_v має вираз $F_v = mV^2k$, де k – кривина траєкторії, стає зрозуміло, що вирішальну роль в

формуванні траєкторії відіграє швидкість V руху частинки. Чим більша швидкість частинки, тим більше траєкторія її руху наближається до прямої лінії. Для поверхні відбувається те ж саме, з тією різницею, що відцентрова сила, яка змушує траєкторію частинки наближатися до геодезичної лінії, виникає від геодезичної складової кривини (рис. 1.8). Вона діє в дотичній до поверхні площині (рис. 1.10,б), тобто рух точки по поверхні в околі точки A можна розглядати, як рух по дотичній до поверхні площині μ . Наприклад, для циліндричної поверхні, яка показана на рис. 1.10,б, граничною траєкторією руху частинки буде гвинтова лінія, тобто геодезична для циліндра. У ґрунтообробних робочих органах частинки ґрунту змушені примусово рухатися по поверхні з приблизно сталою швидкістю, рівною швидкості руху агрегату. При великих швидкостях їх траєкторії практично можуть збігатися із геодезичними лініями. Тому для великих робочих швидкостей ґрунтообробних знарядь складання диференціальних рівнянь руху частинок можна замінити відшукуванням геодезичних ліній в заданому напрямі вступу частинки на поверхню [17].

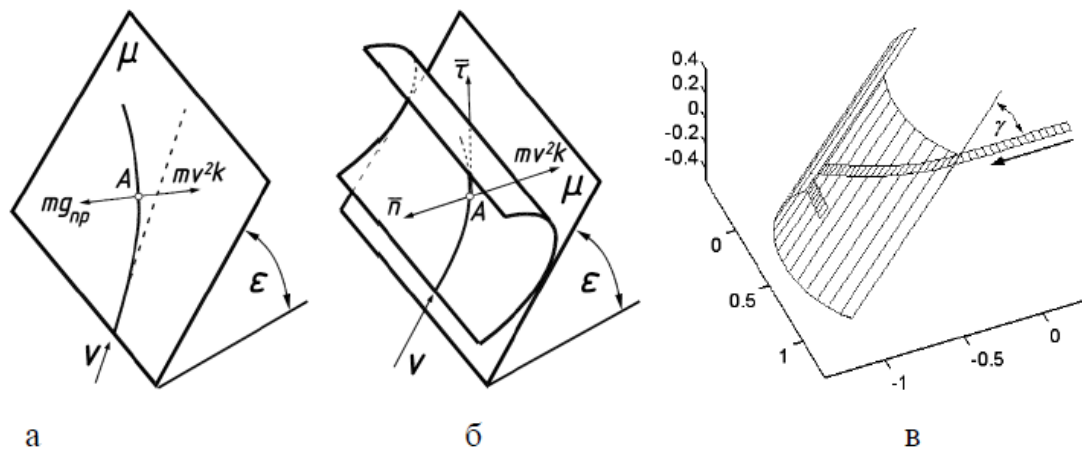


Рис. 1.10. Ілюстрації до руху частинки по геодезичних лініях на поверхні при їх великих швидкостях:

- а) рух частинки по похилій площині; б) по циліндрі;
- в) визначення геодезичної лінії в заданому напрямі паперовою стрічкою

Ідея застосування кінематичних властивостей геодезичної лінії продемонстрована на прикладі руху гнучкої (наприклад, паперової) стрічки по поверхні циліндра в заданому напрямі (рис. 1.10,в). Стрічка, якій властива певна пружність, змушена рухатися по геодезичній лінії незалежно від її швидкості. Ця властивість може бути використана для практичного знаходження геодезичної лінії на поверхні в заданому напрямі. Якщо скиба ґрунту має певну пружність (наприклад, переплетена корінням рослин) і вступає на полицю плуга, то вона подібно стрічці, намагатиметься рухатися по геодезичній лінії. Цьому сприятимуть два фактори: пружність скиби і швидкість її руху.

Ця властивість може бути покладена в основу проектування поверхонь робочих органів за бажаною траєкторією руху частинок технологічного матеріалу по них, тобто в розв'язуванні оберненої задачі – конструюванні поверхні за заданою геодезичною лінією. Ця ідея запропонована в праці [27]. Із диференціальної геометрії відомо, що через задану лінію можна провести одну розгортну і безліч нерозгортних поверхонь, за умови, щоб вона була геодезичною для цих поверхонь. Отже, задача побудови поверхні за заданою геодезичною лінією має багато розв'язків. Це можна показати на прикладі плоскої кривої – кола. Через нього проходить тільки одна розгортна поверхня (в нашому випадку циліндр, рис. 1.11,а) і безліч нерозгортних, як лінійчатих (рис. 1.11,в), так і нелінійчатих (рис. 1.11,г,д). На розгортці розгортних поверхонь геодезична лінія перетворюється на пряму (рис. 1.11,б).

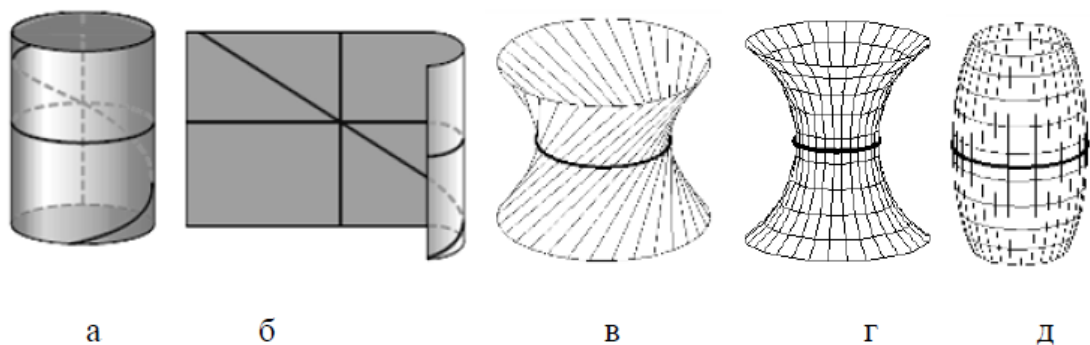


Рис. 1.11. Приклади поверхонь, для яких коло є геодезичною лінією

Оскільки стрічка, якою може бути скиба ґрунту, намагається рухатися по геодезичній лінії, то бажану форму траєкторії руху скиби, тобто геодезичної лінії, можна задати наперед. За заданою геодезичною лінією можна побудувати одну розгортну поверхню і інші нерозгортні, які в більшій або меншій мірі відрізнятимуться від розгортної.

1.5. Висновки

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 непереривною, що забезпечує властивості матеріалів, які переважають властивості класичних метала, 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.

Задача знаходження і побудови геодезичних ліній на поверхнях не є простою. Для її розв'язання потрібно складати і розв'язувати диференціальні рівняння другого порядку, які тільки в окремих випадках (наприклад, для поверхонь обертання) можуть бути зведені до інтегралу. Проте і в цьому випадку геодезична лінія не може бути побудована в повному обсязі із-за специфіки чисельного інтегрування, а можуть бути побудовані тільки окремі її фрагменти.

РОЗДІЛ 2. ЗАСОБИ МОДЕЛЮВАННЯ ВИРОБІВ, ОТРИМАНИХ МЕТОДОМ НАМОТУВАННЯ

2.1. Огляд засобів моделювання

Розвиток методу скінченних елементів обумовлений взаємозв'язком трьох факторів: наявністю високопродуктивної обчислювальної техніки; розробленням математичних моделей досліджуваних явищ, які адекватні реальним процесам з достатнім ступенем точності; особливостями самого методу [6].

Перші програмні комплекси, в яких реалізований метод скінченних елементів, були розроблені в 60-х роках. До них належать STRUDL-II, SAP-IV, NONSAP, ASKA, NASTRAN, SESAM-69 та інші. Появі цих універсальних програмних систем через особливості методу скінченних елементів передувало створення високопродуктивних електронно-обчислювальних машин, таких, наприклад, як IBM-370. Починаючи з кінця 70-х років XX ст., в СРСР з'явилося декілька десятків програмних комплексів для різних ЕОМ, в яких був реалізований МСЕ. До них відносять МІРАЖ, МОРЕ, КАСКАД-2, МІЦНІСТЬ-75, МСЕ/20, МАРС, ПАРСЕК, ЛІРУ, СПРИНТ, FEA і ряд інших програм.

У США і ряді інших країн подальший розвиток МСЕ і необхідність у проведенні розрахунків конструкцій на міцність також сприяли подальшому розвитку вже створених програмних комплексів і розробленню нових. Були розроблені сотні програмних комплексів, які призначені для наближеного вирішення найрізноманітніших задач не тільки з області механіки деформованного твердого тіла, але й із таких областей, як гідродинаміка, акустика, електротехніка і т.д. Найбільшого поширення з них набули: ABAQUS, ADINA, ANSYS, MARC, MSC/NASTRAN, COSMOS, LS-DYNA та інші.

Відзначимо, що розроблення програмних комплексів є дорогою справою. Тому, як правило, організації і фірми - власники розроблених програм розглядають їх як комерційний науково-технічний продукт.

Регулярно друкуються огляди існуючих програмних комплексів і їх характеристик, відомості про програми у галузевих фондах алгоритмів і програм дозволяють користувачам програмної продукції цілеспрямовано вибирати необхідні для їх діяльності програми розрахунку.

У кожної програми є свої сильні і слабкі сторони при розрахунку конкретної конструкції. Вибір програми розрахунку залежить від підготовленості користувача в своїй науковій області, типу вирішуваної задачі, типу доступної ЕОМ, розмірності задачі і інших факторів.

До критеріїв, що допомагають зробити вибір, необхідно віднести такі фактори:

- 1) програма широко використовується;
- 2) у програмі використовуються новітні наукові досягнення;
- 3) програма комерційно цілком доступна;
- 4) є докладна і зрозуміла документація.

Ознайомлення з програмною документацією і доступною літературою з описом програми і її елементів дозволяє зробити остаточний висновок про доцільність вибору програмного комплексу.

Для МСЕ характерні особливості, які необхідно враховувати при виборі і розробленні програми розрахунку. Такими особливостями є великі об'єми початкових даних, проміжних і остаточних результатів розрахунку. Тому розрахунок за МСЕ складається з трьох основних етапів:

- 1) розроблення розрахункової скінченно-елементної схеми і підготовки початкових даних;
- 2) перевірки самого розрахунку;
- 3) обробки результатів розрахунку.

Кожен етап є самостійною задачею. Перший етап полягає у створенні початкової скінченно-елементної розрахункової моделі, виходячи з інженерної інтуїції про поведінку конструкції. Надалі ця модель може коригуватися на основі аналізу результатів розрахунку. Коригування моделі може виконуватися і програмним шляхом, якщо така можливість реалізована

у використовуваному програмному комплексі. Підготовка початкових даних здійснюється, як правило, за допомогою програм генераторів сіток скінченних елементів, які створюють блок підготовки даних.

Проведення розрахунку (етап 2) здійснюється розрахунковим блоком, у якому використовується той або інший алгоритм розрахунку методом скінченних елементів. Як правило, розрахунковий блок складається з ряду програмних модулів, кожний з яких виконується на певному кроці алгоритму. У простому випадку програмної реалізації МСЕ для лінійної статичної крайової задачі теорії пружності розрахунковий блок містить таку послідовність кроків:

- 1) введення початкових даних (наприклад, підготовлених програмою-генератором в окремому файлі);
- 2) обчислення матриць жорсткостей скінченних елементів;
- 3) формування глобальної матриці жорсткості повної структури;
- 4) формування глобального вектора навантажень;
- 5) розв'язання системи лінійних алгебраїчних рівнянь;
- 6) обчислення переміщень вузлів сітки скінченних елементів, деформацій і напружень у довільних точках скінченних елементів.

На різних кроках розрахункового блоку включаються перевірки правильності початкових даних і результатів проміжних обчислень (діагностика помилок), програмні модулі вибору поєднань навантажень, які діють на конструкцію, визначення площі перетинів та інше. Діагностика помилок на етапі виконання програми є важливою, оскільки при своєчасному виявленні помилки припиняються обчислення, що приводить до економного використання ресурсів ЕОМ.

Ефективне використання ЕОМ досягається також за рахунок розроблення спеціальних методів розв'язання стандартних математичних задач, що враховують специфіку МСЕ і в першу чергу стрічковість і розрідженість матриці жорсткості розрахункової моделі конструкції.

Розрахунок напружено-деформованого стану конструкції в рамках лінійної теорії пружності при дії на неї статичних навантажень зводиться до розв'язання системи лінійних алгебраїчних рівнянь. У скінченно-елементних комплексах програм використовуються різноманітні методи розв'язання великих систем рівнянь.

Різні варіанти методу Гаусса реалізовані в програмах ADINA (блочний метод Гаусса), NASTRAN (LTDL - декомпозиція). Ефективним є фронтальний метод, реалізований у програмах ABAQUS, ANSYS та ін. Методи суперелементів і редукції базису дозволяють суттєво скоротити час обчислення. Ефективними є також ітераційні методи.

Розрахунок власних коливань конструкції виконується методами ітерації в підпросторі (SAP-7), обчислення кореня характеристичного визначника (NASTRAN), Хаусхолдера з використанням методу Якобі (ASKA), Гівенса і QR-методу (NASTRAN), Hiber-Hughes (ABAQUS), Ланцоша (ANSYS). При розрахунку динамічного відгуку використовуються методи: подання розв'язання у вигляді суперпозиції форм власних коливань, крокові - Вілсона, Ньюмарка (ABAQUS, ADINA, SAP-7, NASTRAN). Розв'язання геометрично і фізично нелінійних задач здійснюється, як правило, ітераційними методами, основу яких складає метод Ньютона-Рафсона в поєднанні із кроковими методами (ABAQUS, ADINA, NASTRAN, ANSYS та ін.).

Необхідно відзначити, що принцип модульного програмування, який використовується в програмних комплексах, дозволяє створювати як універсальні обчислювальні програми, так і промислові для вирішення вузького класу задач. На перших етапах освоєння МСЕ розроблялися в основному промислові обчислювальні програми. Вони ефективні, якщо розв'язується велика кількість варіантів однотипних задач або виконується великий об'єм обчислень для якісного і кількісного дослідження явищ, які пов'язані з новим поставленням задачі. Тенденція розвитку обчислювальної техніки, що привела до створення персональних ЕОМ і нових інформаційних

технологій, зробила вплив на розроблення програмного забезпечення МСЕ. Програмні комплекси за МСЕ активно використовуються у системах автоматизованого проектування, які базуються на персональних ЕОМ (AutoCAD/MechanicalDesktop, Pro-Engineer).

Сучасні програмні комплекси, в яких використовується МСЕ, дозволяють одержувати наближені чисельні розв'язання при розрахунку конструкцій на статичні і динамічні навантаження для широкого класу матеріалів з різними механічними характеристиками і поведінкою. Розрахунок конструкцій на статичні навантаження може проводитися з урахуванням фізичної і геометричної нелінійності, температурних полів, взаємодії з іншими середовищами (наприклад, з рідиною). Проводиться розрахунок критичних навантажень, при яких конструкція або її елементи втрачають стійкість та поведінку конструкції після втрати стійкості. МСЕ дозволяє також визначити навантаження, при яких відбувається руйнування конструкції. Враховуються такі властивості матеріалу, як анізотропія, нелінійна пружність, пластичність, текучість. Враховуються види геометричної нелінійності: великі деформації і великі переміщення.

Як було показано в [10] 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).

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 опрацювання 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 тел.

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 «недоторканною» 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 утоми і процедурами

оптимізації. Єдина система команд й єдина база даних повністю виключає труднощі інтеграції й взаємного обміну між зазначеними сферами. Більше того, шляхом використання в програмі спеціалізованих кінцевих елементів, що мають крім переміщень і поворотів у вузлах також і ступеня 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. Загальні можливості ANSYS

2.2.1. Загальна методика розрахунку на міцність композиційних конструкцій методом скінченних елементів в програмі ANSYS

Розрахунок на міцність композиційних конструкцій методом кінцевих елементів використовується для аналізу напружено-деформованого та граничного стану різних конструкційних елементів. Використання даного методу дозволяє більш повно дослідити процеси деформування та руйнування особливо у елементах із складною геометрією. Можливо комплексно задати умови навантаження та закріплення, розрахунок можна проводити одразу із задаванням багатьох кроків дослідження. Відображення результатів за допомогою полів, графіків, таблиць побудованих за результатами в точці, вздовж лінії чи вибраної площини дозволяє провести детальний аналіз стану конструкції. Для композитів використовується окремий модуль в якому можна задати шаруватість та спрямованість матеріалу.

У даній роботі використано програму ANSYS. Розрахунок на міцність у програмі ANSYS передбачає наступні загальні кроки спільні для інших програм також:

1. Задавання характеристик матеріалу: границі міцності розтяг/стиск, модулі пружності при розтягу, модулі зсуву, границі міцності на зсув, коефіцієнти Пуасона відносно кожної вісі анізотропії матеріалу, а

також визначення, при необхідності, густини, інших фізичних характеристик, параметрів втомлюваності та повзучості при необхідних температурах.

2. Створення за допомогою вбудованого редактора геометрії деталі чи, як правило, імпорт геометрії деталі із інших програм (SolidWorks, NX, Catia).
3. Створення допоміжних площин та систем координат для виконання подальших операцій.
4. Задавання та утворення розрахункової сітки. Обирається тип сітки, форма елементів, розмір елементів. Визначаються місця з необхідною підвищеною точністю розрахунків: місця датчиків, перегинів, з'єднань, тріщин та ін.. В таких місцях задається подрібнена сітка. Проводиться утворення та оптимізація розрахункової сітки.
5. Для композиційних деталей додатково визначаються шари, їх матеріал, розміщення, послідовність укладання та напрямки.
6. Застосовуються додаткові параметри (вставляються тріщини).
7. Визначаються місця та типи фіксації.
8. Визначаються параметри навантаження.
9. Визначаються параметри навколишнього середовища (температура).
10. Задаються параметри розрахунку (кількості кроків та час, зокрема).
11. Задаються параметри відображення результатів та спосіб їх відображення. Показуються деформації та напруження різних типів (максимальні, вздовж напрямку, по Мізесу та ін.) в точці, вздовж лінії, на площині чи поверхні елементу.
12. Задаються додаткові критерії (запас міцності, КІН та ін.)
13. Для композитів додатково задаються параметри відображення результатів по шарам чи волокнам (напруження в шарі, між шарові напруження, руйнування волокон та ін.).

14. Виконується розрахунок, після якого результати автоматично відображаються за заданими налаштуваннями.

Після ряду вказаних кроків можна проводити аналіз отриманих даних і проводити подальші числові розрахунки.

2.2.2. Призначення ANSYS

Програма ANSYS – це гнучкий засіб проектування й аналізу. Вона працює в середовищі операційних систем найпоширеніших комп'ютерів – від PC до робочих станцій і суперкомп'ютерів. Особливістю програми є файлова сумісність всіх членів сімейства ANSYS для всіх використовуваних платформ. Багатоцільова спрямованість програми (тобто реалізація в ній засобів для опису відгуку системи на впливи різної фізичної природи) дозволяє використовувати ту саму модель для розв'язування таких зв'язаних завдань, як міцність при тепловому навантаженні, вплив магнітних полів на міцність конструкції, тепломасоперенесення в електромагнітному полі. Модель, створена на PC, може використовуватися на суперкомп'ютері. Це забезпечує всім користувачам програми можливості розв'язування широкого кола інженерних завдань [13].

Програма ANSYS пропонує безупинно зростаючий перелік розрахункових засобів, які [12]:

- можуть урахувати різноманітні конструктивні нелінійності;
- дають можливість вирішити самий загальний випадок контактного завдання для поверхонь;
- допускають наявність великих (кінцевих) деформацій і кутів повороту;
- дозволяють виконати інтерактивну оптимізацію й аналіз впливу електромагнітних полів, одержати розв'язання завдань гідроаеродинаміки
- і багато чого іншого – разом з параметричним моделюванням, адаптивним перебудуванням сітки, використанням *p*-елементів і

великих можливостей створення макрокоманд за допомогою мови параметричного проектування програми ANSYS (APDL).

2.2.3. 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.

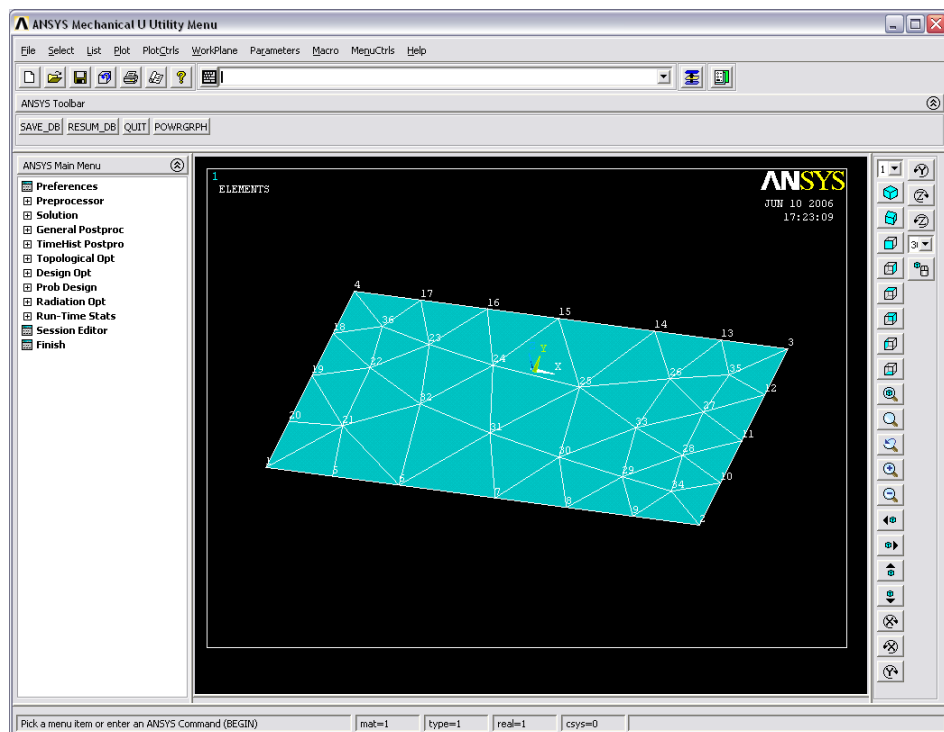


Рис. 2.1. Інтерфейс Motif Standard

Існують чотири загальних методи уведення 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 меню (Рис. 2.2) і вікон списку, допомагаючи користувачеві управляти програмою.

2.2.4. Графічні можливості

Засоби твердотілого 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 щирої форми й поперечного переріза трубопроводів, патрубків, балок і магнітів;
- показ окремих шарів композитних матеріалів й 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.

2.2.5. 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.

Процесори (підпрограми), доступні в програмі ANSYS.

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

2.2.6. База даних

У програмі ANSYS використовується одна, центральна, база даних для всього набору відомостей, що стосуються моделі й результатів обчислень (Рис. 2.3). 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.

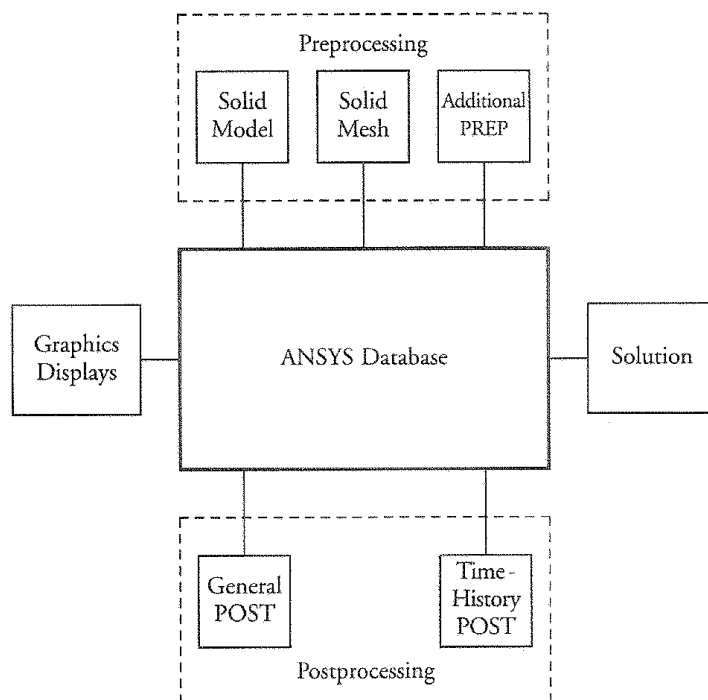


Рис. 2.2. Діаграма зв'язків центральної бази дані програми ANSYS

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

2.2.7. Побудова сітки

Програма ANSYS уможливорює швидку й зручну побудову сітки високої якості для CAD-моделей. При цьому використовуються численні засоби керування якістю сітки.

Після того як побудована твердотільна модель, її кінцево-елементний аналог (тобто сітка вузлів й елементів) може бути створена всього лише одним звертанням до меню програми. У програмі ANSYS передбачено чотири способи генерації сітки: використання методу екструзії, створення впорядкованої сітки, створення довільної сітки (автоматично) і адаптивна побудова.

Метод екструзії (видавлювання) використається для перетворення областей двовимірної сітки в тривимірні об'єкти, що складаються з паралелепіпедів, клиноподібних елементів або їхньої комбінації. Процес

екструзії здійснюється за допомогою процедур зсуву із площини, буксирування, поступального й обертального переміщень.

Побудова впорядкованої сітки вимагає попередньої розбивки моделі на окремі складові частини із простою геометрією, а потім вибору таких атрибутів елемента й відповідних команд керування якістю сітки, щоб можна було побудувати кінцево-елементну модель із упорядкованою сіткою. Створювана програмою ANSYS упорядкована сітка може складатися із шестикутних, чотирикутних і трикутних елементів. Для одержання трикутної сітки програма виділяє області моделі, призначені для нанесення впорядкованої сітки, створює спочатку чотирикутну сітку, а потім перетворює її в сітку із трикутних елементів. Такий спосіб побудови сітки є досить зручним як при створенні сітки з тетраедрів (описується нижче), так і при нанесенні сітки на твердих поверхнях при рішенні контактних завдань, де переважно зводити число кінцевих елементів до мінімуму.

Як додатковий спосіб побудови впорядкованої сітки на деякій поверхні використається розподіл протилежних граничних ліній цієї поверхні таким чином, щоб можна було здійснити перехід від одного розміру сітки до іншого. Побудова впорядкованої сітки змінного розміру можливо тільки для поверхонь, обмежених чотирма лініями (з використанням або без використання конкатенації, тобто операції об'єднання двох послідовностей граничних ліній в одну, при цьому перша послідовність стає початком результуючої, а друга - її хвостом).

Програма ANSYS надає в розпорядження користувача надійні генератори довільної сітки, за допомогою яких вона може наноситися безпосередньо на модель досить складної геометрії - без необхідності будувати сітку для окремих частин і потім збирати їх у єдину модель. Довільну сітку можна будувати із трикутних, чотирикутних і чотиригранних елементів. Генератори довільної сітки мають у наявності основної й додатковий побудовники, що підвищує гнучкість вибору для користувача й імовірність одержання сітки задовільної якості.

Генератор тетраедної сітки дозволяє одержати об'ємну сітку високої якості. Він забезпечує однокрокову побудову сітки після інтенсивної перевірки моделі, а потім виконує етап поліпшення готової сітки. На підготовчій стадії перевіряється, немає чи взаємо перетину граничних граней і правильність їхньої орієнтації, а користувачеві повідомляється інформація про можливі проблеми або неможливість побудови сітки.

Генератори довільної сітки мають широкий набір внутрішніх і зовнішніх опцій керування якістю сітки. Так, наприклад, реалізований алгоритм розумного вибору розмірів кінцевого елемента, що дозволяє будувати сітку елементів з урахуванням кривизни поверхні моделі й найкращого відображення її реальної геометрії (Рис. 2.3). Користувач може вибрати дрібну або велику сітку елементів, указавши як керуючий параметр будь-яке число з діапазону від одиниці до десяти. Крім того, кожний із цих генераторів постачений найсучаснішими алгоритмами згладжування й рафінування розмірів сітки, що знижує число елементів незадовільної форми й переривань процедури побудови сітки.

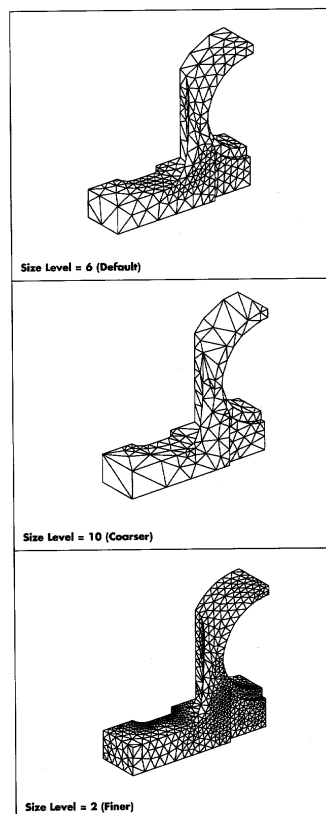


Рис. 2.3. Зміна розміру сітки для однієї й тієї ж моделі

Інші доступні засоби керування якістю сітки включають: вказівка загального розміру елемента, розподіл граничної лінії, розміри в околиці заданих геометричних крапок, коефіцієнти розтягання або стиску вдалині від границь, обмеження на кривизну й можливість завдання “твердих” крапок (тобто завдання точного положення вузла разом з розмірами сітки в такій крапці). Оскільки тетраедний генератор працює з урахуванням розміру сітки на сусідніх частинах моделі, користувач має можливість указати потрібну сітку на границі області до звертання до генератора.

Ще один вид контролю якості сітки реалізований для двовимірних областей, складних у геометричному відношенні, для яких ручна розбивка на частини й наступну побудову впорядкованої сітки є обтяжним (до них ставляться моделювання деяких зон потоків рідини або газу). Для таких випадків програмний засіб пошарової побудови сітки дає можливість автоматично генерувати довільну сітку, профільовану уздовж деякої перешкоди й зручну для моделювання явищ, що мають шарувату структуру (Рис. 2.4). Пошарова побудова сітки особливо зручно при рішенні завдань обчислювальної гідроаеродинаміки за допомогою модуля ANSYS/FLOTTRAN для обліку прикордонних ефектів або для моделювання поверхневих ефектів в електромагнітному аналізі. Користувач має можливість автоматично генерувати сітку, профільовану уздовж потрібних границь моделі й, що складаються з елементів заданого розміру.

Перехід від шестигранної сітки до чотиригранного з використанням пірамідальних елементів являє собою коштовний засіб для моделювання геометрії на стику областей з різною сіткою (Рис. 2.4). У користувача їсти можливість автоматично зістиковувати такі області без необхідності вводити умови-обмеження або пропускати серединні вузли елементів й уникати математичних розривів у шуканих функціях. Така можливість досить зручна, оскільки деякі області моделі можна досить просто розбити на частині, для яких будується впорядкована сітка, тоді як інші її області можуть бути більше

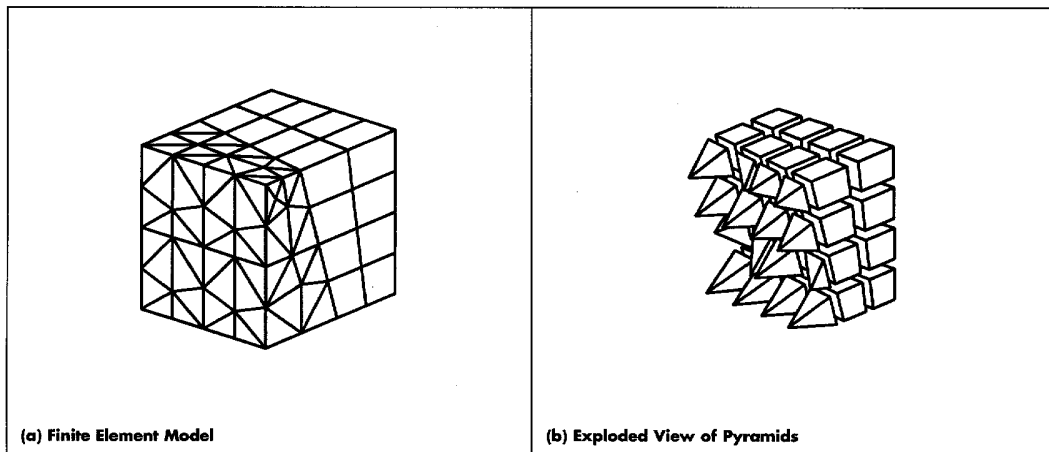


Рис. 2.4. Пірамідальні елементи в перехідній області моделі

складними в геометричному відношенні. Для більш простих областей моделі можна використати шестигранні елементи, а для інших - тетраедри. Області моделі зі значними градієнтами шуканих величин можуть зажадати введення сітки із шестиграних елементів, а для менш критичних областей придатними можуть виявитися тетраедри.

Адаптивна побудова сітки полягає в тому, що після створення твердотільної моделі й завдання граничних умов програма, за вказівкою користувача, генерує кінцево-елементну сітку, виконує аналіз, оцінює помилку за рахунок сіткової дискретизації й міняє розмір сітки від рішення до рішення доти, поки розрахункова погрішність не стане менше якоїсь наперед заданої величини (або поки не буде досягнуте встановлене число ітерацій).

Адаптивна побудова моделі можливо використати для статичного лінійного аналізу або стаціонарного теплового розрахунку. Така процедура створення сітки може бути виконана й при наявності декількох різних умов навантаження. Крім того, користувач має можливість указати ті області розрахункової моделі, для яких зменшення помилки дискретизації не настільки важливо, і виключити їх з адаптивної процедури. Користувач може настроїти процедуру адаптивної побудови сітки, виходячи з індивідуального підходу до проведення аналізу.

Можливості програми ANSYS допускають проведення модифікації кінцево-елементної сітки. Наприклад, можуть бути змінені атрибути вузлів й

елементів. Якщо модель складається з повторюваних областей, то можна створити сітку тільки для деякої області моделі, а потім зробити копію цієї області. Після того як геометрична модель покривається сіткою кінцевих елементів, програма автоматично забезпечує їхній взаємо-перекресний контроль, щоб гарантувати правильність виконуваних користувачем видозмін сіткової моделі. Такі перевірки запобігають некоректне знищення або псування даних, що ставляться до твердотельної і сіткової моделей. Так, наприклад, ключові крапки, лінії, поверхні або обсяги сіткової моделі не можна знищити або перемістити доти, поки користувач явно не зажадає від програми скасувати їхній автоматичний контроль.

2.3. Мова APDL у середовищі ANSYS

У системі ANSYS розроблено мову параметричного проектування, за допомогою якого можна створювати користувацькі моделі матеріалів, описувати складні геометричні конструкції, використовувати власні функції навантажень і геометричних параметрів, а також дистанційно керувати системою. Люба дія в інтерфейсі системи ANSYS дублюється відповідною командою на вбудованій мові проектування, тому створення нової моделі матеріалу можна автоматизувати. Для цього достатньо створити текстовий файл із команди та параметрів, який міститиме всі необхідні процедури для створення нової моделі матеріалу, та імпортувати його в систему ANSYS. За допомогою вкладених циклів і логічних операторів визначаються структура і зміст необхідного набору команд.

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, такі як **/PREP7**, **/POST26**, і ***ENDDO**). Деякі команди

(наприклад, **ADAPT** й **ETEXR**) у дійсності є *макрокомандами*. Імена макрокоманд повинні бути уведені 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. Замовчення

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 попереджень, 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 взаємодія 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 повинна 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 потрібна зв'язок з комп'ютером.

2.3.4 Робота LOG файлу

Кожний 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 тім 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 потрібно 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 введення.

2.4. Висновки

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).

РОЗДІЛ 3. ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ НАМОТУВАННЯ ДЛЯ ВИРОБІВ З КОМПОЗИЦІЙНИХ МАТЕРІАЛІВ

В системах автоматизованого проектування, що мають справу з такими геометричними об'єктами, як криві, поверхні, тіла, велике застосування знаходять геодезичні лінії. Особливо плідні застосування геодезичні лінії знаходять у геометричному моделюванні різних технологічних процесів, зокрема, у моделюванні процесу намотування при одержанні виробів з композиційних матеріалів. Це порозумівається тим, що геодезичні лінії на поверхні мають чудові властивості. По-перше, вони дають найкоротшу відстань між двома точками поверхні серед всіх кривих на поверхні, що з'єднують ці точки. По-друге, головна нормаль у кожній точці геодезичної збігається з напрямком нормалі до поверхні в цій точці. Тому велике поширення одержало намотування стрічки з композиційних матеріалів на поверхню оправлення по геодезичних лініях, при якій автоматично досягається мінімізація витрати композита й рівноважне положення стрічки (точніше кажучи, того волокна, що укладається по геодезичній лінії) на поверхні оправлення. Крім того, у загальній концепції геометричних моделей процесу намотування, у яких ураховується розходження геометричних характеристик різних волокон стрічки, з яких вона складається, при їхньому укладанні на поверхню оправлення, одне із центральних місць також займають геодезичні лінії.

Так як в загальному випадку система звичайних диференціальних рівнянь, що задають геодезичні лінії на поверхні, досить громіздка, і її чисельне рішення при моделюванні займає чимало часу, то актуальною є завдання спрощення цієї системи для яких-небудь класів поверхонь, пошуку інтегровальних комбінацій, що заміняють собою цю систему.

3.1. Математичні моделі розрахунку траєкторії геодезичної укладки по даним кінцево – елементної моделі без застосування диференціалів

Програма ANSYS уможлиблює швидко й зручну побудову сітки високої якості для CAD-моделей. При цьому використовуються численні засоби керування якістю сітки.

Створювана програмою ANSYS упорядкована сітка може складатися із шестикутних, чотирикутних і трикутних елементів. Для одержання програма виділяє області моделі, призначені для нанесення впорядкованої сітки, створює спочатку чотирикутну сітку, а потім перетворює її в сітку із трикутних елементів.

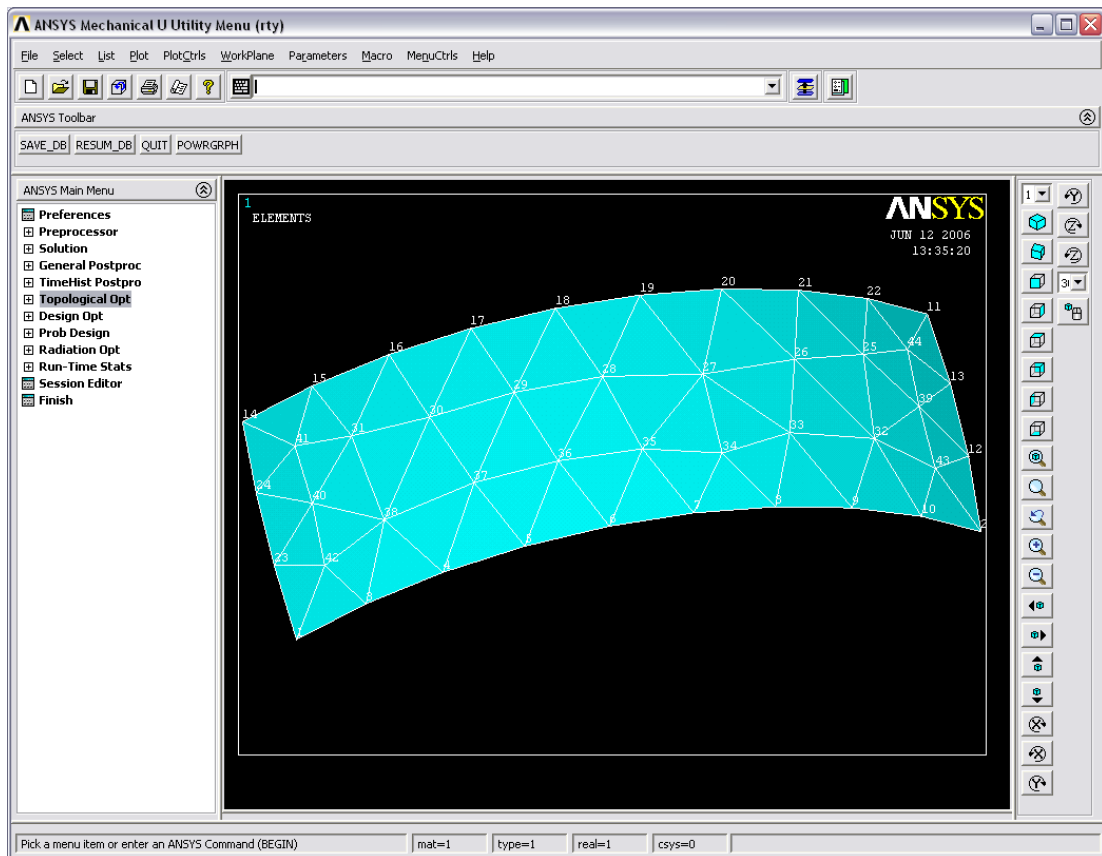


Рис.3.1. упорядкована сітка

В створеній моделі розглядається робота з трикутним кінцевим елементом SHELL 63-упруга оболонка. Геометрія елемента наведена на рис.3.2.

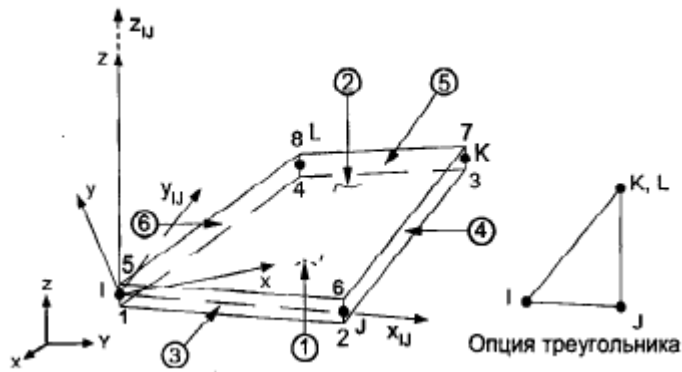


Рис.3.2. Елемент SHELL 63-упруга оболонка

Після того, як зроблена трикутна сітка, виводимо в файли данні о координатах вузлових точок кінцевих елементів (файл NLIST) та список вузлів які формують кінцеві елементи (файл ELIST).

ELIST Command

File

LIST ALL SELECTED ELEMENTS. (LIST NODES)

ELEM	MAT	TYP	REL	ESY	SEC	NODES
1	1	1	1	0	1	42 40 23 23
2	1	1	1	0	1	42 38 40 40
3	1	1	1	0	1	43 32 10 10
4	1	1	1	0	1	43 39 32 32
5	1	1	1	0	1	41 40 31 31
6	1	1	1	0	1	41 24 40 40
7	1	1	1	0	1	44 25 39 39
8	1	1	1	0	1	44 22 25 25
9	1	1	1	0	1	13 44 39 39
10	1	1	1	0	1	44 11 22 22
11	1	1	1	0	1	13 11 44 44
12	1	1	1	0	1	39 43 12 12
13	1	1	1	0	1	43 2 12 12
14	1	1	1	0	1	10 2 43 43
15	1	1	1	0	1	38 42 3 3
16	1	1	1	0	1	42 1 3 3
17	1	1	1	0	1	23 1 42 42
18	1	1	1	0	1	15 41 31 31
19	1	1	1	0	1	41 14 24 24
20	1	1	1	0	1	15 14 41 41

Рис.3.3. файл ELIST

NLIST Command

File

LIST ALL SELECTED NODES. DSYS= 0
SORT TABLE ON: NODE NODE NODE

***** ANSYS - ENGINEERING ANALYSIS SYSTEM RELEASE 8.0 *****
ANSYS Mechanical U
00265231 VERSION=INTEL NT 14:11:43 JUN 12, 2006 CP= 3.688

NODE	X	Y	Z
1	0.5000000000	1.0000000000	0.0000000000
2	0.5000000000	0.0000000000	1.0000000000
3	0.5000000000	0.984807753007	0.173648177700
4	0.5000000000	0.939692620747	0.342020143432
5	0.5000000000	0.866025403687	0.500000000168
6	0.5000000000	0.766044442942	0.642787609897
7	0.5000000000	0.642787609414	0.766044443347
8	0.5000000000	0.499999999623	0.866025404002
9	0.5000000000	0.342020142840	0.939692620963
10	0.5000000000	0.173648177079	0.984807753116
11	1.0000000000	0.0000000000	1.0000000000
12	0.666472413303	0.0000000000	1.01394154103
13	0.833527586696	0.0000000000	1.01394154103
14	1.0000000000	1.0000000000	0.0000000000
15	1.0000000000	0.984807753007	0.173648177700
16	1.0000000000	0.939692620747	0.342020143432
17	1.0000000000	0.866025403687	0.500000000168
18	1.0000000000	0.766044442942	0.642787609897
19	1.0000000000	0.642787609414	0.766044443347
20	1.0000000000	0.499999999623	0.866025404002
21	1.0000000000	0.342020142840	0.939692620963
22	1.0000000000	0.173648177079	0.984807753116
23	0.666472413303	1.01394154103	0.0000000000
24	0.833527586696	1.01394154103	0.0000000000

Рис.3.4. файл NLIST

Програма ANSYS розбиває поверхню кінцевими елементами так, що ми можемо розглядати окремий кінцевий елемент як поверхню першого порядку, а не другого. Таким чином перехід від поверхні другого порядку до поверхні першого порядку спрощує математичну модель розрахунку траєкторії геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі.

3.1.1 Перша математична модель визначення координат геодезичної лінії

У вибраному кінцевому елементі визначаємо:

- рівняння площини по 3-м точкам:

Рівняння в загальному виді:

$$\mathbf{Ax} + \mathbf{By} + \mathbf{Cz} + \mathbf{D} = 0 \quad (1)$$

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0$$

$$(\mathbf{x} - \mathbf{x}_1)\mathbf{A}_{11} + (\mathbf{y} - \mathbf{y}_1)\mathbf{A}_{12} + \mathbf{A}_{13}(\mathbf{z} - \mathbf{z}_1) = 0 \quad (*)$$

$$\mathbf{A}_{11} = (-1)^2 \begin{vmatrix} y_2 - y_1 & z_2 - z_1 \\ y_3 - y_1 & z_3 - z_1 \end{vmatrix} = (-1)^2 ((y_2 - y_1) * (z_3 - z_1) - (y_3 - y_1) * (z_2 - z_1))$$

$$\mathbf{A}_{12} = (-1)^3 \begin{vmatrix} x_2 - x_1 & z_2 - z_1 \\ x_3 - x_1 & z_3 - z_1 \end{vmatrix} = (-1)^3 ((x_2 - x_1) * (z_3 - z_1) - (x_3 - x_1) * (z_2 - z_1))$$

$$\mathbf{A}_{13} = (-1)^4 \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = (-1)^4 ((x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1))$$

Підставимо коефіцієнти $\mathbf{A}_{11}, \mathbf{A}_{12}, \mathbf{A}_{13}$ у рівняння (*) і одержимо:

$$\begin{aligned} & (x - x_1) * (-1)^2 * ((y_2 - y_1) * (z_3 - z_1) - (y_3 - y_1) * (z_2 - z_1)) + \\ & + (y - y_1) * (-1)^3 * ((x_2 - x_1) * (z_3 - z_1) - (x_3 - x_1) * (z_2 - z_1)) + \quad (**) \\ & + (z - z_1) * (-1)^4 * ((x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)) = 0 \end{aligned}$$

Спростивши рівняння (**) знайдемо коефіцієнти **A, B, C, D** для рівняння (1)

$$\mathbf{A} = (y_2 - y_1) * (z_3 - z_1) - (y_3 - y_1) * (z_2 - z_1)$$

$$\mathbf{B} = (x_3 - x_1) * (z_2 - z_1) - (x_2 - x_1) * (z_3 - z_1)$$

$$\mathbf{C} = (x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)$$

$$\begin{aligned} \mathbf{D} = & x_1 * ((y_3 - y_1) * (z_2 - z_1) - (y_2 - y_1) * (z_3 - z_1)) + \\ & + y_1 * ((x_2 - x_1) * (z_3 - z_1) - (x_3 - x_1) * (z_2 - z_1)) + \\ & + z_1 * ((x_3 - x_1) * (y_2 - y_1) - (x_2 - x_1) * (y_3 - y_1)) \end{aligned}$$

Отже, вектор нормалі $\vec{N} = (A, B, C)$.

Нормальне рівняння площини:

$$x \cos \alpha + y \cos \beta + z \cos \gamma - p = 0$$

$$\begin{aligned}\cos \alpha &= \frac{A}{\pm \sqrt{A^2 + B^2 + C^2}} \\ \cos \beta &= \frac{B}{\pm \sqrt{A^2 + B^2 + C^2}} \\ \cos \gamma &= \frac{C}{\pm \sqrt{A^2 + B^2 + C^2}} \\ p &= \frac{D}{\pm \sqrt{A^2 + B^2 + C^2}}\end{aligned}$$

де
знак множника, що $\frac{1}{\pm \sqrt{A^2 + B^2 + C^2}}$ нормує, береться протилежним знаку D, якщо D=0 знак довільний. Знак перед коренем вибирається так, щоб дотримувалася умова $p > 0$.

- рівняння прямої через дві точки кінцевого елемента, одна з якої вузлова крапка, обрана користувачем:

$$(1^*) \frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}, \text{ где } \vec{R} = (l, m, n) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

або $\frac{x - x_1}{\cos \alpha'} = \frac{y - y_1}{\cos \beta'} = \frac{z - z_1}{\cos \gamma'}$, де α', β', γ' -кути утворені прямої з осями координат

OX, OY, OZ. Тоді напрямні косинуси прямої обчислюються:

$$\begin{aligned}\cos \alpha' &= \frac{l}{\sqrt{l^2 + m^2 + n^2}} \\ \cos \beta' &= \frac{m}{\sqrt{l^2 + m^2 + n^2}} \\ \cos \gamma' &= \frac{n}{\sqrt{l^2 + m^2 + n^2}}\end{aligned}$$

- Рівняння прямої через дві точки кінцевого елемента

$$\frac{x - x_1}{x_3 - x_1} = \frac{y - y_1}{y_3 - y_1} = \frac{z - z_1}{z_3 - z_1} (1^{**}), \text{ де } \vec{M} = (l_1, m_1, n_1) = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$$

Знайдемо кут між прямими (1*) і (1**) $\vec{R} = (l, m, n) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$ і

$$\vec{M} = (l_1, m_1, n_1) = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$$

$$\cos \varepsilon = \frac{l * l_1 + m * m_1 + n * n_1}{\sqrt{l^2 + m^2 + n^2} * \sqrt{l_1^2 + m_1^2 + n_1^2}}$$

$$\varepsilon = \arccos\left(\frac{l * l_1 + m * m_1 + n * n_1}{\sqrt{l^2 + m^2 + n^2} * \sqrt{l_1^2 + m_1^2 + n_1^2}}\right)$$

- Проекція осі ОХ на дану площину (одержати рівняння прямої)

Усяка площина, що проходить через вісь ОХ, виражається рівнянням $B_1y + C_1z = 0$ (2). Ця площина повинна бути перпендикулярна площини (1) з вектором нормалі $\vec{N} = (A, B, C)$. Отже, рівняння (2) буде мати вигляд

$$\frac{-C}{B}y + z = 0 \quad (3), \text{ де вектор нормалі } \vec{N}' = (0, \frac{-C}{B}, 1).$$

Рівняння прямої задане перетинанням двох площин

$$\begin{cases} Ax + By + Cz + D = 0 \\ \frac{-C}{B}y + z = 0 \end{cases}$$

$$\vec{N} = (A, B, C), \quad \vec{N}' = (0, \frac{-C}{B}, 1).$$

$$\vec{S} = \vec{N} \times \vec{N}' = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & \frac{-C}{B} & 1 \\ A & B & C \end{vmatrix}.$$

Рівняння прямої:

$$\frac{x - x'_1}{\begin{vmatrix} \frac{-C}{B} & 1 \\ B & C \end{vmatrix}} = \frac{y - y'_1}{\begin{vmatrix} 1 & 0 \\ C & A \end{vmatrix}} = \frac{z - z'_1}{\begin{vmatrix} 0 & \frac{-C}{B} \\ A & B \end{vmatrix}}; \quad (2^*).$$

$$\frac{x - x'_1}{\frac{-C^2}{B} - B} = \frac{y - y'_1}{A} = \frac{z - z'_1}{\frac{C * A}{B}}$$

Напрямний вектор прямиї (4) $\vec{S} = (\frac{-C^2}{B} - B, A, \frac{C * A}{B})$.

- Виконаємо перехід від однієї системи координат до іншої (одночасний перенос і поворот осей).

Для цього потрібно провести дві перпендикулярні прями до прямиї (1*)

пряма минаюча через точку (x_1, y_1, z_1) перпендикулярно площини (1) задається рівнянням:

$$\cos \alpha'' = \frac{A}{\sqrt{A^2 + B^2 + C^2}}$$

$$\frac{x - x_1}{A} = \frac{y - y_1}{B} = \frac{z - z_1}{C} \quad (2), \text{ тоді напрямні косинуси (2): } \cos \beta'' = \frac{B}{\sqrt{A^2 + B^2 + C^2}}$$

$$\cos \gamma'' = \frac{C}{\sqrt{A^2 + B^2 + C^2}}$$

Проведемо площину перпендикулярну прямій $\frac{x - x_1}{l} = \frac{y - y_1}{m} = \frac{z - z_1}{n} \quad (1^*)$

минаючу через точку (x_1, y_1, z_1) :

$$l(x - x_1) + m(y - y_1) + n(z - z_1) = 0$$

$$lx + my + nz - (lx_1 + my_1 + nz_1) = 0$$

Знайдемо рівняння прямої минаючої через точку (x_1, y_1, z_1) і перетинання двох площин:

$$\begin{cases} Ax + By + Cz + D = 0 \\ lx + my + nz - (lx_1 + my_1 + nz_1) = 0 \end{cases}$$

$$\vec{N} = (A, B, C), \quad \vec{N}'' = (l, m, n)$$

$$\vec{S} = \vec{N} \times \vec{N}'' = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ l & m & n \\ A & B & C \end{vmatrix}$$

Рівняння прямої:

$$\frac{x - x_0}{\begin{vmatrix} m & n \\ B & C \end{vmatrix}} = \frac{y - y_0}{\begin{vmatrix} n & l \\ C & A \end{vmatrix}} = \frac{z - z_0}{\begin{vmatrix} l & m \\ A & B \end{vmatrix}}; \quad (3^*),$$

$$\frac{x - x_0}{mc - Bn} = \frac{y - y_0}{nA - lC} = \frac{z - z_0}{lB - Am}$$

тоді напрямні косинуси (3):

$$\cos \alpha''' = \frac{mC - Bn}{\sqrt{(mC - Bn)^2 + (nA - lC)^2 + (lB - Am)^2}}$$

$$\cos \beta'' = \frac{nA - lC}{\sqrt{(mC - Bn)^2 + (nA - lC)^2 + (lB - Am)^2}}$$

$$\cos \gamma'' = \frac{lB - Am}{\sqrt{(mC - Bn)^2 + (nA - lC)^2 + (lB - Am)^2}}$$

Виконаємо одночасний перенос і поворот осей у початкову крапку (x_1, y_1, z_1) , так щоб вісь ou' збігалася із прямою (1^*) а x координата рівнялася 0:

$$\left. \begin{aligned} \tilde{x} &= \cos \alpha''(x - x_0) + \cos \alpha'(y - y_0) + \cos \alpha'''(z - z_0) \\ \tilde{y} &= \cos \beta''(x - x_0) + \cos \beta'(y - y_0) + \cos \beta'''(z - z_0) \\ \tilde{z} &= \cos \gamma''(x - x_0) + \cos \gamma'(y - y_0) + \cos \gamma'''(z - z_0) \\ \text{или} \\ x &= \cos \alpha''\tilde{x} + \cos \alpha'\tilde{y} + \cos \alpha'''\tilde{z} \\ y &= \cos \beta''\tilde{x} + \cos \beta'\tilde{y} + \cos \beta'''\tilde{z} \\ z &= \cos \gamma''\tilde{x} + \cos \gamma'\tilde{y} + \cos \gamma'''\tilde{z} \end{aligned} \right\}$$

(Перехід з однієї прямолінійної координатної системи в тривимірному просторі до іншої описується в загальному випадку в такий спосіб:

$$\begin{aligned} x^* &= \alpha_1 x + \alpha_2 y + \alpha_3 z + \lambda \\ y^* &= \beta_1 x + \beta_2 y + \beta_3 z + \mu \\ z^* &= \gamma_1 x + \gamma_2 y + \gamma_3 z + \nu \end{aligned}$$

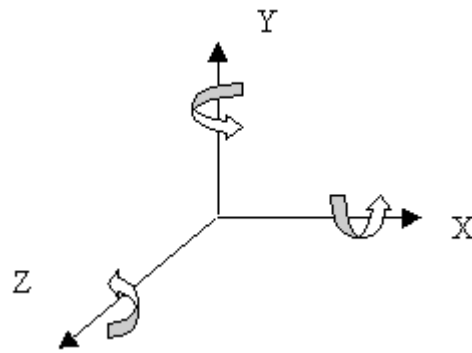
або в матричному виді:

$$[x^* y^* z^* 1] = [xyz1][A]$$

$$A = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & 0 \\ \alpha_2 & \beta_2 & \gamma_2 & 0 \\ \alpha_3 & \beta_3 & \gamma_3 & 0 \\ \lambda & \mu & \nu & 1 \end{bmatrix}$$

Розглянемо матриці, що відповідають наступним базовим геометричним перетворенням:

1. Повороти



навколо осі X на кут φ

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

навколо осі Y на кут ψ

$$[R_y] = \begin{bmatrix} \cos \psi & 0 & -\sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

навколо осі Z на кут χ

$$[R_z] = \begin{bmatrix} \cos \chi & \sin \chi & 0 & 0 \\ -\sin \chi & \cos \chi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Розтягання (стиск):

$$[D] = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

якщо $\alpha, \beta, \gamma > 1$ - розтягання, $0 < \alpha, \beta, \gamma < 1$ - стиск

3. Відбиття (зеркалювання)

щодо площини XOY

$$[M_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

щодо площини YOZ

$$[M_x] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

щодо площини ZOX

$$[M_y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Перенос (зрушення, переміщення) на вектор (λ, μ, ν)

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \lambda & \mu & \nu & 1 \end{bmatrix}$$

Важливо: перетворення точки (зі збереженням розташування вихідної системи координат) відповідає виконанню зворотної операції стосовно перетворення системи координат. Наприклад, **поворот точки** на деякий кут **по годинній стрілці** навколо осі X відповідає **повороту системи координат проти вартовий стрілки** на той же кут).

Знайдемо кут між прямими (1*) і (2*) $\vec{R} = (l, m, n) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$ і

$$\vec{S} = \left(\frac{-C^2}{B} - B, A, \frac{C * A}{B} \right) :$$

$$\cos \delta = \frac{l * \left(\frac{-C^2}{B} - B \right) + m * A + n * \frac{C * A}{B}}{\sqrt{l^2 + m^2 + n^2} * \sqrt{\left(\frac{-C^2}{B} - B \right)^2 + A^2 + \left(\frac{C * A}{B} \right)^2}}$$

$$\delta = \arccos\left(\frac{l * \left(\frac{-C^2}{B} - B\right) + m * A + n * \frac{C * A}{B}}{\sqrt{l^2 + m^2 + n^2} * \sqrt{\left(\frac{-C^2}{B} - B\right)^2 + A^2 + \left(\frac{C * A}{B}\right)^2}}\right)$$

У новій системі координат площина (1) буде лежати в площині OYZ, отже, можна перейти від рівнянь прямих у просторі до рівнянь прямих у площини, що належить площини (1).

Знайдемо рівняння прямих (2*) минаючої через точку (x_1, y_1, z_1) у новій системі координат (урахуємо, що координати точки $O'(x_1, y_1, z_1)$ у новій системі координат $(0,0,0)$):

$$\tilde{z} = tg \delta * \tilde{y} \quad (2**).$$

Знайдемо рівняння прямої минаючої через точку O' під кутом ϕ (обраним довільно) до прямих (2**):

$$\tilde{z} = tg(\phi + \delta) * \tilde{y} \quad (3**).$$

Якщо $\phi + \delta < \varepsilon$, то знаходимо рівняння прямої минаючої через дві точки $A(x_2, y_2, z_2)$ і $B(x_3, y_3, z_3)$ попередньо порівнявши їхньої координати й урахувавши, що координата $\tilde{x}_2 = 0$ й $\tilde{x}_3 = 0$

$$\tilde{x}_2 = \cos \alpha''(x_2 - x_0) + \cos \alpha'(y_2 - y_0) + \cos \alpha'''(z_2 - z_0)$$

$$\tilde{y}_2 = \cos \beta''(x_2 - x_0) + \cos \beta'(y_2 - y_0) + \cos \beta'''(z_2 - z_0)$$

$$\tilde{z}_2 = \cos \gamma''(x_2 - x_0) + \cos \gamma'(y_2 - y_0) + \cos \gamma'''(z_2 - z_0)$$

I

$$\tilde{x}_3 = \cos \alpha''(x_3 - x_0) + \cos \alpha'(y_3 - y_0) + \cos \alpha'''(z_3 - z_0)$$

$$\tilde{y}_3 = \cos \beta''(x_3 - x_0) + \cos \beta'(y_3 - y_0) + \cos \beta'''(z_3 - z_0)$$

$$\tilde{z}_3 = \cos \gamma''(x_3 - x_0) + \cos \gamma'(y_3 - y_0) + \cos \gamma'''(z_3 - z_0)$$

$$\frac{\tilde{y} - \tilde{y}_2}{\tilde{y}_3 - \tilde{y}_2} = \frac{\tilde{z} - \tilde{z}_2}{\tilde{z}_3 - \tilde{z}_2}, \text{ или } \tilde{z} = \tilde{y} * \frac{\tilde{z}_3 - \tilde{z}_2}{\tilde{y}_3 - \tilde{y}_2} + \tilde{z}_2 - \tilde{y}_2 * \frac{\tilde{z}_3 - \tilde{z}_2}{\tilde{y}_3 - \tilde{y}_2},$$

$$\text{где } k = \frac{\tilde{z}_3 - \tilde{z}_2}{\tilde{y}_3 - \tilde{y}_2}, \text{ b} = \tilde{z}_2 - \tilde{y}_2 * \frac{\tilde{z}_3 - \tilde{z}_2}{\tilde{y}_3 - \tilde{y}_2} \quad (3***)$$

$$\tilde{z} = k\tilde{y} + b$$

- Знайдемо точку перетинання прямих (3**) і (3***):

$$\begin{cases} \tilde{z} = k\tilde{y} + b \\ \tilde{z} = \operatorname{tg}(\phi + \delta) * \tilde{y} \end{cases}$$

$$\begin{cases} \tilde{z} = \frac{-k * b}{k - \operatorname{tg}(\phi + \delta)} + b ; \quad \text{крапка } M_1\left(\frac{-b}{k - \operatorname{tg}(\phi + \delta)}, \frac{-k * b}{k - \operatorname{tg}(\phi + \delta)} + b\right) \\ \tilde{y} = \frac{-b}{k - \operatorname{tg}(\phi + \delta)} \end{cases}$$

- Знайдемо кут між прямими (3**) і (3***):

Якщо $\operatorname{tg}(\phi + \delta) * k + 1 = 0$, то прямі перпендикулярні якщо ні, те

$$\eta = \operatorname{arctg}\left(\frac{\operatorname{tg}(\phi + \delta) - k}{1 + \operatorname{tg}(\phi + \delta) * k}\right)$$

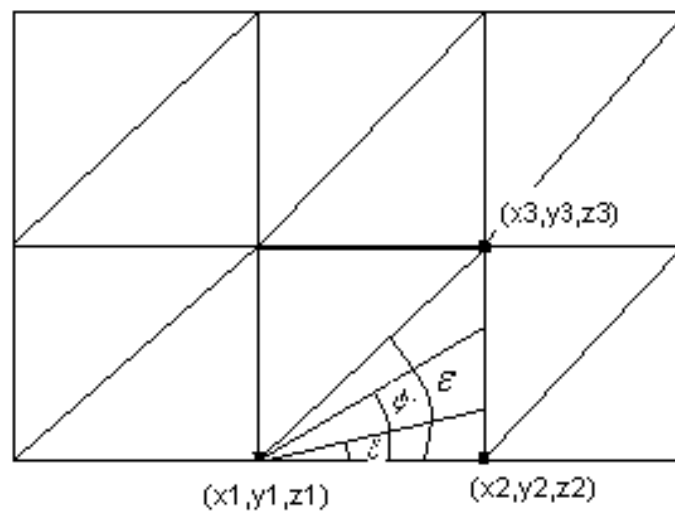


Рис.3.5. Розташування кутів

Якщо $\phi + \delta > \varepsilon$, то пряма не буде належати даному кінцевому елементу й потрібно вибрати кінцевий елемент який буде містити точки з координатами (x_1, y_1, z_1) і (x_3, y_3, z_3) , і повторити всю пройдену операцію.

Якщо $\phi + \delta = \varepsilon$, то крапка перетинання прямих (3**) і (3***) буде крапка з координатами (x_3, y_3, z_3) а кут між прямими:

$$\eta = \operatorname{arctg}\left(\frac{\operatorname{tg}(\varepsilon) - k}{1 + \operatorname{tg}(\varepsilon) * k}\right).$$

3.1.2. Друга математична модель визначення координат геодезичної лінії

У вибраному кінцевому елементі визначаємо:

- рівняння площини по 3-м точкам:

Рівняння в загальному виді:

$$\mathbf{Ax} + \mathbf{By} + \mathbf{Cz} + \mathbf{D} = 0 \quad (1)$$

$$\begin{vmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix} = 0$$

$$(\mathbf{x} - \mathbf{x}_1)\mathbf{A}_{11} + (\mathbf{y} - \mathbf{y}_1)\mathbf{A}_{12} + \mathbf{A}_{13}(\mathbf{z} - \mathbf{z}_1) = 0 \quad (*)$$

$$A_{11} = (-1)^2 \begin{vmatrix} y_2 - y_1 & z_2 - z_1 \\ y_3 - y_1 & z_3 - z_1 \end{vmatrix} = (-1)^2 ((y_2 - y_1) * (z_3 - z_1) - (y_3 - y_1) * (z_2 - z_1))$$

$$A_{12} = (-1)^3 \begin{vmatrix} x_2 - x_1 & z_2 - z_1 \\ x_3 - x_1 & z_3 - z_1 \end{vmatrix} = (-1)^3 ((x_2 - x_1) * (z_3 - z_1) - (x_3 - x_1) * (z_2 - z_1))$$

$$A_{13} = (-1)^4 \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = (-1)^4 ((x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1))$$

Підставимо коефіцієнти A_{11}, A_{12}, A_{13} у рівняння (*) і одержимо:

$$\begin{aligned} & (x - x_1) * (-1)^2 * ((y_2 - y_1) * (z_3 - z_1) - (y_3 - y_1) * (z_2 - z_1)) + \\ & + (y - y_1) * (-1)^3 * ((x_2 - x_1) * (z_3 - z_1) - (x_3 - x_1) * (z_2 - z_1)) + \quad (**) \\ & + (z - z_1) * (-1)^4 * ((x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)) = 0 \end{aligned}$$

Спростивши рівняння (**) знайдемо коефіцієнти $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ для рівняння

(1)

$$\mathbf{A} = (y_2 - y_1) * (z_3 - z_1) - (y_3 - y_1) * (z_2 - z_1)$$

$$\mathbf{B} = (x_3 - x_1) * (z_2 - z_1) - (x_2 - x_1) * (z_3 - z_1)$$

$$\mathbf{C} = (x_2 - x_1) * (y_3 - y_1) - (x_3 - x_1) * (y_2 - y_1)$$

$$\begin{aligned} \mathbf{D} = & x_1 * ((y_3 - y_1) * (z_2 - z_1) - (y_2 - y_1) * (z_3 - z_1)) + \\ & + y_1 * ((x_2 - x_1) * (z_3 - z_1) - (x_3 - x_1) * (z_2 - z_1)) + \\ & + z_1 * ((x_3 - x_1) * (y_2 - y_1) - (x_2 - x_1) * (y_3 - y_1)) \end{aligned}$$

Отже, вектор нормалі $\vec{N} = (A, B, C)$.

Нормальне рівняння площини:

$$x \cos \alpha + y \cos \beta + z \cos \gamma - p = 0$$

$$\begin{aligned}\cos \alpha &= \frac{A}{\pm \sqrt{A^2 + B^2 + C^2}} \\ \cos \beta &= \frac{B}{\pm \sqrt{A^2 + B^2 + C^2}} \\ \cos \gamma &= \frac{C}{\pm \sqrt{A^2 + B^2 + C^2}} \\ p &= \frac{D}{\pm \sqrt{A^2 + B^2 + C^2}}\end{aligned}$$

де
знак множника, що $\frac{1}{\pm \sqrt{A^2 + B^2 + C^2}}$ нормує, береться протилежним знаку D, якщо D=0 знак довільний. Знак перед коренем вибирається так, щоб дотримувалася умова $p > 0$.

- рівняння прямої через дві точки кінцевого елемента, одна з якої вузлова крапка, обрана користувачем:

$$(1^*) \frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}, \text{ где } \vec{R} = (l, m, n) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

$$\text{або } \frac{x - x_1}{\cos \alpha'} = \frac{y - y_1}{\cos \beta'} = \frac{z - z_1}{\cos \gamma'}, \text{ де } \alpha', \beta', \gamma' \text{ - кути утворені прямої з осями координат}$$

OX, OY, OZ. Тоді напрямні косинуси прямої обчислюються:

$$\begin{aligned}\cos \alpha' &= \frac{l}{\sqrt{l^2 + m^2 + n^2}} \\ \cos \beta' &= \frac{m}{\sqrt{l^2 + m^2 + n^2}} \\ \cos \gamma' &= \frac{n}{\sqrt{l^2 + m^2 + n^2}}\end{aligned}$$

- Рівняння прямої через дві точки кінцевого елемента

$$\frac{x - x_1}{x_3 - x_1} = \frac{y - y_1}{y_3 - y_1} = \frac{z - z_1}{z_3 - z_1} \quad (1^{**}), \text{ де } \vec{M} = (l_1, m_1, n_1) = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$$

Знайдемо кут між прямими (1*) і (1**) $\vec{R} = (l, m, n) = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$ і

$$\vec{M} = (l_1, m_1, n_1) = (x_3 - x_1, y_3 - y_1, z_3 - z_1)$$

$$\cos \varepsilon = \frac{l * l_1 + m * m_1 + n * n_1}{\sqrt{l^2 + m^2 + n^2} * \sqrt{l_1^2 + m_1^2 + n_1^2}}$$

$$\varepsilon = \arccos\left(\frac{l * l_1 + m * m_1 + n * n_1}{\sqrt{l^2 + m^2 + n^2} * \sqrt{l_1^2 + m_1^2 + n_1^2}}\right)$$

- Проекція осі ОХ на дану площину (одержати рівняння прямої)

Усяка площина, що проходить через вісь ОХ, виражається рівнянням $B_1y + C_1z = 0$ (2). Ця площина повинна бути перпендикулярна площини (1) з вектором нормалі $\vec{N} = (A, B, C)$. Отже, рівняння (2) буде мати вигляд

$$\frac{-C}{B}y + z = 0 \quad (3), \text{ де вектор нормалі } \vec{N}' = (0, \frac{-C}{B}, 1).$$

Рівняння прямої задане перетинанням двох площин

$$\begin{cases} Ax + By + Cz + D = 0 \\ \frac{-C}{B}y + z = 0 \end{cases}$$

$$\vec{N} = (A, B, C), \quad \vec{N}' = (0, \frac{-C}{B}, 1).$$

$$\vec{S} = \vec{N} \times \vec{N}' = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & \frac{-C}{B} & 1 \\ A & B & C \end{vmatrix}.$$

Рівняння прямої:

$$\frac{x - x'_1}{\begin{vmatrix} \frac{-C}{B} & 1 \\ B & C \end{vmatrix}} = \frac{y - y'_1}{\begin{vmatrix} 1 & 0 \\ C & A \end{vmatrix}} = \frac{z - z'_1}{\begin{vmatrix} 0 & \frac{-C}{B} \\ A & B \end{vmatrix}}; \quad (2^*).$$

$$\frac{x - x'_1}{\frac{-C^2}{B} - B} = \frac{y - y'_1}{A} = \frac{z - z'_1}{\frac{C * A}{B}}$$

Напрямний вектор прямій (4) $\vec{S} = (\frac{-C^2}{B} - B, A, \frac{C * A}{B})$.

- Обмежуємо вибір тієї точки в якій проекція осі ОХ на дану площину пролягає всередині трикутника з вузлових точок.

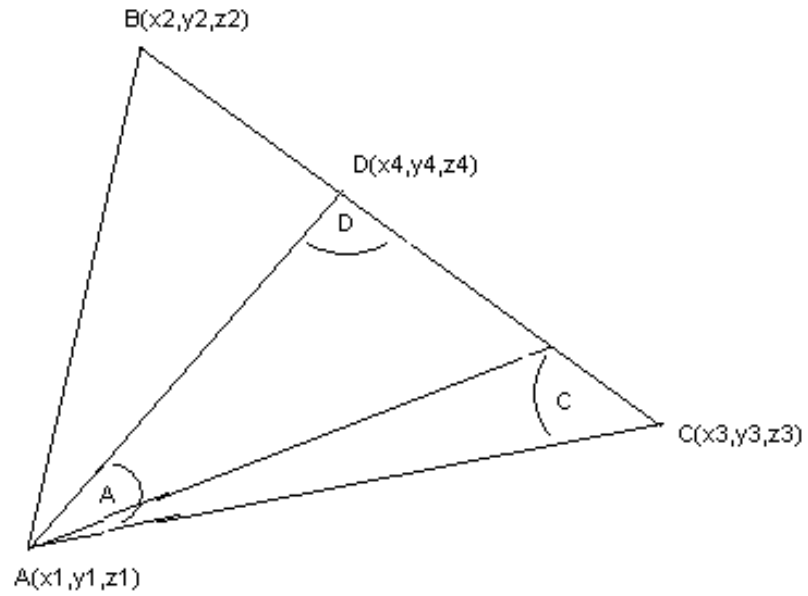


Рис.3.6. Розташування кутів

- З теореми синусів : $\frac{DC}{\sin(a)} = \frac{AC}{\sin(d)}$

$$\angle D = 180^\circ - \angle A - \angle C;$$

$$AC = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2};$$

$$DC = \frac{AC \sin(a)}{\sin(d)}$$

- Знайдемо координати точки D:

$$x_4 = \frac{x_2 - \lambda x_3}{1 + \lambda}; \quad y_4 = \frac{y_2 - \lambda y_3}{1 + \lambda};$$

$$z_4 = \frac{z_2 - \lambda z_3}{1 + \lambda}; \quad \text{де } \lambda = \frac{BD}{DC};$$

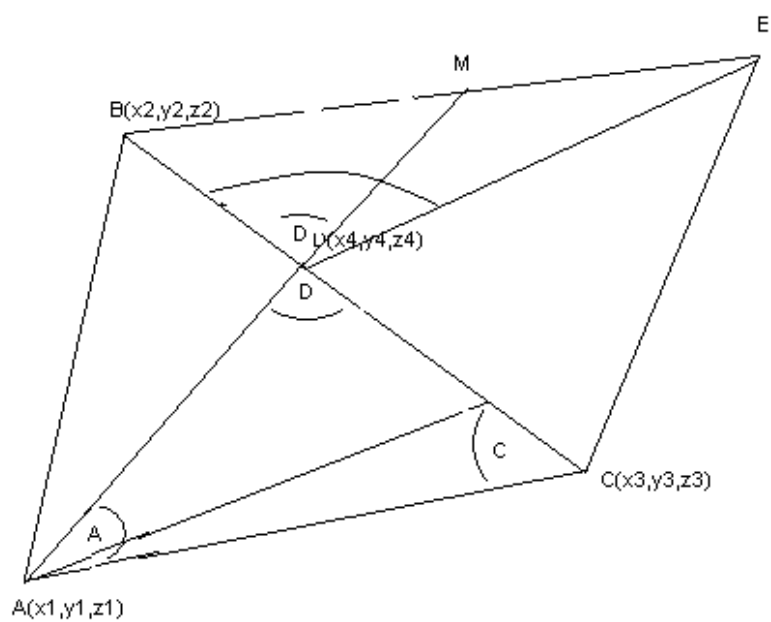


Рис.3.7. Розташування кутів

Якщо $\angle D < \angle BDE$ то точка М належить стороні ВЕ і розрахунок повторюємо , якщо $\angle D > \angle BDE$ то точка М належить стороні ЕС і розрахунок повторюємо.

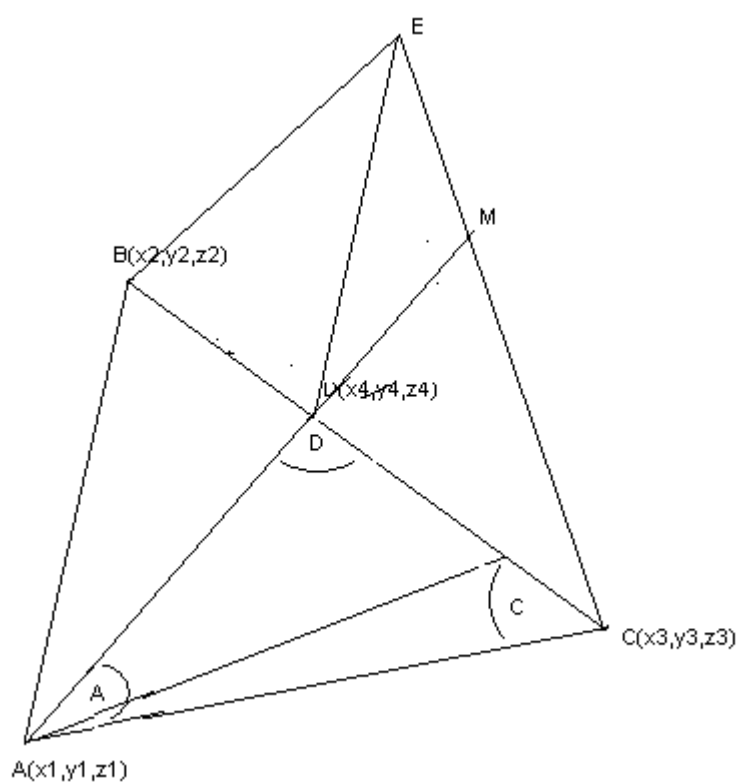


Рис.3.8. Розташування кутів

3.2. Розробка програмного засобу

Вхідний файл для CAD/CAE-системи ANSYS повинен реалізувати два етапи.

Перший етап.

Для реалізації першого кроку – побудови точок – доцільно використовувати спеціальний список програми ANSYS, так званий P51X. Слід відзначити, що порядок роботи з цим списком мало документован. 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*, *ITEMY*, *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.

ITEMY, Y та Z координати.
ITEMZ

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

Другий етап.

Генерація ліній виконується командою **LSTR, P1, P2** [10]:

P1, P2, Список точок, по яким будуватиметься лінія.

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**

Розробка програмного засобу

Для розробки оболонки, що служить генератором вхідного файлу ANSYS автор 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.

Організація інтерфейсу

Після запуску програми з'являється форма проекту в якій треба відкрити два файли ELIST та NLIST, в них містяться данні о координатах вузлових точок кінцевих елементів (файл NLIST) та список вузлів які формують кінцеві елементи (файл ELIST) рис.3.9.

Вибираємо номер кінцевого елементу, вузлову точку та вводимо кут армування. Натискаємо на кнопку «Сформувати макрос» і отримуємо макрос на мові APDL рис.3.10.

чтение данных с файла

чтение с файла Elist.lis

открыть файл Elist

чтение с файла Nlist.lis

открыть файл Nlist

Ввод данных пользователем

Выбери номер конечного элемента

Выбери узловую точку конечного элемента

СФОРМИРОВАТЬ МАКРОС

Рис.3.9.Форма запуску

чтение данных с файла

чтение с файла Elist.lis

открыть файл Elist

чтение с файла Nlist.lis

открыть файл Nlist

Ввод данных пользователем

Выбери номер конечного элемента

3

Выбери узловую точку конечного элемента

22

Введи угол меньше 57 градусов

10

СФОРМИРОВАТЬ МАКРОС

```

/prep7
k,1, 1.37395360940000E+0000, 7.09953030356000E-0001, 7.86495843106000E-0001
k,2, 2.00000000000000E+0000, 2.13762458956083E-0001, 9.71857638384592E-0001
k,3, 1.52413829485332E+0000, 5.65062126211950E-0001, 8.95681474706497E-0001
k,4, 1.69602815180740E+0000, 5.25554359144878E-0001, 9.10286852179141E-0001
k,5, 1.65580711044688E+0000, 3.59766209612420E-0001, 9.89711114441088E-0001
k,6, 1.67352624372209E+0000, 3.27032436045940E-0001, 1.002069394859034E+0000
k,7, 1.81961477010815E+0000, 1.83985289168607E-0001, 1.01749904913055E+0000
k,8, 1.94943096528122E+0000, 2.26785734426177E-0001, 9.83098166143048E-0001
k,9, 2.00000000000000E+0000, 4.54063021708374E-0002, 9.94022146902154E-0001
LSTR,1,2
LSTR,1,3
LSTR,3,4
LSTR,4,5
LSTR,5,6

```

Рис.3.10.Одержані результати

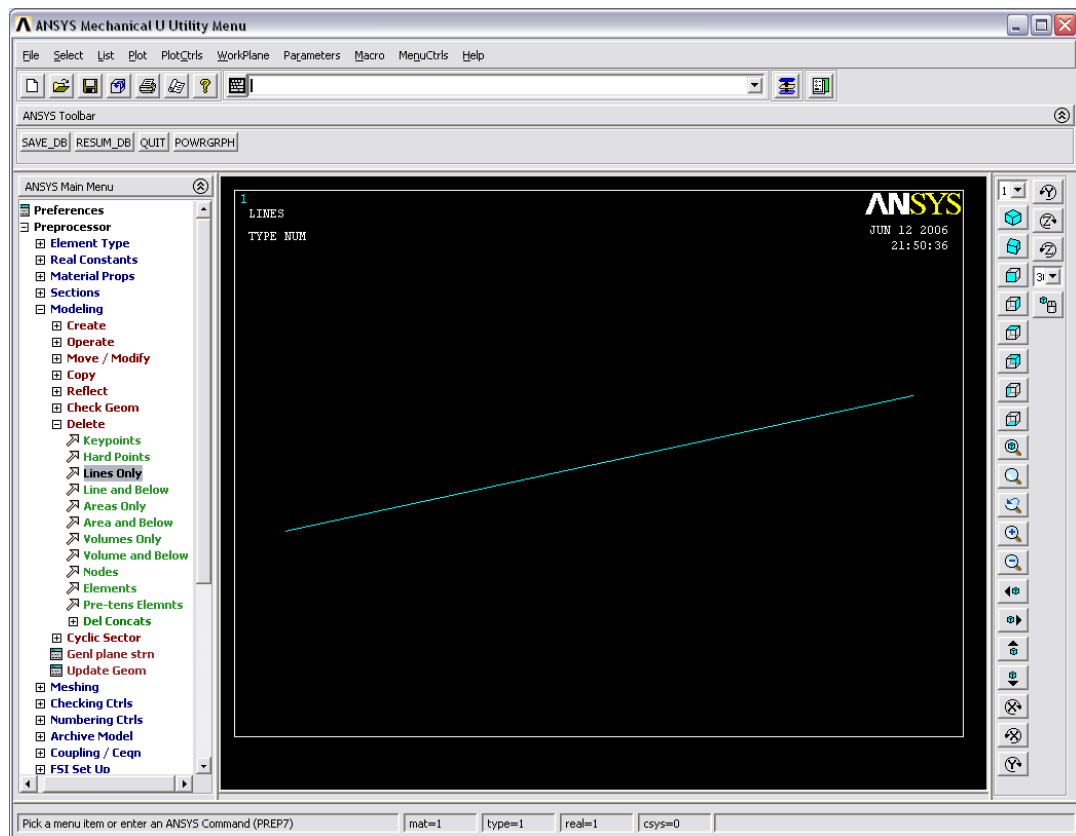


Рис.3.11.Одержані результати в ANSYS

3.3. Висновки

Таким чином, в процесі виконання магістерської роботи було створено програмний засіб, що генерує траєкторію геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі в середовищі ANSYS та забезпечує можливості вибору початку траєкторії з довільного кінцевого елемента. Ця програма формує макрос для програми ANSYS на мові APDL. Запуск цього макросу в ANSYS буде траєкторію геодезичної укладки по даним кінцево – елементної моделі намотаного виробу.

ВИСНОВКИ

У магістерській роботі було досліджено та розроблено модель траєкторії геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі та параметрам процесу намотки у середовищі ANSYS.

В межах завдання створення математичної моделі розрахунку траєкторії укладання армуючого матеріалу дана робота запропонувала вирішення першого кроку – перехід від поверхні другого порядку до поверхні першого порядку що спрощує математичну модель розрахунку траєкторії геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі, та програмний засіб, який генерує траєкторію геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі. За основу взято систему інженерного аналізу ANSYS [11], як програмний комплекс не залежний від програмно-апаратної платформи, який налічує багату бібліотеку можливостей фізико-механічного аналізу, в якого є засоби моделювання багаточисельних КМ.

В результаті виконання роботи були отримані наступні результати:

- 1) виконано огляд практичного застосування властивостей геодезичних ліній в прикладній геометрії поверхонь, способи їх відшукування і побудови та конструювання поверхонь за заданою геодезичною лінією. З'ясовано, що в силу чисельного інтегрування диференціальних рівнянь геодезичні лінії на поверхнях часто будувалися наближено. Дослідження стосовно конструювання лінійчатих поверхонь за заданою геодезичною лінією є обмеженими;
- 2) зроблено аналіз можливостей системи ANSYS з урахуванням особливостей геометрії та кінцево елементного розподілу моделі;
- 3) розроблена математична модель розрахунку траєкторії геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі;

- 4) створено програму – генератор файлу макроса інженерної системи ANSYS для побудови траєкторії геодезичної укладки армуючого матеріалу на оправку виробу по даним кінцево – елементної моделі.
Це – перший етап створення середовища для моделювання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Калинин В.А., Якунин В.И. Геометрическое моделирование технологического процесса намотки в производстве ЛА. - М.: изд-во МАИ, 1995.- 68с.
2. 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.
3. Дубровин Б.А., Новиков С.П., Фоменко А.Т. Современная геометрия. - М.: Наука, 1979.- 719с.
4. Рашевский П.К. Курс дифференциальной геометрии. - М.: Гостехиздат, 1946.- 420с.
5. 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.
6. Буров А.К., Андреевская Г.Д. Синтетические волокнистые анизотропные структуры. М.: Изд-во АН СССР, 1952.
7. Андреевская Г.Д. Высокопрочные ориентированные стеклопластики. М.: Химия, 1966.
8. 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.
9. Donald L., Hunston M. In: Composites update (The Newsletter of the University of Delaver Center for Composite Materials), Fall, 1992.
10. Donald L., Hunston M. In: Composites update (The Newsletter of the University of Delaver Center for Composite Materials), Fall, 1992.
11. Могильний Г.А., Текучев О.Ю.. Особливості моделювання виробів з композиційних матеріалів, що випускаються методом намотування // Вісник СНУ – 2003– №8 – С. 96 – 99.

- 12.ANSYS. Basic Analysis Procedures Guide. Rel. 5.7. / ANSYS Inc. Houston, 2000.
- 13.ANSYS. Commands Reference. Rel. 5.7. / ANSYS Inc. Houston, 2000.
- 14.ANSYS. Elements Reference. Rel. 5.7. / ANSYS Inc. Houston, 2000.
- 15.ANSYS. Theory Reference. Rel. 5.7. Ed. P. Kothnke / ANSYS Inc. Houston, 2000.
- 16.ANSYS. Verification Manual. Rel. 5.7. / ANSYS Inc. Houston, 2000.
- 17.Росато Д.В., Грове К.С. Намотка стеклонитью: Пер. с англ. – М.: Машиностроение, 1969. – 310 с.
- 18.Калинчев В.А., Макаров М.С. Намоточные стеклопластики. – М.: Химия, 1986. – 272 с.
- 19.Шукшунов В.Е., Жуковский В.Г., Евченко А.И. и др. Автоматизированные системы управления намоточными станками. – М.: Машиностроение, 1985. – 208 с.
- 20.Забашта В.Ф., Кривов Г.О., Бондарь В.Г. Полимерные композиционные материалы конструкционного назначения. – К.: Техника, 1993 – 160 с.
- 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.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.
- 23.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.
- 24.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.

25. Любутин О.С. Автоматизация производства стеклопластиков. – М.: Химия, 1969. – 256 с.
26. 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.
27. Рыбников С.И. Автоматическое управление намоткой. – М.: Энергия, 1972. – 112 с.
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. Ashok K. Munjal. Optimization of design allowables for composite structures // SAMPE Quart., 1987. – 18. – No2. – p.18-27.
30. Hollingsworth R.D., Osment D.R. Filament wound thermoplastic matrix pressure vessels // International SAMPE Symposium. – 1987, April 6-9. – p.662-669.
31. 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.
32. 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.
33. 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.
34. Гуняев Г.М. Проектирование высококомодульных полимерных композитов с заданными свойствами // Композиционные материалы – М.: Машиностроение, 1981. – С. 24-28.

35. 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).
36. Hayashi R. Future of composite material assessment techniques // Frans. JSCM.- V. 11.- № 2.- P. 41-44.
37. 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.
38. 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.
39. 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.
40. 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.
41. 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).
42. 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.

- 43.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.
- 44.Кривелли-Висконти И. Конструирование деталей из композиционных материалов // Достижения в области композиционных материалов.- М., 1982.- С. 58-69.
- 45.Гуняев Г.М. Проектирование высокомодульных полимерных композитов с заданными свойствами // Композиционные материалы.- М., 1981.- С. 24-28.
- 46.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.
- 47.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.
- 48.Брайан Р. Нотон. О некоторых трудностях при использовании новых материалов // Композиционные материалы. Т. 3. Применение композиционных материалов в технике. - М., 1978.-С. 491-495.
- 49.Перегудов Ф.И., Тарасенко Ф.П. Введение в системный анализ.- М.: Высш. шк., 1989.- 367 с.
- 50.Корб Л. Дж. Космические летательные аппараты // Композиционные материалы. Т. 3. Применение композиционных материалов в технике. - М., 1978.- С. 78-129.
- 51.Рач В.А. Создание корпусов малогабаритных РДТТ одноразового и кратковременного действия из армированных пластиков: Автореф. дис....д-ра техн. наук.- Харьков,1992.- 42с.
- 52.Рач В.А. Инновационная деятельность: системные аспекты// Вісник Східноукраїнського державного університету.-1997.- №2(6).- С.120-127.

ДОДАТОК А

Лістинг коду.

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Math, StdCtrls;
```

```
type
```

```
Node=record
```

```
    Nomer:word;
```

```
    x,y,z:real;
```

```
end;
```

```
t2=array of node;
```

```
Tohka=record
```

```
    nomer:word;
```

```
    Node1,Node2,Node3:word;
```

```
end;
```

```
Node1=record
```

```
    a,b,c,d:real;
```

```
end;
```

```
Node3=record
```

```
    l,m,n:real;
```

```
end;
```

```
Node5=record
```

```
    x1,y1,z1:real;
```

```
end;
```

```
TForm1 = class(TForm)
```

```
    GroupBox1: TGroupBox;
```

```
    GroupBox2: TGroupBox;
```

```
    Button1: TButton;
```

```
    Button2: TButton;
```

```
    Label1: TLabel;
```

```
    Label2: TLabel;
```

```
    Label3: TLabel;
```

```
    ComboBox1: TComboBox;
```

```
    ComboBox2: TComboBox;
```

```
    Edit1: TEdit;
```

```
    Label4: TLabel;
```

```
    Label5: TLabel;
```

```
    Button3: TButton;
```

```
    OpenFileDialog: TOpenDialog;
```

```

SaveDialog1: TSaveDialog;
Memo1: TMemo;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure ComboBox2Change(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;

implementation
  var
    f           :text;
    mass,Arr,mas1,ccc,yu      :t2;
    e           :string;
    a,b         :set of byte;
    nom_ke,nomer_uzl,i,j,uu1,ttt,uuu   :integer;
    ugol_arm    :string;
    cry         :array of tohka;

{$R *.dfm}
procedure file_lok (filename:string);
var
  s:string;
  k,n:integer;
  x,y,z:real;
begin
  AssignFile(f,filename);
  Reset(f);
  for k:=1 to 1 do
    Readln(f,s);
    SetLength(arr,1);
    i:=0;
  while not eof(f) do
    begin
      SetLength(arr,i+1);
      Read(f,arr[i].nomer);
      Read(f,arr[i].x,arr[i].y,arr[i].z);
    end;
  end;
end;

```

```

    i:=i+1;
end;
    CloseFile(f);
end;
procedure file_tohka (filename:string);
var
    s:string;
    k,n:integer;
    Node1, Node2, Node3:integer;
begin
    AssignFile(f,filename);
    Reset(f);
    for k:=1 to 1 do
        Readln(f,s);
        i:=0;
    while not eof(f) do
        begin
            SetLength(qry,i+1);
            Read(f,qry[i].nomer);
            for k:=1 to 5 do
                Read(f,n);
                Read(f,qry[i].Node1,qry[i].Node2,qry[i].Node3);
                read(f,n);
                i:=i+1;
            end ;
            CloseFile(f);
        end;
    end;

```

```

procedure mass_abcd ( var a_1:node; b_1:node; c_1:node;var d_1:node1);
    begin
        d_1.a:=(b_1.y-a_1.y)*(c_1.z-a_1.z)-(c_1.y-a_1.y)*(b_1.z-a_1.z);
        d_1.b:=- (b_1.x-a_1.x)*(c_1.z-a_1.z)+(c_1.x-a_1.x)*(b_1.z-a_1.z);
        d_1.c:=(b_1.x-a_1.x)*(c_1.y-a_1.y)-(c_1.x-a_1.x)*(b_1.y-a_1.y);
        d_1.d:=d_1.a*a_1.x+d_1.b*a_1.y+d_1.c*a_1.z;
    end;

```

```

procedure mas_lmn (var c_1:node; c_2:node; var c_3:node3);
    begin
        c_3.l:=c_2.x-c_1.x;
        c_3.m:=c_2.y-c_1.y;
        c_3.n:=c_2.z-c_1.z;
    end;
procedure poisk_ke (var l1,l2:integer;var r2:integer);
    var

```

```

yy1:integer;
begin
  b:=[];
  a:=[];
  a:=[l1,l2];
  yy1:=-1;
  for i:=low(qry) to high(qry)-1 do
    begin
      if i<>nom_ke then
        begin
          include(b,qry[i].Node1);
          include(b,qry[i].Node2);
          include(b,qry[i].Node3);
          if a<=b
            then yy1:=i;
          end;
        b:=[];
        end;
      r2:=yy1;
    end;

    ///

    procedure poisk_th (var l11,l22:integer;var r2:integer);
  var
  ty, yy1,yu:integer;
  begin
    b:=[];
    a:=[];
    a:=[l11,l22];
    yy1:=-1;
    for i:=low(qry) to high(qry)-1 do
      begin
        if i<>nom_ke then
          begin
            include(b,qry[i].Node1);
            include(b,qry[i].Node2);
            include(b,qry[i].Node3);
            if a<=b
              then
                begin
                  yy1:=i;
                  for j:=1 to 200 do
                    if (j in b)and (j<>l11)and(j<>l22)then ty:=j;
                  end;
                end;
            end;
          end;
        end;
      end;
    end;
  end;

```

```

    //
    end;
    b:=[];
    end;
    yu:=yy1;
    if yu<>-1 then r2:=ty
    else
    r2:=-1;
    end;
    ///
procedure ras_tohk(var a_1:node; b_1:node; var d:real);
begin
    d:=sqrt(sqr(b_1.x-a_1.x)+sqr(b_1.y-a_1.y)+sqr(b_1.z-a_1.z));
end;
procedure toh_pr( var b_1:node; c_1:node;d1,d2,q1,q2:real;var
eww:node5);
var
    d3,t1,t2:real;
begin
    d3:=d1*sin(q1)/sin(q2);
    t1:=d2-d3;
    t2:=t1/d3;
    eww.x1:=(c_1.x+t2*b_1.x)/(1+t2);
    eww.y1:=(c_1.y+t2*b_1.y)/(1+t2);
    eww.z1:=(c_1.z+t2*b_1.z)/(1+t2);
end;

function ugol_pr (var c_11:node3; c_12:node3):real;
var
    e1:real;
begin
    e1:=ArcCos((c_11.l*c_12.l+c_11.m*c_12.m+c_11.n*c_12.n)/(sqrt(sqr(c_11.l)+sqr
(c_11.m)+sqr(c_11.n))*sqrt(sqr(c_12.l)+sqr(c_12.m)+sqr(c_12.n))));
    ugol_pr:=e1;
end;
    ///
procedure poisk_ke_kolih(var vb:integer; var ui:integer);
var
    ty:integer;

begin
    ty:=0;
    for i:=low(qry) to high(qry)-1 do

```



```

    if (vb=qry[i].Node1)or (vb=qry[i].Node2) or (vb=qry[i].Node3) then ty:=ty+1;

    ui:=ty;
    end;
    ///
procedure for_mass (var e_1:t2);
    var

    k_g,k_h,g_h,ck,ck1,ck2,ck3,ck4          :node3;
    eww                                     :node5;
    ugol,ugol_gkh,ugol_ghk,d1,d2,ugol_pr1,w,w1,w2,d3,d4,ugol3 :real;
    uzel,k,g,h,u,uu,uz1,uz2                :integer;
    t3:byte;
    begin

        ugol:=strtofloat(ugol_arm)*pi/180;
        SetLength(mass,2);
        mass[0].Nomer:=1;
        mass[0].x:= arr[nomer_uzl-1].x;
        mass[0].y:= arr[nomer_uzl-1].y;
        mass[0].z:= arr[nomer_uzl-1].z;
        if qry[nom_ke].Node1=nomer_uzl then
        begin
            k:=qry[nom_ke].Node1;
            g:= qry[nom_ke].Node2;
            h:=qry[nom_ke].Node3;
        end
        else
        if qry[nom_ke].Node2=nomer_uzl then
        begin
            k:=qry[nom_ke].Node2;
            g:=qry[nom_ke].Node1;
            h:=qry[nom_ke].Node3;
        end
        else
        begin
            k:=qry[nom_ke].Node3;
            g:=qry[nom_ke].Node2;
            h:=qry[nom_ke].Node1;
        end;

        mas_lmn (arr[k-1],arr[g-1],k_g);
        mas_lmn (arr[k-1],arr[h-1],k_h);

```

```

ugol_gkh:=trunc(ugol_pr(k_g,k_h)*180/pi)*pi/180;

mas_lmn (arr[g-1],arr[h-1],g_h);
ugol_ghk:=trunc(ugol_pr(g_h,k_h)*180/pi)*pi/180;
ras_tohk(arr[k-1],arr[h-1],d1);
ras_tohk(arr[h-1],arr[g-1],d2);
ugol_pr1:=pi-ugol-ugol_ghk;
toh_pr(arr[g-1],arr[h-1], d1,d2,ugol ,ugol_pr1,eww);

mass[1].Nomer:=2;
mass[1].x:=eww.x1;
mass[1].y:=eww.y1;
mass[1].z:=eww.z1;
poisk_ke(g,h,uzel);
poisk_th(g,h,u);
t3:=2;
while uzel<>-1 do
begin

/////
mas_lmn (arr[h-1],arr[g-1],ck);
mas_lmn (arr[h-1],arr[u-1],ck1);
w:=trunc(ugol_pr(ck,ck1)*180/pi)*pi/180;
mas_lmn (arr[u-1],arr[g-1],ck2);
w1:=trunc(ugol_pr(ck2,ck)*180/pi)*pi/180;
///
mas_lmn (mass[t3-1],arr[u-1],ck3);
mas_lmn (mass[t3-1],arr[h-1],ck4);
w2:=trunc(ugol_pr(ck3,ck4)*180/pi)*pi/180;
/////
if w2>ugol_pr1 then
begin
ugol3:=pi-w-ugol_pr1;
ras_tohk(arr[h-1],arr[u-1],d3);
ras_tohk(arr[h-1],mass[t3-1],d4);
toh_pr(arr[u-1],arr[h-1], d4,d3,ugol_pr1,ugol3,eww);
SetLength(mass,t3+1);
mass[t3].Nomer:=t3+1;
mass[t3].x:=eww.x1;
mass[t3].y:=eww.y1;
mass[t3].z:=eww.z1;
g:=u;
h:=h;
ugol_pr1:= ugol3;
end

```

```

else
begin
    ugol_pr1:=pi-ugol_pr1;
    ugol3:=pi-w1-ugol_pr1;
    ras_tohk(arr[g-1],arr[u-1],d3);
    ras_tohk(arr[g-1],mass[t3-1],d4);
    toh_pr(arr[u-1],arr[g-1], d4,d3,ugol_pr1,ugol3,eww);
    SetLength(mass,t3+1);
    mass[t3].Nomer:=t3+1;
    mass[t3].x:=eww.x1;
    mass[t3].y:=eww.y1;
    mass[t3].z:=eww.z1;
    g:=g;
    h:=u;
    ugol_pr1:= ugol3;
end;
t3:=t3+1;
nom_ke:=uzel;
poisk_ke(g,h,uzel);
poisk_th(g,h,u);
end;
setlength(e_1,t3);
e_1:=copy(mass);
end;
////////////////////////////////////

```

```

procedure tr_prob(var e2:t2);
var
    k_g,k_h,g_h,ck,ck1,ck2,ck3,ck4      :node3;
    eww                                  :node5;
    k,g,h,uu,uzel1,t3,u                  :integer;
    ugol1,ugol_gkh,ugol_ghk,d1,d2,ugol_pr1,w,w1,w2,ugol3,d3,d4 :real;
begin
    poisk_ke_kolih( nomer_uzl,uu);
    if uu>3 then
    begin
        ugol1:=strtofloat(ugol_arm)*pi/180;

        if qry[nom_ke].Node1=nomer_uzl then
begin
    k:=qry[nom_ke].Node1;
    g:= qry[nom_ke].Node2;
    h:=qry[nom_ke].Node3;
end
else

```

```

if qry[nom_ke].Node2=nomer_uzl then
begin
k:=qry[nom_ke].Node2;
g:=qry[nom_ke].Node1;
h:=qry[nom_ke].Node3;
end
else
begin
k:=qry[nom_ke].Node3;
g:=qry[nom_ke].Node2;
h:=qry[nom_ke].Node1;
end;

end;
if uu=5 then

begin

mas_lmn (arr[k-1],arr[g-1],k_g);
mas_lmn (arr[k-1],arr[h-1],k_h);
ugol_gkh:=ugol_pr(k_g,k_h);
if ugol_gkh/2 < ugol1 then
begin

poisk_ke(k,g,uzel1);
poisk_th(k,g,u);
g:=u;
nom_ke:=uzel1;
poisk_th(k,g,u);
h:=u;
mas_lmn (arr[k-1],arr[g-1],k_g);
mas_lmn (arr[k-1],arr[h-1],k_h);
ugol_gkh:=trunc(ugol_pr(k_g,k_h)*180/pi)*pi/180;

mas_lmn (arr[g-1],arr[h-1],g_h);
ugol_ghk:=trunc(ugol_pr(g_h,k_h)*180/pi)*pi/180;
ras_tohk(arr[k-1],arr[h-1],d1);
ras_tohk(arr[h-1],arr[g-1],d2);
ugol_pr1:=pi-ugol1-ugol_ghk;
toh_pr(arr[g-1],arr[h-1], d1,d2,ugol1 ,ugol_pr1,eww);
SetLength(mass1,2);
mass1[0].Nomer:=1;
mass1[0].x:=eww.x1;

```

```

mass1[0].y:=eww.y1;
mass1[0].z:=eww.z1;
nom_ke:=uzel1;
poisk_ke(g,h,uzel1);
poisk_th(g,h,u);
t3:=1;
while uzel1<>-1 do
begin

/////
mas_lmn (arr[h-1],arr[g-1],ck);
mas_lmn (arr[h-1],arr[u-1],ck1);
w:=trunc(ugol_pr(ck,ck1)*180/pi)*pi/180;
mas_lmn (arr[u-1],arr[g-1],ck2);
w1:=trunc(ugol_pr(ck2,ck)*180/pi)*pi/180;
///
mas_lmn (mass1[t3-1],arr[u-1],ck3);
mas_lmn (mass1[t3-1],arr[h-1],ck4);
w2:=trunc(ugol_pr(ck3,ck4)*180/pi)*pi/180;
/////
if w2>ugol_pr1 then
begin
ugol3:=pi-w-ugol_pr1;
ras_tohk(arr[h-1],arr[u-1],d3);
ras_tohk(arr[h-1],mass1[t3-1],d4);
toh_pr(arr[u-1],arr[h-1], d4,d3,ugol_pr1,ugol3,eww);
SetLength(mass1,t3+1);
mass1[t3].Nomer:=t3+1;
mass1[t3].x:=eww.x1;
mass1[t3].y:=eww.y1;
mass1[t3].z:=eww.z1;
g:=u;
h:=h;
ugol_pr1:= ugol3;
end
else
begin
ugol_pr1:=pi-ugol_pr1;
ugol3:=pi-w1-ugol_pr1;
ras_tohk(arr[g-1],arr[u-1],d3);
ras_tohk(arr[g-1],mass1[t3-1],d4);
toh_pr(arr[u-1],arr[g-1], d4,d3,ugol_pr1,ugol3,eww);
SetLength(mass1,t3+1);
mass1[t3].Nomer:=t3+1;
mass1[t3].x:=eww.x1;

```

```

        mass1[t3].y:=eww.y1;
        mass1[t3].z:=eww.z1;
        g:=g;
        h:=u;
        ugol_pr1:= ugol3;
    end;
    t3:=t3+1;
    nom_ke:=uzel1;
    poisk_ke(g,h,uzel1);
    poisk_th(g,h,u);
end;
end

else

    poisk_ke(k,h,uzel1);
poisk_th(k,h,u);
h:=u;
    nom_ke:=uzel1;
poisk_th(k,h,u);
g:=u;
    mas_lmn (arr[k-1],arr[g-1],k_g);
    mas_lmn (arr[k-1],arr[h-1],k_h);
    ugol_gkh:=trunc(ugol_pr(k_g,k_h)*180/pi)*pi/180;

    mas_lmn (arr[g-1],arr[h-1],g_h);
    ugol_ghk:=trunc(ugol_pr(g_h,k_h)*180/pi)*pi/180;
    ras_tohk(arr[k-1],arr[h-1],d1);
    ras_tohk(arr[h-1],arr[g-1],d2);
    ugol_pr1:=pi-ugol1-ugol_ghk;
    toh_pr(arr[g-1],arr[h-1], d1,d2,ugol1 ,ugol_pr1,eww);
    SetLength(mass1,2);
    mass1[0].Nomer:=1;
mass1[0].x:=eww.x1;
mass1[0].y:=eww.y1;
mass1[0].z:=eww.z1;
    nom_ke:=uzel1;
    poisk_ke(g,h,uzel1);
    poisk_th(g,h,u);
t3:=1;
    while uzel1<>-1 do
begin

        /////

```

```

mas_lmn (arr[h-1],arr[g-1],ck);
mas_lmn (arr[h-1],arr[u-1],ck1);
w:=trunc(ugol_pr(ck,ck1)*180/pi)*pi/180;
mas_lmn (arr[u-1],arr[g-1],ck2);
w1:=trunc(ugol_pr(ck2,ck)*180/pi)*pi/180;
///
mas_lmn (mass1[t3-1],arr[u-1],ck3);
mas_lmn (mass1[t3-1],arr[h-1],ck4);
w2:=trunc(ugol_pr(ck3,ck4)*180/pi)*pi/180;
/////
if w2>ugol_pr1 then
begin
  ugol3:=pi-w-ugol_pr1;
  ras_tohk(arr[h-1],arr[u-1],d3);
  ras_tohk(arr[h-1],mass1[t3-1],d4);
  toh_pr(arr[u-1],arr[h-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=u;
  h:=h;
  ugol_pr1:= ugol3;
end
else
begin
  ugol_pr1:=pi-ugol_pr1;
  ugol3:=pi-w1-ugol_pr1;
  ras_tohk(arr[g-1],arr[u-1],d3);
  ras_tohk(arr[g-1],mass1[t3-1],d4);
  toh_pr(arr[u-1],arr[g-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=g;
  h:=u;
  ugol_pr1:= ugol3;
end;
t3:=t3+1;
nom_ke:=uzel1;
poisk_ke(g,h,uzel1);
poisk_th(g,h,u);

```

```

end;

    end;
    if uu=6 then
        begin
            poisk_ke(k,g,uzel1);
            poisk_th(k,g,u);
            g:=u;
            nom_ke:=uzel1;
            poisk_th(k,g,u);
            g:=u;
            poisk_ke(k,g,uzel1);
            nom_ke:=uzel1;
            poisk_th(k,g,u);
h:=u;
            mas_lmn (arr[k-1],arr[g-1],k_g);
            mas_lmn (arr[k-1],arr[h-1],k_h);
            ugol_gkh:=trunc(ugol_pr(k_g,k_h)*180/pi)*pi/180;

            mas_lmn (arr[g-1],arr[h-1],g_h);
            ugol_ghk:=trunc(ugol_pr(g_h,k_h)*180/pi)*pi/180;
            ras_tohk(arr[k-1],arr[h-1],d1);
            ras_tohk(arr[h-1],arr[g-1],d2);
            ugol_pr1:=pi-ugol1-ugol_ghk;
            toh_pr(arr[g-1],arr[h-1], d1,d2,ugol1 ,ugol_pr1,eww);
            SetLength(mass1,2);
            mass1[0].Nomer:=1;
            mass1[0].x:=eww.x1;
            mass1[0].y:=eww.y1;
            mass1[0].z:=eww.z1;
            nom_ke:=uzel1;
            poisk_ke(g,h,uzel1);
            poisk_th(g,h,u);
            t3:=1;
            while uzel1<>-1 do
                begin

                    /////
                    mas_lmn (arr[h-1],arr[g-1],ck);
                    mas_lmn (arr[h-1],arr[u-1],ck1);
                    w:=trunc(ugol_pr(ck,ck1)*180/pi)*pi/180;
                    mas_lmn (arr[u-1],arr[g-1],ck2);
                    w1:=trunc(ugol_pr(ck2,ck)*180/pi)*pi/180;
                    ///
                    mas_lmn (mass1[t3-1],arr[u-1],ck3);

```



```

mas_lmn (mass1[t3-1],arr[h-1],ck4);
w2:=trunc(ugol_pr(ck3,ck4)*180/pi)*pi/180;
/////
if w2>ugol_pr1 then
begin
  ugol3:=pi-w-ugol_pr1;
  ras_tohk(arr[h-1],arr[u-1],d3);
  ras_tohk(arr[h-1],mass1[t3-1],d4);
  toh_pr(arr[u-1],arr[h-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=u;
  h:=h;
  ugol_pr1:= ugol3;
end
else
begin
  ugol_pr1:=pi-ugol_pr1;
  ugol3:=pi-w1-ugol_pr1;
  ras_tohk(arr[g-1],arr[u-1],d3);
  ras_tohk(arr[g-1],mass1[t3-1],d4);
  toh_pr(arr[u-1],arr[g-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=g;
  h:=u;
  ugol_pr1:= ugol3;
end;
t3:=t3+1;
nom_ke:=uzel1;
poisk_ke(g,h,uzel1);
poisk_th(g,h,u);
end;
end;
if uu=7 then
begin

```

```

mas_lmn (arr[k-1],arr[g-1],k_g);

```

```

mas_lmn (arr[k-1],arr[h-1],k_h);
ugol_gkh:=ugol_pr(k_g,k_h);
  if ugol_gkh/2 < ugol1 then
    begin
      poisk_ke(k,g,uzel1);
      poisk_th(k,g,u);
      g:=u;
      nom_ke:=uzel1;
      poisk_th(k,g,u);
      g:=u;
      poisk_ke(k,g,uzel1);
      nom_ke:=uzel1;
      poisk_th(k,g,u);
h:=u;

mas_lmn (arr[k-1],arr[g-1],k_g);
mas_lmn (arr[k-1],arr[h-1],k_h);
ugol_gkh:=trunc(ugol_pr(k_g,k_h)*180/pi)*pi/180;

mas_lmn (arr[g-1],arr[h-1],g_h);
ugol_ghk:=trunc(ugol_pr(g_h,k_h)*180/pi)*pi/180;
ras_tohk(arr[k-1],arr[h-1],d1);
ras_tohk(arr[h-1],arr[g-1],d2);
ugol_pr1:=pi-ugol1-ugol_ghk;
toh_pr(arr[g-1],arr[h-1], d1,d2,ugol1 ,ugol_pr1,eww);
  SetLength(mass1,2);
  mass1[0].Nomer:=1;
mass1[0].x:=eww.x1;
mass1[0].y:=eww.y1;
mass1[0].z:=eww.z1;
  nom_ke:=uzel1;
  poisk_ke(g,h,uzel1);
  poisk_th(g,h,u);
t3:=1;
  while uzel1<>-1 do
    begin

      /////
      mas_lmn (arr[h-1],arr[g-1],ck);
      mas_lmn (arr[h-1],arr[u-1],ck1);
      w:=trunc(ugol_pr(ck,ck1)*180/pi)*pi/180;
      mas_lmn (arr[u-1],arr[g-1],ck2);
      w1:=trunc(ugol_pr(ck2,ck)*180/pi)*pi/180;
      ///
      mas_lmn (mass1[t3-1],arr[u-1],ck3);

```

```

mas_lmn (mass1[t3-1],arr[h-1],ck4);
w2:=trunc(ugol_pr(ck3,ck4)*180/pi)*pi/180;
/////
if w2>ugol_pr1 then
begin
  ugol3:=pi-w-ugol_pr1;
  ras_tohk(arr[h-1],arr[u-1],d3);
  ras_tohk(arr[h-1],mass1[t3-1],d4);
  toh_pr(arr[u-1],arr[h-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=u;
  h:=h;
  ugol_pr1:= ugol3;
end
else
begin
  ugol_pr1:=pi-ugol_pr1;
  ugol3:=pi-w1-ugol_pr1;
  ras_tohk(arr[g-1],arr[u-1],d3);
  ras_tohk(arr[g-1],mass1[t3-1],d4);
  toh_pr(arr[u-1],arr[g-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=g;
  h:=u;
  ugol_pr1:= ugol3;
end;
t3:=t3+1;
nom_ke:=uzel1;
poisk_ke(g,h,uzel1);
poisk_th(g,h,u);
end;
end

else
  poisk_ke(k,h,uzel1);
  poisk_th(k,h,u);

```

```

h:=u;
nom_ke:=uzel1;
poisk_th(k,h,u);
h:=u;
poisk_ke(k,h,uzel1);
nom_ke:=uzel1;
poisk_th(k,h,u);
g:=u;

mas_lmn (arr[k-1],arr[g-1],k_g);
mas_lmn (arr[k-1],arr[h-1],k_h);
ugol_gkh:=trunc(ugol_pr(k_g,k_h)*180/pi)*pi/180;

mas_lmn (arr[g-1],arr[h-1],g_h);
ugol_ghk:=trunc(ugol_pr(g_h,k_h)*180/pi)*pi/180;
ras_tohk(arr[k-1],arr[h-1],d1);
ras_tohk(arr[h-1],arr[g-1],d2);
ugol_pr1:=pi-ugol1-ugol_ghk;
toh_pr(arr[g-1],arr[h-1], d1,d2,ugol1 ,ugol_pr1,eww);
SetLength(mass1,2);
mass1[0].Nomer:=1;
mass1[0].x:=eww.x1;
mass1[0].y:=eww.y1;
mass1[0].z:=eww.z1;
nom_ke:=uzel1;
poisk_ke(g,h,uzel1);
poisk_th(g,h,u);
t3:=1;
while uzel1<>-1 do
begin

/////
mas_lmn (arr[h-1],arr[g-1],ck);
mas_lmn (arr[h-1],arr[u-1],ck1);
w:=trunc(ugol_pr(ck,ck1)*180/pi)*pi/180;
mas_lmn (arr[u-1],arr[g-1],ck2);
w1:=trunc(ugol_pr(ck2,ck)*180/pi)*pi/180;
///
mas_lmn (mass1[t3-1],arr[u-1],ck3);
mas_lmn (mass1[t3-1],arr[h-1],ck4);
w2:=trunc(ugol_pr(ck3,ck4)*180/pi)*pi/180;
/////
if w2>ugol_pr1 then
begin
ugol3:=pi-w-ugol_pr1;

```

```

ras_tohk(arr[h-1],arr[u-1],d3);
ras_tohk(arr[h-1],mass1[t3-1],d4);
toh_pr(arr[u-1],arr[h-1], d4,d3,ugol_pr1,ugol3,eww);
  SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=u;
  h:=h;
  ugol_pr1:= ugol3;
end
else
begin
  ugol_pr1:=pi-ugol_pr1;
  ugol3:=pi-w1-ugol_pr1;
  ras_tohk(arr[g-1],arr[u-1],d3);
  ras_tohk(arr[g-1],mass1[t3-1],d4);
toh_pr(arr[u-1],arr[g-1], d4,d3,ugol_pr1,ugol3,eww);
SetLength(mass1,t3+1);
  mass1[t3].Nomer:=t3+1;
  mass1[t3].x:=eww.x1;
  mass1[t3].y:=eww.y1;
  mass1[t3].z:=eww.z1;
  g:=g;
  h:=u;
  ugol_pr1:= ugol3;
end;
  t3:=t3+1;
  nom_ke:=uzel1;
poisk_ke(g,h,uzel1);
poisk_th(g,h,u);
end;

end;
setlength(e2,t3);
e2:=copy(mass1);
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
  if OpenFileDialog1.Execute then
    file_tohka(OpenDialog1.FileName);
    for i:=low(qry) to high(qry)-1 do
      combobox1.Items.add(inttostr(qry[i].nomer));
end;

```

procedure TForm1.Button2Click(Sender: TObject);

```
begin
  if OpenFileDialog1.Execute then
    file_lok(OpenDialog1.FileName);
end;
```

procedure TForm1.ComboBox1Change(Sender: TObject);

```
begin
  combobox2.Items.Clear;
  combobox2.ItemIndex:=1;
  combobox2.items.Add(inttostr(qry[strtoint(combobox1.Text)-1].Node1));
  combobox2.items.Add(inttostr(qry[strtoint(combobox1.Text)-1].Node2));
  combobox2.items.Add(inttostr(qry[strtoint(combobox1.Text)-1].Node3));
  nom_ke:=strtoint(combobox1.Items[ComboBox1.ItemIndex])-1;
```

end;

procedure TForm1.ComboBox2Change(Sender: TObject);

```
var
  ugol_gkh,f1 :real;
  k_g,k_h     :node3;
  k,g,h       :integer;
begin
  nomer_uzl:=strtoint(combobox2.Items[ComboBox2.ItemIndex]);
  if qry[nom_ke].Node1=nomer_uzl then
    begin
      k:=qry[nom_ke].Node1;
      g:= qry[nom_ke].Node2;
      h:=qry[nom_ke].Node3;
    end
  else
    if qry[nom_ke].Node2=nomer_uzl then
      begin
        k:=qry[nom_ke].Node2;
        g:=qry[nom_ke].Node1;
        h:=qry[nom_ke].Node3;
      end
    else
      begin
        k:=qry[nom_ke].Node3;
        g:=qry[nom_ke].Node2;
        h:=qry[nom_ke].Node1;
      end
    end;
  mas_lmn (arr[k-1],arr[g-1],k_g);
```

```

    mas_lmn (arr[k-1],arr[h-1],k_h);
    ugol_gkh:=ugol_pr(k_g,k_h);
    fl:= trunc((ugol_gkh)*180/pi);
    label4.Caption:='Введи угол меньше'+ floattostr(fl)+'градусов';
end;

```

```

procedure TForm1.Edit1Change(Sender: TObject);
begin
    ugol_arm:=edit1.text;
end;

```

```

procedure TForm1.Button3Click(Sender: TObject);
var Pi,i,j:integer;
f1:TextFile;
begin
    ttt:=length(mass);
    uuu:= length(mass1);
    setlength(ccc,ttt);
    tr_prob(ccc);
    setlength(yu,uuu);
    for_mass(yu);
    if SaveDialog1.Execute then
        begin
            AssignFile(f1,SaveDialog1.FileName);
            rewrite(f1);
            Pi:=Length(mass);
            Writeln(f1,'/prep7');
            for i:=0 to Pi-1 do
                writeln(f1,'k,',mass[i].Nomer,',',Mass[i].x,',',mass[i].y,',',mass[i].z,',');
            for i:=0 to Length(mass1)-1 do
                writeln(f1,'k,',mass1[i].Nomer+Pi,',',Mass1[i].x,',',mass1[i].y,',',mass1[i].z,',');
            for i:=1 to Pi-1 do
                writeln(f1,'LSTR,',mass[i-1].Nomer,',',mass[i].nomer);
                writeln(f1,'LSTR,',mass[0].Nomer,',',mass1[0].nomer+pi);
            for i:=1 to Length(mass1)-1 do
                writeln(f1,'LSTR,',mass1[i-1].Nomer+pi,',',mass1[i].nomer+pi);
            CloseFile(f1);
            Memo1.Lines.Clear;
            Memo1.Lines.LoadFromFile(SaveDialog1.FileName);
        end;
end;

end.

```