

Міністерство освіти і науки України

Донбаська державна машинобудівна академія

В.А.Овчаренко, С.В.Подлесний, С.М.Зінченко

**ОСНОВИ МЕТОДУ КІНЦЕВИХ ЕЛЕМЕНТІВ
І ЙОГО ЗАСТОСУВАННЯ В ІНЖЕНЕРНИХ
РОЗРАХУНКАХ**

НАВЧАЛЬНИЙ ПОСІБНИК

Рекомендовано

Міністерством освіти і науки України

як навчальний посібник для студентів технічних вузів

Краматорськ 2008

УДК 621.539.4
ББК 34.4
О 35

Рецензенти:

Львов Г.І., д-р техн. наук, професор, завідувач кафедри динаміки та міцності машин Національного технічного університету «Харківський політехнічний інститут»;

Авраменко К.В., д-р техн. наук, професор, провідний науковий співробітник відділу нестационарних механічних процесів Інституту проблем Машинобудування НАН України ім. А.М.Підгорного;

Бейгельзімер Я.Ю., д-р техн. наук, професор, провідний науковий співробітник Донецького фізико-технічного інституту НАН України.

Гриф надано Міністерством освіти і науки України
Лист № 1.4/18-Г-1935 від 08.11.2007

Овчаренко В.А., Подлесний С.В., Зінченко С.М.

О 35 **Основи методу кінцевих елементів і його застосування в інженерних розрахунках: Навчальний посібник.** – Краматорськ: ДДМА, 2008. – 380 с.
ISBN 978-966-379-224-8

У навчальному посібнику дані основні відомості про метод кінцевих елементів для різних видів вантаження стрижнів, товстостінних циліндрів, пластин і оболонок. Наведені практичні розрахунки машинобудівних конструкцій з контрольними прикладами, блок-схеми і тексти програм їх реалізації на ПЕВМ. Може бути корисним студентам механічних спеціальностей, аспірантам та інженерам.

УДК 621.539.4
ББК 34.4

ISBN 978-966-379-224-8

© В.А.Овчаренко,
С.В.Подлесний
С.М.Зінченко, 2008
© ДДМА, 2008

ЗМІСТ

Вступ	8
1 Основи методу кінцевих елементів	8
1.1 Поняття про метод кінцевих елементів	8
1.2 Опис властивостей кінцевого елемента.....	9
1.2.1 Функція переміщень	9
1.2.2 Функція деформацій	10
1.2.3 Функція напружень.....	11
1.3 Система рівнянь МКЕ для конструкції в цілому	11
2 Розрахунок ступінчастого стрижня, який працює на розтягання	13
2.1 Основні залежності опору матеріалів при розтяганні – стисканні ...	13
2.2 Функція переміщень	14
2.3 Вектори деформацій, напружень і зусиль	14
2.4 Матриця жорсткості для кінцевого елемента.....	15
2.5 Приклад	16
3 Розрахунок стрижневих систем, що працюють на розтягання	18
3.1 Розрахунок плоских ферм	18
3.2 Розрахунок просторових ферм.....	21
3.3 Приклад	22
4 Розрахунок на міцність круглих валів	27
4.1 Основні залежності опору матеріалів при крученні	27
4.2 Функція переміщень для кінцевого елемента	29
4.3 Вектори деформацій, напружень і зусиль	30
4.4 Матриця жорсткості кінцевого елемента	31
4.5 Приклад	32
5 Розрахунок балок	35
5.1 Поняття про згинання. Основні залежності при згинанні	35
5.2 Функція переміщень при згинанні.....	38
5.3 Вектори деформацій, напружень і зусиль	40
5.4 Матриця жорсткості кінцевого елемента	42
5.5 Врахування впливу розподіленого навантаження на величину навантажень у вузлах.....	43
5.6 Приклад	44
6 Розрахунок плоских рам.....	48

6.1 Згинання з розтяганням плоского стрижня	48
6.2 Матриця жорсткості в локальних координатах для кінцевого елемента при згинанні з розтяганням	49
6.3 Матриця жорсткості для стрижнів плоскої рами	51
6.4 Приклад	53
7 Розрахунок стрижнів, що випробовують складне згинання	60
7.1 Основні залежності опору матеріалів при складному згинанні	60
7.2 Матриця жорсткості кінцевого елемента при складному згинанні	62
7.3 Приклад	63
8 Розрахунок стрижнів, що випробовують складне згинання з крученням	71
8.1 Основні залежності при згинанні з крученням	71
8.2 Матриця жорсткості кінцевого елемента при згинанні з крученням .	72
8.3 Приклад	74
9 Розрахунок просторових стрижневих систем	81
9.1 Основні залежності МКЕ для просторової стрижневої системи	81
9.2 Реалізація розрахунку просторової стрижневої системи МКЕ на ПЕОМ	85
9.2.1 Блок-схеми програми	85
9.2.2 Введення початкових даних	89
9.2.3 Результати розрахунку і їх аналіз	92
10 Розрахунок плоскої задачі теорії пружності	95
10.1 Поняття про плоску задачу теорії пружності	95
10.2 Узагальнений закон Гука	95
10.3 Плоский напружений стан	96
10.4 Плоский деформований стан	98
10.5 Зв'язок між деформаціями і переміщеннями	99
10.6 Зв'язок між напруженнями і переміщеннями	101
10.7 Функція переміщень для кінцевого трикутного елемента	102
10.8 Матриця жорсткості для трикутного кінцевого елемента	105
10.9 Функція переміщення для прямокутного кінцевого елемента	106
10.10 Приклад	107
10.11 Реалізація напружено-деформованого стану конструкції за допомогою ПЕОМ	111
10.11.1 Введення початкових даних	111

10.11.2	Виведення початкових даних на екран і у файл.....	113
10.11.3	Визначення коефіцієнтів матриці жорсткості елемента ..	113
10.11.4	Формування матриці жорсткості конструкції	114
10.11.5	Рішення системи рівнянь.....	114
10.11.6	Визначення зусиль і напружень	115
10.11.7	Виведення результатів розрахунків на екран і у файл.....	115
11	Розрахунок вісесиметричної задачі теорії пружності.....	115
11.1	Особливості вісесиметричної задачі теорії пружності	115
11.2	Функція переміщення	117
11.3	Вектори деформацій і напружень	118
11.4	Матриця жорсткості кінцевого елемента.....	119
11.5	Приклад	120
11.6	Тестове завдання	125
11.7	Блок-схеми програми для ПЕОМ вісесиметричної задачі теорії пружності.....	126
11.8	Розрахунок пресових посадок	129
12	Розрахунок вісесиметричних тіл при довільному навантаженні	131
12.1	Симетричне навантаження	132
12.2	Кососиметричне навантаження	136
13	Розрахунок пластин	140
13.1	Поняття про пластини. Гіпотеза Кирхгофа	140
13.2	Зв'язок між переміщеннями, деформаціями і зусиллями	142
13.3	Функція переміщень для прямокутного елемента	145
14	Напружено-деформований стан тонкостінних конструкцій	147
14.1	Визначення параметрів кінцевого елемента.....	148
14.2	Жорсткість плоского трикутного елемента в глобальній системі координат	157
14.3	Програма визначення деформацій і напружень в тонкостінних конструкціях	161
14.3.1	Сутність програми	161
14.3.2	Апробація програми.....	161
14.3.3	Висновки з апробації програми.....	170
15	Напружено-деформований стан тонкостінних тіл обертання при довільному вантаженні.....	170
15.1	Представлення зовнішнього навантаження.....	171

15.2	Формування матриць жорсткості кінцевого елемента	176
15.3	Формування матриці жорсткості конструкції і системи рівнянь рівноваги вузлів	191
15.4	Врахування граничних умов	196
15.5	Визначення переміщень і напружень	197
15.6	Блок-схеми програми.....	199
15.7	Апробації методики на класичних прикладах розрахунків.....	211
15.7.1	Розрахунок кільцевої пластини під дією розподіленого навантаження	211
15.7.2	Розрахунок конічної кришки під дією кільцевого розподіленого навантаження	213
15.7.3	Розрахунок ступінчастої циліндрової кришки під дією рівномірного тиску.....	215
15.7.4	Висновки з апробації методики	215
16	Розрахунок об'ємної задачі теорії пружності	217
16.1	Поняття про об'ємну задачу теорії пружності	217
16.2	Функція переміщення для тетраедра.....	219
16.3	Вектори деформацій і напружень.....	221
16.4	Матриця жорсткості для тетраедра	222
16.5	Функція переміщень і матриця жорсткості для паралелепіпеда ...	223
17	Розрахунок віброізолюваного фундаменту шаботного молота	223
17.1	Модель розрахунку віброізолюваного фундаменту.....	224
17.2	Приклад розрахунку і аналіз коливань	224
18	Розрахунок процесу ущільнення дискретного матеріалу в умовах нелінійної нестационарної теплопровідності	243
18.1	Отримання математичної моделі об'єкта	244
18.2	Рішення задачі ущільнення дискретного матеріалу в умовах одновимірної нелінійної нестационарної теплопровідності	249
18.3	Розрахунок поля температур і рішення задачі теплопровідності методом кінцевих елементів.....	252
18.4	Розрахунок поля щільності	261
18.5	Визначення оптимальних параметрів процесу пресування при охолодженні нагрітої заготовки	266
Додаток А. Текст програми розрахунку просторових стрижневих систем STERGV		270

Додаток Б. Текст програми плоскої задачі теорії пружності	288
Додаток В. Текст програми розрахунку вісесиметричних товстостінних циліндрів	304
Додаток Г. Текст програми автоматичного розбиття конструкції на елементи.....	312
Додаток Д. Текст програми розрахунку тонкостінних просторових систем	329
Додаток Е. Текст програми розрахунку процесу ущільнення пористого матеріалу.....	356
Додаток Ж. Інтегрування функцій форми.....	376
Література	378

ВСТУП

Навчальний посібник «Основи методу кінцевих елементів і його застосування в інженерних розрахунках» призначений для студентів технічних вузів, аспірантів та інженерів. Дана робота є результатом багаторічної науково-дослідної роботи авторів з розрахунку реальних машинобудівних конструкцій заводів. Окремі розділи роботи викладалися студентам спеціальностей «Інформаційні технології проектування» і «Підйомно-транспортні машини» [26].

1 ОСНОВИ МЕТОДУ КІНЦЕВИХ ЕЛЕМЕНТІВ

1.1 Поняття про метод кінцевих елементів

МКЕ є ефективним чисельним методом рішення інженерних і фізичних задач. Область його застосування тягнеться від аналізу напружень в конструкціях літаків або автомобілів до розрахунку таких складних систем, як атомна електростанція. З його допомогою розглядається рух рідини по трубах, через дамби, в пористих середовищах, досліджується перебіг газу, що стискається, вирішуються завдання електростатики і мастила, аналізуються коливання систем. МКЕ є чисельним методом рішення диференціальних рівнянь, що зустрічаються у фізиці й техніці.

Виникнення цього методу пов'язане з рішенням космічних задач (1950 р.). Область застосування МКЕ істотно поширилася, коли було показано, що рівняння, які описують елементи у задачах будівельної механіки, розповсюдження тепла і гідромеханіки аналогічні. МКЕ з чисельної процедури рішення задач будівельної механіки перетворився на загальний метод чисельного рішення диференціальних рівнянь.

Основна ідея МКЕ полягає в тому, що будь-яку безперервну величину, таку як температура, тиск і переміщення, можна апроксимувати дискретною моделлю, яка будується на безлічі шматково-безперервних функцій. У загальному випадку безперервна величина наперед не відома, і потрібно визначити значення цієї величини в деяких внутрішніх точках області. Дискретну модель дуже легко побудувати, якщо спочатку

припустити, що числові значення цієї величини в кожній внутрішній області відомі. Після цього можна перейти до загального випадку. Отже, при побудові дискретної моделі безперервної величини поступають таким чином:

- 1 У даній області фіксується кінцева кількість точок. Ці точки називаються вузловими точками або вузлами.
- 2 Значення безперервної величини в кожній точці вважається змінним, яке має бути визначеним.
- 3 Область визначення безперервної величини розбивається на кінцеву кількість областей, які називаються елементами. Ці елементи мають загальні вузлові точки і в сукупності апроксимують форму області.
- 4 Безперервна величина апроксимується на кожному елементі поліномом, який визначається за допомогою вузлових значень цієї величини. Для кожного елемента визначається свій поліном, але поліноми підбираються так, щоб збереглася безперервність величини уздовж меж елемента (його називають функцією елемента). Вибір форми елементів і їх функцій для конкретних завдань визначає точність наближеного рішення і залежить від винахідливості і майстерності інженера.

1.2 Опис властивостей кінцевого елемента

1.2.1 Функція переміщень

Щоб полегшити розуміння загальних співвідношень, розглянемо характеристики кінцевого елемента на прикладі ступінчастого стрижня, що випробовує розтягання-стискання (рис. 1.1).

У цьому випадку як кінцевий елемент береться стрижень (рис.1.2).

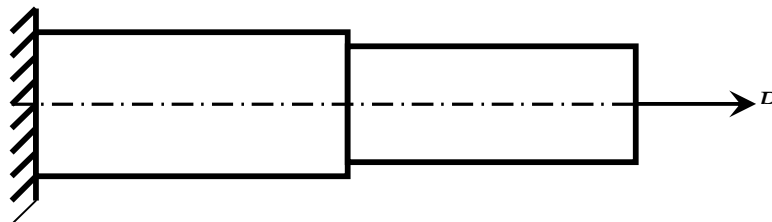


Рисунок 1.1 - Ступінчастий стрижень

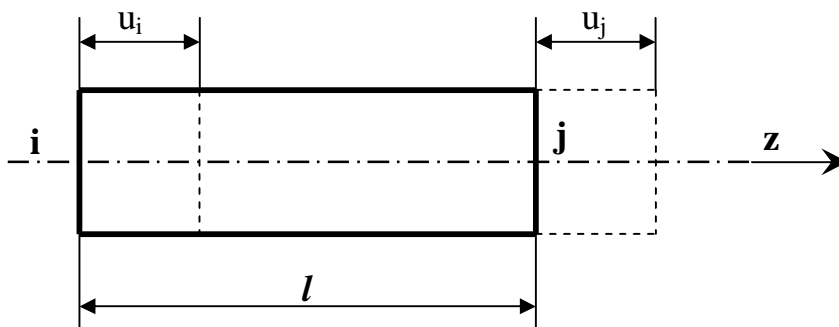


Рисунок 1.2 – Кінцевий елемент

Функція переміщення $\{f\}$ для цього випадку має вигляд

$$\{f\} = \left[\frac{l-z}{l}; \frac{z}{l} \right] \cdot \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}, \quad (1.1)$$

де u_i і u_j – переміщення вузлів i і j ;

$\frac{l-z}{l} = N_1$ і $\frac{z}{l} = N_2$ – функції форми; вони мінються від 1 до 0.

Функція переміщень $\{f\}$ залежить від форми кінцевого елемента.

Як вона вибирається, буде показано нижче.

1.2.2 Функція деформацій

Функція деформацій або вектор деформацій виражається через функцію переміщень.

При розтяганні відносно подовження стрижня

$$\{e\} = \left\{ \frac{\partial f}{\partial z} \right\} = \frac{1}{l} \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}. \quad (1.2)$$

Вираз $\frac{1}{l} \begin{bmatrix} -1 & 1 \end{bmatrix}$ вважаємо матрицею, тоді

$$\{e\} = \mathbf{B} \{d\}^e, \quad (1.3)$$

де $\{d\}^e = \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}$ – вектор переміщень вузлів елемента.

1.2.3 Функція напружень

Функція напружень (вектор напружень) виражається через вектор деформацій:

$$\{s\} = |D|(\{e\} - \{e_0\}) + \{s_0\} \quad (1.4)$$

де $|D|$ – матриця пружності (зв'язує між собою напруження і деформації), яка залежить від даної задачі;

$\{e_0\}$ – вектор початкових деформацій;

$\{s_0\}$ – вектор початкових напружень.

1.3 Система рівнянь МКЕ для конструкції в цілому

Визначимо роботу зовнішніх і внутрішніх сил на можливих (віртуальних) переміщеннях $d\{d\}^e$.

Робота внутрішніх сил в одиниці об'єму на віртуальному переміщенні дорівнює $d(\{e\}^T)\{s\}$, де $\{e\}^T$ – вектор деформацій, транспонований вектору $\{e\}$.

Робота внутрішніх сил по всьому кінцевому елементі:

$$\int_V d(\{e\}^T)\{s\}dV = \int_V d(\{e\}^T)(|D|(\{e\} - \{e_0\}) + \{s_0\})dV =$$

$$\int_V d(\{d\}^e)^T |B|^T |D| |B| dV \{d\}^e - \int_V d(\{d\}^e)^T |B|^T |D| \{e_0\} dV + \int_V d(\{d\}^e)^T |B|^T \{s_0\} dV.$$

Робота зовнішніх вузлових сил $\{F\}^e$ на віртуальних переміщеннях елемента дорівнює

$$(d\{d\}^e)^T \{F\}^e.$$

Прирівнявши роботу зовнішніх і внутрішніх сил на можливих переміщеннях елемента, одержимо:

$$\int_V d(\{d\}^e)^T |B|^T |D||B| dV \{d\}^e - \int_V d(\{d\}^e)^T |B|^T |D|\{e_0\} dV + \int_V d(\{d\}^e)^T |B|^T \{S_0\} dV = (d\{d\}^e)^T \cdot \{F\}^e,$$

скоротивши на $(d\{\delta\}^e)^T$, одержимо:

$$\left(\int_V |B|^T |D||B| dV - \int_V |B|^T |D|\{e_0\} dV + \int_V |B|^T \{S_0\} dV \right) \cdot \{d\}^e = \{P\}^e. \text{ або}$$

$$|K|^e \{d\}^e = \{P\}^e, \quad (1.5)$$

де

$$|K|^e = \int_V |B|^T |D| |B| dV - \int_V |B|^T |D| \{e_0\} dV + \int_V |B|^T \{S_0\} dV \quad (1.6)$$

- матриця жорсткості кінцевого елемента.

Для конструкції в цілому рівняння методу кінцевих елементів має вигляд

$$|K|\{d\} = \{R\}, \quad (1.7)$$

де $|K|$ – матриця жорсткості конструкції в цілому, вона є сумою матриць жорсткості кінцевих елементів, з яких складається конструкція;

$\{d\}$ – вектор переміщень всіх вузлів;

$\{R\}$ – вектор вузлових навантажень.

Будь-яка задача МКЕ в кінцевому підсумку зводиться до системи рівнянь (1.7). Її порядок дорівнює добутку кількості вузлів на кількість степенів вільності вузла. Отже, в задачах МКЕ замість диференціальних рівнянь доводиться вирішувати систему алгебраїчних рівнянь.

На закінчення слід зазначити, що при отриманні матриці жорсткості кінцевого елемента (1.6) не врахована дія температури.

2 РОЗРАХУНОК СТУПІНЧАСТОГО СТРИЖНЯ, ЯКИЙ ПРАЦЮЄ НА РОЗТЯГАННЯ

2.1 Основні залежності опору матеріалів при розтяганні – стисканні

Розтяганням називається вид вантаження, при якому в поперечному перерізі зі всіх внутрішніх силових факторів виникають тільки подовжні сили (рис.2.1).

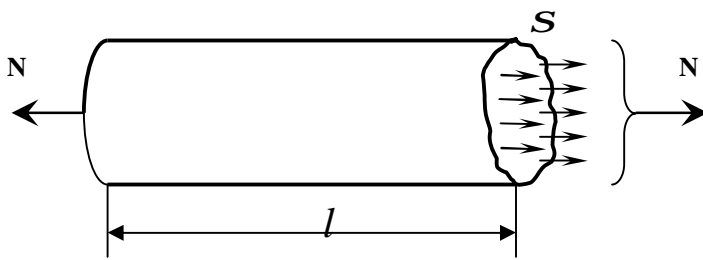


Рисунок 2.1- Розтягнутий стрижень

Умова міцності при розтяганні має вигляд

$$S = \frac{N}{F} \leq [S], \quad (2.1)$$

де S – нормальні напруження, МПа;

F – площа поперечного перерізу;

$[S]$ – допустимі напруження, які дорівнюють

$$[S] = \frac{S_0}{n},$$

де S_0 – небезпечні напруження:

а) $S_0 = S_{\sigma}$ – для крихкого матеріалу;

б) $S_0 = S_T$ – для пластичного матеріалу.

Тут S_{σ} – межа міцності, S_T – межа текучості, n – коефіцієнт запасу міцності.

Абсолютне подовження визначається за допомогою закону Гука:

$$\Delta l = \frac{Nl}{EF}, \quad (2.2)$$

де E – модуль пружності при розтяганні, для сталей $E = 2 \cdot 10^5$ МПа.

Зв'язок між деформацією ε і напруженнями

$$S = eE \quad (2.3)$$

також називається законом Гука.

2.2 Функція переміщень

Для ступінчастого стрижня (див. рис.1.1) кінцевим елементом є стрижень, показаний на рис.1.2.

У цьому випадку функція переміщення береться у вигляді полінома

$$u = a_0 + a_1 z. \quad (2.4)$$

Коефіцієнти a_0 і a_1 визначаються через переміщення вузлів u_i і u_j .

$$\text{При } z = 0 \quad u = u_i \rightarrow u_i = a_0.$$

$$\text{При } z = l \quad u = u_j \rightarrow u_j = a_0 + a_1 l = u_i + a_1 l,$$

$$\text{звідки } a_1 = \frac{u_j - u_i}{l}.$$

Підставивши значення a_0 і a_1 , одержимо:

$$u = u_i + \frac{u_j - u_i}{l} z \quad \text{або} \quad u = \left(1 - \frac{z}{l}\right) u_i + \frac{z}{l} u_j.$$

У матричній формі це запишеться так:

$$\{u\} = \left[1 - \frac{z}{l}; \frac{z}{l} \right] \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} \quad (2.5)$$

2.3 Вектори деформацій, напружень і зусиль

Під дією сил нескінченно мала ділянка стрижня деформується (рис.2.2).

Визначимо відносне подовження цієї ділянки

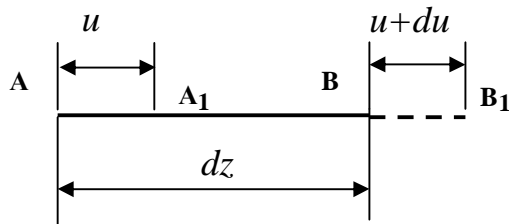


Рисунок 2.2 - Переміщення вузлів елемента

$$e = \frac{\Delta l}{l} = \frac{l_1 - l}{l} = \frac{A_1 B_1 - AB}{AB},$$

$$AB = dz, \quad A_1 B_1 = AB + BB_1 - AA_1,$$

$$A_1 B_1 = dz + u + du - u = dz + du.$$

$$e = \frac{dz + du - dz}{dz} = \frac{du}{dz},$$

Підставивши сюди вираз (2.5), одержимо вектор деформацій

$$\{e\} = \frac{1}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}. \quad (2.6)$$

На підставі закону Гука $\mathbf{S} = eE$ маємо вектор напружень

$$\{S\} = E\{e\} = \frac{E}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}. \quad (2.7)$$

Помноживши вектор напруження на площу поперечного перерізу елемента F , одержимо вектор зусиль

$$\{N\} = F\{S\} = \frac{EF}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}. \quad (2.8)$$

2.4 Матриця жорсткості для кінцевого елемента

Відповідно до виразу (1.6) матриця жорсткості для будь-якого кінцевого елемента дорівнює

$$|K|^e = \int_V |B|^T |D| |B| dV,$$

для розтягання $|B| = \frac{1}{l} \begin{vmatrix} -1 & 1 \end{vmatrix}$ з виразу (2.6), $|D| = E$ – із закону Гука, тоді

$$|K|^e = \int_0^l \frac{1}{l} \begin{vmatrix} -1 \\ 1 \end{vmatrix} \cdot E \cdot \frac{1}{l} \begin{vmatrix} -1 \\ 1 \end{vmatrix} \cdot F dz = \frac{EF}{l^2} \int_0^l \begin{vmatrix} -1 \\ 1 \end{vmatrix} \begin{vmatrix} -1 \\ 1 \end{vmatrix} dz.$$

Після інтегрування і перемноження матриць маємо

$$|K|^e = \frac{EF}{l} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}. \quad (2.9)$$

2.5 Приклад

Для стрижня, показаного на рис. 2.3,а, необхідно за допомогою МКЕ побудувати епюри подовжніх сил і напружень.

Розв'язання. Нумеруємо вузли (бажано послідовно) і елементи (рис.2.3,а). Обчислюємо матрицю жорсткості кожного елемента (за допомогою виразу (2.9)).

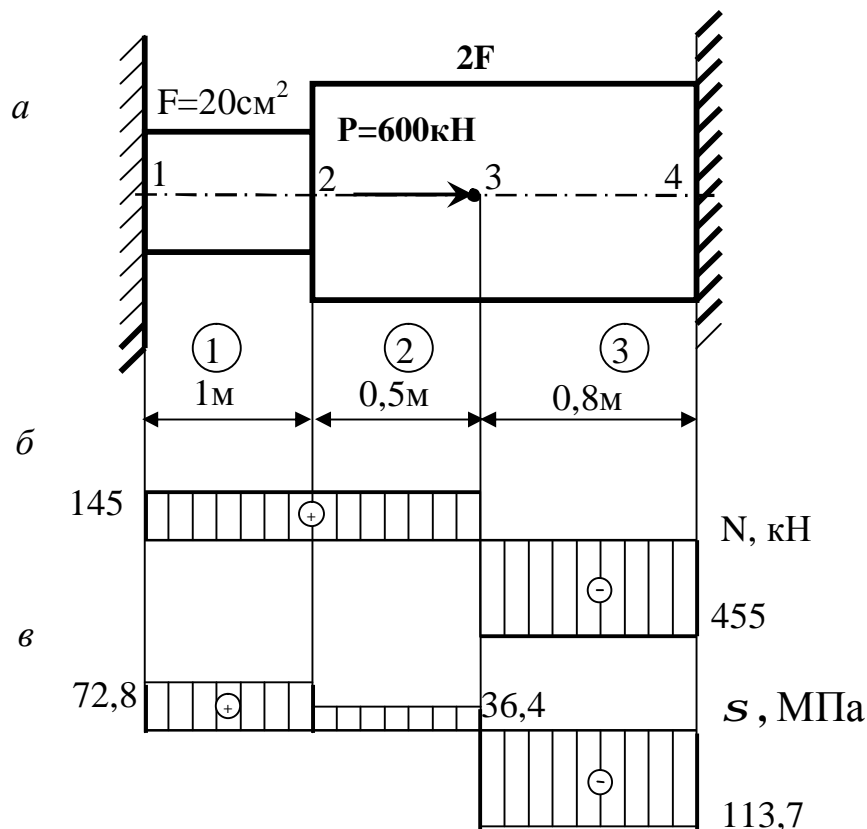


Рисунок 2.3 - Ступінчастий стрижень з епюрами подовжніх сил (б) і нормальних напружень(в)

$$|K|_I^e = \frac{EF}{l} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = EF \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix};$$

$$|K|_{II}^e = \frac{E \cdot 2F}{0,5} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = EF \begin{vmatrix} 4 & -4 \\ -4 & 4 \end{vmatrix};$$

$$|K|_{III}^e = \frac{E \cdot 2F}{0,8} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = EF \begin{vmatrix} 2,5 & -2,5 \\ -2,5 & 2,5 \end{vmatrix}.$$

Тепер формуємо матрицю жорсткості конструкції. Оскільки маємо чотири вузли, а кожен вузол має одну ступінь вільності, то розмірність матриці жорсткості дорівнює 4x4.

Коефіцієнти в одному і тому ж осередку складаються алгебраїчно.

У правому стовпці ставляться величини навантажень, що діють у відповідних вузлах.

У даному випадку сила прикладена в третьому вузлі справа, отже, в правому стовпці в третьому рядку записуємо 600.

$$EF \begin{vmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -4 & 0 \\ 0 & -4 & 4 & -2,5 \\ 0 & 0 & 2,5 & 2,5 \end{vmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 600 \\ 0 \end{Bmatrix}. \quad (2.10)$$

Враховуємо граничні умови: оскільки стрижень по кінцях затиснений, то $u_1 = 0$, $u_4 = 0$. На підставі цього викреслюємо перший і четвертий стовпці і перший і четвертий рядки. Після цього одержимо:

$$EF \begin{vmatrix} 5 & -4 \\ -4 & 6,5 \end{vmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 600 \end{Bmatrix} \quad \text{звідки } u_2 = 0,8u_3.$$

Підставивши значення u_2 у другий рядок, маємо:

$$EF(-4 \cdot 0,8u_3 + 6,5u_3) = 600,$$

$$u_3 = \frac{182}{EF}, \quad u_2 = \frac{182}{EF} \cdot 0,8 = \frac{145,6}{EF}.$$

Визначаємо величини подовжніх сил:

$$\{N\}_I = \frac{EF}{l_1}|-1; \quad 1\left\{\begin{matrix} u_1 \\ u_2 \end{matrix}\right\} = \frac{EF}{1}|-1; \quad 1\left\{\begin{matrix} 0 \\ 145,6 \\ EF \end{matrix}\right\} = 145,6 \text{ кН},$$

$$\{N\}_{II} = \frac{E \cdot 2F}{0,5}|-1; \quad 1\left\{\begin{matrix} 145,6 \\ EF \\ EF \end{matrix}\right\} = 145,6 \text{ кН},$$

$$\{N\}_{III} = \frac{E \cdot 2F}{0,8}|-1; \quad 1\left\{\begin{matrix} 182 \\ EF \\ 0 \end{matrix}\right\} = -455 \text{ кН}.$$

За набутими значеннями будуємо епюру N (рис. 2.3, б).

Розділивши на кожній ділянці подовжню силу на площу, одержимо величини напружень:

$$s_I = \frac{N_I}{F_I} = \frac{145,6 \cdot 10^{-3}}{20 \cdot 10^{-4}} = 72,8 \text{ МПа},$$

$$s_{II} = \frac{N_{II}}{F_{II}} = \frac{145,6 \cdot 10^{-3}}{2 \cdot 20 \cdot 10^{-4}} = 36,4 \text{ МПа},$$

$$s_{III} = \frac{N_{III}}{F_{III}} = \frac{-455 \cdot 10^{-3}}{2 \cdot 20 \cdot 10^{-4}} = 113,7 \text{ МПа}.$$

Будуємо епюру нормальних напруження (рис. 2.3, в).

3 РОЗРАХУНОК СТРИЖНЕВИХ СИСТЕМ, ЩО ПРАЦЮЮТЬ НА РОЗТЯГАННЯ

3.1 Розрахунок плоских ферм

Ферма – це стрижнева система, стрижні якої між собою сполучені за допомогою шарнірів. Якщо всі стрижні і навантаження розташовані в одній площині, то ферма називається плоскою (рис. 3.1).

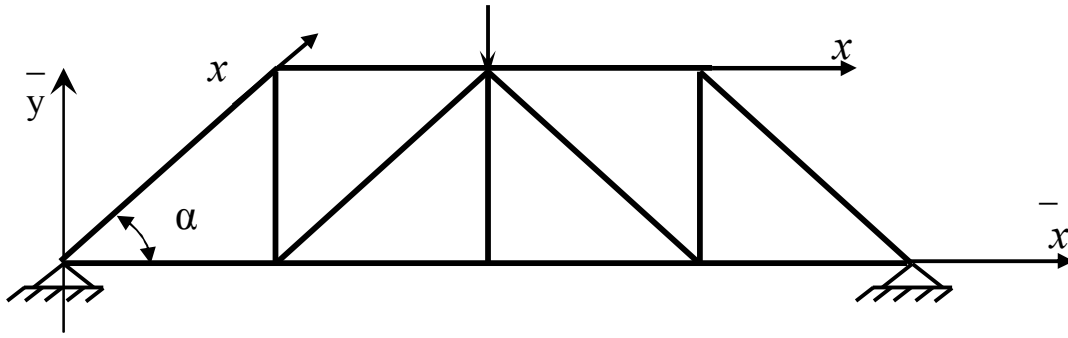


Рисунок 3.1 – Плоска ферма з глобальною і локальною системами

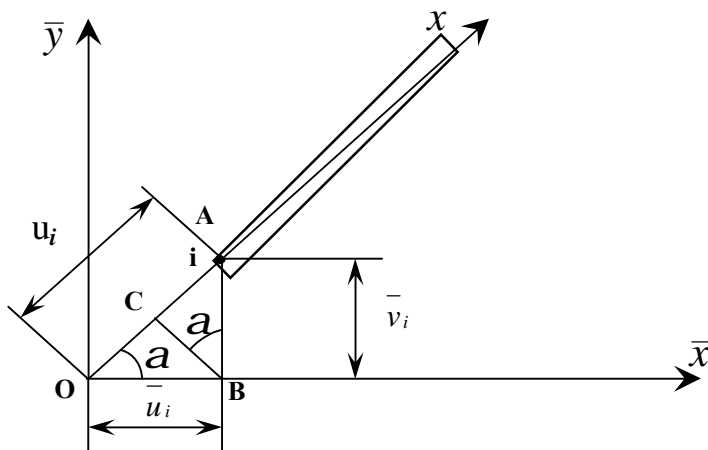
Особливість ферми в порівнянні із ступінчастим стрижнем полягає в тому, що стрижні ферми у відношенні один до одного повернені. У зв'язку з цим необхідно вводити дві системи координат: локальну – місцеву для кожного стрижня і глобальну – загальну для всіх стрижнів.

Для ступінчастого стрижня була одержана матриця жорсткості у такому вигляді:

$$|K|^e = \frac{EF}{l} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}. \quad (3.1)$$

Тепер необхідно одержати таку ж матрицю жорсткості елемента, але в глобальній системі координат.

Під дією подовжніх сил вузол i (рис. 3.2, точка А) одержить переміщення в локальній системі координат u_i , а в глобальній системі



координат – $u_i\text{-bar}, v_i\text{-bar}$

$$u_i = OA = OC + CA,$$

$$OC = u_i\text{-bar} \cos a, \quad CA = v_i\text{-bar} \sin a,$$

$$u_i = u_i\text{-bar} \cos a + v_i\text{-bar} \sin a,$$

де a – кут нахилу стрижня до осі $x\text{-bar}$.

Аналогічно

$$u_j = u_j\text{-bar} \cos a + v_j\text{-bar} \sin a.$$

Рисунок 3.2 – Переміщення i -го вузла

У матричній формі це запишеться так

$$\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \begin{vmatrix} \cos a & \sin a & 0 & 0 \\ 0 & 0 & \cos a & \sin a \end{vmatrix} \begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \\ \bar{u}_j \\ \bar{v}_j \end{Bmatrix} \quad (3.2)$$

або $\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = |I| \begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \\ \bar{u}_j \\ \bar{v}_j \end{Bmatrix}$, де $|I|$ – матриця направляючих косинусів,

$$|I| = \begin{vmatrix} \cos a & \sin a & 0 & 0 \\ 0 & 0 & \cos a & \sin a \end{vmatrix}. \quad (3.3)$$

У цьому випадку вектор деформацій запишеться так:

$$\{e\} = \frac{1}{l}|-1; \quad 1|\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \frac{1}{l}|-1; \quad 1||I| \begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \\ \bar{u}_j \\ \bar{v}_j \end{Bmatrix}. \quad (3.4)$$

Тепер матриця $|B|$ має вигляд

$$|B| = \frac{1}{l}|-1; \quad 1||I|. \quad (3.5)$$

Після підстановки у вираз (1.6) одержимо:

$$\begin{aligned} |\bar{K}|^e &= \int_V |B|^T |D| |B| dV = \int_0^l |I|^T \cdot \frac{1}{l} \begin{vmatrix} -1 \\ 1 \end{vmatrix} \cdot E \cdot \frac{1}{l} \begin{vmatrix} -1 \\ 1 \end{vmatrix} \cdot 1 ||I| \cdot F \cdot dz = \\ &= \frac{EF}{l} |I|^T \begin{vmatrix} -1 \\ 1 \end{vmatrix} \begin{vmatrix} -1 \\ 1 \end{vmatrix} |I| = |I|^T \cdot \frac{EF}{1} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} |I|, \end{aligned}$$

остаточно

$$|\bar{K}|^e = |I|^T |K|^e |I|, \quad (3.6)$$

де $|K|^e$ і $|\bar{K}|^e$ – матриці жорсткості кінцевого елемента в локальній і глобальній системах координат.

Вектори напружень і зусиль відповідно запишуться

$$\{S\} = \frac{E}{l} | -1; \quad 1 || I | \left\{ \begin{array}{c} \bar{u}_i \\ \bar{v}_i \\ \bar{u}_j \\ \bar{v}_j \end{array} \right\}, \quad (3.7)$$

$$\{N\} = \frac{EF}{l} | -1; \quad 1 || I | \left\{ \begin{array}{c} \bar{u}_i \\ \bar{v}_i \\ \bar{u}_j \\ \bar{v}_j \end{array} \right\}. \quad (3.8)$$

3.2 Розрахунок просторових ферм

За аналогією з плоскими фермами виконується розрахунок просторових ферм.

Вводимо дві системи координат: локальну – вісь x і глобальну – осі $\bar{x}, \bar{y}, \bar{z}$. Тепер кожен вузол має три степені вільності, і переміщення вузлів в глобальній системі координат запишуться так:

$$\left\{ \begin{array}{c} u_i \\ v_j \end{array} \right\} = \left| \begin{array}{cccccc} \cos a_1 & \cos a_2 & \cos a_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos a_1 & \cos a_2 & \cos a_3 \end{array} \right| \left\{ \begin{array}{c} \bar{w}_i \\ \bar{u}_j \\ \bar{v}_j \\ \bar{w}_j \end{array} \right\}, \quad (3.9)$$

$$\text{де} \quad \cos a_1 = \frac{\bar{x}_j - \bar{x}_i}{l}; \quad \cos a_2 = \frac{\bar{y}_j - \bar{y}_i}{l}; \quad \cos a_3 = \frac{\bar{z}_j - \bar{z}_i}{l};$$

$$l = \sqrt{(\bar{x}_j - \bar{x}_i)^2 + (\bar{y}_j - \bar{y}_i)^2 + (\bar{z}_j - \bar{z}_i)^2}.$$

Матриця жорсткості кінцевого елемента в глобальній системі координат, як і для плоскої ферми, визначається за формулою (3.6), але при цьому матриця направляючих косинусів має вигляд

$$|I| = \begin{vmatrix} \cos a_1 & \cos a_2 & \cos a_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos a_1 & \cos a_2 & \cos a_3 \end{vmatrix}. \quad (3.10)$$

3.3 Приклад

На стрижневу систему (рис. 3.3) діє сила $P=500$ кН. Визначити зусилля

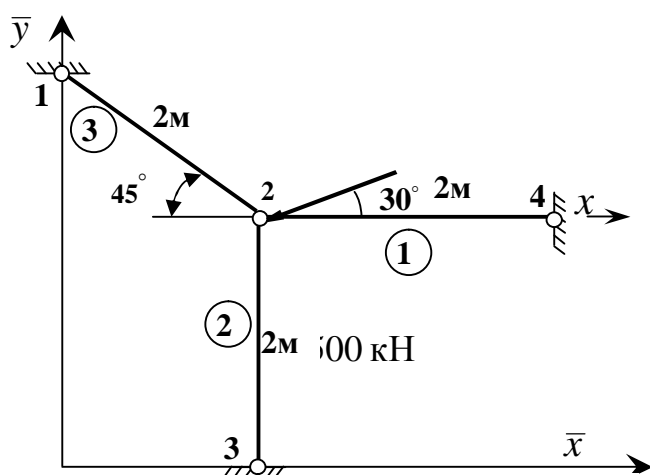


Рисунок 3.3 – Стрижньова система

і напруження в стрижнях, прийнявши площі всіх стрижнів однаковими і рівними $F=30$ см².

Розв'язання. Вводимо глобальну систему координат \bar{x} \bar{y} . Нумеруємо вузли і елементи.

Матриці жорсткості стрижнів визначаємо за формулою (3.6).

Елемент 1: його зв'язки 2–4, вісь x йде від вузла 2 до вузла 4, отже, кут між осями x і \bar{x} $\alpha=0$. Матриця направляючих косинусів для цього елемента запишеться так

$$|I| = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix}.$$

Обчислюємо матрицю жорсткості елемента:

$$|\bar{K}|_1^e = \frac{EF}{2} \begin{vmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} = EF \begin{vmatrix} 0,5 & 0 & -0,5 & 0 \\ 0 & 0 & 0 & 0 \\ -0,5 & 0 & 0,5 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}.$$

Елемент 2: стрижень 2–3, $a = -90^\circ$ (напрямок осі від вузла 2 до 3).

$$|I| = \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix}.$$

$$|\bar{K}|_{II}^e = \frac{EF}{2} \begin{vmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{vmatrix} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix} = EF \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0,5 & 0 & -0,5 \\ 0 & 0 & 0 & 0 \\ 0 & -0,5 & 0 & 0,5 \end{vmatrix}.$$

Елемент 3: стрижень 2-1, $\alpha = 135^\circ$.

$$|I| = \begin{vmatrix} -0,71 & 0,71 & 0 & 0 \\ 0 & 0 & -0,71 & 0,71 \end{vmatrix}.$$

$$|\bar{K}|_{III}^e = \frac{EF}{2} \begin{vmatrix} -0,71 & 0 \\ 0,71 & 0 \\ 0 & -0,71 \\ 0 & 0,71 \end{vmatrix} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} \cdot \begin{vmatrix} -0,71 & 0,71 & 0 & 0 \\ 0 & 0 & -0,71 & 0,71 \end{vmatrix} =$$

$$= EF \begin{vmatrix} 0,25 & -0,25 & -0,25 & 0,25 \\ -0,25 & 0,25 & 0,25 & -0,25 \\ -0,25 & 0,25 & 0,25 & -0,25 \\ 0,25 & -0,25 & -0,25 & 0,25 \end{vmatrix}.$$

Матриця жорсткості кожного елемента має бути симетричною, а за головною діагоналлю повинні стояти додатні коефіцієнти.

Переходимо до формування матриці жорсткості конструкції. Оскільки вузлів 4, а кожен вузол в глобальній системі координат має дві степені вільності, то матриця жорсткості конструкції має розмірність 8×8 .

При заповненні матриці жорсткості конструкції необхідно враховувати, які вузли входять до елемента. Наприклад, елемент 1. До нього входять вузли 2 і 4, отже, необхідно коефіцієнти матриці жорсткості цього елемента заносити в стовпці і рядки 2 і 4.

У правий стовпець заносимо значення навантажень. Оскільки навантаження прикладене тільки в другому вузлі, то тільки до цього вузла заносимо величини:

$$P_{2x}^- = -P \cos 30^\circ = -500 \cdot 0,866 = -433 \text{ кН},$$

$$P_{2y}^- = -P \sin 30^\circ = -500 \cdot 0,5 = -250 \text{ кН}.$$

Навантаження узяті з мінусом, оскільки вони направлені проти додатного напрямку осей.

Тепер врахуємо граничні умови. Вузли 1, 3 і 4 закріплені, отже, $\bar{u}_1, \bar{v}_1, \bar{u}_3, \bar{v}_3$ і \bar{u}_4, \bar{v}_4 дорівнюють нулю.

Після врахування граничних умов одержимо:

	1	2	3	4					
EF	1	0,25	-0,25	-0,25	0,25	0	0	0	0
		-0,25	0,25	0,25	-0,25	0	0	0	0
	2	-0,25	0,25	0,5	0	0	0	-0,5	0
				0	0	0	0	0	0
	3	0,25	-0,25	-0,25	0,25	0	-0,5	0	0
				0	0,5	0	0	0	0
	4	0	0	0	0	0	0	0	0
		0	0	0	-0,5	0	0,5	0	0
	0	0	-0,5	0	0	0	0,5	0	
	0	0	0	0	0	0	0	0	

$$\begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -433 \\ -250 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

$$EF \begin{vmatrix} 0,75 & -0,25 \\ -0,25 & 0,75 \end{vmatrix} \begin{Bmatrix} \bar{u}_2 \\ \bar{v}_2 \end{Bmatrix} = \begin{Bmatrix} -433 \\ -250 \end{Bmatrix},$$

помноживши другий рядок на 3, маємо:

$$EF \begin{vmatrix} 0,75 & -0,25 \\ -0,75 & 2,25 \end{vmatrix} \begin{Bmatrix} \overline{u_2} \\ \overline{v_2} \end{Bmatrix} = \begin{Bmatrix} -433 \\ -750 \end{Bmatrix}.$$

Склавши рядки, знайдемо:

$$2,0\overline{v_2} \cdot EF = -1183,$$

$$\overline{v_2} = -\frac{591,5}{EF}.$$

Підставивши у друге рівняння значення, $\overline{v_2}$ знайдемо $\overline{u_2}$:

$$-0,25\overline{u_2} + 0,75\frac{(-591,5)}{EF} = -\frac{250}{EF};$$

$$\overline{u_2} = -\frac{775}{EF}.$$

Визначаємо зусилля в стрижнях за формулою (3.8):

$$\{N\}_I = \frac{EF}{2} \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} \begin{Bmatrix} -\frac{775}{EF} \\ -\frac{591,5}{EF} \\ 0 \\ 0 \end{Bmatrix} = \frac{EF}{2} \begin{vmatrix} -1 & 0 & 1 & 0 \end{vmatrix} \begin{Bmatrix} -\frac{775}{EF} \\ -\frac{591,5}{EF} \\ 0 \\ 0 \end{Bmatrix} = 387,5кН,$$

$$\{N\}_{II} = \frac{EF}{2} \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix} \begin{Bmatrix} -\frac{775}{EF} \\ -\frac{591,5}{EF} \\ 0 \\ 0 \end{Bmatrix} = \frac{EF}{2} \begin{vmatrix} 0 & 1 & 0 & -1 \end{vmatrix} \begin{Bmatrix} -\frac{775}{EF} \\ -\frac{591,5}{EF} \\ 0 \\ 0 \end{Bmatrix} = -296кН,$$

$$\{N\}_{III} = \frac{EF}{2} \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & -0,71 & 0,71 & 0 \\ 0 & 0 & 0 & -0,71 \\ 0 & 0,71 & 0 & 0 \end{vmatrix} \begin{Bmatrix} -\frac{775}{EF} \\ -\frac{591,5}{EF} \\ 0 \\ 0 \end{Bmatrix} =$$

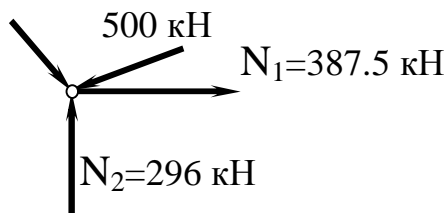
$$= \frac{EF}{2} \begin{bmatrix} 0,71 & -0,71 & -0,71 & 0,71 \end{bmatrix} \begin{Bmatrix} -\frac{775}{EF} \\ \frac{591,5}{EF} \\ 0 \\ 0 \end{Bmatrix} = -65,1 \text{ кН}.$$

Покажемо зусилля в стрижнях (якщо зусилля від'ємне, воно показане напрямком до вузла, якщо додатне – від вузла) (рис. 3.4).

Перевірка: $\sum P_x = 0$.

$$N_3 = 65,1 \text{ кН}$$

$$N_3 \cos 45^\circ - 500 \cos 30^\circ + N_1 = 0;$$



$$65,1 \cdot 0,707 - 500 \cdot 0,866 + 387,5 = 0;$$

$$433,7 - 433 = 0.$$

Рисунок 3.4 – Схема сил, які діють у вузлі 2

Погрішність 0,2%.

$$\sum P_y = 0,$$

$$-N_3 \sin 45^\circ + N_2 - 500 \cdot \sin 30^\circ = 0,$$

$$-65,1 \cdot 0,707 + 296 - 500 \cdot 0,5 = 0,$$

$$-296,2 + 296 = 0.$$

Погрішність 0,2%, отже, зусилля знайдені вірно.

Визначимо напруження в стрижнях:

$$s_1 = \frac{N_1}{F} = \frac{387,5 \cdot 10^{-3}}{30 \cdot 10^{-4}} = 129 \text{ МПа},$$

$$s_2 = \frac{N_2}{F} = \frac{296 \cdot 10^{-3}}{30 \cdot 10^{-4}} = 98,7 \text{ МПа},$$

$$s_3 = \frac{N_3}{F} = \frac{65,1 \cdot 10^{-3}}{30 \cdot 10^{-4}} = 21,7 \text{ МПа}.$$

4 РОЗРАХУНОК НА МІЦНІСТЬ КРУГЛИХ ВАЛІВ

4.1 Основні залежності опору матеріалів при крученні

Напруження в довільній точці перерізу визначаються за формулою

$$t = \frac{M_{кр} r}{J_r}, \quad (4.1)$$

де $M_{кр}$ - крутний момент в перерізі, де визначаються дотичні напруження t (рис. 4.1);

r - відстань від точки, де визначаються напруження, до центра вала;

J_r - полярний момент інерції перерізу вала:

$$J_r = \frac{\rho D^4}{32} \text{ - для суцільного вала;}$$

$$J_r = \frac{\rho D^4}{32} (1 - a^4) \text{ - для порожнистого вала;}$$

D, d - зовнішній і внутрішній діаметри вала $a = d/D$.

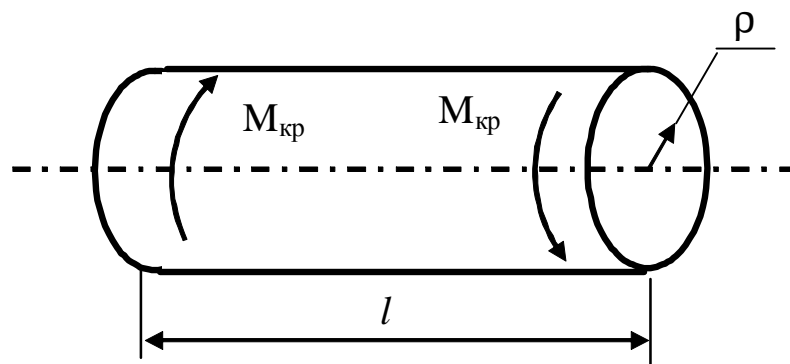


Рисунок 4.1 – Вал з круглим поперечним перерізом

Епюра напружень при крученні показана на рис. 4.2.

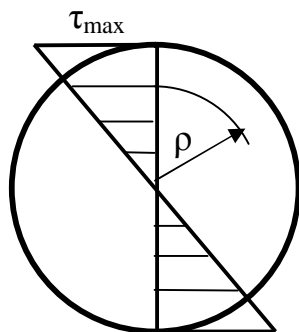


Рисунок 4.2 – Етюра напружень при крученні

Умова міцності при крученні

$$t_{\max} = \frac{M_{кр}}{W_r} \leq [t], \quad (4.2)$$

де W_r – полярний момент опору перерізу.

Для суцільного вала $W_r = \frac{\rho D^3}{16}$, а для порожнистого $W_r = \frac{\rho D^3}{16}(1 - a^4)$.

Вал розраховується не тільки на міцність, але і на жорсткість:

$$\Theta = \frac{M_{кр}}{GJ_r} \leq [\Theta] \quad (4.3)$$

де Θ – відносний кут закручування;

G – модуль пружності другого роду: для сталі $G=8 \cdot 10^4$ МПа,

$$G = \frac{E}{2(1 + m)};$$

$[\Theta]$ – допустимий відносний кут закручування.

Повний кут закручування ділянки валу постійного перерізу обчислюється за формулою

$$j = \frac{M_{кр} l}{GJ_r}. \quad (4.4)$$

Закон Гука при крученні

$$t = gG, \quad (4.5)$$

де g – відносний зсув.

4.2 Функція переміщень для кінцевого елемента

Для розрахунку вала, показаного на рис. 4.3, як кінцевий приймається елемент постійного перерізу (рис.4.4).

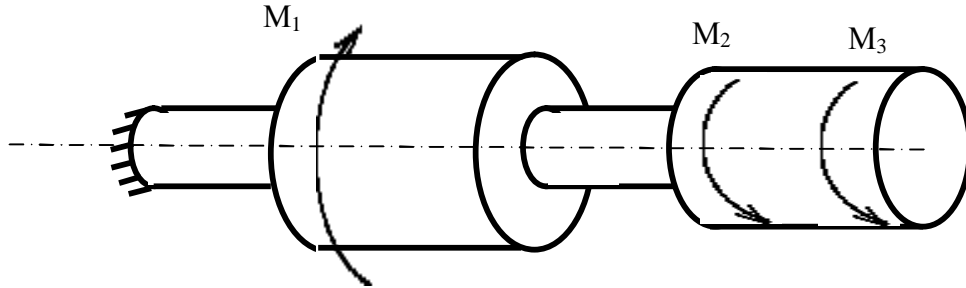


Рисунок 4.3 – Схема вала

Оскільки кожен вузол має одну ступінь свободи, то функція переміщень для кінцевого елемента має вигляд

$$\dot{j} = a_0 + a_1 z. \quad (4.6)$$

Коефіцієнти a_0 і a_1 визначаємо через переміщення j_i, j_j вузлів. При $z=0$ $j = j_i$, а при $z=l$ $j = j_j$.

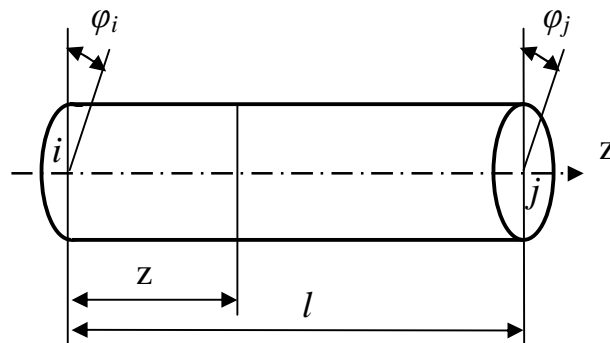


Рисунок 4.4 – Кінцевий елемент вала

Підставивши ці значення у вираз (4.6), одержимо:

$$j_i = a_0, \quad j_j = a_0 + a_1 l = j_i + a_1 l, \quad \text{звідки } a_1 = \frac{j_j - j_i}{l}.$$

Підставляємо значення a_0 і a_1 у вираз (4.6):

$$\mathbf{j} = \mathbf{j}_i + \frac{\mathbf{j}_j - \mathbf{j}_i}{l} z = \left(1 - \frac{z}{l}\right) \mathbf{j}_i + \frac{z}{l} \mathbf{j}_j.$$

У матричній формі маємо:

$$\{\mathbf{j}\} = \left| 1 - \frac{z}{l}; \frac{z}{l} \right| \begin{Bmatrix} \mathbf{j}_i \\ \mathbf{j}_j \end{Bmatrix}. \quad (4.7)$$

4.3 Вектори деформацій, напружень і зусиль

У результаті дії крутних моментів елемент вала довжиною dz закручується на кут dj (рис. 4.5).

Встановимо залежність між g і dj .

З одного боку, $AA_1 = \rho \cdot d\varphi$, з іншого – $AA_1 = z dz$.

Прирівнявши ці значення, одержимо $r dj = g dz$, звідки

$$\{\mathbf{g}\} = \frac{dj}{dz} \mathbf{r}. \quad (4.8)$$

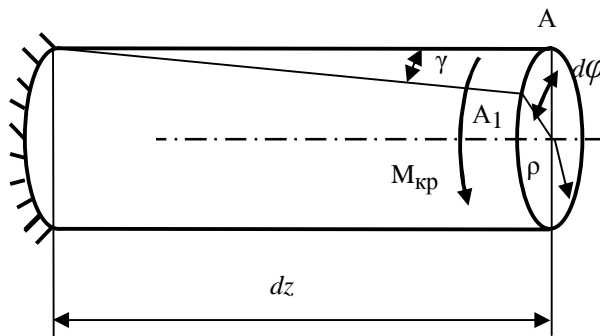


Рисунок 4.5 – Деформація вала при крученні

Використовуючи вираз (4.7) і, узявши похідну $\frac{dj}{dz}$, одержуємо:

$$\{\mathbf{g}\} = \frac{\mathbf{r}}{\mathbf{l}} \left| -1; 1 \right| \begin{Bmatrix} \mathbf{j}_i \\ \mathbf{j}_j \end{Bmatrix}. \quad (4.9)$$

Тепер матриця $|\mathbf{B}|$ запишеться так:

$$|B| = \frac{r}{l} \begin{vmatrix} -1 & 1 \end{vmatrix}. \quad (4.10)$$

Вектор напруження на підставі закону Гука при крученні запишеться так:

$$\{t\} = G\{g\} = \frac{Gr}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} \begin{Bmatrix} j_i \\ j_j \end{Bmatrix}. \quad (4.11)$$

Вектор зусиль визначаємо з виразу (4.1):

$$\{M_{kp}\} = \frac{tJ_r}{r} = \frac{GJ_{kp}}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} \begin{Bmatrix} j_i \\ j_j \end{Bmatrix}. \quad (4.12)$$

4.4 Матриця жорсткості кінцевого елемента

Матрицю жорсткості визначаємо за формулою (1.6)

$$|K|^e = \int_V |B|^T |D| |B| dV = \int_F \int_0^l \frac{r}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} G \frac{r}{l} \begin{vmatrix} -1 & 1 \end{vmatrix} dz dF,$$

де $dV = dz dF$, $|D| = G$.

Після інтегрування одержимо:

$$|K|^e = G \frac{1}{l^2} \begin{vmatrix} -1 & 1 \\ 1 & -1 \end{vmatrix} \cdot \begin{vmatrix} -1 & 1 \end{vmatrix} \int_F r^2 dF \int_0^l dz = \frac{GJ_r}{l} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}.$$

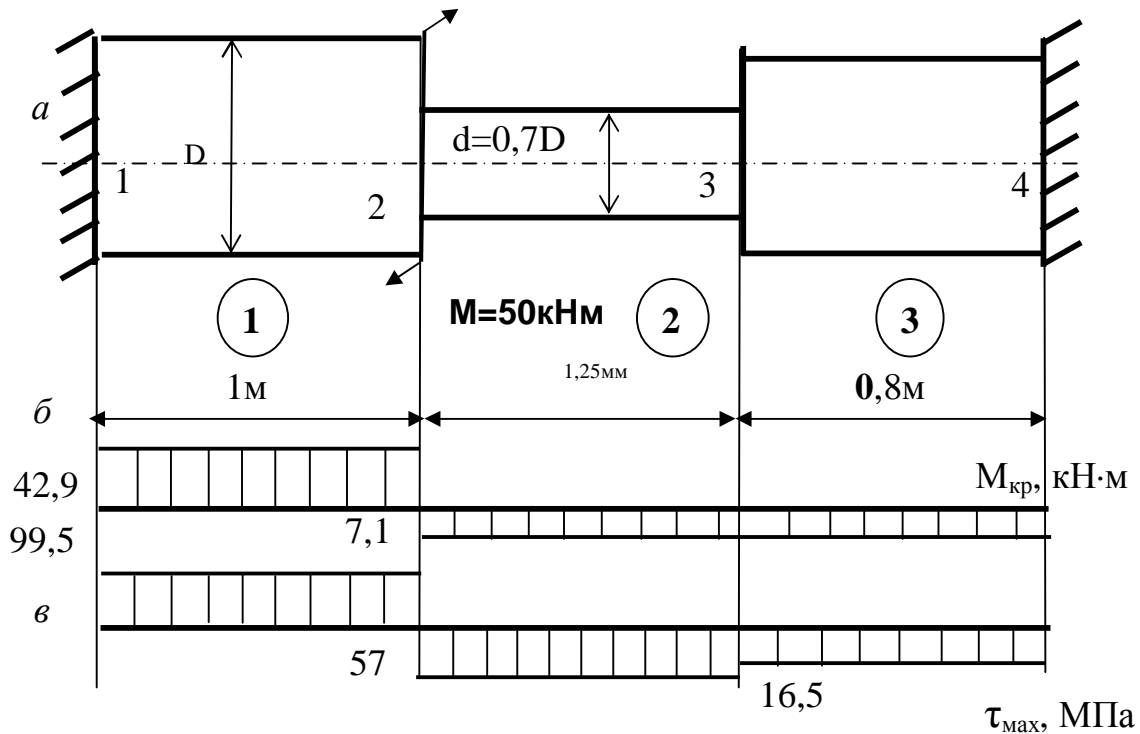
Таким чином, матриця кінцевого елемента дорівнює

$$|K|^e = \frac{GJ_r}{l} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix}. \quad (4.13)$$

Як бачимо, не дивлячись на відмінність задач розтягання і кручення, їх матриці жорсткості аналогічні.

4.5 Приклад

Для вала, показаного на рис.4.6, а, побудувати епюру крутних моментів, визначити діаметри вала, якщо $[\tau]=120 \text{ МПа}$, побудувати епюру найбільших дотичних напружень.



*а – схема вала; б – епюра крутних моментів;
в – епюра максимальних дотичних напружень*

Рисунок 4.6 – Вал і епюри крутних моментів і максимальних дотичних напружень

Розв'язання. Нумеруємо вузли і елементи. Матриці жорсткості елементів визначаємо за формулою (4.12). Заздалегідь визначаємо полярні моменти інерції перерізів вала:

$$J_r^I = \frac{\rho D^4}{32},$$

$$J_r^{II} = \frac{\rho d^4}{32} = \frac{\rho (0,7D)^4}{32} = \frac{\rho D^4}{32} 0,24 = 0,24 J_r^I,$$

$$|K|_I^e = \frac{GJ_r^I}{1} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = GJ_r^I \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix},$$

$$|K|_{II}^e = \frac{0,24GJ_r^I}{1,25} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = GJ_r^I \begin{vmatrix} 0,19 & -0,19 \\ -0,19 & 0,19 \end{vmatrix},$$

$$|K|_{III}^e = \frac{GJ_r^I}{0,8} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix} = GJ_r^I \begin{vmatrix} 1,25 & -1,25 \\ -1,25 & 1,25 \end{vmatrix}.$$

Формуємо матрицю жорсткості конструкції. Її розмірність 4x4 (кількість вузлів – 4, степенів вільності вузла –1).

$$GJ_r^I \begin{vmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & -0,19 & 0 \\ 0 & -0,19 & 0,19 & -1,25 \\ 0 & 0 & -1,25 & 1,25 \end{vmatrix} \begin{Bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 50 \\ 0 \\ 0 \end{Bmatrix}. \quad (4.14)$$

Враховуємо граничні умови: $\varphi_1=0$, $\varphi_4=0$.

Викреслюючи 1- і 4-й рядки і відповідні стовпці, одержимо:

$$GJ_r^I \begin{vmatrix} 1,19 & -0,19 \\ -0,19 & 1,44 \end{vmatrix} \begin{Bmatrix} j_2 \\ j_3 \end{Bmatrix} = \begin{Bmatrix} 50 \\ 0 \end{Bmatrix},$$

$$-0,19\varphi_2 + 1,44\varphi_3 = 0 \quad \text{а} \quad j_2 = \frac{1,44}{0,19} j_3 = 7,58j_3.$$

Підставляємо в перше рівняння значення φ_2 :

$$1,19 \cdot 7,53j_2 - 0,19j_3 = \frac{50}{GJ_r^I},$$

звідки

$$j_3 = \frac{50}{8,829GJ_r^I} = \frac{5,66}{GJ_r^I} \quad i \quad j_2 = \frac{42,9}{GJ_r^I}.$$

Обчислюємо крутні моменти:

$$\{M_{кр}\}_I = \frac{GJ_r^I}{1} \cdot 1; \quad 1 \left\{ \begin{array}{c} 0 \\ 42,93 \\ \frac{GJ_r^I}{GJ_r^I} \end{array} \right\} = 42,9 \text{ кН м};$$

$$\{M_{кр}\}_{II} = \frac{GJ_r^{II}}{1,25} \cdot 1; \quad 1 \left\{ \begin{array}{c} 42,93 \\ \frac{GJ_r^I}{GJ_r^I} \\ 5,66 \\ \frac{GJ_r^I}{GJ_r^I} \end{array} \right\} = \frac{0,24GJ_r^I}{1,25} \cdot 1; \quad 1 \left\{ \begin{array}{c} 42,4 \\ \frac{GJ_r^I}{GJ_r^I} \\ 5,66 \\ \frac{GJ_r^I}{GJ_r^I} \end{array} \right\} = -7,1 \text{ кН м};$$

$$\{M_{кр}\}_{III} = \frac{GJ_r}{0,8} \cdot 1; \quad 1 \left\{ \begin{array}{c} 5,66 \\ \frac{GJ_r^I}{GJ_r^I} \\ 0 \end{array} \right\} = -7,1 \text{ кН м}.$$

За цими значеннями будуюмо епюру крутних моментів (див. рис. 4.6,б).

З умови міцності визначаємо необхідний момент опору перерізу

$$W_r \geq \frac{M_{кр}}{[t]} = \frac{42,9 \cdot 10^{-3}}{120} 10^6 = 357,5 \text{ см}^3.$$

З іншого боку

$$W_r = \frac{pD^3}{16} = 357,5,$$

звідки

$$D = \sqrt[3]{\frac{16 \cdot 357,5}{p}} = \sqrt[3]{1821,7} = 12,2 \text{ см}.$$

Беремо $D=130 \text{ мм}$, $d=0,7D=0,7 \cdot 130=91 \text{ мм}$.

Визначимо d по моменту на другій ділянці:

$$W_r^{II} = \frac{M_{кр}}{[t]} = \frac{7,1 \cdot 10^{-3}}{120} 10^6 = 59,2 \text{ см}^3;$$

$$\frac{pd^3}{16} = 59,2 \text{ а } d = \sqrt[3]{\frac{59,2 \cdot 16}{p}} = \sqrt[3]{301,7} = 6,7 \text{ см.}$$

Беремо $d = 70 \text{ мм}$ $D = \frac{70}{0,7} = 100 \text{ мм.}$

Як розрахункові приймаємо більші діаметри: $D = 130 \text{ мм}$, $d = 91 \text{ мм}$.

Визначаємо максимальні дотичні напруження:

$$t_{max}^I = \frac{M_{кр}^I}{W_r^I} = \frac{42,9 \cdot 10^{-3}}{431 \cdot 10^{-6}} = 99,5 \text{ МПа,}$$

$$W_r^I = \frac{pD^3}{16} = \frac{p \cdot 13^3}{16} = 431 \text{ см}^3,$$

$$t_{max}^{II} = \frac{M_{кр}^{II}}{W_r^{II}} = \frac{7,1 \cdot 10^{-3}}{124,5 \cdot 10^{-6}} = 57 \text{ МПа,}$$

$$W_r^{II} = \frac{p \cdot 9,1^3}{16} = 124,5 \text{ см}^3,$$

$$t_{max}^{III} = \frac{M_{кр}^{III}}{W_r^{III}} = \frac{7,1 \cdot 10^{-3}}{431 \cdot 10^{-6}} = 16,5 \text{ МПа.}$$

Будуємо епюру дотичних напружень (рис.4.6, в).

5 РОЗРАХУНОК БАЛОК

5.1 Поняття про згинання. Основні залежності при згинанні

Розрізняють чисте і поперечне згинання.

Чистим згинанням називається вид вантаження, при якому в поперечних перерізах виникають тільки згинальні моменти.

Якщо окрім згинальних моментів виникають і поперечні сили, то *згинання називають поперечним*.

Стрижень, що працює на згинання, називається *балкою* (рис. 5.1).

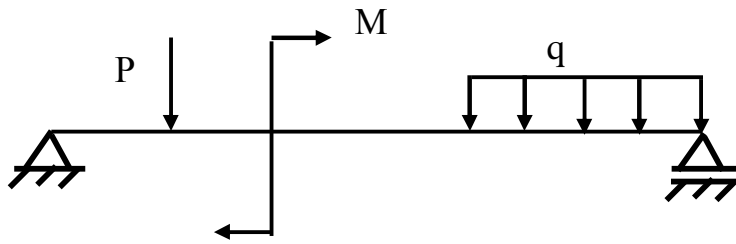


Рисунок 5.1 - Схема балки

При згинанні в балці виникають нормальні і дотичні напруження. Нормальні напруження визначаються за формулою

$$S = \frac{M_x y}{J_x}, \quad (5.1)$$

де M_x – згинальний момент відносно осі x (рис. 5.2);

y – відстань від головної центральної осі x до точки, в якій визначається напруження;

J_x – момент інерції перерізу відносно осі x . У загальному випадку він рівний

$$J_x = \int_F y^2 dF.$$

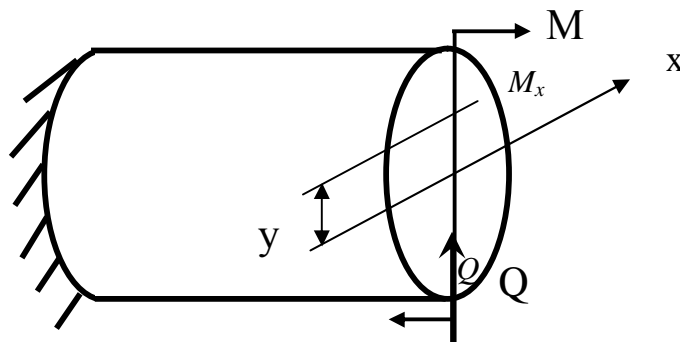


Рисунок 5.2 – Внутрішні зусилля балки

Для простих перерізів J_x рівний:

– для прямокутника $J_x = \frac{bh^3}{12}$; – для круга $J_x = \frac{pd^4}{64}$.

Дотичні напруження при згинанні визначаються за формулою Журавського:

$$t = \frac{QS_x^*}{bJ_x}, \quad (5.2)$$

де Q – поперечна сила в перерізі, де визначаються дотичні напруження;

$S_x^* = y_c^* F^*$ – статичний момент відсіченої частини перерізу, розташованої вище за рівень y , де визначається напруження (рис.5.3);

b – ширина перерізу в тому місці, де визначається напруження.

При розрахунку балки необхідно виконати наступні умови:

а) основна умова міцності

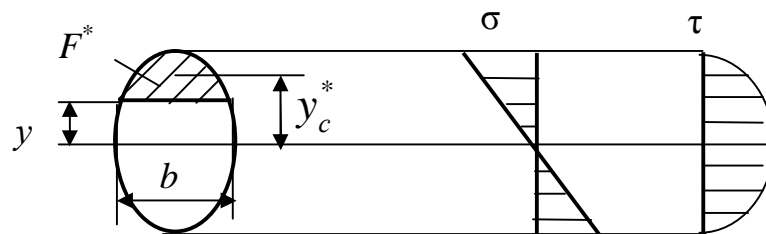


Рисунок 5.3 – Епюри нормальних і дотичних напружень при поперечному згинанні

$$s_{max} = \frac{M_x}{W_x} \leq [s]. \quad (5.3)$$

де W_x – момент опору перерізу відносно осі x .

У загальному випадку $W_x = \frac{J_x}{y_{max}}$,

для прямокутника $W_x = \frac{bh^2}{6}$,

для круга $W_x = \frac{pd^3}{32} = 0,1d^3$;

б) умова міцності за дотичними напруженнями

$$t_{max} = \frac{Q_{max} S_x^{omc}}{bJ_x} \leq [t], \quad (5.4)$$

де $[t] = (0,5..0,6)[\sigma]$ – допустимі дотичні напруження;

в) за еквівалентним напруженням:

– за третьою теорією міцності

$$s_{\text{екв}}^{\text{III}} = \sqrt{s^2 + 4t^2} \leq [s]; \quad (5.5)$$

– за четвертою теорією міцності

$$s_{\text{екв}}^{\text{IV}} = \sqrt{s^2 + 3t^2} \leq [s]. \quad (5.6)$$

5.2 Функція переміщень при згинанні

Для балки, яка представлена на рис 5.4, як кінцевий елемент береться стрижень довжиною l , розташований між характерними перерізами, а навантаження прикладені по кінцях елемента (рис.5.5).

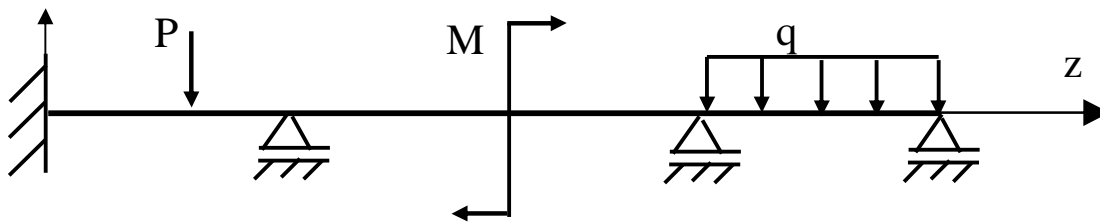


Рисунок 5.4 - Схема балки

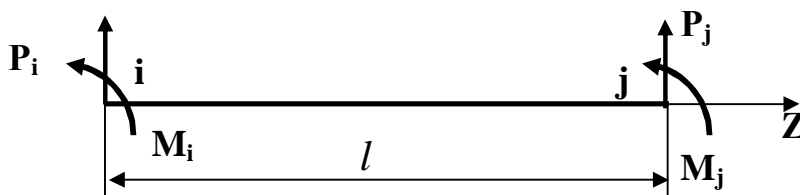


Рисунок 5.5 - Елемент балки з додатними напрямками вузлових навантажень

Покажемо елемент після деформації (рис.5.6).

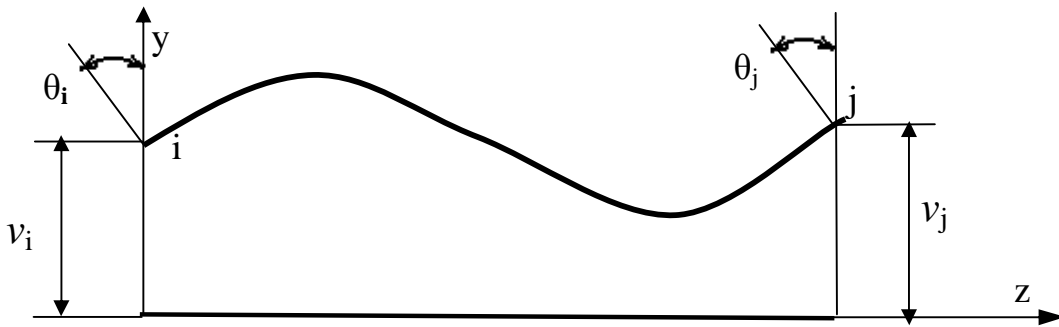


Рисунок 5.6 - Кінцевий елемент після деформації

Як бачимо, кожен вузол має два степені вільності, а в елементі два вузли. Значить, функція переміщень для елемента є поліномом (5.7), що містить чотири коефіцієнти:

$$v = a_0 + a_1 z + a_2 z^2 + a_3 z^3. \quad (5.7)$$

Коефіцієнти a_0 , a_1 , a_2 , a_3 визначаємо через переміщення вузлів v_i , u_i , v_j і u_j .

При $z=0$ $v=v_i$, $u=u_i$. При $z=l$ $v=v_j$, $u=u_j$.

При згинанні

$$q = \frac{dv}{dz}, \quad u = a_1 + 2a_2 z + 3a_3 z^2. \quad (5.8)$$

Підставивши у вирази (5.7) і (5.8) значення переміщень для вузлів i і j , одержимо:

$$v_i = a_0 \longrightarrow a_0 = v_i, \quad \theta_i = a_1 \longrightarrow a_1 = \theta_i,$$

$$v_j = v_i + q_s l + a_2 l^2 + a_3 l^3,$$

$$q_j = q_i + 2a_2 l + 3a_3 l^2,$$

$$\begin{cases} a_2 l^2 + a_3 l^3 = v_j - v_i - q_i l; \\ 2a_2 l + 3a_3 l^2 = q_j - q_i. \end{cases}$$

За допомогою визначників знаходимо:

$$a_2 = \frac{\begin{vmatrix} v_j - v_i - q_i l & l^3 \\ q_j - q_i & 3l^2 \end{vmatrix}}{\begin{vmatrix} l^2 & l^3 \\ 2l & 3l^2 \end{vmatrix}} = \frac{1}{l^4} [3l^2(v_j - v_i - q_i l) - l^3(q_j - q_i)],$$

$$a_3 = \frac{1}{l^4} \begin{vmatrix} l^2 & v_j - v_i - q_i l \\ 2l & q_j - q_i \end{vmatrix} = \frac{1}{l^4} [(q_j - q_i)l^2 - 2l(v_j - v_i - q_i l)].$$

Підставляємо знайдені коефіцієнти у вираз (5.7):

$$v = v_i + q_i z + \frac{1}{l^4} [3l^2(v_j - v_i - q_i l) - l^3(q_j - q_i)] z^2 + \\ + \frac{1}{l^4} [(q_j - q_i)l^2 - 2l(v_j - v_i - q_i l)] z^3$$

або

$$v = \left[1 - 3\frac{z^2}{l^2} + 2\frac{z^3}{l^3} \right] \cdot v_i + \left[z - 2\frac{z^2}{l} + \frac{z^3}{l^2} \right] q_i + \\ + \left[3\frac{z^2}{l^2} - 2\frac{z^3}{l^3} \right] \cdot v_j + \left[-\frac{z^2}{l} + \frac{z^3}{l^2} \right] q_j.$$

Запишемо цей вираз в матричній формі:

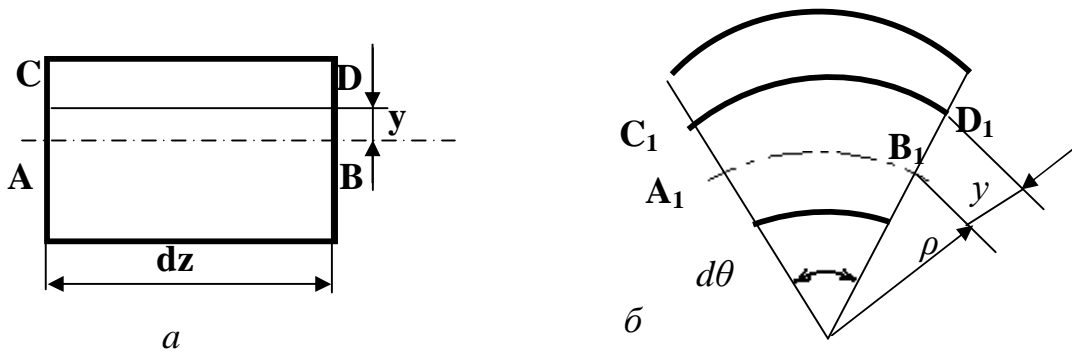
$$v = \left[1 - 3\frac{z^2}{l^2} + 2\frac{z^3}{l^3}; \quad z - 2\frac{z^2}{l} + \frac{z^3}{l^2}; \quad 3\frac{z^2}{l^2} - 2\frac{z^3}{l^3}; \quad -\frac{z^2}{l} + \frac{z^3}{l^2} \right] \cdot \begin{Bmatrix} v_i \\ q_i \\ v_j \\ q_j \end{Bmatrix}. \quad (5.9)$$

5.3 Вектори деформацій, напружень і зусиль

Встановимо зв'язок між деформаціями і переміщеннями при згинанні. Для цього покажемо елемент балки довжиною dz до і після деформації (рис.5.7).

Відносне подовження довільного волокна CD дорівнює

$$e = \frac{C_1 D_1 - CD}{CD},$$



a – до деформації, *б* – після деформації.
Рисунок 5.7- Елемент балки довжиною *dz*

$$CD=AB=A_1B_1=\rho \cdot d\theta,$$

$$C_1D_1 = (\rho + y) d\theta,$$

$$e = \frac{(r + y) \cdot dq - r \cdot dq}{r \cdot dq} = \frac{y}{r}.$$

З математики відома залежність

$$\frac{1}{r} = \frac{\frac{d^2v}{dz^2}}{\left[1 + \left(\frac{dv}{dz}\right)^2\right]^{3/2}} \approx \frac{d^2v}{dz^2}.$$

Таким чином, маємо

$$e = y \frac{d^2v}{dz^2}.$$

Узявши другу похідну виразу (5.8), одержимо:

$$\{e\} = y \left| -\frac{6}{l^2} + \frac{12z}{l^3}; -\frac{4}{l} + \frac{6z}{l^2}; \frac{6}{l^2}; \frac{6}{l^2} - \frac{12z}{l^3}; -\frac{2}{l} + \frac{6z}{l^2} \right| \begin{Bmatrix} v_i \\ q_i \\ v_j \\ q_j \end{Bmatrix}, \quad (5.10)$$

$$\{e\} = |B| \cdot \begin{Bmatrix} v_i \\ \mathbf{q}_i \\ v_j \\ \mathbf{q}_j \end{Bmatrix},$$

тут

$$|B| = y \cdot \left| -\frac{6}{l^2} + \frac{12z}{l^3}; \quad -\frac{4}{l} + \frac{6z}{l^2}; \quad \frac{6}{l^2} - \frac{12z}{l^3}; \quad -\frac{2}{l} + \frac{6z}{l^2} \right|. \quad (5.11)$$

На підставі закону Гука одержимо вектор напруження

$$\{S\} = E\{e\} = yE \left| -\frac{6}{l^2} + \frac{12z}{l^3}; \quad -\frac{4}{l} + \frac{6z}{l^2}; \quad \frac{6}{l^2} - \frac{12z}{l^3}; \quad -\frac{2}{l} + \frac{6z}{l^2} \right| \begin{Bmatrix} v_i \\ \mathbf{q}_i \\ v_j \\ \mathbf{q}_j \end{Bmatrix}. \quad (5.12)$$

З формули (5.1) згинальний момент дорівнює

$$M_x = J_x \{S\} \frac{1}{y}.$$

Підставивши сюди вираз (5.10), одержимо вектор зусиль

$$\{M_x\} = EJ_x \left| -\frac{6}{l^2} + \frac{12z}{l^3}; \quad -\frac{4}{l} + \frac{6z}{l^2}; \quad \frac{6}{l^2} - \frac{12z}{l^3}; \quad -\frac{2}{l} + \frac{6z}{l^2} \right| \begin{Bmatrix} v_i \\ \mathbf{q}_i \\ v_j \\ \mathbf{q}_j \end{Bmatrix}. \quad (5.13)$$

5.4 Матриця жорсткості кінцевого елемента

Відповідно до виразу (1.6) знаходимо:

$$|K|^e = \int_V |B|^T |D| |B| dV = \int_0^l \int_F y \begin{vmatrix} -\frac{6}{l^2} + \frac{12z}{l^3} \\ \frac{4}{l} + \frac{6z}{l^2} \\ \frac{6}{l^2} - \frac{12z}{l^3} \\ -\frac{2}{l} + \frac{6z}{l^2} \end{vmatrix} E y \times \\ \times \left| -\frac{6}{l^2} + \frac{12z}{l^3}; -\frac{4}{l} + \frac{6z}{l^2}; \frac{6}{l^2}; \frac{6}{l^2} - \frac{12z}{l^3}; -\frac{2}{l} + \frac{6z}{l^2} \right| dz dF ;$$

$$\int_F y^2 dF = J_x - \text{момент інерції перерізу відносно осі } x.$$

Перемноживши матриці $|B|^T$, $|D|$ і $|B|$ і узявши інтеграли, одержимо матрицю жорсткості кінцевого елемента в наступному вигляді:

$$|K|^e = EJ_x \begin{vmatrix} \frac{12}{l^3} & \frac{6}{l^2} & -\frac{12}{l^3} & \frac{6}{l^2} \\ \frac{6}{l^2} & \frac{4}{l} & -\frac{6}{l^2} & \frac{2}{l} \\ -\frac{12}{l^3} & -\frac{6}{l^2} & \frac{12}{l^3} & -\frac{6}{l^2} \\ \frac{6}{l^2} & \frac{2}{l} & -\frac{6}{l^2} & \frac{4}{l} \end{vmatrix}. \quad (5.14)$$

5.5 Врахування впливу розподіленого навантаження на величину навантажень у вузлах

При складанні рівнянь МКЕ у вектор навантажень вузлів записуються значення сил і моментів у відповідних вузлах. За наявності розподіленого навантаження, яке розподілене за довжиною елемента (рис. 5.8, а), його вплив необхідно замінити вузловими силами. Для цього визначимо роботу, яку виконує навантаження q на переміщенні v .

Матриця кінцевого елемента дорівнює

$$\int_0^l q v dz = \int_0^l q \left| 1 - 3\frac{z^2}{l^2} + 2\frac{z^3}{l^3}; z - 2\frac{z^2}{l} + 3\frac{z^2}{l^2}; 3\frac{z^2}{l^2} - 2\frac{z^3}{l^3}; -\frac{z^2}{l} - \frac{z^3}{l^2} \right| dz.$$

Якщо $q = \text{const}$, то

$$\int_0^l qVdz = \left[\frac{ql}{2}; \frac{ql^2}{12}; \frac{ql}{2}; -\frac{ql^2}{12} \right].$$

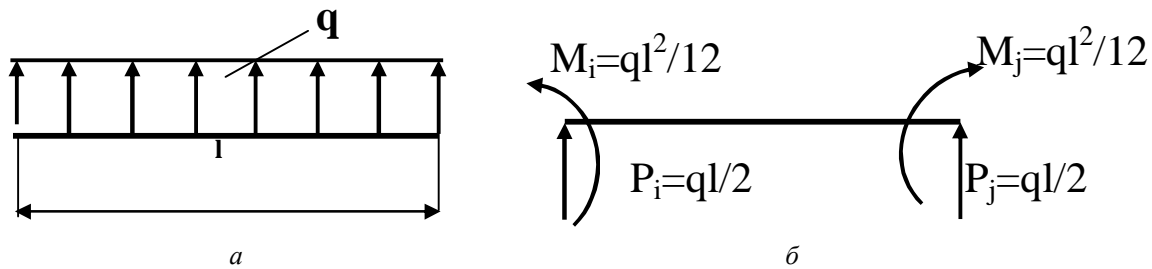


Рисунок 5.8 – Заміна дії рівномірного навантаження вузловими силами

Це означає, що вплив розподіленого навантаження з енергетичної точки зору рівносильний впливу вузлових навантажень: у лівому вузлі $P_i = ql/2$, $M_i = ql^2/12$, в правому $P_j = ql/2$, $M_j = -ql^2/12$ (рис. 5.8, б).

Після рішення задачі МКЕ і визначення переміщень вузлів навантаження за вузлами кінцевих елементів визначаються так:

$$\begin{Bmatrix} P_i \\ M_i \\ P_j \\ M_j \end{Bmatrix} = |K|^e \begin{Bmatrix} v_i \\ q_i \\ v_j \\ q_j \end{Bmatrix} - \begin{Bmatrix} \frac{ql}{2} \\ \frac{ql^2}{12} \\ \frac{ql}{2} \\ -\frac{ql^2}{12} \end{Bmatrix}. \quad (5.15)$$

5.6 Приклад

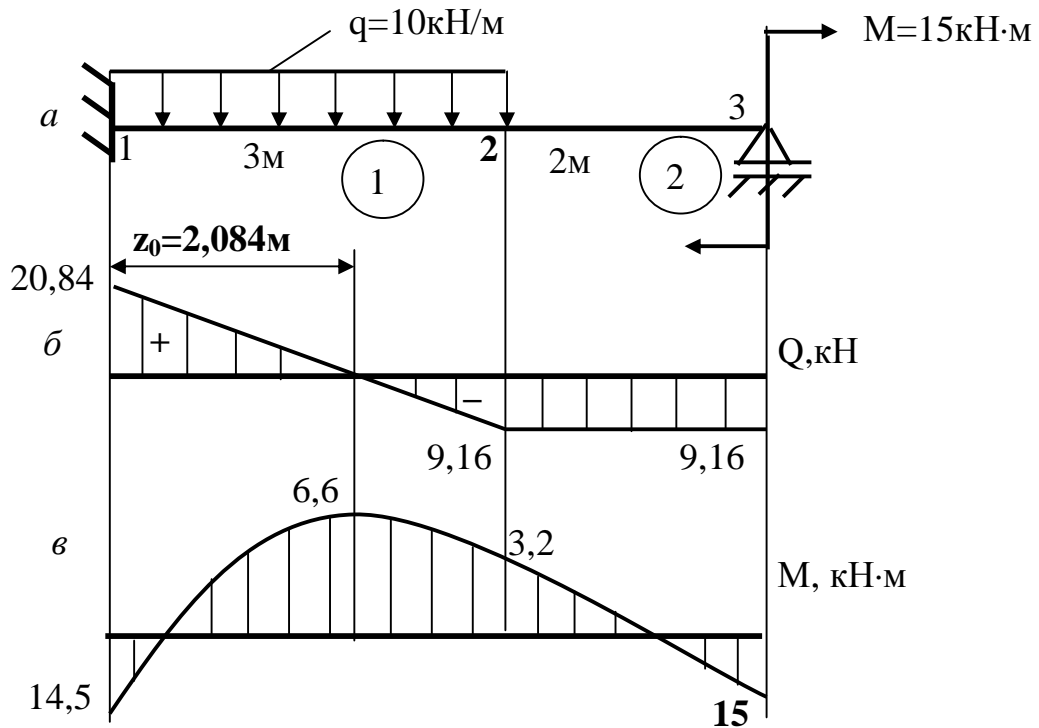
Для балки, показаної на рис.5.9, а, побудувати епюри поперечних сил і згинальних моментів.

Розв'язання. Нумеруємо вузли і елементи (елементи обведені кухлями). Обчислюємо матриці жорсткості елементів за формулою (5.13). До елемента 1 входять вузли 1 і 2, його довжина – 3м.

$$|K|_I^e = EJ_x \begin{vmatrix} 0,44 & 0,67 & -0,44 & 0,67 \\ 0,67 & 1,33 & -0,67 & 0,67 \\ -0,44 & -0,67 & 0,44 & -0,67 \\ 0,67 & 0,67 & -0,67 & 1,33 \end{vmatrix}.$$

Елемент 2: входять вузли 2 і 3, його довжина – 2м.

$$|K|_{II}^e = EJ_x \begin{vmatrix} 1,5 & 1,5 & -1,5 & 1,5 \\ 1,5 & 2 & -1,5 & 1 \\ -1,5 & -1,5 & 1,5 & -1,5 \\ 1,5 & 1 & -1,5 & 2 \end{vmatrix}.$$



*a – задана балка; б – епюра поперечних сил;
в – епюра згинальних моментів*

Рисунок 5.9 – Схема балки і епюри внутрішніх зусиль

Тепер формуємо матрицю жорсткості конструкції. Оскільки кожен вузол має два степені вільності, а вузлів 3, то розмірність матриці жорсткості конструкції 6×6 .

$$EJ_x \begin{bmatrix} 0,44 & 0,67 & -0,44 & 0,67 & 0 & 0 \\ 0,67 & 1,33 & -0,67 & 0,67 & 0 & 0 \\ -0,44 & -0,67 & 0,44 & -0,67 & -1,5 & 1,5 \\ 0,67 & 0,67 & -0,67 & 1,33 & -1,5 & 1 \\ 0 & 0 & -1,5 & -1,5 & 1,5 & -1,5 \\ 0 & 0 & 1,5 & 1 & -1,5 & 2 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ q_1 \\ v_2 \\ q_2 \\ v_3 \\ q_3 \end{bmatrix} = \begin{bmatrix} -\frac{ql}{2} \\ \frac{ql^2}{12} \\ -\frac{ql}{2} \\ \frac{ql^2}{12} \\ 0 \\ -15 \end{bmatrix} = \begin{bmatrix} -15 \\ -7,5 \\ -15 \\ 7,5 \\ 0 \\ -15 \end{bmatrix}$$

При записі вектора навантажень враховуємо, що на перший елемент діє розподілене навантаження. Його дію замінюємо вузловими навантаженнями, як було сказано вище. У даному випадку знак розподіленого навантаження беремо з мінусом, оскільки воно направлено вниз.

Враховуємо граничні умови: $v_1=0$, $\theta_1=0$, $v_3=0$.

Після їх врахування одержимо систему рівнянь:

$$EJ_x \begin{bmatrix} 1,94 & 0,83 & 1,5 \\ 0,83 & 3,33 & 1 \\ 1,5 & 1 & 2 \end{bmatrix} \begin{bmatrix} v_2 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} -15 \\ 7,5 \\ -15 \end{bmatrix}$$

Її рішення:

$$v_2 = -\frac{5,1}{EJ_x}; \quad q_2 = \frac{5,4}{EJ_x}; \quad q_3 = -\frac{6,4}{EJ_x}$$

Визначаємо зусилля по кінцях елементів.

Розглядаємо 1-й елемент:

$$\begin{bmatrix} P_1 \\ M_1 \\ P_2 \\ M_2 \end{bmatrix} = |K|_I^e \cdot \begin{bmatrix} v_1 \\ q_1 \\ v_2 \\ q_2 \end{bmatrix} = \begin{bmatrix} -\frac{ql}{2} \\ -\frac{ql^2}{12} \\ -\frac{ql}{2} \\ -\frac{ql^2}{12} \end{bmatrix}$$

$$= EJ_x \begin{vmatrix} 0,44 & 0,67 & -0,44 & 0,67 \\ 0,67 & 1,33 & -0,67 & 0,67 \\ -0,44 & -0,67 & 0,44 & -0,67 \\ 0,67 & 0,67 & -0,67 & 1,33 \end{vmatrix} \begin{Bmatrix} 0 \\ 0 \\ -\frac{5,1}{EI_x} \\ \frac{5,4}{EI_x} \end{Bmatrix} - \begin{Bmatrix} -15 \\ -7,5 \\ -15 \\ 7,5 \end{Bmatrix} = \begin{Bmatrix} 20,84 \\ 14,5 \\ 9,16 \\ 3,2 \end{Bmatrix}$$

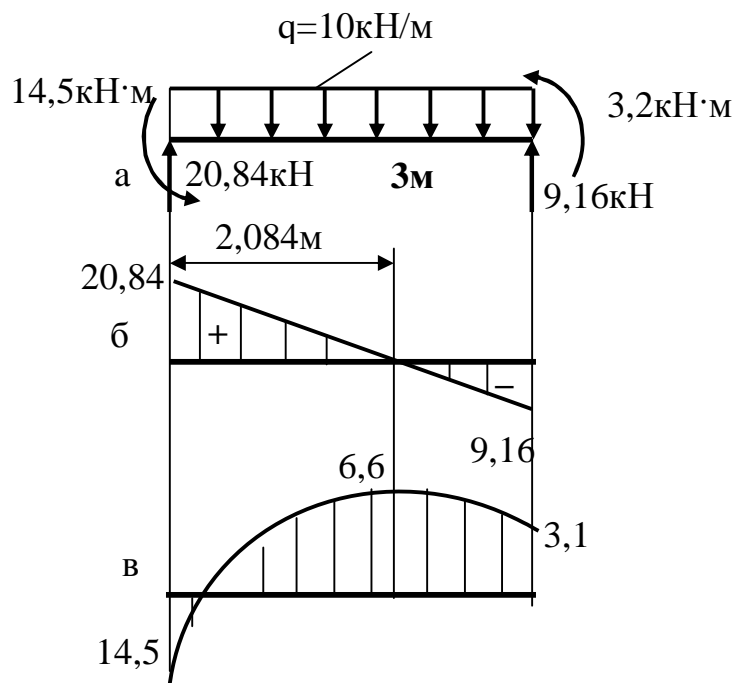
Прикладаємо одержані вузлові сили до вузлів 1-го елемента (рис.5.10, а).

Для контролю необхідно виконати перевірки: $\sum P_y = 0$, $\sum M_a = 0$.

$$20,84 + 9,16 - 10 \cdot 3 = 0,$$

$$14,5 - 20,84 \cdot 3 + 10 \cdot 3 \cdot 1,5 + 3,2 = 0,$$

$$14,5 - 62,52 + 45 + 3,2 = 0.$$



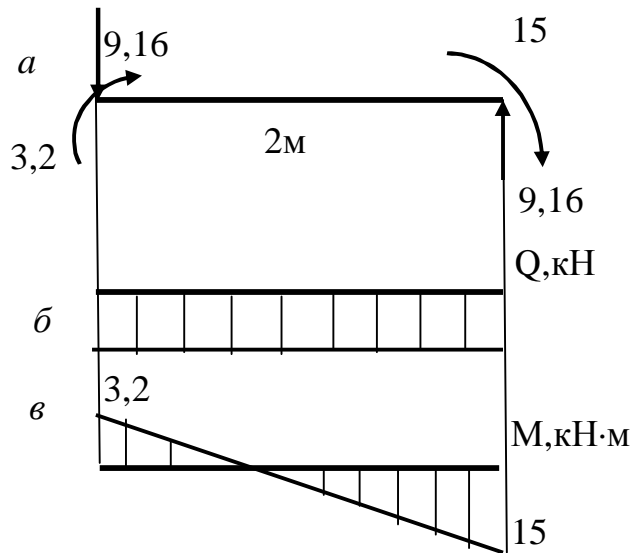
а – 1-й елемент з зовнішнім навантаженням і вузловими зусиллями;

б – еюра поперечних сил; в – еюра згинальних моментів

Рисунок 5.10 – Елемент 2 з еярами Q і M_x

Розглядаємо 2-й елемент:

$$\begin{Bmatrix} P_2 \\ M_2 \\ P_3 \\ M_3 \end{Bmatrix} = |K|_{II}^I \begin{Bmatrix} v_2 \\ q_2 \\ v_3 \\ q_3 \end{Bmatrix} = EJ_x \begin{vmatrix} 1,5 & 1,5 & -1,5 & 1,5 \\ 1,5 & 2 & -1,5 & 1 \\ -1,5 & -1,5 & 1,5 & -1,5 \\ 1,5 & 1 & -1,5 & 2 \end{vmatrix} \begin{Bmatrix} \frac{5,1}{EI_x} \\ \frac{5,4}{EI_x} \\ \frac{EI_x}{EI_x} \\ \frac{6,4}{EI_x} \end{Bmatrix} = \begin{Bmatrix} -9,16 \\ -3,2 \\ 9,16 \\ -15 \end{Bmatrix}$$



*a – елемент з діючими навантаженнями; б – еюра поперечних сил;
в – еюра згинальних моментів*

Рисунок 5.11– Елемент 2 з еюрами Q і M

Прикладаємо одержані вузлові сили до вузлів 2-го елемента (рис.5.11, а).

Зістикувавши еюри за ділянками, одержуємо остаточні еюри поперечних сил (див. рис.5.9, б) і згинальних моментів (див. рис.5.9, в).

6 РОЗРАХУНОК ПЛОСКИХ РАМ

6.1 Згинання з розтяганням плоского стрижня

Стрижні рами випробовують поперечне згинання з розтяганням (рис. 6.1).

Напруження в довільній точці перерізу дорівнюють

$$S = \frac{N}{F} + \frac{M_x y}{J_x} \quad (6.1)$$

Цей вид вантаження слід розглядати як комбінацію розтягання і згинання.

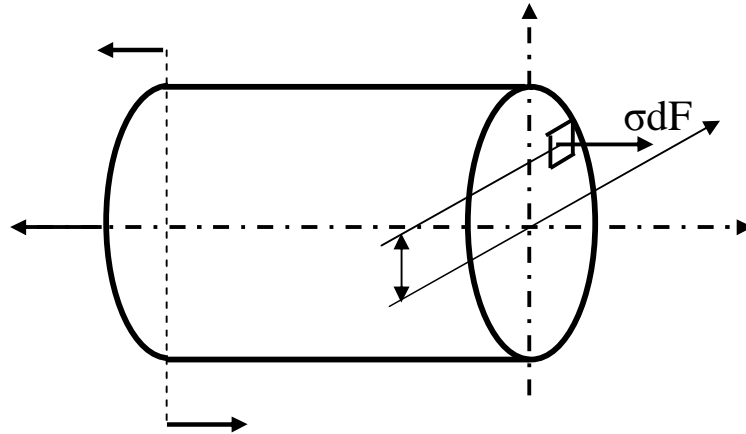


Рисунок 6.1 – Розрахункова схема стрижня, який знаходиться під дією згинання з розтяганням

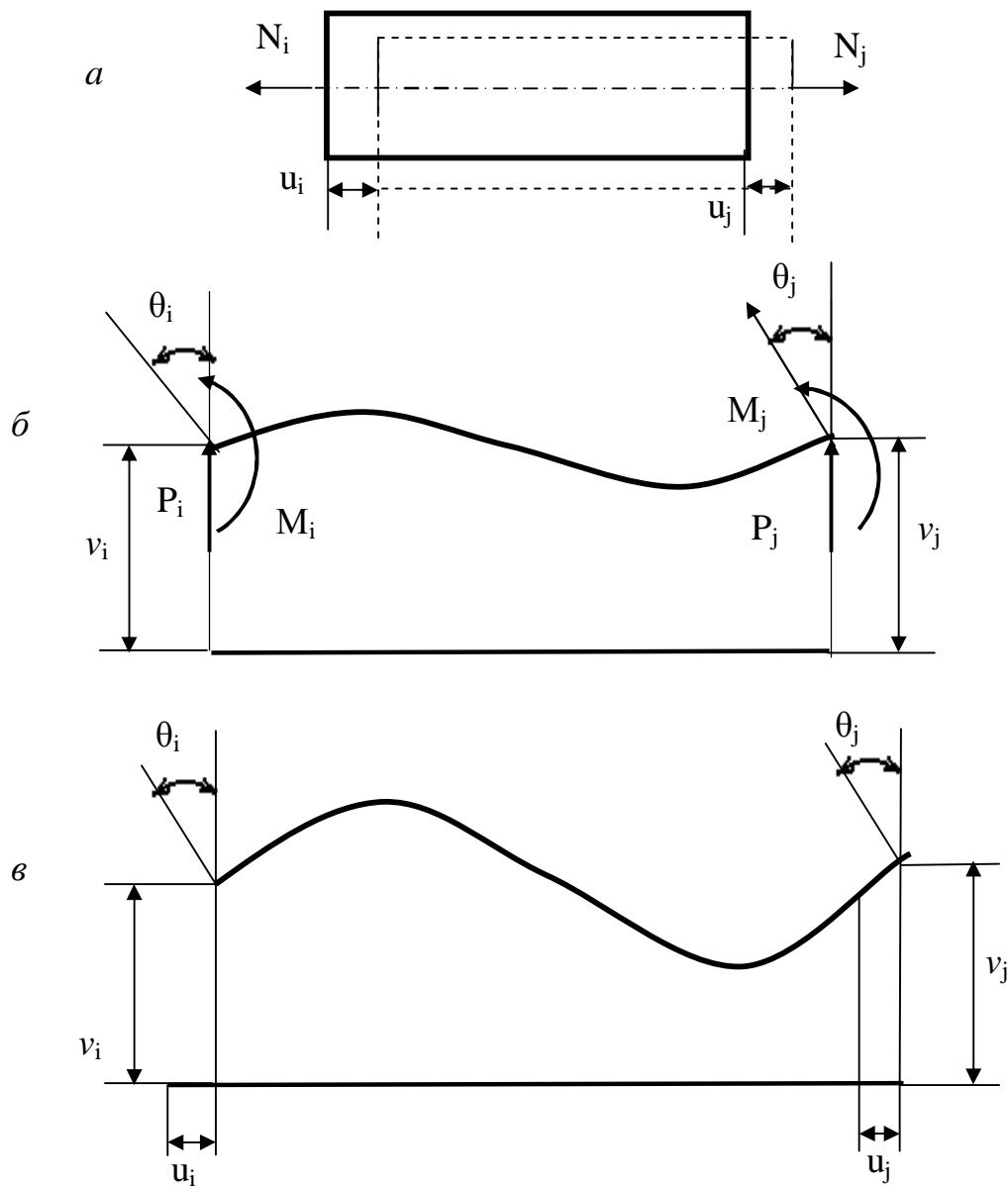
6.2 Матриця жорсткості в локальних координатах для кінцевого елемента при згинанні з розтяганням

При розтяганні (рис.6.2, а) вектор переміщення вузлів елемента і матриця жорсткості мають вигляд

$$\{d\}^e = \begin{Bmatrix} u_i \\ u_j \end{Bmatrix}, \quad |K|^e = \frac{EF}{l} \begin{vmatrix} 1 & -1 \\ -1 & 1 \end{vmatrix},$$

а при згинанні (рис. 6.2, б) вектор переміщення вузлів і матриця жорсткості виглядають так

$$\{d\}^e = \begin{Bmatrix} u_i \\ q_i \\ u_j \\ q_j \end{Bmatrix}, \quad |K|^e = EJ_x \begin{vmatrix} \frac{12}{l^3} & \frac{6}{l^2} & -\frac{12}{l^3} & \frac{6}{l^2} \\ \frac{6}{l^2} & \frac{4}{l} & -\frac{6}{l^2} & \frac{2}{l} \\ -\frac{12}{l^3} & -\frac{6}{l^2} & \frac{12}{l^3} & -\frac{6}{l^2} \\ \frac{6}{l^2} & \frac{2}{l} & -\frac{6}{l^2} & \frac{4}{l} \end{vmatrix}.$$



*a – розтягання стрижня; б – згинання стрижня;
в – сумісна дія згинання з розтяганням*

Рисунок 6.2 – Деформація кінцевого елемента при сумісній дії розтягання і згинання

При сумісній дії згинання і розтягання вектор переміщень і матрицю жорсткості одержуємо шляхом накладання відповідно векторів переміщень і матриць жорсткості при розтяганні і при згинанні

$$\{d\}^e = \begin{Bmatrix} u_i \\ v_i \\ q_i \\ u_j \\ v_j \\ q_j \end{Bmatrix};$$

$$|K|^e = \begin{pmatrix} \frac{EF}{l} & 0 & 0 & -\frac{EF}{l} & 0 & 0 \\ 0 & \frac{12EJ_x}{l^3} & \frac{6EJ_x}{l^2} & 0 & -\frac{12EJ_x}{l^3} & \frac{6EJ_x}{l^2} \\ 0 & \frac{6EJ_x}{l^2} & \frac{4EJ_x}{l} & 0 & -\frac{6EJ_x}{l^2} & \frac{2EJ_x}{l} \\ -\frac{EF}{l} & 0 & 0 & \frac{EF}{l} & 0 & 0 \\ 0 & -\frac{12EJ_x}{l^3} & -\frac{6EJ_x}{l^2} & 0 & \frac{12EJ_x}{l^3} & -\frac{6EJ_x}{l^2} \\ 0 & \frac{6EJ_x}{l^2} & \frac{2EJ_x}{l} & 0 & -\frac{6EJ_x}{l^2} & \frac{4EJ_x}{l} \end{pmatrix}. \quad (6.2)$$

6.3 Матриця жорсткості для стрижнів плоскої рами

Рама (рис. 6.3) – це стрижнева система, стрижні якої у всіх або деяких вузлах жорстко сполучені один з одним. Якщо стрижнева система знаходиться в одній площині і в цій площині діють навантаження, то рама називається плоскою.

У цьому випадку вводимо дві системи координат: глобальну і загальну \bar{y} і \bar{z} для всієї рами і локальну (місцеву) y , z для кожного стрижня.

Як і для ферми, матриця жорсткості в глобальній системі координат виражається через матрицю жорсткості в локальних координатах (6.2) і матрицю направляючих косинусів.

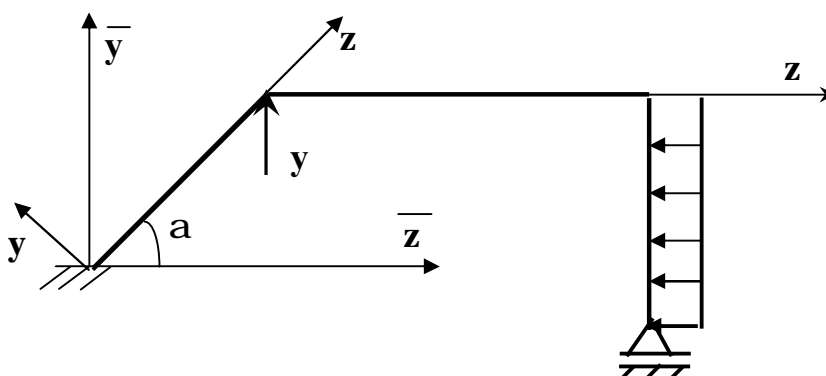


Рисунок 6.3 – Плоска рама

$$|\bar{K}|^e = |I|^T |K|^e |I|. \quad (6.3)$$

Встановимо зв'язок між переміщеннями в локальній і глобальній системах координат для стрижня загального положення (рис.6.4). З рис. 6.4 видно, лівий вузол стрижня після деформації перейде в точку i , і його переміщення в локальній системі координат будуть u_i і v_i , а в глобальній - \bar{u} і \bar{v}_i

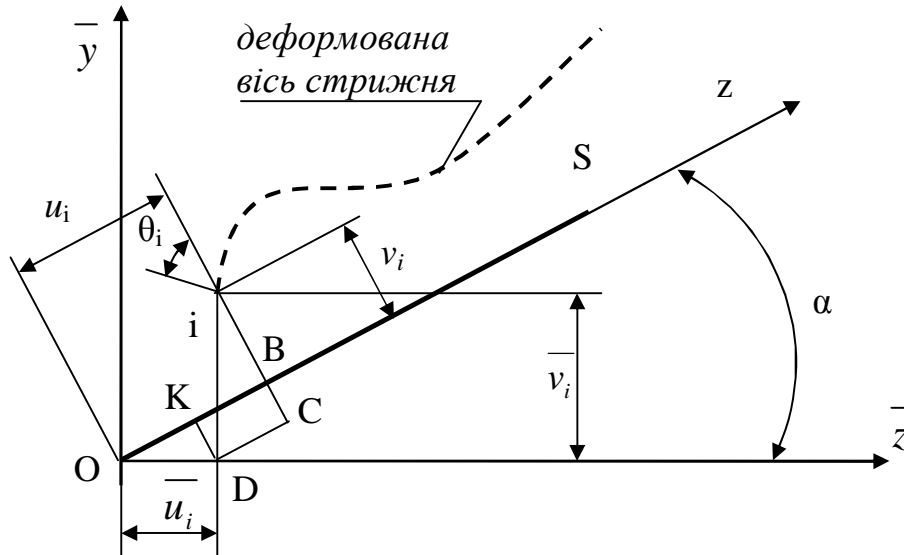


Рисунок 6.4 – Переміщення елемента в локальній і глобальній системах координат

$$u_i = OB = OK + KB,$$

$$OK = \bar{u} \cos \alpha, \quad BK = DC = \bar{v}_i \sin \alpha, \quad u_i = \bar{u} \cos \alpha + \bar{v}_i \sin \alpha, \quad v_i = AB = AC - BC$$

$$AC = \bar{v}_i \cos \alpha, \quad BC = KD = \bar{u}_i \sin \alpha, \quad v_i = -\bar{u}_i \sin \alpha + \bar{v}_i \cos \alpha.$$

Аналогічний зв'язок для вузла j :

$$u_j = \bar{u}_j \cos \alpha + \bar{v}_j \sin \alpha, \quad v_j = -\bar{u}_j \sin \alpha + \bar{v}_j \cos \alpha.$$

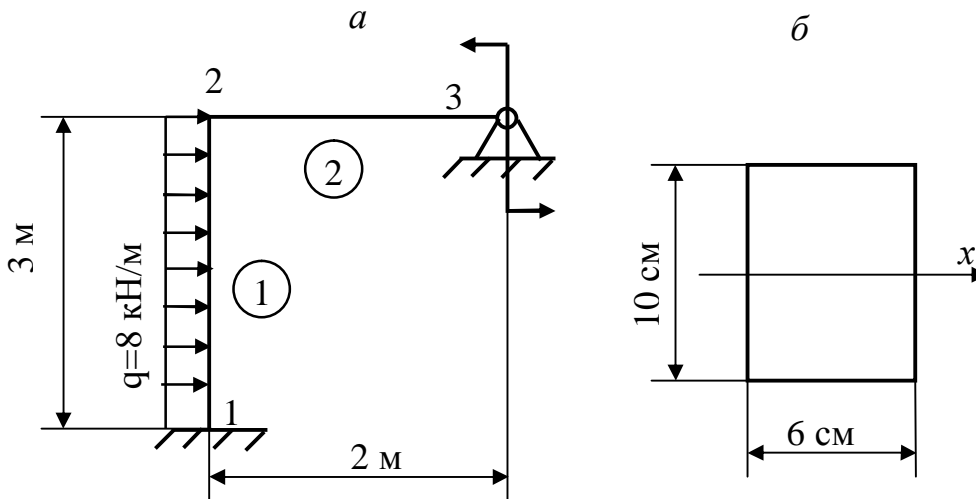
Кути повороту в обох системах координат однакові, тобто $\bar{q}_i = q_i$, $\bar{q}_j = q_j$.

У матричній формі маємо

$$|I| = \begin{vmatrix} \cos \alpha & \sin \alpha & 0 & 0 & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & 0 & 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}. \quad (6.4)$$

6.4 Приклад

Для рами, показаної на рис.6.5, побудувати епюри подовжніх і поперечних сил і згинальних моментів. Визначити максимальні нормальні напруження, якщо рама має прямокутний поперечний переріз ($h=10\text{см}$, $b=6\text{см}$).



*а – схема рами; б – поперечний переріз рами
Рисунок 6.5 – Рами з навантаженням*

Розв'язання. Нумеруємо вузли і елементи. Визначаємо геометричні характеристики перерізу (рис.6.5,б).

$$F=6 \cdot 10=60 \text{ см}^2 \quad J_x=6 \cdot 10^3:12=500 \text{ см}^4.$$

Визначаємо жорсткості елементів при розтяганні і згинанні:

$$EF=2 \cdot 10^5 \cdot 60 \cdot 10^{-4}=1200 \text{ МН}, \quad EI_x=2 \cdot 10^5 \cdot 500=1 \text{ МН} \cdot \text{м}^2.$$

Переходимо до визначення матриць жорсткості елементів.

1-й елемент (входять вузли 1-2): $\alpha = 90^\circ$.

Матриця направляючих косинусів для нього:

$$|I| = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix},$$

$$|K|_I^e = \begin{vmatrix} 400 & 0 & 0 & -400 & 0 & 0 \\ 0 & 0,44 & 0,67 & 0 & -0,44 & 0,67 \\ 0 & 0,67 & 1,33 & 0 & -0,67 & 0,67 \\ -400 & 0 & 0 & 400 & 0 & 0 \\ 0 & -0,44 & -0,67 & 0 & 0,44 & -0,67 \\ 0 & 0,67 & 0,67 & 0 & -0,67 & 1,33 \end{vmatrix},$$

$$|\overline{K}|^e = |I|^T |K|^e |I| = \begin{vmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \times$$

$$\times \begin{vmatrix} 400 & 0 & 0 & -400 & 0 & 0 \\ 0 & 0,44 & 0,67 & 0 & -0,44 & 0,67 \\ 0 & 0,67 & 1,33 & 0 & -0,67 & 0,67 \\ -400 & 0 & 0 & 400 & 0 & 0 \\ 0 & -0,44 & -0,67 & 0 & 0,44 & -0,67 \\ 0 & 0,67 & 0,67 & 0 & -0,67 & 1,33 \end{vmatrix} \cdot \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} =$$

$$= \begin{vmatrix} 0,44 & 0 & -0,67 & -0,44 & 0 & -0,67 \\ 0 & 400 & 0 & 0 & -400 & 0 \\ -0,67 & 0 & 1,33 & 0,67 & 0 & 0,67 \\ -0,44 & 0 & 0,67 & 0,44 & 0 & 0,67 \\ 0 & -400 & 0 & 0 & 400 & 0 \\ -0,67 & 0 & 0,67 & 0,67 & 0 & 1,33 \end{vmatrix}.$$

2-й элемент (узлы 2-3): $\alpha=0^0$.

$$|I| = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix};$$

$$|K|_{II}^e = \begin{vmatrix} 600 & 0 & 0 & -600 & 0 & 0 \\ 0 & 1,5 & 1,5 & 0 & -1,5 & 1,5 \\ 0 & 1,5 & 2 & 0 & -1,5 & 1 \\ -600 & 0 & 0 & 600 & 0 & 0 \\ 0 & -1,5 & -1,5 & 0 & 1,5 & -1,5 \\ 0 & 1,5 & 1 & 0 & -1,5 & 2 \end{vmatrix}.$$

Оскільки $|\lambda|$ є одиничною матрицею, то $|\overline{K}|_{II}^e = |K|_{II}^e$.

Тепер формуємо матрицю жорсткості конструкції. У рамі маємо три вузли, кожний з яких має три степені свободи. Отже, матриця жорсткості конструкції має розмірність 9×9 .

0,44	0	-0,67	-0,44	0	-0,67	0	0	0	$\left. \begin{matrix} \overline{u_1} \\ \overline{v_1} \\ \overline{q_1} \\ \overline{u_2} \\ \overline{v_2} \\ \overline{q_2} \\ \overline{u_3} \\ \overline{v_3} \\ \overline{q_3} \end{matrix} \right\} = \left\{ \begin{matrix} 12 \\ 0 \\ -6 \\ 12 \\ 0 \\ 6 \\ 0 \\ 0 \\ 10 \end{matrix} \right\} \cdot 10^{-3}.$
0	400	0	0	-400	0	0	0	0	
-0,67	0	1,33	0,67	0	0,67	0	0	0	
-0,44	0	0,67	0,44	0	0,67	-600	0	0	
0	-400	0	0	400	0	0	-1,5	1,5	
-0,67	0	0,67	0,67	0	1,33	0	-1,5	1	
0	0	0	-600	0	0	600	0	0	
0	0	0	0	-1,5	-1,5	0	1,5	-1,5	
0	0	0	0	1,5	1	0	-1,5	2	

При записі вектора вузлових сил враховуємо, що на 1-й елемент діє розподілене навантаження. Отже, у першому і другому вузлах повинні бути прикладені сили в горизонтальному напрямку $ql/2=8\cdot3/2=12\text{кН}$ і моменти $-ql^2/12=-8\cdot3^2/12=-6\text{кН}\cdot\text{м}$ і $ql^2/12=8\cdot3^2/12=6\text{кН}\cdot\text{м}$.

У третьому вузлі прикладений момент 15 кН·м. Оскільки коефіцієнти матриці жорсткості мають одиниці вимірювання Меганьютон (МН) і метр (м), то вектор вузлових сил помножимо на 10^{-3} .

Граничні умови: $\bar{u}_1 = 0$; $\bar{v}_1 = 0$; $q_1 = 0$; $\bar{u}_3 = 0$; $\bar{v}_3 = 0$.

Після врахування граничних умов маємо:

$$\begin{vmatrix} 600,44 & 0 & 0,67 & 0 \\ 0 & 401,5 & 1,5 & 1,5 \\ 0,67 & 1,5 & 3,33 & 1 \\ 0 & 1,5 & 1 & 2 \end{vmatrix} \begin{Bmatrix} \bar{u}_2 \\ \bar{v}_2 \\ q_2 \\ q_3 \end{Bmatrix} = \begin{Bmatrix} 12 \\ 0 \\ 6 \\ 10 \end{Bmatrix} \cdot 10^{-3}.$$

Після рішення системи рівнянь одержимо:

$$\bar{u}_2 = 19,6 \cdot 10^{-6} \text{ м}; \quad \bar{v}_2 = -19,4 \cdot 10^{-6} \text{ м}; \quad q_2 = 353 \cdot 10^{-6}; \quad q_3 = 4858 \cdot 10^{-6}.$$

Визначаємо зусилля на кінцях елементів за формулою

$$\begin{Bmatrix} P_{i\bar{z}}^- \\ P_{iy}^- \\ M_i \\ P_{j\bar{z}}^- \\ P_{jy}^- \\ M_j \end{Bmatrix} = |K|^e \cdot \begin{Bmatrix} \bar{u}_i \\ \bar{v}_i \\ q_i \\ \bar{u}_j \\ \bar{v}_j \\ q_j \end{Bmatrix} = \begin{Bmatrix} \text{зусилля} \\ \text{у} \\ \text{вузлах} \\ \text{елемента} \\ \text{від} \\ \text{інтенсивного} \\ \text{навантаження} \end{Bmatrix}.$$

1-й елемент:

$$\begin{Bmatrix} P_{1\bar{z}}^- \\ P_{1y}^- \\ M_1 \\ P_{2\bar{z}}^- \\ P_{2y}^- \\ M_2 \end{Bmatrix} = \begin{vmatrix} 0,44 & 0 & 0,67 & -0,44 & 0 & -0,67 \\ 0 & 400 & 0 & 0 & -400 & 0 \\ -0,67 & 0 & 1,33 & 0,67 & 0 & 0,67 \\ -0,44 & 0 & 0,67 & 0,44 & 0 & 0,67 \\ 0 & -400 & 0 & 0 & 400 & 0 \\ -0,67 & 0 & 0,67 & 0,67 & 0 & 1,33 \end{vmatrix} \cdot \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 19,6 \\ -19,4 \\ 353,5 \end{Bmatrix} \cdot 10^{-6}.$$

$$-\begin{Bmatrix} 12 \\ 0 \\ -6 \\ 12 \\ 0 \\ 6 \end{Bmatrix} \cdot 10^{-3} = \begin{Bmatrix} -12,25 \\ 7,7 \\ 6,2 \\ -11,75 \\ -7,7 \\ -5,5 \end{Bmatrix} \cdot 10^{-3}.$$

Одержимо значення вектора вузлових навантажень по кінцях елемента (сили – в меганьютонах (МН), моменти – в меганьютоно-метрах (МН·м)).

Перевіряємо рівновагу елемента (див. рис. 6.6, а):

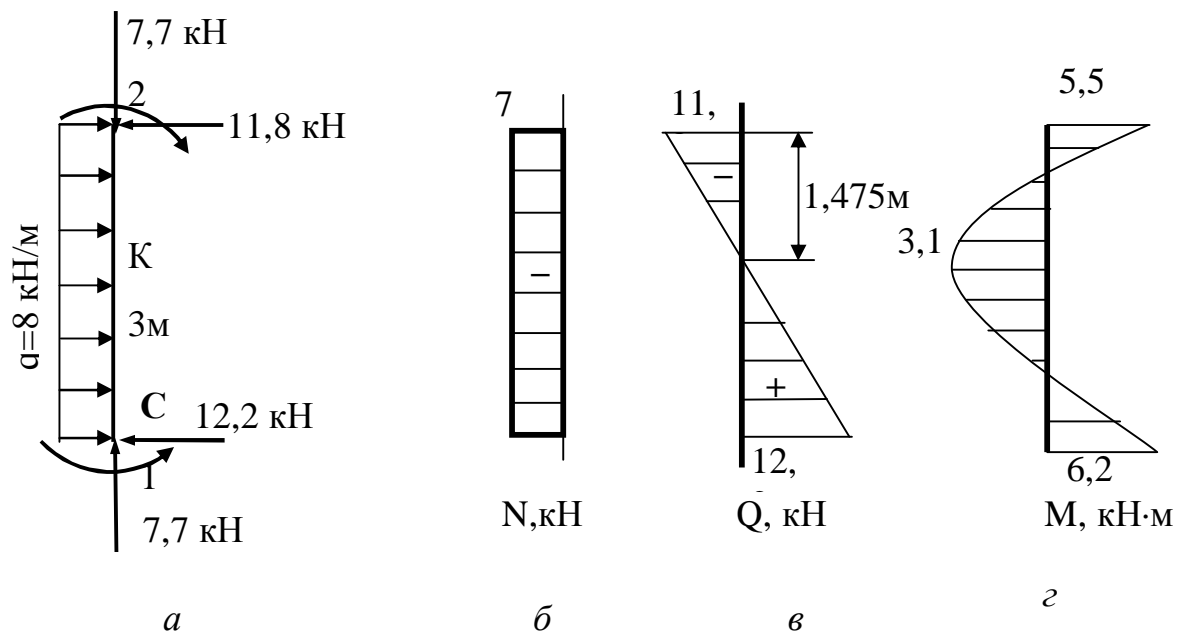
$$\sum P_z = 0 \quad -12,2 - 11,8 + 3 \cdot 8 = 0; \quad -24 + 24 = 0,$$

$$\sum P_v = 0 \quad -7,7 + 7,7 = 0,$$

$$\sum M_c = 0 \quad -5,5 - 8 \cdot 3 \cdot 1,5 + 11,8 \cdot 3 + 6,2 = 0; \quad -41,5 + 41,6 = 0.$$

Будуємо епюри подовжніх сил для елемента 1 (рис.6.6, б), поперечних сил (рис.6.6, в) і епюру згинальних моментів (рис.6.6, г).

$$M_k = -11,8 \cdot 1,475 + 5,5 + 8 \cdot 1,475 \cdot 1,475 / 2 = -3,1 \text{ кН}\cdot\text{м}.$$

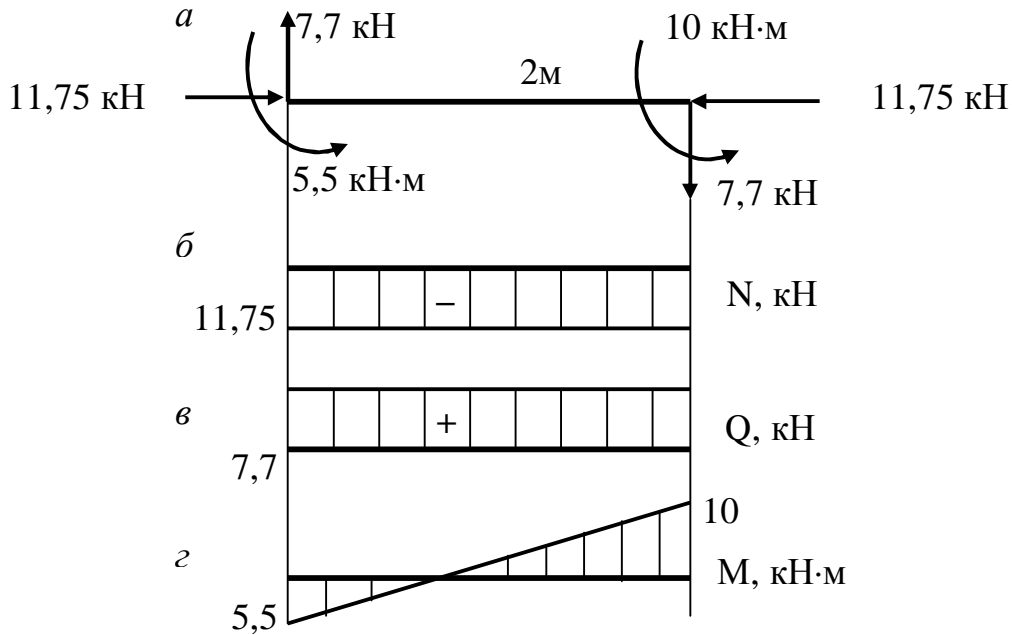


а – елемент з навантаженнями; б – епюра подовжніх сил; в – епюра поперечних сил; г – епюра згинальних моментів

Рисунок 6.6 – Елемент 1

Аналогічно будуюмо епюри N, Q і M для елемента 2 (рис. 6.7). Зістикувавши відповідні епюри елементів 1 і 2, одержуємо підсумкові епюри для рами у цілому (рис. 6.8).

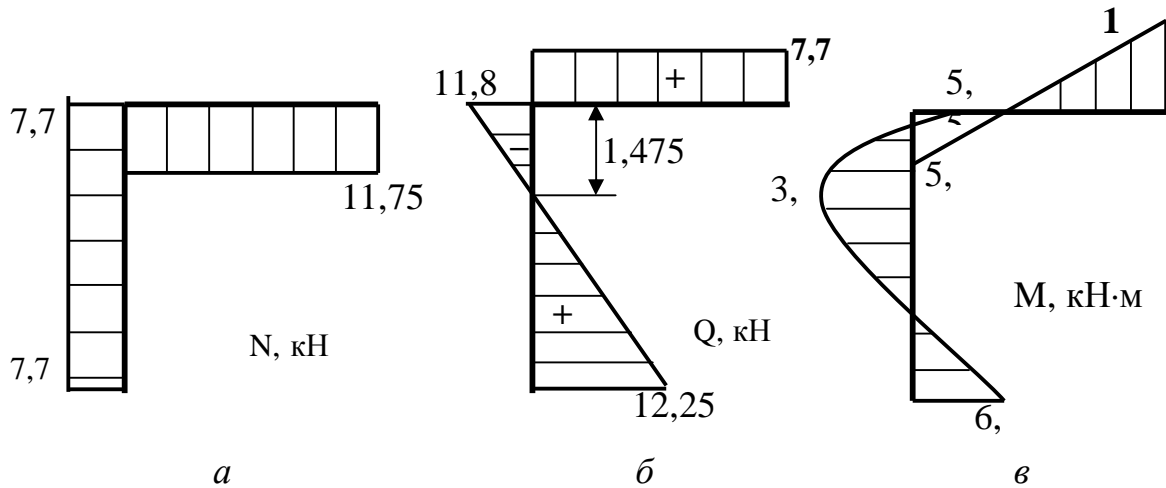
2-й елемент:



*a – елемент 2 з навантаженнями; б – епюра подовжніх сил;
в – епюра поперечних сил; г – епюра згинальних моментів*

Рисунок 6.7 – Елемент 2

$$\begin{Bmatrix} P_{2z}^- \\ P_{2y}^- \\ M_2 \\ P_{3z}^- \\ P_{3y}^- \\ M_3 \end{Bmatrix} = \begin{bmatrix} 600 & 0 & 0 & -600 & 0 & 0 \\ 0 & 1,5 & 1,5 & 0 & -1,5 & 1,5 \\ 0 & 1,5 & 2 & 0 & -1,5 & 1 \\ -600 & 0 & 0 & 600 & 0 & 0 \\ 0 & -1,5 & -1,5 & 0 & 1,5 & -1,5 \\ 0 & 1,5 & 1 & 0 & -1,5 & 2 \end{bmatrix} \cdot \begin{Bmatrix} 196 \\ -194 \\ 3535 \\ 0 \\ 0 \\ 4858 \end{Bmatrix} \cdot 10^{-6} = \begin{Bmatrix} 1175 \\ 7,7 \\ 5,5 \\ -1175 \\ -7,7 \\ 10 \end{Bmatrix} \cdot 10^{-3}.$$



*a – епюра подовжніх сил; б – епюра поперечних сил;
в – епюра згинальних моментів
Рисунок 6.8 – Підсумкові епюри для рами*

Перевіряємо рівновагу другого елемента:

$$\sum P_z = 0, -11,8+11,8=0,$$

$$\sum P_y = 0, 7,7-7,7=0$$

$$\sum M_E = 0 \quad 5,5-7,7 \cdot 2+10=0, \quad 15,5-15,4 \approx 0.$$

Визначаємо максимальні нормальні напруження.

Небезпечним є переріз, де $M_{\max}=10$ кН·м.

У цьому перерізі $N=7,7$ кН.

$$\sigma = \frac{N}{F} + \frac{M_x}{W_x} = \frac{7,7 \cdot 10^{-3}}{60 \cdot 10^{-4}} + \frac{10 \cdot 10^{-3}}{100 \cdot 10^{-6}} = 102 \text{ МПа},$$

$$\text{де } W_x = \frac{bh^2}{6} = \frac{6 \cdot 10^2}{6} = 100 \text{ см}^3.$$

7 РОЗРАХУНОК СТРИЖНІВ, ЩО ВИПРОБОВУЮТЬ СКЛАДНЕ ЗГИНАННЯ

7.1 Основні залежності опору матеріалів при складному згинанні

Складним згинанням називається вид вантаження, при якому сили, що згинають стрижень, діють в різних силових площинах (рис.7.1).

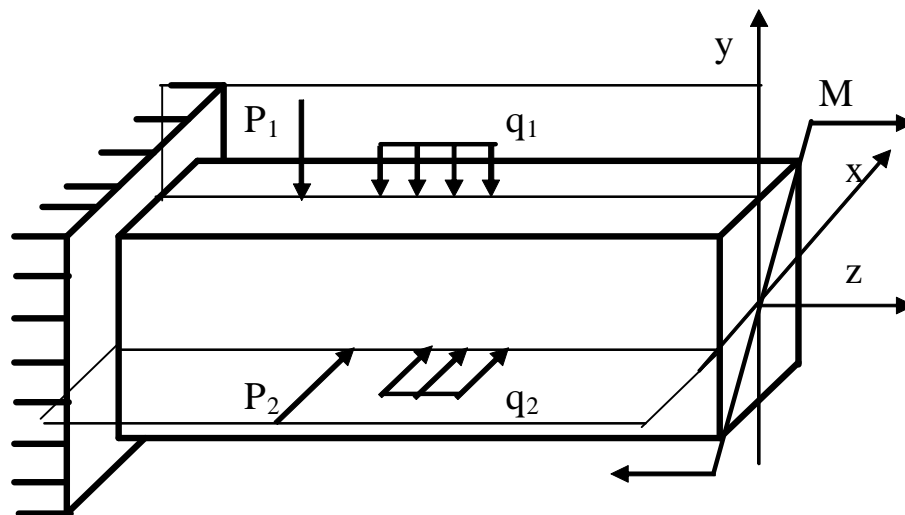


Рисунок 7.1 – Схема стрижня, що випробовує складне згинання

Складне згинання слід розглядати як комбінацію двох плоских згинань. У довільному поперечному перерізі виникають дві поперечні сили Q_x і Q_y і два згинальні моменти – M_x і M_y (рис.7.2).

Як правило, дотичні напруження, які виникають від поперечних сил, незначні, ними нехтуємо.

Нормальні напруження в довільній точці перерізу дорівнюють

$$S = \frac{M_x y}{J_x} + \frac{M_y x}{J_y}, \quad (7.1)$$

де M_x і M_y – згинальні моменти відносно осей x і y ;

J_x і J_y – моменти інерції перерізу відносно осей x і y ;

x і y – координати точки, в якій визначаються напруження, відносно головних центральних осей.

Умова міцності для стрижнів з бісиметричним перерізом і виступаючими кутами (прямокутника, двотавра, коробчатого перерізу) має вигляд

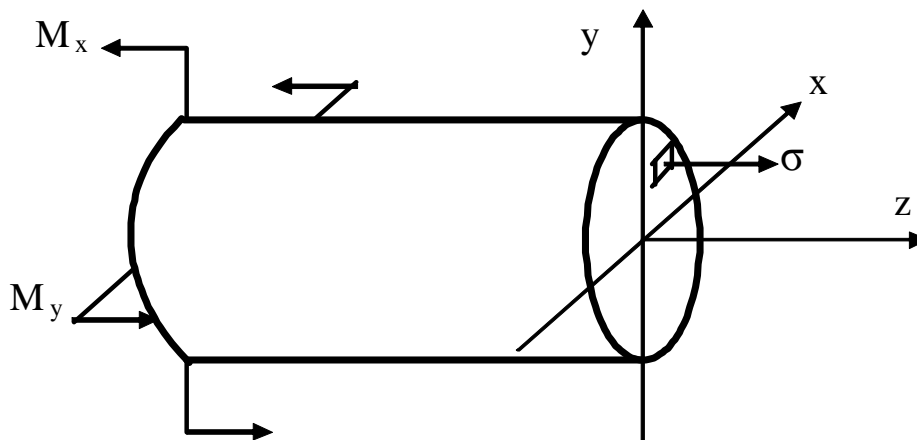


Рисунок 7.2 – Стрижень, що випробовує складне згинання

$$S_{\max} = \frac{M_x}{W_x} + \frac{M_y}{W_y} \leq [S] \quad (7.2)$$

де W_x і W_y – моменти опору перерізу відносно осей x і y .

Для стрижня з довільним перерізом положення нейтральної осі $n-n$ визначається за формулою

$$tgb = -\frac{M_x J_x}{M_y J_y} = -\frac{J_x}{J_y} tga, \quad (7.3)$$

де α – кут площини $\alpha-\alpha$ дії результуючого згинального моменту M (див. рис. 7.1)

Знаходиться найбільш віддалена точка (рис.7.3), для якої умова міцності має вигляд

$$S_{\max} = \frac{M_x y_A}{J_x} + \frac{M_y x_A}{J_y} \leq [S]. \quad (7.4)$$

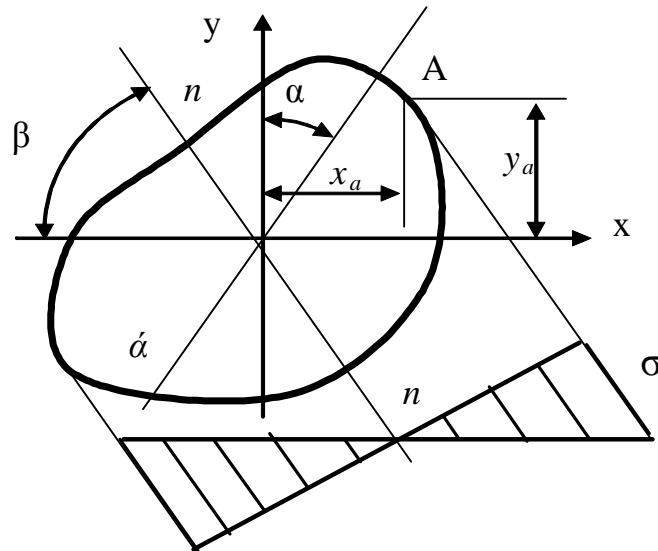


Рисунок 7.3 – Епюра напружень при складному згинанні

7.2 Матриця жорсткості кінцевого елемента при складному згинанні

При складному згинанні (рис. 7.4) будь-яке навантаження розкладається за головними силовими площинами yz і xz .

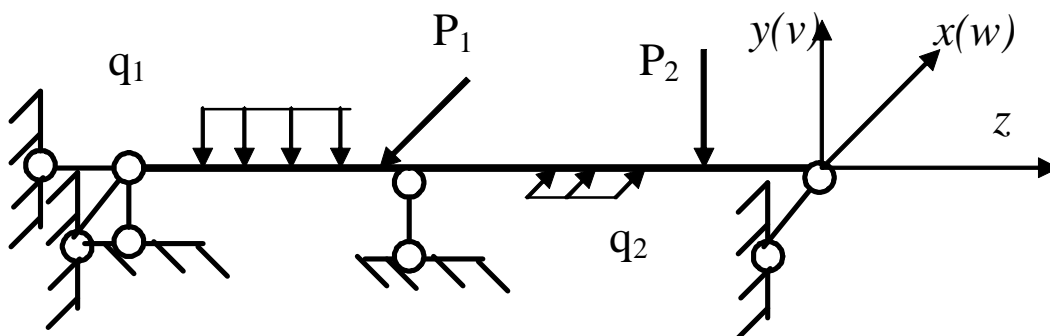


Рисунок 7.4 – Розбивка сил по головним площинам

У цьому випадку вектор переміщень вузлів і матриця жорсткості елемента є відповідно комбінацією векторів переміщення і матриць жорсткості при двох плоских згинаннях.

$$|K|^1 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline \frac{12EJ_x}{l^3} & 0 & \frac{6EJ_x}{l^2} & 0 & \frac{-12EJ_x}{l^3} & 0 & \frac{6EJ_x}{l^2} & 0 \\ \hline 0 & \frac{12EJ_y}{l^3} & 0 & \frac{6EJ_y}{l^2} & 0 & \frac{-12EJ_y}{l^3} & 0 & \frac{6EJ_y}{l^2} \\ \hline \frac{6EJ_x}{l^2} & 0 & \frac{4EJ_x}{l} & 0 & \frac{-6EJ_x}{l^2} & 0 & \frac{2EJ_x}{l} & 0 \\ \hline 0 & \frac{6EJ_y}{l^2} & 0 & \frac{4EJ_y}{l} & 0 & \frac{-6EJ_y}{l^2} & 0 & \frac{2EJ_y}{l} \\ \hline \frac{-12EJ_x}{l^3} & 0 & \frac{-6EJ_x}{l^2} & 0 & \frac{12EJ_x}{l^3} & 0 & \frac{-6EJ_x}{l^2} & 0 \\ \hline 0 & \frac{-12EJ_y}{l^3} & 0 & \frac{-6EJ_y}{l^2} & 0 & \frac{12EJ_y}{l^3} & 0 & \frac{-6EJ_y}{l^2} \\ \hline \frac{6EJ_x}{l^2} & 0 & \frac{2EJ_x}{l} & 0 & \frac{-6EJ_x}{l^2} & 0 & \frac{4EJ_x}{l} & 0 \\ \hline 0 & \frac{6EJ_y}{l^2} & 0 & \frac{2EJ_y}{l} & 0 & \frac{-6EJ_y}{l^2} & 0 & \frac{4EJ_y}{l} \\ \hline \end{array} \quad (7.5)$$

Оскільки кожен вузол має чотири степені вільності, то вектор переміщень вузла має чотири складові, а матриця жорсткості елемента буде 8x8.

$$\{d\}_i^e = \begin{Bmatrix} v_i \\ w_i \\ q_{xi} \\ q_{yi} \end{Bmatrix}. \quad (7.6)$$

7.3 Приклад

Для стрижня, показано на рис.7.5, за допомогою МКЕ побудувати епюри поперечних сил і згинальних моментів. Визначити σ_{\max} .

Розв'язання. Нумеруємо вузли і елементи. визначаємо матриці жорсткості елементів (вирази (7.7) і (7.8)).

$$|K|_1^1 = E \quad (7.7)$$

0,44J _x	0	0,67J _x	0	-0,44J _x	0	0,67J _x	0
0	0,44J _y	0	0,67J _y	0	-0,44J _y	0	0,67J _y
0,67J _x	0	1,33J _x	0	-0,67J _x	0	0,67J _x	0
0	0,67J _y	0	1,33J _y	0	-0,67J _y	0	0,67J _y
-0,44J _x	0	-0,67J _x	0	0,44J _x	0	-0,67J _x	0
0	-0,44J _y	0	-0,67J _y	0	0,67J _y	0	-0,67J _y
0,67J _x	0	0,67J _x	0	-0,67J _x	0	1,33J _x	0
0	0,67J _y	0	0,67J _y	0	-0,67J _y	0	1,33J _y

$$|K|_2^1 = E \quad (7.8)$$

1,5J _x	0	1,5J _x	0	-1,5J _x	0	1,5J _x	0
0	1,5J _y	0	1,5J _y	0	-1,5J _y	0	1,5J _y
1,5J _x	0	2J _x	0	-1,5J _x	0	J _x	0
0	1,5J _y	0	2J _y	0	-1,5J _y	0	J _y
-1,5J _x	0	-1,5J _x	0	1,5J _x	0	-1,5J _x	0
0	-1,5J _y	0	-1,5J _y	0	1,5J _y	0	-1,5J _y
1,5J _x	0	J _x	0	-1,5J _x	0	2J _x	0
0	1,5J _y	0	J _y	0	-1,5J _y	0	2J _y

Визначаємо J_x і J_y : $J_x = \frac{16 \cdot 18^3}{12} - \frac{16^3 \cdot 12}{12} = 3680 \text{ см}^4$,

$$J_y = \frac{18 \cdot 16^3}{12} - \frac{12^3 \cdot 16}{12} = 3840 \text{ см}^4.$$

Визначаємо моменти опору W_x і W_y :

$$W_x = \frac{3680}{9} = 409 \text{ см}^3; \quad W_y = \frac{3840}{8} = 480 \text{ см}^3.$$

Формуємо матрицю жорсткості конструкції (7.9).

Маємо три вузли, кожен вузол має чотири степені вільності, значить, розмірність матриці жорсткості 12x12. У вектор вузлових навантажень записуємо наступні значення: на перший елемент діє розподілене навантаження, від нього – вузлові навантаження:

$$-\frac{ql}{2} = -\frac{20 \cdot 3}{2} = -30 \text{ кН}, \quad -\frac{ql^2}{2} = -\frac{20 \cdot 9}{12} = -15 \text{ кН} \cdot \text{м},$$

$$-\frac{ql}{2} = -\frac{20 \cdot 3}{2} = -30 \text{ кН}, \quad \frac{ql^2}{2} = \frac{20 \cdot 9}{12} = 15 \text{ кН} \cdot \text{м}.$$

Вони запишуться в перший, третій, п'ятий і сьомий рядки.

У другому вузлі прикладений зосереджений момент в горизонтальній площині – його значення заносимо до восьмого рядка. У решті рядків ставимо нулі.

Таким чином, одержали матрицю жорсткості конструкції (7.9).

Тепер враховуємо граничні умови:

$$v_1=0, w_1=0, \theta_{x1}=0, \theta_{y1}=0, v_3=0, w_3=0, \theta_{x3}=0, \theta_{y3}=0.$$

Після їх врахування одержуємо систему рівнянь (7.10)

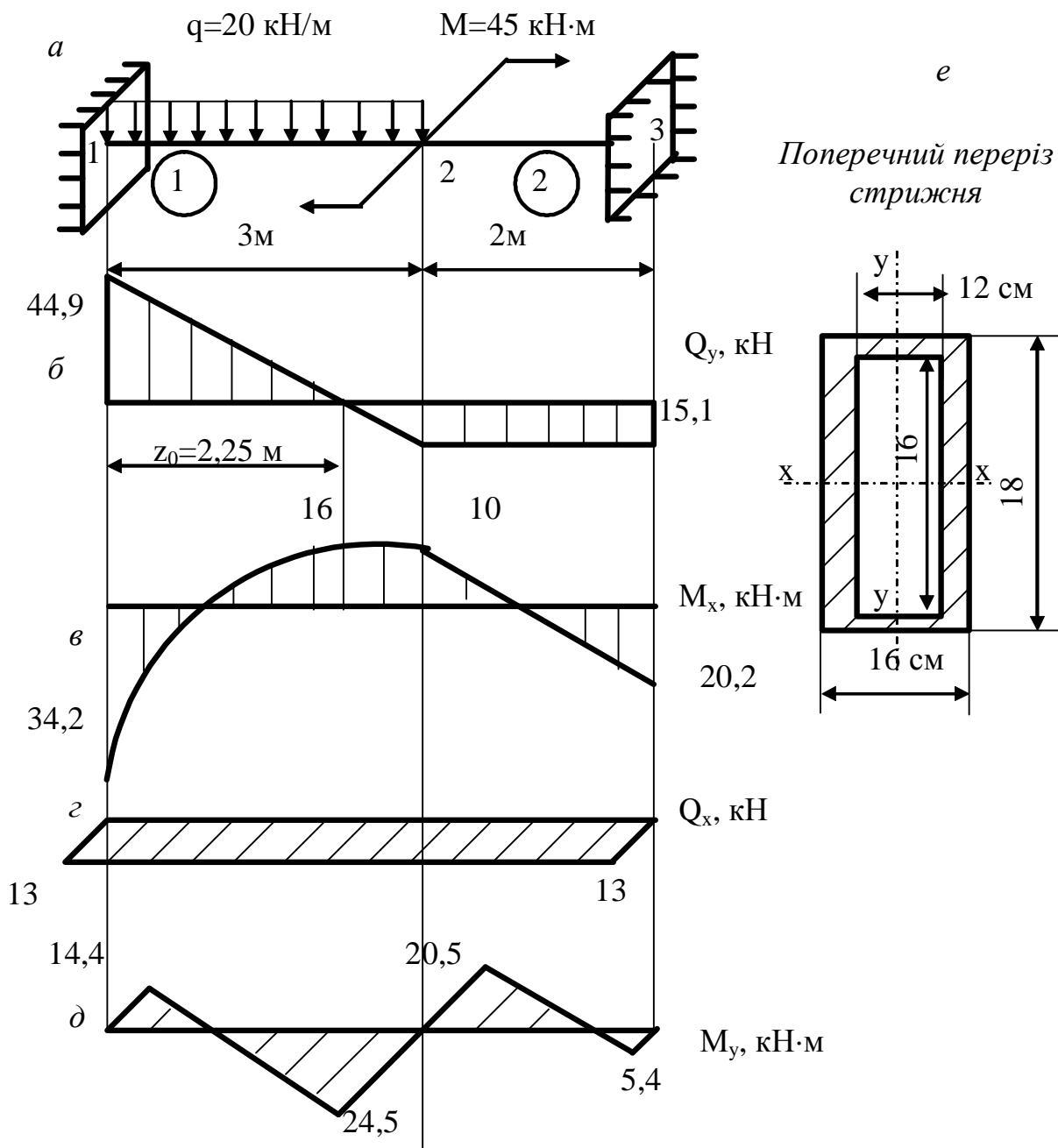


Рисунок 7.5 – епюри внутрішніх зусиль для стрижня, що знаходиться під дією складного згинання

$$\mathbf{E} \begin{array}{|c|c|c|c|} \hline 1,94 J_x & 0 & 0,83 J_x & 0 \\ \hline 0 & 1,94 J_y & 0 & 0,83 J_y \\ \hline 0,83 J_x & 0 & 3,33 J_x & 0 \\ \hline 0 & 0,83 J_y & 0 & 3,33 J_x \\ \hline \end{array} \begin{array}{l} \left. \begin{array}{l} v_2 \\ w_2 \\ q_{x2} \\ q_{y2} \end{array} \right\} = \left. \begin{array}{l} -30 \\ 0 \\ 15 \\ -45 \end{array} \right\}. \quad (7.10)
 \end{array}$$

Її рішення дає наступні значення:

$$v_2 = -\frac{19,4}{EJ_x}, \quad w_2 = \frac{15,1}{EJ_y}, \quad q_{x2} = \frac{9,4}{EJ_x}, \quad q_{y2} = -\frac{15,1}{EJ_y}.$$

Визначаємо вектори вузлових навантажень на кінцях елементів.

1-й елемент:

$$\begin{array}{l} \left. \begin{array}{l} P_{1x} \\ P_{1y} \\ M_{1x} \\ M_{1y} \\ P_{2x} \\ P_{2y} \\ M_{2x} \\ M_{2y} \end{array} \right\} = \left[\mathbf{K} \right] \left[\mathbf{l} \right] \begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \\ -19,4/EJ_x \\ 15,1/EJ_y \\ 9,4/EJ_x \\ -15,1/EJ_y \end{array} - \begin{array}{l} -30 \\ 0 \\ -15 \\ 0 \\ -30 \\ 0 \\ 15 \\ 0 \end{array} = \begin{array}{l} 44,9 \\ -13 \\ 34,2 \\ -14,4 \\ 15,1 \\ 13 \\ 10 \\ -24,5 \end{array} \right\}.
 \end{array}$$

Одержані навантаження прикладаємо до вузлів першого елемента (рис.7.6).

Перевіримо його рівновагу:

$$\sum P_y = 0. \quad 44,9 + 15,1 - 20 \cdot 3 = 0; \quad 60 - 60 = 0; \quad \sum P_x = 0. \quad -13 + 13 = 0;$$

$$\sum M_{xc} = 0. \quad M_{xc} = 34,2 - 44,9 \cdot 3 + 20 \cdot 3 \cdot 1,5 + 10;$$

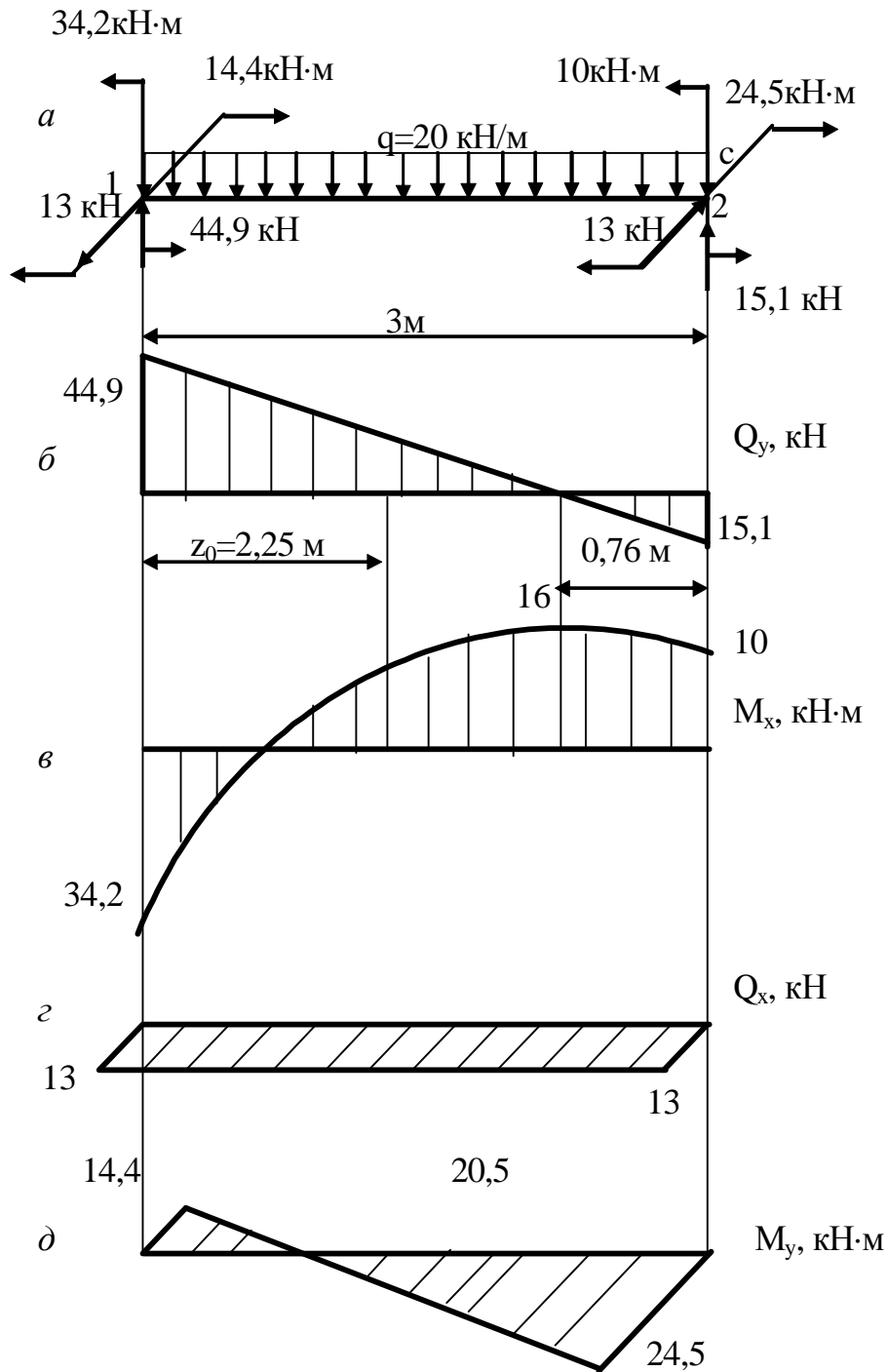


Рисунок 7.6 – Епюри внутрішніх зусиль для 1-го елемента

$$34,2 - 13 \cdot 3 + 90 + 10 = -134,7 + 134,2 \approx 0.$$

Погрішність складає 0,38%, що допустимо.

$$\sum M_{yc} = 0. \quad M_{yc} = -14,4 + 13 \cdot 3 - 24,5 = 0; -38,9 + 39 \approx 0.$$

Погрішність складає 0,25%, що допустимо.

2-й елемент:

$$\begin{Bmatrix} P_{2x} \\ P_{2y} \\ M_{2x} \\ M_{2y} \\ P_{3x} \\ P_{3y} \\ M_{3x} \\ M_{3y} \end{Bmatrix} = |K|_i \begin{Bmatrix} -19,4/EJ_x \\ 15,1/EJ_y \\ 9,4/EJ_x \\ -15,1/EJ_y \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} - \begin{Bmatrix} -30 \\ 0 \\ -15 \\ 0 \\ -30 \\ 0 \\ 15 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -15,1 \\ -13 \\ -10 \\ -20,5 \\ 15,1 \\ 13 \\ 20,2 \\ -5,4 \end{Bmatrix}$$

Прикладаємо знайдені навантаження за вузлами елемента 2 (рис. 7.7).

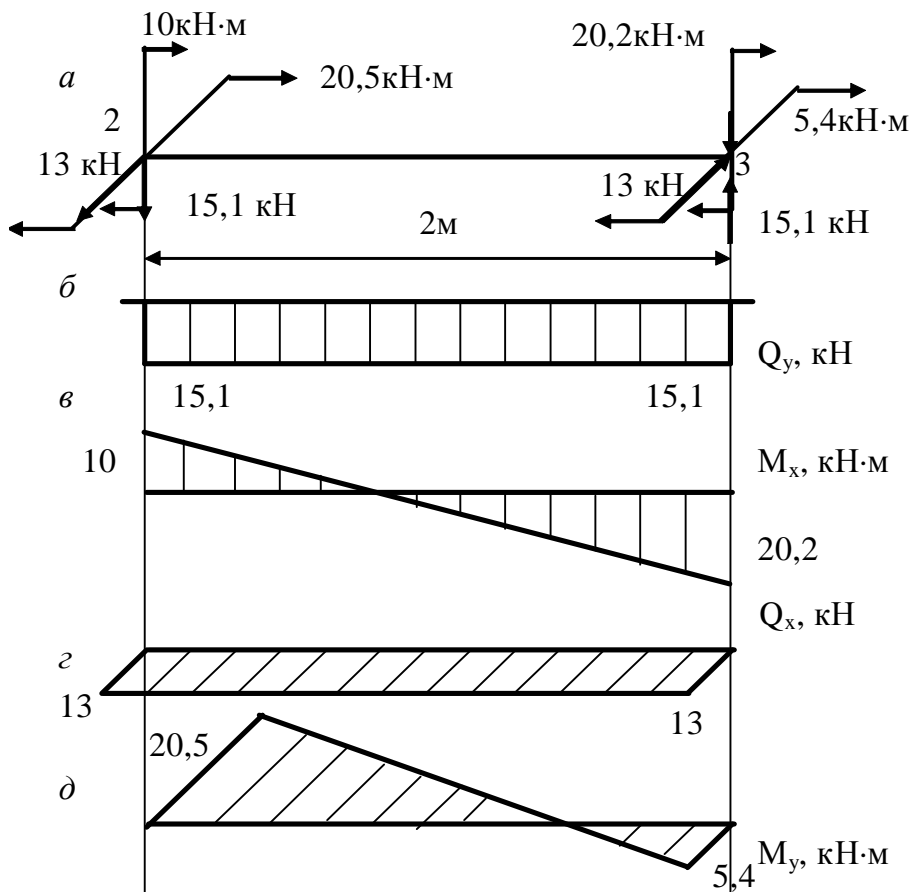


Рисунок 7.7 – Епюри внутрішніх зусиль для 2-го елемента

Перевіряємо рівняння рівноваги.

$$\sum P_y = 0. \quad -15,1 + 15,1 = 0;$$

$$\sum P_x = 0. \quad -13 + 13 = 0;$$

$$\sum M_x = 0. \quad M_x = 15,1 \cdot 2 - 10 - 20,2 = 30,2 - 30,2 = 0;$$

$$\sum M_y = 0. \quad M_y = 13 \cdot 2 - 20,5 - 5,4 = 26 - 25,9 = 0.$$

Усі рівняння виконуються.

Стикування за ділянками дало результуючі епюри (див. рис. 7.5).

Визначаємо σ_{max} .

Небезпечними є перерізи, де $M_{x1} = 34,2 \text{ kH} \cdot \text{м}$, $M_{y1} = 14,4 \text{ kH} \cdot \text{м}$ і $M_{x2} = 10 \text{ kH} \cdot \text{м}$ і $M_{y2} = 24,5 \text{ kH} \cdot \text{м}$.

$$S_{max1} = \frac{M_{x1}}{W_x} + \frac{M_{y1}}{W_y} = \frac{34,2 \cdot 10^{-3}}{409 \cdot 10^{-6}} + \frac{14,4 \cdot 10^{-3}}{480 \cdot 10^{-6}} = 83,6 + 30 = 113,6 \text{ МПа},$$

$$S_{max2} = \frac{M_{x2}}{W_x} + \frac{M_{y2}}{W_y} = \frac{10 \cdot 10^{-3}}{409 \cdot 10^{-6}} + \frac{24,5 \cdot 10^{-3}}{480 \cdot 10^{-6}} = 24,4 + 51 = 85,4 \text{ МПа}.$$

Як бачимо, небезпечним є перший переріз.

8 РОЗРАХУНОК СТРИЖНІВ, ЩО ВИПРОБОВУЮТЬ СКЛАДНЕ ЗГИНАННЯ З КРУЧЕННЯМ

8.1 Основні залежності при згинанні з крученням

При згинанні з крученням (рис.8.1) у довільному перерізі вала виникають три моменти $M_{вер}$, $M_{гор}$ і $M_{кр}$ і дві поперечні сили $Q_{верт}$ і $Q_{гор}$. Дотичні напруження, що виникають від поперечних сил, незначні, ними нехтуємо. За вказаними моментами (рис.8.2) визначається розрахунковий момент:

– за третьою теорією міцності:

$$M_{розр}^{III} = \sqrt{M_{верт}^2 + M_{гор}^2 + M_{кр}^2};$$

– за четвертою теорією міцності:

$$M_{розр}^{IV} = \sqrt{M_{верт}^2 + M_{гор}^2 + 0,75M_{кр}^2}.$$

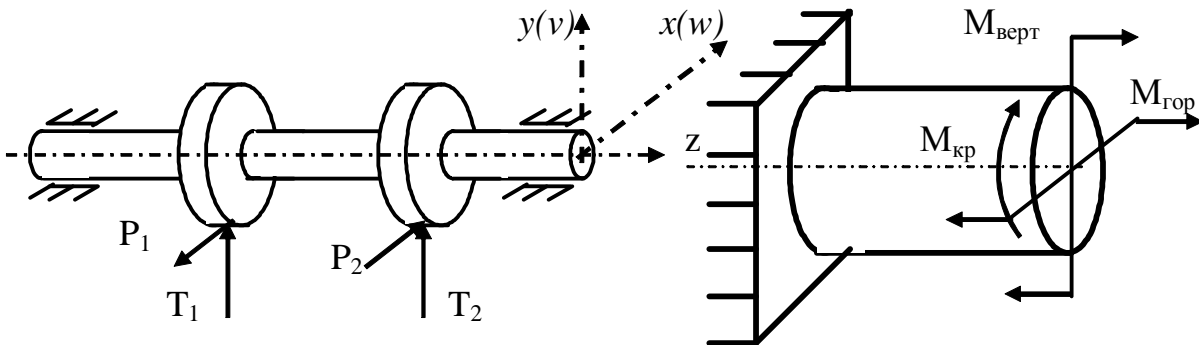


Рисунок 8.1 - Схема вала, що випробовує згинання з крученням

Рисунок 8.2 - Моменти, які виникають в перерізах вала

Умова міцності має вигляд: $S_{екв} = \frac{M_{расч}}{W_0} \leq [S],$ (8.1)

де W_0 – осьовий момент опору вала:

$$W_0 = \frac{\rho D^3}{32} \approx 0,1D^3 \quad \text{– для суцільного вала;}$$

$$W_0 = 0,1D^3(1 - a^4) \quad \text{– для кільцевого вала.}$$

Тут $a = \frac{d}{D},$

де d – внутрішній діаметр кільцевого вала.

8.2 Матриця жорсткості кінцевого елемента при згинанні з крученням

Згинання з крученням можна розглядати як комбінацію складного згинання і кручення. У цьому випадку кожен вузол має п'ять степенів вільності, їм відповідають 5 переміщень вузла (рис. 8.3) і 5 внутрішніх зусиль (рис. 8.4). Вектор переміщень i -го вузла має вигляд:

$$\{d\}_i = \begin{Bmatrix} v_i \\ w_i \\ q_{xi} \\ q_{yi} \\ q_{zi} \end{Bmatrix}. \quad (8.2)$$

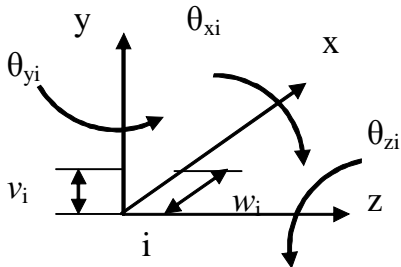


Рисунок 8.3 – Додатні напрямки переміщень

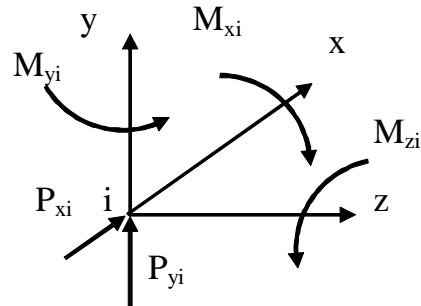


Рисунок 8.4 - Додатні напрямки вузлових зусиль

Матриця жорсткості кінцевого елемента має розмірність 10x10. Оскільки осьові моменти інерції $J_x = J_y = J_o$, то матриця жорсткості запишеться так:

$$|K|^1 = \begin{bmatrix} \frac{12EJ_o}{l^3} & 0 & \frac{6EJ_o}{l^2} & 0 & 0 & \frac{-12EJ_o}{l^3} & 0 & \frac{6EJ_o}{l^2} & 0 & 0 \\ 0 & \frac{12EJ_o}{l^3} & 0 & \frac{6EJ_o}{l^2} & 0 & 0 & \frac{-12EJ_o}{l^3} & 0 & \frac{6EJ_o}{l^2} & 0 \\ \frac{6EJ_o}{l^2} & 0 & \frac{4EJ_o}{l} & 0 & 0 & \frac{-6EJ_o}{l^2} & 0 & \frac{2EJ_o}{l} & 0 & 0 \\ 0 & \frac{6EJ_o}{l^2} & 0 & \frac{4EJ_o}{l} & 0 & 0 & \frac{6EJ_o}{l^2} & 0 & \frac{2EJ_o}{l} & 0 \\ 0 & 0 & 0 & 0 & \frac{GJ_p}{l} & 0 & 0 & 0 & 0 & \frac{-GJ_p}{l} \\ \frac{-12EJ_o}{l^3} & 0 & \frac{-6EJ_o}{l^2} & 0 & 0 & \frac{12EJ_o}{l^3} & 0 & \frac{-6EJ_o}{l^2} & 0 & 0 \\ 0 & \frac{-12EJ_o}{l^3} & 0 & \frac{-6EJ_o}{l^2} & 0 & 0 & \frac{12EJ_o}{l^3} & 0 & \frac{-6EJ_o}{l^2} & 0 \\ \frac{6EJ_o}{l^2} & 0 & \frac{2EJ_o}{l} & 0 & 0 & \frac{-6EJ_o}{l^2} & 0 & \frac{4EJ_o}{l} & 0 & 0 \\ 0 & \frac{6EJ_o}{l^2} & 0 & \frac{2EJ_o}{l} & 0 & 0 & \frac{-6EJ_o}{l^2} & 0 & \frac{4EJ_o}{l} & 0 \\ 0 & 0 & 0 & 0 & \frac{-GJ_p}{l} & 0 & 0 & 0 & 0 & \frac{GJ_p}{l} \end{bmatrix} \quad (8.3)$$

8.3 Приклад

Для стрижня, показано на рис.8.5,а, за допомогою МКЕ побудувати епюри $M_{верт}$, $M_{гор}$ і $M_{кр}$.

Визначити діаметр вала за IV теорією міцності, якщо $[\sigma]=120$ МПа.

Розв'язання. Нумеруємо вузли і елементи.

Обчислюємо матрицю жорсткості кожного елемента (вирази (8.4) і (8.5)), при цьому виноситься загальний множник EJ_0 . Коефіцієнт від

кручення GJ_p необхідно розділити на EJ_0

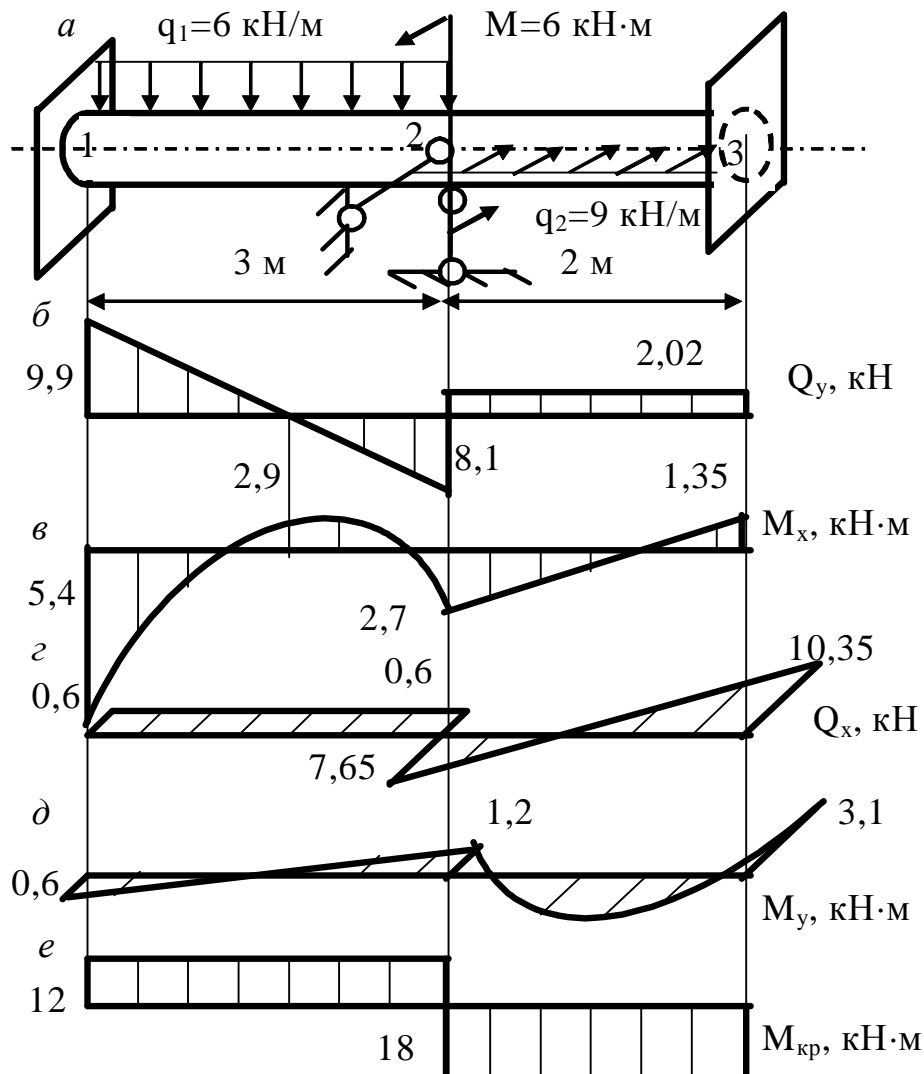
$$\frac{GJ_p}{EJ_0} = \frac{8 \cdot 10^4 \cdot 2J_0}{2 \cdot 10^5 \cdot J_0} = 0,8.$$


Рисунок 8.5 – Стрижень, що випробовує згинання з крученням, і його результуючі епюри

Формуємо матрицю жорсткості всієї конструкції (8.6).

До вектора вузлових навантажень заносимо значення від $q_1=6$ кН/м:

$$\text{1-й рядок} \quad -\frac{q_1 \mathbf{l}_1}{2} = -\frac{6 \cdot 3}{2} = -9 \text{кН};$$

$$\text{3-й рядок} \quad -\frac{q_1 \mathbf{l}_1^2}{12} = -\frac{6 \cdot 3^2}{12} = -4,5 \text{кН} \cdot \text{м};$$

$$\text{6-й рядок} \quad -\frac{q_1 \mathbf{l}_1}{2} = -\frac{6 \cdot 3}{2} = -9 \text{кН};$$

$$\text{8-й рядок} \quad \frac{q_1 \mathbf{l}_1^2}{12} = \frac{6 \cdot 3^2}{12} = 4,5 \text{кН} \cdot \text{м};$$

від навантаження $q_2=9$ кН/м:

$$\text{7-й рядок} \quad \frac{q_2 \mathbf{l}_2}{2} = \frac{9 \cdot 2}{2} = 9 \text{кН};$$

$$\text{9-й рядок} \quad \frac{q_2 \mathbf{l}_2^2}{12} = \frac{9 \cdot 2^2}{12} = 3 \text{кН} \cdot \text{м};$$

$$\text{12-й рядок} \quad \frac{q_2 \mathbf{l}_2}{2} = \frac{9 \cdot 2}{2} = 9 \text{кН};$$

$$\text{14-й рядок} \quad -\frac{q_2 \mathbf{l}_2^2}{12} = -\frac{9 \cdot 2^2}{12} = -3 \text{кН} \cdot \text{м}.$$

Від $M = 30$ кН·м в 10-й рядок заносимо 30 кН·м.

$$|K|_I^1 = EJ_0 \quad (8.4)$$

0,44	0	0,67	0	0	-0,44	0	0,67	0	0
0	0,44	0	0,67	0	0	-0,44	0	0,67	0
0,67	0	1,33	0	0	-0,67	0	0,67	0	0
0	0,67	0	1,33	0	0	-0,67	0	0,67	0
0	0	0	0	0,27	0	0	0	0	-0,27
-0,44	0	-0,67	0	0	0,44	0	-0,67	0	0
0	-0,44	0	-0,67	0	0	0,44	0	-0,67	0
0,67	0	0,67	0	0	-0,67	0	1,33	0	0
0	0,67	0	0,67	0	0	-0,67	0	1,33	0
0	0	0	0	-0,27	0	0	0	0	0,27

$$|K|_{II}^e = EJ_0 \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline 1,5 & 0 & 1,5 & 0 & 0 & -1,5 & 0 & -1,5 & 0 & 0 \\ \hline 0 & 1,5 & 0 & 1,5 & 0 & 0 & -1,5 & 0 & -1,5 & 0 \\ \hline 1,5 & 0 & 2 & 0 & 0 & -1,5 & 0 & 1 & 0 & 0 \\ \hline 0 & 1,5 & 0 & 2 & 0 & 0 & -1,5 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0,4 & 0 & 0 & 0 & 0 & -0,4 \\ \hline -1,5 & 0 & -1,5 & 0 & 0 & 1,5 & 0 & -1,5 & 0 & 0 \\ \hline 0 & -1,5 & 0 & -1,5 & 0 & 0 & 1,5 & 0 & -1,5 & 0 \\ \hline 1,5 & 0 & 1 & 0 & 0 & -1,5 & 0 & 2 & 0 & 0 \\ \hline 0 & 1,5 & 0 & 1 & 0 & 0 & -1,5 & 0 & 2 & 0 \\ \hline 0 & 0 & 0 & 0 & -0,4 & 0 & 0 & 0 & 0 & 0,4 \\ \hline \end{array} \quad (8.5)$$

Граничні умови:

$$v_1 = 0; \quad v_3 = 0; \quad w_1 = 0; \quad q_{y1} = 0; \quad q_{x1} = 0; \quad q_{z1} = 0;$$

$$v_2 = 0; \quad w_2 = 0; \quad w_3 = 0; \quad q_{x3} = 0; \quad q_{y3} = 0; \quad q_{z3} = 0.$$

Після врахування граничних умов у виразі (8.6) одержимо систему рівнянь

$$EJ_0 = \begin{array}{|c|c|c|} \hline 3,33 & 0 & 0 \\ \hline 0 & 3,33 & 0 \\ \hline 0 & 0 & 0,67 \\ \hline \end{array} \begin{array}{l} \left[q_{x2} \right] \\ \left[q_{y2} \right] \\ \left[q_{z2} \right] \end{array} = \left\{ \begin{array}{l} 4,5 \\ 3 \\ 30 \end{array} \right\}.$$

Її рішення:

$$q_{x2} = \frac{1,35}{EJ_0}; \quad q_{y2} = \frac{0,9}{EJ_0}; \quad q_{z2} = \frac{45}{EJ_0}.$$

Визначаємо зусилля у вузлах елементів, прикладаємо ці зусилля до відповідних елементів і будуємо епюри внутрішніх зусиль для елементів (рис. 8.6 і 8.7).

1-й елемент:

$$\begin{Bmatrix} P_{y1} \\ P_{x1} \\ M_{x1} \\ M_{y1} \\ M_{z1} \\ P_{y2} \\ P_{x2} \\ M_{x2} \\ M_{y2} \\ M_{z2} \end{Bmatrix} = |K|_I^1 \begin{Bmatrix} v_1 \\ w_1 \\ q_{x1} \\ q_{y1} \\ q_{z1} \\ v_2 \\ w_2 \\ q_{x2} \\ q_{y2} \\ q_{z2} \end{Bmatrix} = |K|_I^1 \begin{Bmatrix} -q_1 \mathbf{l}/2 \\ 0 \\ -q_1 \mathbf{l}^2/12 \\ 0 \\ 0 \\ -q_1 \mathbf{l}/2 \\ 0 \\ -q_1 \mathbf{l}^2/12 \\ 0 \\ 0 \end{Bmatrix} = |K|_I^1 \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1,35/EJ_0 \\ 0,9/EJ_0 \\ 45/EJ_0 \end{Bmatrix} \begin{Bmatrix} -9 \\ 0 \\ -4,5 \\ 0 \\ 0 \\ -9 \\ 0 \\ 4,5 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 9,9 \\ 0,6 \\ 5,4 \\ 0,6 \\ -1,2 \\ 8,1 \\ -0,6 \\ -2,7 \\ 1,2 \\ 12 \end{Bmatrix} .$$

2-й елемент:

$$\begin{Bmatrix} P_{y2} \\ P_{x2} \\ M_{x2} \\ M_{y2} \\ M_{z2} \\ P_{y3} \\ P_{x3} \\ M_{x3} \\ M_{y3} \\ M_{z3} \end{Bmatrix} = |K|_{II}^1 \begin{Bmatrix} v_2 \\ w_2 \\ q_{x2} \\ q_{y2} \\ q_{z2} \\ v_3 \\ w_3 \\ q_{x3} \\ q_{y3} \\ q_{z3} \end{Bmatrix} = |K|_{II}^1 \begin{Bmatrix} 0 \\ q_2 \mathbf{l}/2 \\ 0 \\ q_2 \mathbf{l}^2/12 \\ 0 \\ 0 \\ q_2 \mathbf{l}/2 \\ 0 \\ -q_2 \mathbf{l}^2/12 \\ 0 \end{Bmatrix} = |K|_{II}^1 \begin{Bmatrix} 0 \\ 0 \\ 1,35/EJ_0 \\ 0,9/EJ_0 \\ 45/EJ_0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 9 \\ 0 \\ 3 \\ 0 \\ 0 \\ 9 \\ 0 \\ -3 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 2,02 \\ -7,65 \\ 2,7 \\ -1,2 \\ 18 \\ -2,02 \\ -10,35 \\ 1,35 \\ 3,1 \\ -18 \end{Bmatrix} .$$

Зістикування епюр за ділянками дає результуючі епюри (див. рис. 8.5).

Найбільш небезпечними є перерізи в защемленнях, де діють моменти:

а) ліве защемлення: $M_x = 5,4 \text{ кН} \cdot \text{м}, \quad M_y = 0,6 \text{ кН} \cdot \text{м};$
 $M_z = 12 \text{ кН} \cdot \text{м},$

б) праве зацімлення: $M_x = 1,35 \text{ кН} \cdot \text{м}$, $M_y = 3,1 \text{ кН} \cdot \text{м}$,
 $M_z = 18 \text{ кН} \cdot \text{м}$.

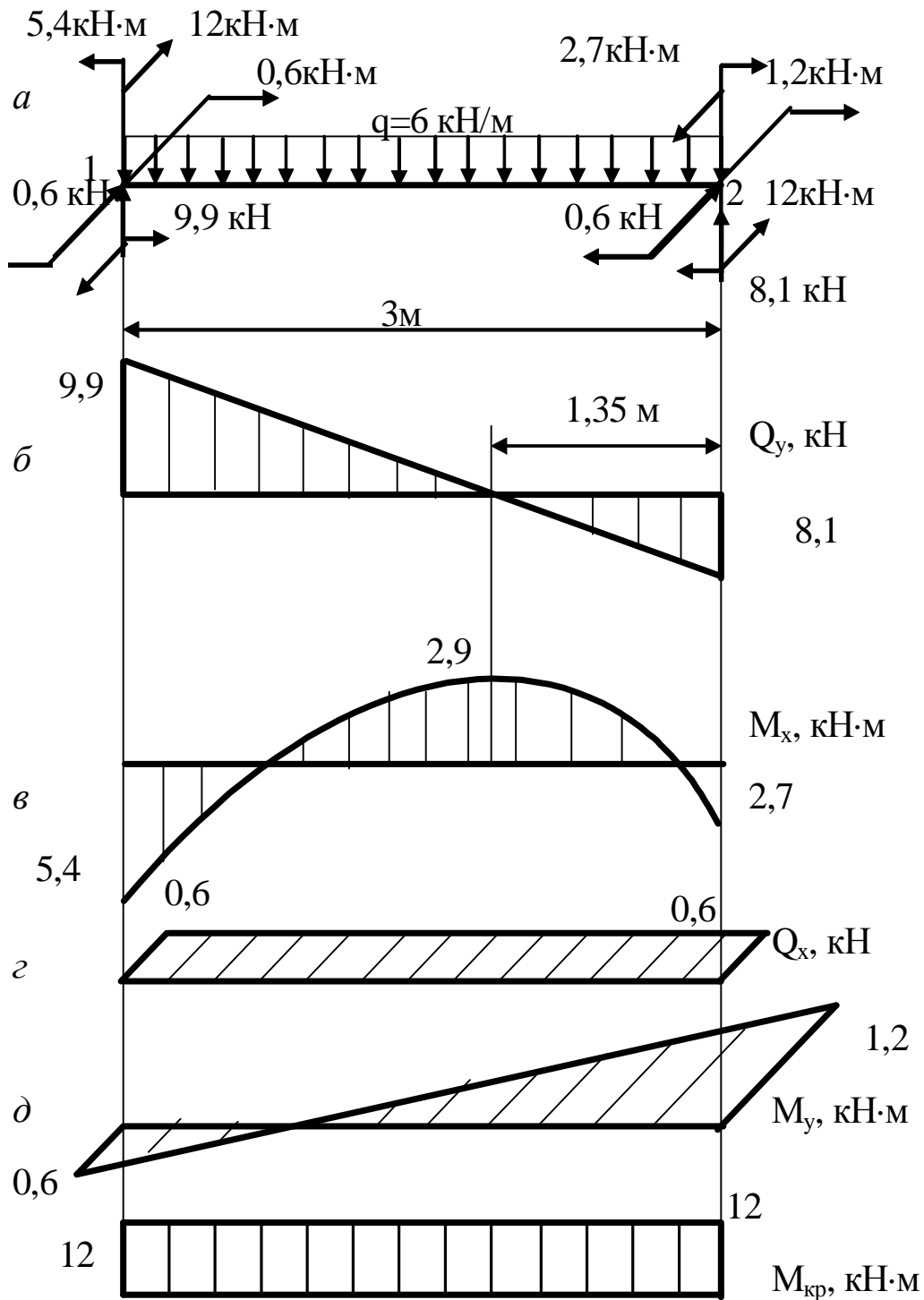


Рисунок 8.6 – Епюри внутрішніх зусиль для 1-го елемента

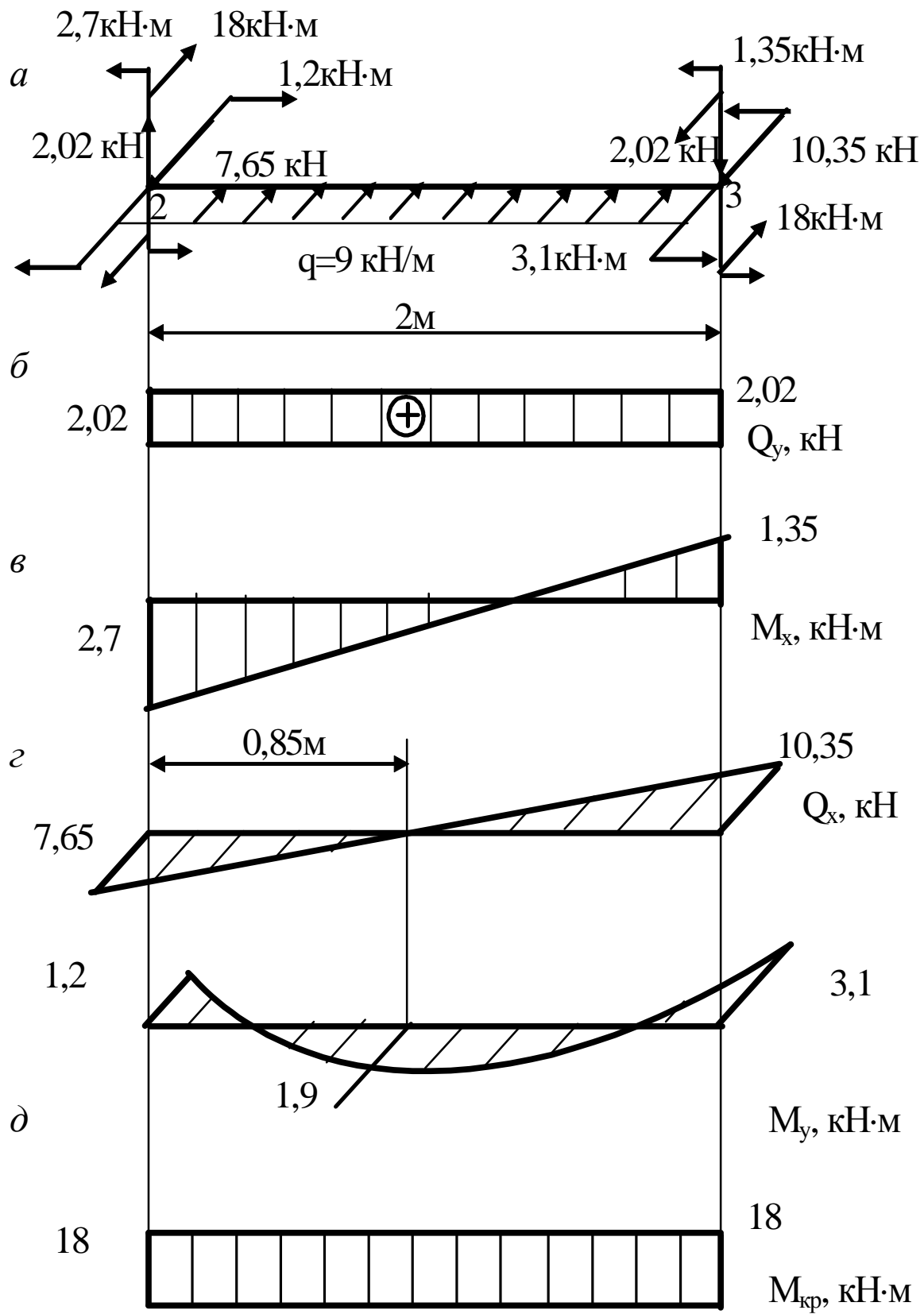


Рисунок 8.7 – Епюри внутрішніх зусиль для 2-го елемента

Як, бачимо, найбільш небезпечним є переріз в правому защемленні.

Для цього перерізу визначаємо розрахунковий момент:

$$M_{расч}^{IV} = \sqrt{M_{верт}^2 + M_{гор}^2 + 0,75M_{кр}^2} = \sqrt{1,35^2 + 3,1^2 + 0,75 \cdot 18^2} = 15,9 \text{ кН} \cdot \text{м}.$$

Визначаємо необхідний момент опору перерізу:

$$W_o = \frac{M_{расч}^{IV}}{[\sigma]} = \frac{15,9 \cdot 10^{-3}}{120} \cdot 10^6 = 132,5 \text{ см}^3.$$

$$\text{З іншого боку, } W_o = 0,1D^3, \text{ звідки } D = \sqrt[3]{\frac{W_o}{0,1}} = \sqrt[3]{\frac{132,5}{0,1}} = 10,95 \text{ см}.$$

Беремо діаметр вала $D = 110$ мм.

9 РОЗРАХУНОК ПРОСТОРОВИХ СТРИЖНЕВИХ СИСТЕМ

9.1 Основні залежності МКЕ для просторової стрижневої системи

Стосовно просторових стрижневих систем МКЕ є подальшим розвитком методу переміщень при розкритті статичної невизначуваності, де за невідомі бралися лінійні і кутові переміщення. Аналогічно і в МКЕ за невідомі беруться лінійні й кутові переміщення в глобальній системі координат, пов'язаній зі всією конструкцією. У разі просторової системи незакріплений вузол має шість невідомих переміщень: три лінійні зсуви уздовж координатних осей і три кутові переміщення щодо осей, паралельних координатним осям.

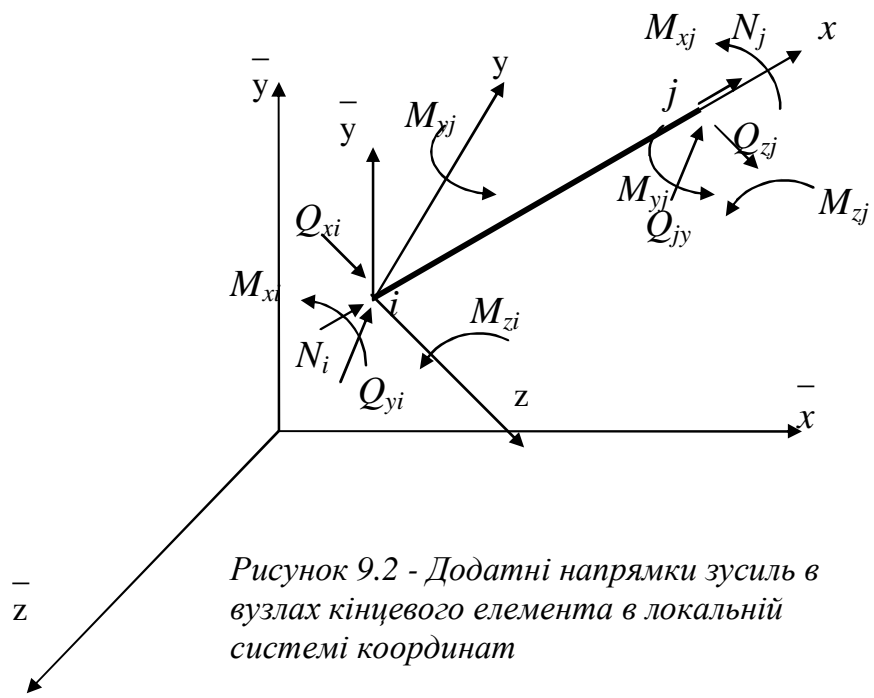
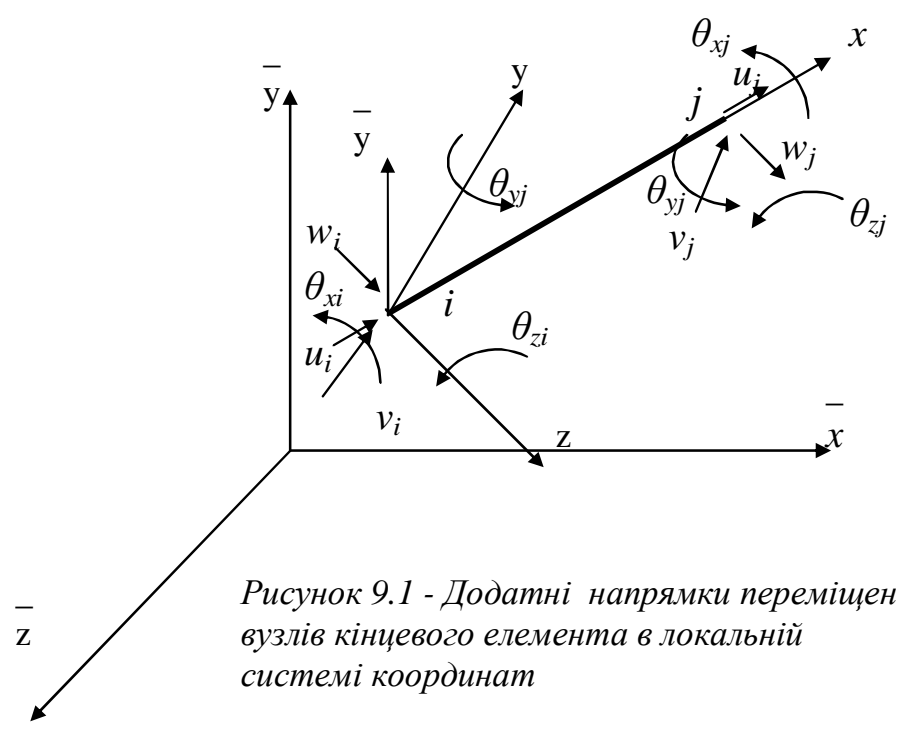
Для просторової стрижневої системи складові, що входять до рівняння (1.7), набувають наступного значення:

[K] – матриця жорсткості всієї конструкції в глобальній (загальній) системі координат – реакції в i -му напрямку від j -го одиничного переміщення;

{ δ } – шуканий вектор вузлових переміщень в глобальній системі координат;

{R} – вектор вузлових навантажень в глобальній системі координат. Додатні напрямки переміщень і зусиль у місцевій системі координат приведені на рис. 9.1 і 9.2.

Фізичне значення будь-якого рівняння системи (1.7) полягає в тому, що сумарна реакція за напрямком будь-якого шуканого переміщення від зовнішнього навантаження і невідомих переміщень дорівнює нулю.



Матриця [K] завжди симетрична, квадратна, її порядок рівний добутку кількості вузлів на число степенів вільності вузла. Її коефіцієнти визначаються як сума матриць жорсткості всіх кінцевих елементів, що входять до системи:

$$[\bar{K}] = \sum_{i=1}^n [K]^e. \quad (9.1)$$

Тут $[\bar{K}]$ – матриця жорсткості кінцевого елемента в глобальній системі координат, вона рівна

$$[\bar{K}^e] = [I]^T [K]^e [I], \quad (9.2)$$

де $[K]^e$ – матриця жорсткості кінцевого елемента в локальній (місцевій) системі координат, її коефіцієнти визначаються за допомогою виразу (9.5).

До матриці (9.5) входять наступні параметри:

E і G – модулі пружності матеріалу стрижня, відповідно першого і другого роду;

J_y і J_z – моменти інерції поперечного перерізу стрижня відносно головних центральних осей;

$J_{кр}$ – момент інерції перерізу при крученні для тонкостінного стрижня відкритого профілю, що складається з прямокутних смуг довжиною h_k і товщиною δ_k , визначається за формулою

$$J_{кр} = \frac{1}{3} \sum_1^m h_k d_k^3. \text{ Тут } k - \text{ номер смужки } (k = 1, 2, \dots, m);$$

m – кількість смуг, що входять до перерізу.

Для коробчатого прямокутного перерізу $J_{кр}$ визначається за формулою

$$J_{кр} = \frac{h_0^2 b_0^2 t_1 t_2}{h_0 t_2 + b_0 t_1 - t_1^2 - t_2^2}.$$

Позначення вхідних до цієї формули величин див. на рис. 9.11.

$[\lambda]$ – матриця направляючих косинусів кутів нахилу осей елемента до осей глобальної системи координат;

$[\lambda]^T$ – матриця транспонована матриці $[\lambda]$.

$$[I] = \begin{bmatrix} [T] & 0 & 0 & 0 \\ 0 & [T] & 0 & 0 \\ 0 & 0 & [T] & 0 \\ 0 & 0 & 0 & [T] \end{bmatrix}. \quad (9.3)$$

$$[T] = \begin{bmatrix} \cos(x, \bar{x}) & \cos(x, \bar{y}) & \cos(x, \bar{z}) \\ \cos(y, \bar{x}) & \cos(y, \bar{y}) & \cos(y, \bar{z}) \\ \cos(z, \bar{x}) & \cos(z, \bar{y}) & \cos(z, \bar{z}) \end{bmatrix}. \quad (9.4)$$

$$\cos(x, \bar{x}) = \frac{\bar{x}_j - \bar{x}_i}{l}; \quad \cos(x, \bar{y}) = \frac{\bar{y}_j - \bar{y}_i}{l}; \quad \cos(x, \bar{z}) = \frac{\bar{z}_j - \bar{z}_i}{l},$$

де $\bar{x}_i, \bar{y}_i, \bar{z}_i$ і $\bar{x}_j, \bar{y}_j, \bar{z}_j$ – координати i - і j -тої точок елемента в глобальній системі координат.

$$l = \sqrt{(\bar{x}_j - \bar{x}_i)^2 + (\bar{y}_j - \bar{y}_i)^2 + (\bar{z}_j - \bar{z}_i)^2} - \text{довжина стрижня};$$

$\cos(y, \bar{y}) = \cos \gamma$; $\cos(z, \bar{y}) = \cos \beta$, де γ і β – відповідно кути нахилу головних центральних осей поперечного перерізу y і z до осі \bar{y} (рис.9.3).

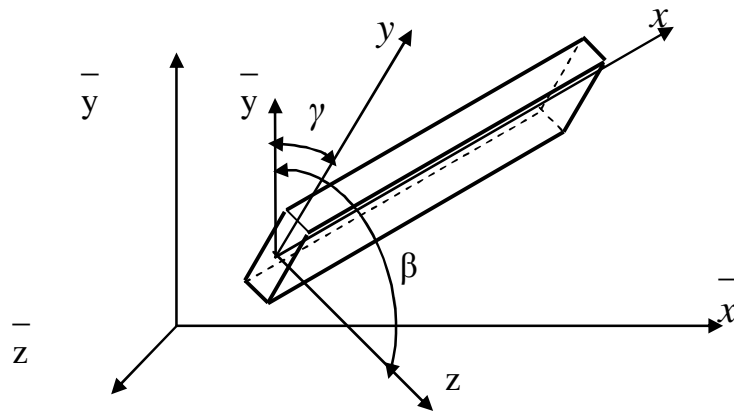


Рисунок 9.3 - Положення елемента в просторі, локальна і глобальна системи координат

Інші чотири напрямні косинуси в матриці (9.5) визначаються з умови взаємної перпендикулярності осей:

$$\cos^2(y, \bar{x}) + \cos^2(y, \bar{y}) + \cos^2(y, \bar{z}) = 1;$$

$$\cos^2(z, \bar{x}) + \cos^2(z, \bar{y}) + \cos^2(z, \bar{z}) = 1;$$

$$\cos(x, \bar{x})\cos(y, \bar{x}) + \cos(x, \bar{y})\cos(y, \bar{y}) + \cos(x, \bar{z})\cos(y, \bar{z}) = 0;$$

$$\cos(x, \bar{x})\cos(z, \bar{x}) + \cos(x, \bar{y})\cos(z, \bar{y}) + \cos(x, \bar{z})\cos(z, \bar{z}) = 0.$$

9.2 Реалізація розрахунку просторової стрижневої системи МКЕ на ПЕОМ

9.2.1 Блок-схеми програми

Для розрахунку просторової стрижневої конструкції розроблена програма STERGV.EXE. Її блок-схеми наведені на рис. 9.4...9.8.

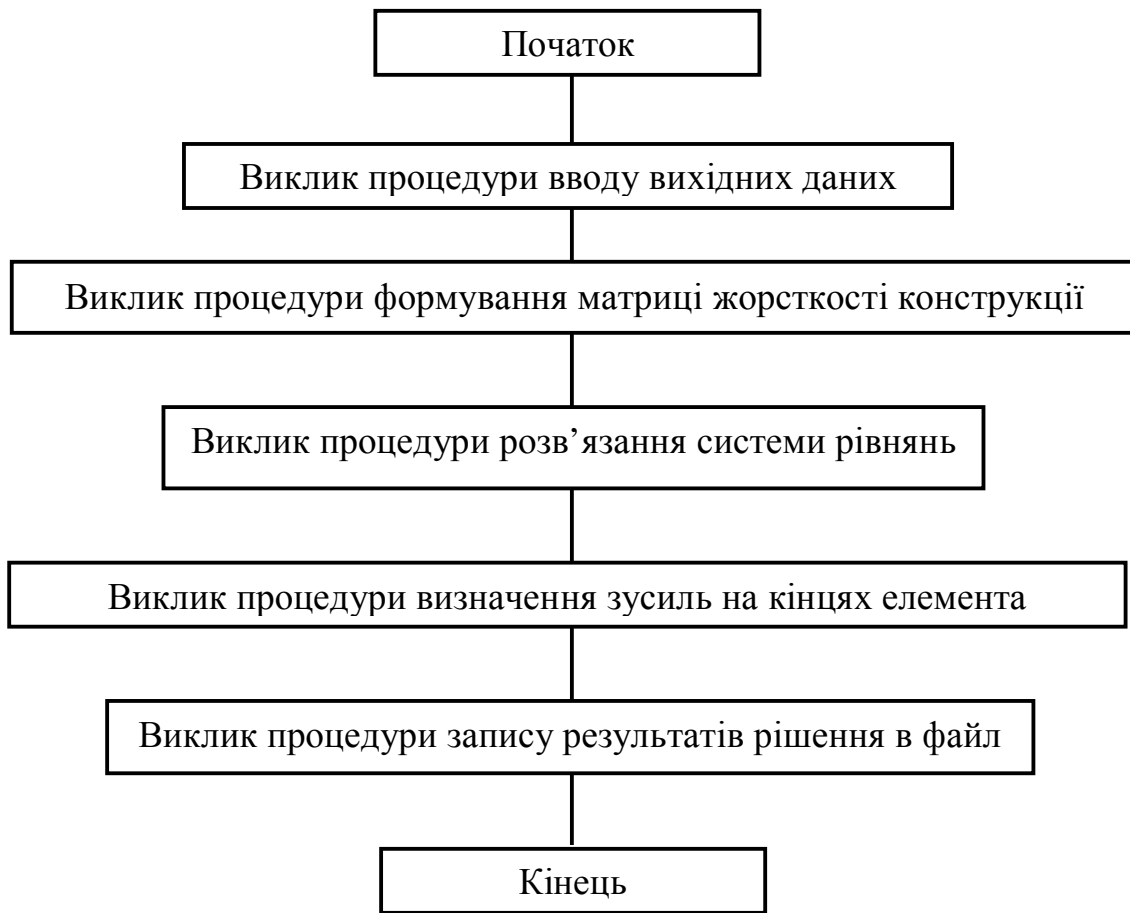


Рисунок 9.4 – Блок-схема основної програми

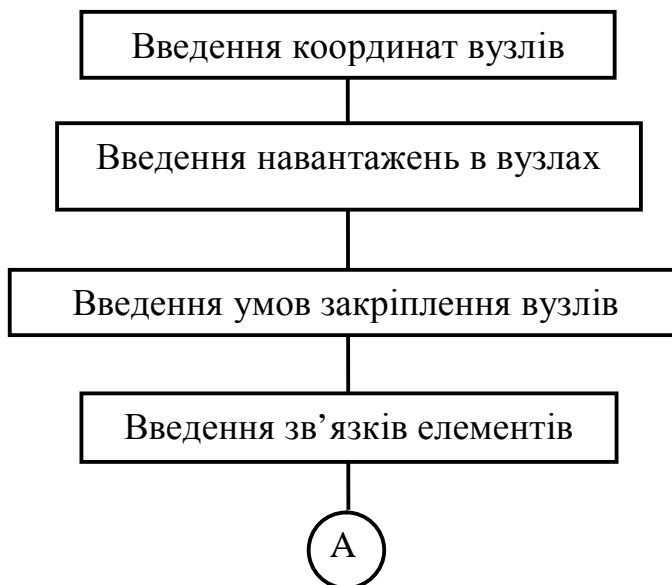


Рисунок 9.5 – Блок-схема процедури введення вихідних даних

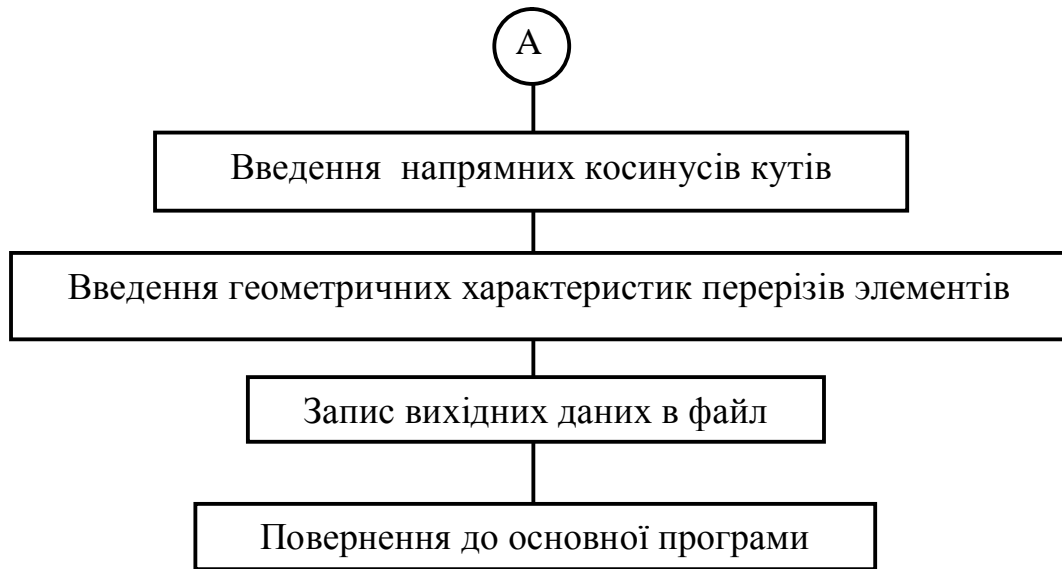


Рисунок 9.5, аркуш 2



Рисунок 9.6 – Блок-схема процедури матриці жорсткості елемента



Рисунок 9.7 – Блок-схема процедури матриці жорсткості конструкції



Рисунок 9.8 – Блок-схема процедури визначення зусиль на кінцях елемента

9.2.2 Введення початкових даних

З появою МКЕ і могутньої обчислювальної техніки стало можливим досліджувати напружено-деформований стан найскладніших конструкцій без яких-небудь спрощень. Але для того, щоб ввести в комп'ютер математичну модель конструкції, необхідно підготувати відповідні початкові дані. Як приклад візьмемо консольну просторову раму (рис. 9.9) з коробчатим прямокутним перерізом (рис. 9.10).

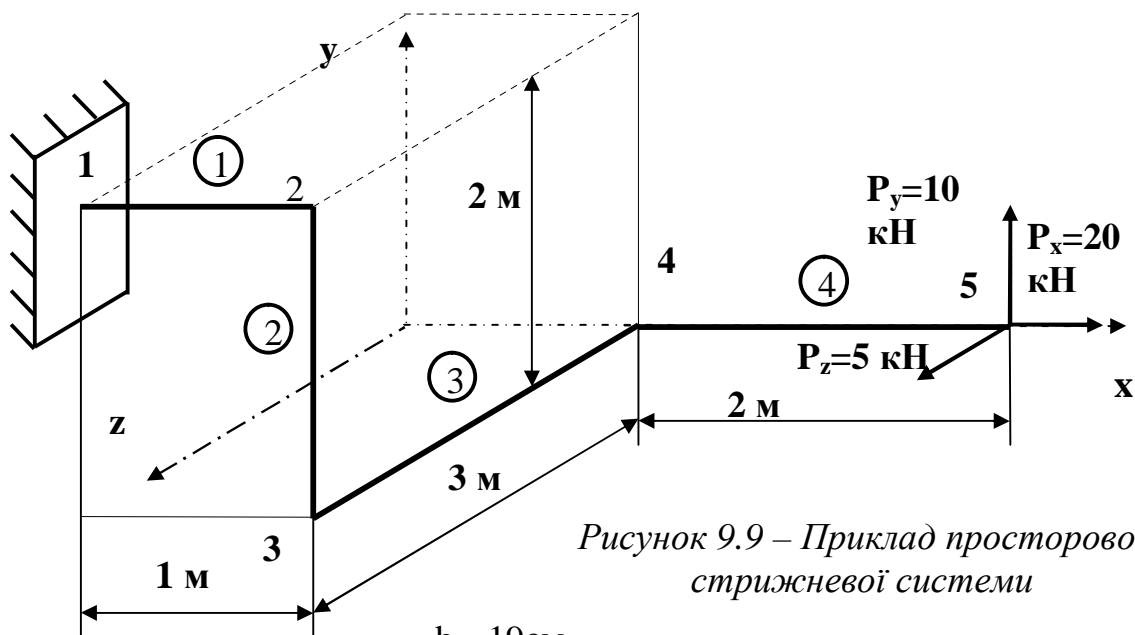


Рисунок 9.9 – Приклад просторової стрижневої системи

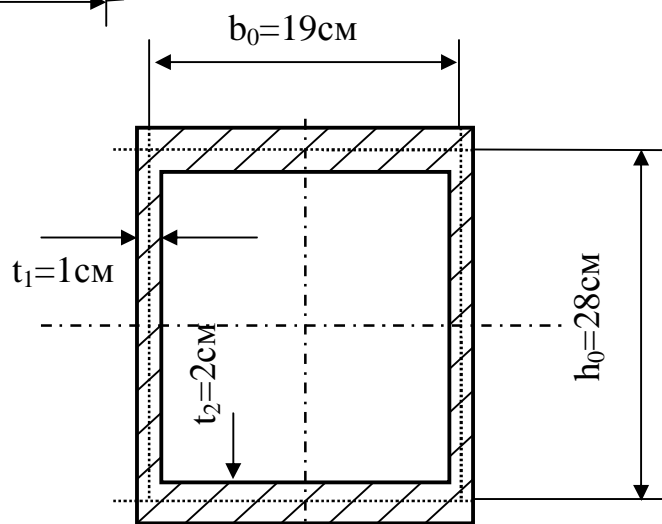


Рисунок 9.10 – Поперечний переріз рами

Тут навмисно взята проста рама, результати рахунку для якої легко можна перевірити без особливих зусиль.

Найбільш складним зі всіх етапів рішення задачі МКЕ на ЕОМ є введення початкових даних.

При їх введенні необхідно ввести кількість вузлів, елементів, їх зв'язки, навантаження, граничні умови, властивості матеріалу.

При запуску програми STERGV.EXE з'являється наступне меню:

- Введення початкових даних з файла
- Введення початкових даних з дисплея
- Інструкція
- Вихід з програми

Якщо дане завдання вирішується вперше, то дані вводяться з дисплея, при цьому заповнюється наступна таблиця 9.1, що з'явилася на екрані дисплея.

Таблиця 9.1 – Введення початкових даних з дисплея

N %	Координати вузлів			Гр. умови	Навантаження у вузлах					
	X, м	Y, м	Z, м		P _x ,кН	P _y ,кН	P _z ,кН	M _x , кНм	M _y ,кНм	M _z , кНм
1	0	2.0	3.0	111111	0	0	0	0	0	0
2	1.0	2.0	3.0	0	0	0	0	0	0	0
3	1.0	0	3.0	0	0	0	0	0	0	0
4	1.0	0	0	0	0	0	0	0	0	0
5	3.0	0	0	0	20	10	5	0	0	0
N / п	Зв'язки елементів		Косинуси кутів		Тип перерізу	Геометричні характеристики				
						F, см ²	J _z , см ⁴	J _y , см ⁴	J _{кр} , см ⁴	
1	1	2	0.000	1.000	1	132.0	18736	7364.0	8026.0	
2	2	3	0.000	0.000	1	132.00	18736	7364.0	8026.0	
3	3	4	1.000	0.000	1	132.0	18736	7364.0	8026.0	
4	4	5	0.000	1.000	1	132.0	18736	7364.0	8026.0	

Як видно, дисплей розподілений на дві частини, перехід з однієї частини в іншу здійснюється за допомогою клавіші F6, додавання рядка – F7, видалення – F8, а вихід – F10.

Украв необхідно введені початкові дані заносити у файл з тим, щоб при їх коректуванні або повторному рішенні даної задачі не повторювати введення даних, а викликати з відповідного файлу. Для цього натискаємо клавішу F10, на екрані з'являється меню:

Продовжити
Дані записати у файл
Редагування даних
Закінчити роботу

Після чого переходимо до режиму «Дані записати у файл», привласнюємо відповідне ім'я файлу і натискаємо «Введення». Переходимо до режиму «Продовжити» – вирішується завдання і початкові дані і результати рішення задачі записуються у файл RES.DAT.

Початкові дані у відповідному файлі (тут файл A1.dat) містяться у такому вигляді, їх можна коректувати безпосередньо у файлі за допомогою клавіші F4

Таблиця 9.2 – Початкові дані з файла A1.dat

5	4							
0.000	2.000	3.000		11111	0	0	0	0
				1				
1.000	2.000	3.000		0	0	0	0	0
1.000	0.000	3.000		0	0	0	0	0
1.000	0.000	0.000		0	0	0	0	0
3.000	0.000	0.000		0	20	10	5	0
1	2	0.000	1.000	1	132.0	18736	7364.0	8026.0
2	3	0.000	0.000	1	132.0	18736	7364.0	8026.0
3	4	1.000	0.000	1	132.0	18736	7364.0	8026.0
4	5	0.000	1.000	1	132.0	18736	7364.0	8026.0

9.2.3 Результати розрахунку і їх аналіз

Результати розрахунку необхідно дивитися у файлі res.dat. Це можна зробити, натиснувши клавішу F4, їх можна перенести у WORD-файл, де вони подані в такій формі:

РОЗРАХУНОК СТРИЖНЕВОЇ ПРОСТОРОВОЇ КОНСТРУКЦІЇ МЕТОДОМ КІНЦЕВИХ ЕЛЕМЕНТІВ

Початкові дані

Модулі пружності: першого і другого роду в МПа: 200000 80000

Координати вузлів і вид їх закріплення

Вузол	X	Y	Z	Гр.умови	Вузол	X	Y	Z	Гр.умови
1	0.000	2.000	3.000	111111	2	1.000	2.000	3.000	0
3	1.000	0.000	3.000	0	4	1.000	0.000	0.000	0
5	3.000	0.000	0.000	0					

Навантаження у вузлах: сили в кН, моменти в кН·м

Вузол	R _x	R _y	R _z	M _x	M _y	M _z
5	20.00	10.00	0.00	0.00	0.00	0.00

Зв'язки елементів, площа, моменти інерції і напрям. косинуси ел-тів

Ел-т	Зв'язки	F	I _x	I _y	I _{кр}	Косинуси е-тів
1	1 2	132.00	18736.00	7364.00	8026.00	0.000 1.000
2	2 3	132.00	18736.00	7364.00	8026.00	0.000 0.000
3	3 4	132.00	18736.00	7364.00	8026.00	1.000 0.000
4	4 5	132.00	18736.00	7364.00	8026.00	0.000 1.000

Результати рахунку – переміщення вузлів

Вузол	X, cm	Y, cm	Z, cm	θ _X	θ _Y	θ _Z
1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0008	0.2263	0.0979	0.0031	-0.0019	0.0044
3	1.1325	0.2271	-0.8420	0.0065	-0.0237	0.0065
4	9.7819	2.4202	-0.8414	0.0077	-0.0319	0.0159
5	9.7834	5.7798	5.5715	0.0077	-0.0322	0.0173

Результати рахунку – внутрішні зусилля в стрижнях

Ел-т	N, кН	Q _y ,кН	Q _z ,кН	M _x ,кН*см	M _y ,кН*см	M _z ,кН*см
1	-20.00	-10.00	-5.00	-2000.00	7500.00	-7000.00
	20.00	10.00	5.00	2000.00	-7000.00	6000.00
2	10.00	-5.00	20.00	-7000.00	-6000.00	2000.00
•	10.00	5.00	-20.00	7000.00	2000.00	-3000.00
3	5.00	-20.00	10.00	2000.00	-3000.00	-7000.00
•	5.00	20.00	-10.00	-2000.00	0.00	1000.00
4	-20.00	-10.00	-5.00	-0.00	1000.00	-2000.00
	20.00	10.00	5.00	0.00	0.00	0.00

Як бачимо, спочатку наводяться початкові дані для контрольного прикладу, поданого на рис. 9.9, а потім результати обчислень.

Додатні напрямки переміщень показані на рис. 9.1, а для внутрішніх зусиль – на рис. 9.2.

Вище наведені зусилля в елементах, де у верхньому рядку наведені зусилля в першому вузлі для даного елемента, а в нижньому – для другого вузла (наприклад для третього елемента першим вузлом є третій вузол, а другим – четвертий).

Нижче (рис. 9.11) наведені зусилля для першого елемента, як для найбільш навантаженого. Аналогічно можна було показати зусилля і для

решти елементів, за якими побудовані епюри подовжніх і поперечних сил, згинальних і крутних моментів(рис.9.12...9.15).

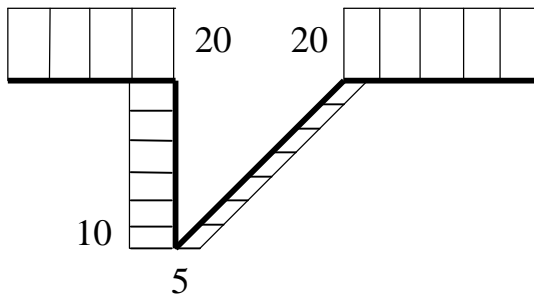


Рисунок 9.12 - Епюра подовжніх сил, кН

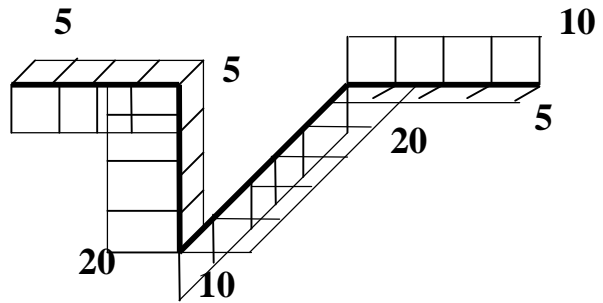


Рисунок 9.13 – Епюра поперечних сил, кН

Маючи епюри внутрішніх зусиль, можна визначити напруження в найбільш навантаженому елементі, використовуючи формули опору матеріалів.

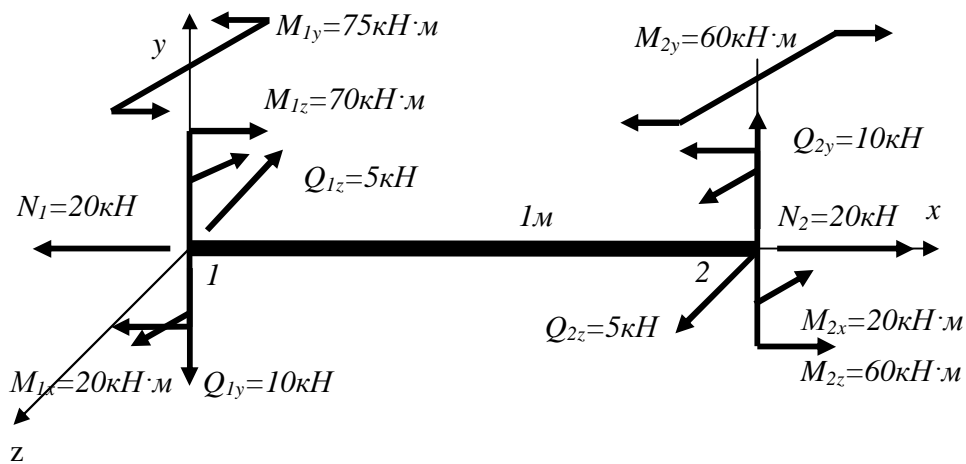


Рисунок 9.11 – Схема зусиль, які виникають в елементі 1

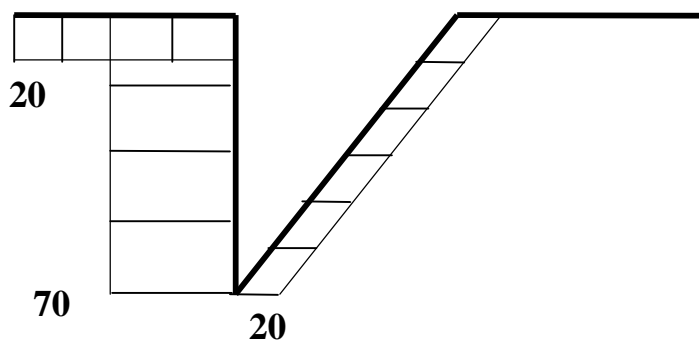


Рисунок 9.14 - Епюра крутних моментів, кН·м

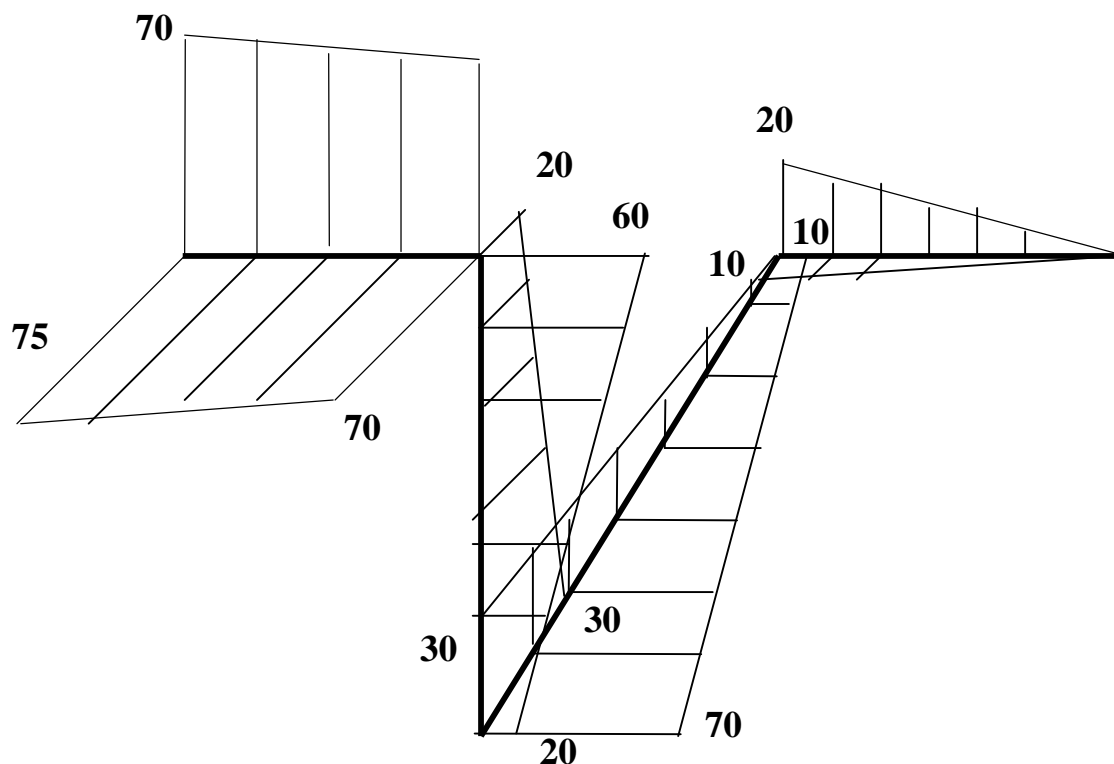


Рисунок 9.15 – Епюра згинальних моментів, $\text{кН}\cdot\text{м}$

10 РОЗРАХУНОК ПЛОСКОЇ ЗАДАЧІ ТЕОРІЇ ПРУЖНОСТІ

10.1 Поняття про плоску задачу теорії пружності

До цих пір розглядалися стрижневі конструкції, для яких МКЕ дає точне рішення. Для задач теорії пружності, в яких сили діють в площині пластини, МКЕ дає наближене рішення, яке залежить від виду кінцевих елементів і їх кількості.

10.2 Узагальнений закон Гука

З опору матеріалів відомі залежності (рис. 10.1) між напруженнями і деформаціями при об'ємному напруженому стані, які називаються узагальненим законом Гука.

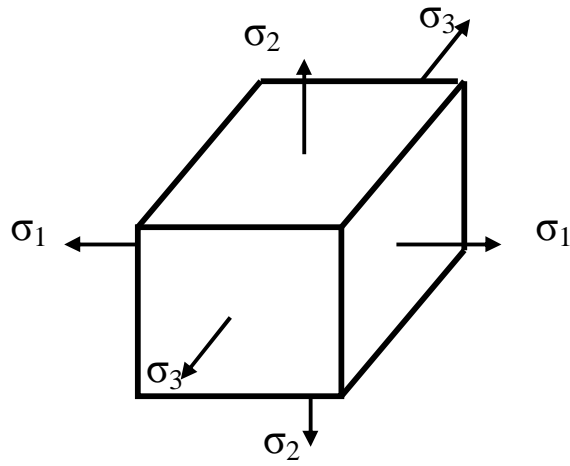


Рисунок 10.1 – Об’ємний напружений стан

Вони мають вигляд:

$$\begin{aligned}
 e_1 &= \frac{1}{E} [s_1 - m(s_2 + s_3)], \\
 e_2 &= \frac{1}{E} [s_2 - m(s_1 + s_3)], \\
 e_3 &= \frac{1}{E} [s_3 - m(s_1 + s_2)],
 \end{aligned}
 \tag{10.1}$$

де s_1, s_2, s_3 – головні нормальні напруження;
 e_1, e_2, e_3 – головні деформації.

10.3 Плоский напружений стан

Для плоскої задачі теорії пружності розрізняють плоский напружений і плоский деформований стани.

При плоскому напруженому стані напруження, перпендикулярні площини, відсутні (рис. 10.2):

$$s_z = 0, t_{xz} = 0, t_{yz} = 0.$$

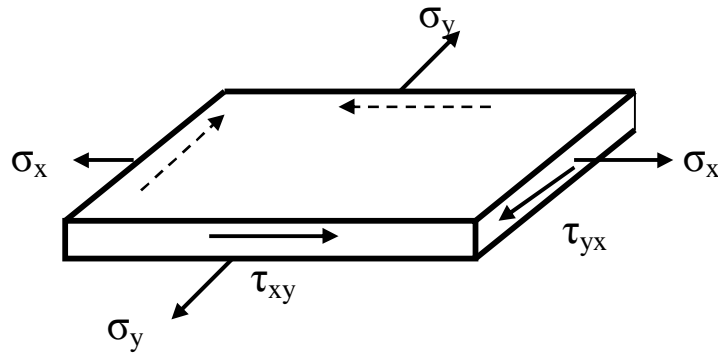


Рисунок 10.2 - Плоский напружений стан

Узагальнений закон Гука для цього випадку має вигляд

$$\left. \begin{aligned} e_x &= \frac{1}{E} (s_x - m s_y) \\ e_y &= \frac{1}{E} (s_y - m s_x) \\ e_z &= -\frac{m}{E} (s_x + s_y) \end{aligned} \right\} \quad (10.2)$$

Виразимо тепер s_x і s_y через e_1 і e_2 :

$$\begin{aligned} s_x &= \frac{E}{1-m^2} (e_1 + m e_2), \\ s_y &= \frac{E}{1-m^2} (e_2 + m e_1), \\ t &= Gg = \frac{E}{2(1+m)} g. \end{aligned}$$

У матричній формі маємо:

$$\begin{Bmatrix} s_x \\ s_y \\ t_{xy} \end{Bmatrix} = \frac{E}{1-m^2} \begin{vmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1-m}{2} \end{vmatrix} \begin{Bmatrix} e_x \\ e_y \\ g \end{Bmatrix} \quad (10.3)$$

10.4 Плоский деформований стан

При плоскому деформованому стані всі деформації знаходяться в одній площині (рис.10.3), $e_z = 0$.Тоді узагальнений закон Гука запишеться так:

$$e_x = \frac{1}{E} [s_x - m(s_y + s_z)],$$

$$e_y = \frac{1}{E} [s_y - m(s_x + s_z)],$$

$$e_z = \frac{1}{E} [s_z - m(s_x + s_y)].$$

Звідси визначаємо s_z і підставляємо в e_x і e_y : $s_z = m(s_x + s_y)$,

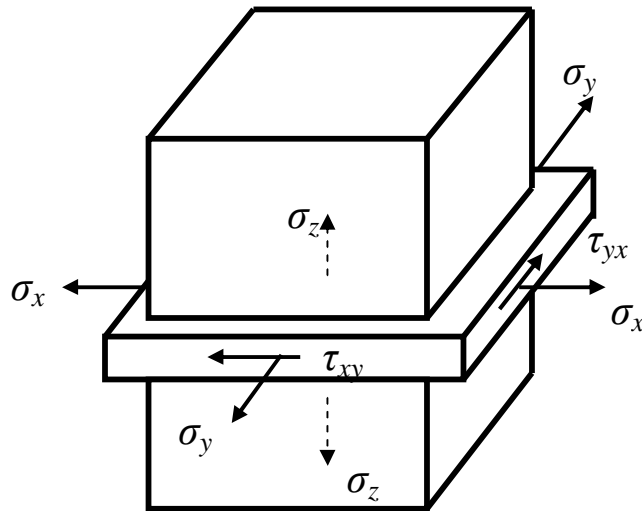


Рисунок 10.3 - Плоский деформований стан

$$\left. \begin{aligned} e_x &= \frac{1}{E} [s_x - ms_y - m^2(s_x + s_y)] \\ e_y &= \frac{1}{E} [s_y - ms_x - m^2(s_x + s_y)] \end{aligned} \right\} \quad (10.4)$$

Напруження, виражені через деформації, дорівнюють

$$s_x = \frac{E(1-m)}{(1-2m)(1+m)} \left(e_x + \frac{1-m}{m} e_y \right)$$

$$s_y = \frac{E(1-m)}{(1-2m)(1+m)} \left(e_y + \frac{1-m}{m} e_x \right)$$

$$t_{xy} = Gg_{xy} = \frac{E}{2(1+m)} g_{xy},$$

або, в матричній формі

$$\begin{Bmatrix} s_x \\ s_y \\ t_{xy} \end{Bmatrix} = \frac{E(1-m)}{(1-2m)(1+m)} \begin{vmatrix} 1 & \frac{1-m}{m} & 0 \\ \frac{1-m}{m} & 1 & 0 \\ 0 & 0 & \frac{1-2m}{2(1-m)} \end{vmatrix} \begin{Bmatrix} e_x \\ e_y \\ g_{xy} \end{Bmatrix}. \quad (10.5)$$

10.5 Зв'язок між деформаціями і переміщеннями

Розглянемо деформацію елемента з розмірами dx і dy (рис. 10.4).

Під дією сил елемент деформується (на рис 10.4 показано пунктиром).

Визначимо відносні деформації e_x , e_y і g_{xy} :

$$e_x = \frac{A_1D_1 - AD}{AD},$$

$$AD = dx,$$

$$A_1D_1 = AD + DD_1 - AA_1 = dx + u + \frac{\partial u}{\partial x} dx - u = dx + \frac{\partial u}{\partial x} dx,$$

(проекції на вісь x визначаємо приблизно),

$$e_x = \frac{dx + \frac{\partial u}{\partial x} dx - dx}{dx} = \frac{\partial u}{\partial x},$$

$$e_y = \frac{A_1 B_1 - AB}{AB}, \quad AB = dy, \quad A_1 B_1 = AB + BB_1 - AA_1,$$

(проекції на вісь y визначаємо приблизно),

$$A_1 B_1 = dy + v + \frac{\partial v}{\partial y} dy - v = dy + \frac{\partial v}{\partial y} dy,$$

$$e_y = \frac{dy + \frac{\partial v}{\partial y} dy}{dy} = \frac{\partial v}{\partial y}.$$

Відносним зсувом називають зменшення прямого кута BAD .

$$g_{xy} = g_1 + g_2,$$

$$g_1 = \frac{D_1 D_2 - A_1 A_3}{A_1 D_3} = \frac{v + \frac{\partial v}{\partial x} dx - v}{dx} = \frac{\partial v}{\partial x},$$

$$g_2 = \frac{B_1 B_2 - A_1 A_2}{B_3 A_1} = \frac{u + \frac{\partial u}{\partial y} dy - u}{dy} = \frac{\partial u}{\partial y},$$

таким чином,

$$g_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}.$$

У матричній формі маємо відомі залежності Коші:

$$\begin{Bmatrix} e_x \\ e_y \\ g_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}. \quad (10.6)$$

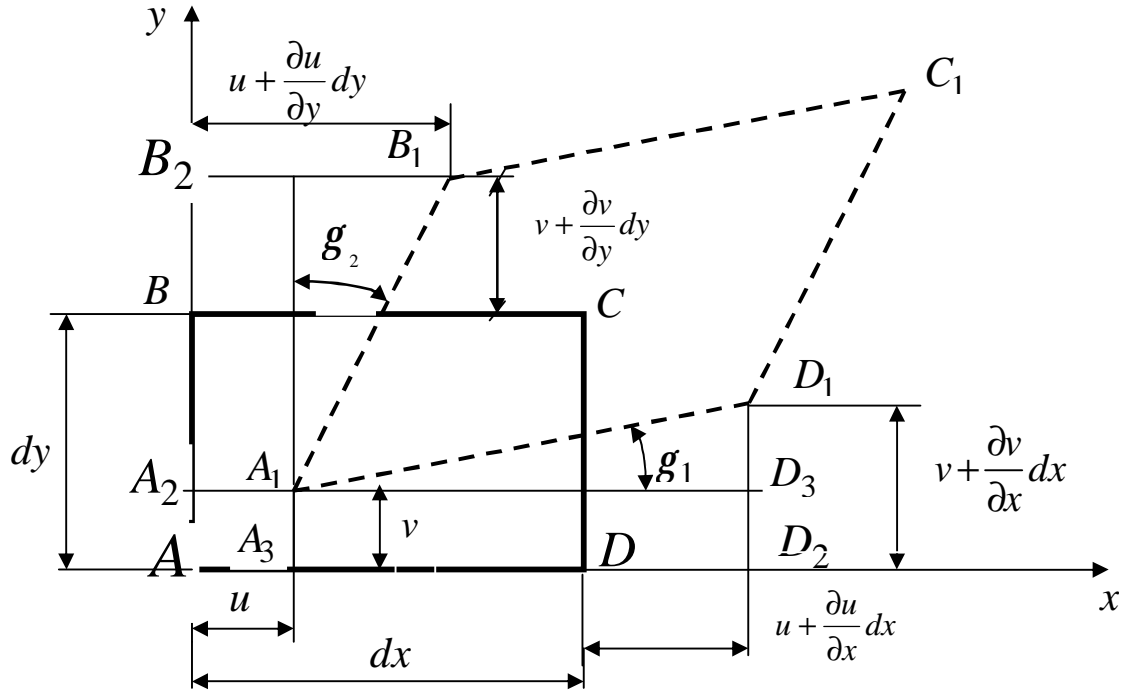


Рисунок 10.4 - Елемент ABCD до і після деформації

10.6 Зв'язок між напруженнями і переміщеннями

Для вирішення задач теорії пружності МКЕ необхідно напруження виразити через переміщення. Для цього у вирази (10.3) і (10.5) необхідно підставити залежність (10.6).

Для плоского напруженого стану

$$\begin{Bmatrix} s_x \\ s_y \\ t_{xy} \end{Bmatrix} = \frac{E}{1-m^2} \begin{vmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1-m}{2} \end{vmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}. \quad (10.7)$$

$$\text{де } |D| = \frac{E}{1-m^2} \begin{vmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1-m}{2} \end{vmatrix} \quad (10.8)$$

матриця пружності для плоского напруженого стану.

Для плоского деформованого стану

$$\begin{Bmatrix} s_x \\ s_y \\ t_{xy} \end{Bmatrix} = \frac{E(1-m)}{(1-2m)(1+m)} \begin{vmatrix} 1 & \frac{m}{1-m} & 0 \\ \frac{m}{1-m} & 1 & 0 \\ 0 & 0 & \frac{1-2m}{2(1-m)} \end{vmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}, \quad (10.9)$$

де

$$|D| = \frac{E(1-m)}{(1-2m)(1+m)} \begin{vmatrix} 1 & \frac{m}{1-m} & 0 \\ \frac{m}{1-m} & 1 & 0 \\ 0 & 0 & \frac{1-2m}{2(1-m)} \end{vmatrix}. \quad (10.10)$$

матриця жорсткості для плоского деформованого стану.

10.7 Функція переміщень для кінцевого трикутного елемента

Найбільш простими є трикутні елементи, якими легко апроксимувати будь-яку конструкцію (рис. 10.5).

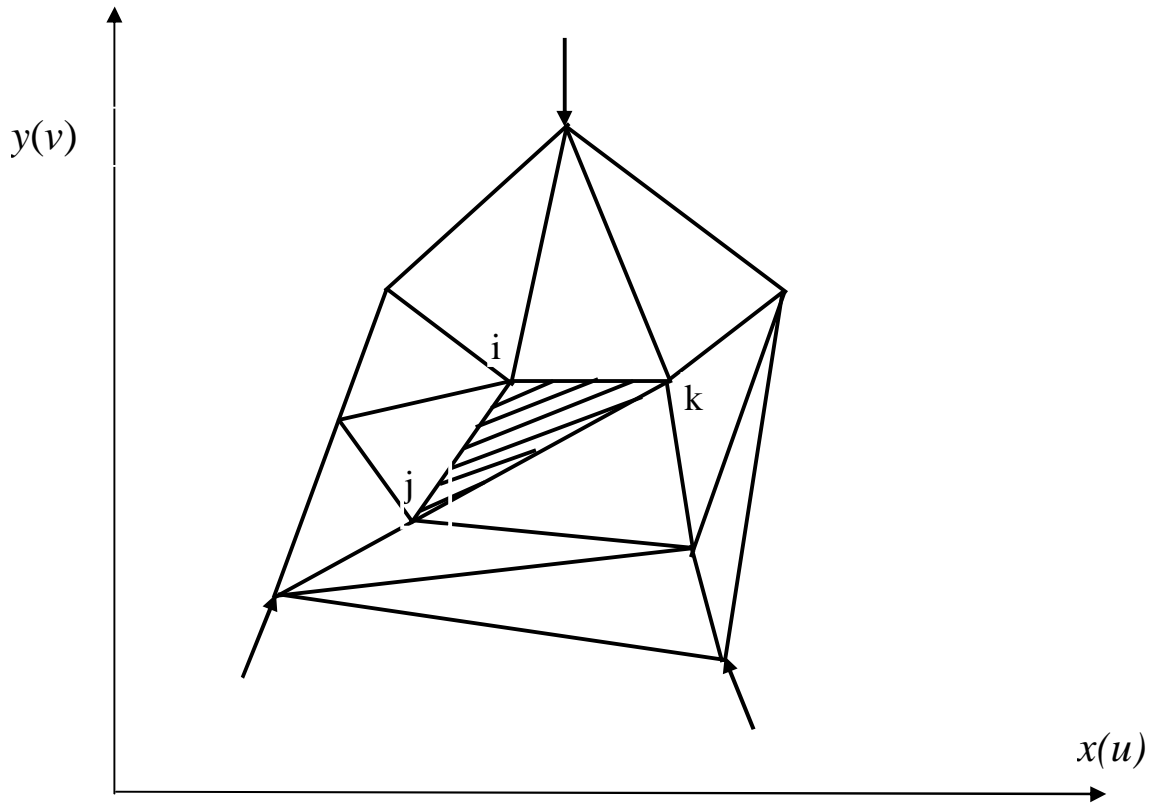


Рисунок 10.5 - Розбивання конструкції на елементи

Оскільки кінцевий елемент має три вузли, то в якості функції переміщень необхідно узяти поліном з трьома коефіцієнтами. У напрямку осі x переміщення u рівні

$$u = a_0 + a_1 x + a_2 y. \quad (10.11)$$

Коефіцієнти a_0, a_1, a_2 визначаються через переміщення вузлів з системи рівнянь

$$\left. \begin{aligned} u_i &= a_0 + a_1 x_i + a_2 y_i \\ u_j &= a_0 + a_1 x_j + a_2 y_j \\ u_k &= a_0 + a_1 x_k + a_2 y_k \end{aligned} \right\}. \quad (10.12)$$

Вирішуючи (10.12) за допомогою визначників, одержуємо:

$$a_0 = \frac{\begin{vmatrix} u_i & x_i & y_i \\ u_j & x_j & y_j \\ u_k & x_k & y_k \end{vmatrix}}{\begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix}} ; \quad \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} = 2\Delta ,$$

де Δ – площа елемента

$$a_1 = \frac{\begin{vmatrix} 1 & u_i & y_i \\ 1 & u_j & y_j \\ 1 & u_k & y_k \end{vmatrix}}{2\Delta} ; \quad a_2 = \frac{\begin{vmatrix} 1 & x_i & u_i \\ 1 & x_j & u_j \\ 1 & x_k & u_k \end{vmatrix}}{2\Delta} .$$

Підставивши ці значення у вираз (10.11) і згрупувавши коефіцієнти, одержимо:

$$u = \left| \frac{a_i + b_i x + c_i y}{2\Delta} ; \frac{a_j + b_j x + c_j y}{2\Delta} ; \frac{a_k + b_k x + c_k y}{2\Delta} \right| \begin{Bmatrix} u_i \\ u_j \\ u_k \end{Bmatrix} . \quad (10.13)$$

Аналогічно, переміщення у напрямку осі y :

$$v = \left| \frac{a_i + b_i x + c_i y}{2\Delta} ; \frac{a_j + b_j x + c_j y}{2\Delta} ; \frac{a_k + b_k x + c_k y}{2\Delta} \right| \begin{Bmatrix} v_i \\ v_j \\ v_k \end{Bmatrix} . \quad (10.14)$$

$$\text{Тут } a_i = x_j y_k - x_k y_j , b_i = y_j - y_k , c_i = x_k - x_j , \quad (10.15)$$

решта коефіцієнтів визначається шляхом кругової заміни індексів відповідно i, j, k .

10.8 Матриця жорсткості для трикутного кінцевого елемента

Підставивши функції переміщень (10.13) і (10.14) у (10.6), одержимо вектор деформацій:

$$\{e\} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = \frac{1}{2\Delta} \begin{vmatrix} b_i & 0 & b_j & 0 & b_k & 0 \\ 0 & c_i & 0 & c_j & 0 & c_k \\ c_i & b_i & c_j & b_j & c_k & b_k \end{vmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix}. \quad (10.16)$$

$$\text{Тут} \quad \frac{1}{2\Delta} \begin{vmatrix} b_i & 0 & b_j & 0 & b_k & 0 \\ 0 & c_i & 0 & c_j & 0 & c_k \\ c_i & b_i & c_j & b_j & c_k & b_k \end{vmatrix} = |B|. \quad (10.17)$$

Для обчислення матриці жорсткості елемента підставляємо всі складові, що входять до формули (1.6). Внаслідок чого маємо:

$$|k|^{\mathbf{1}} = \int_V |B|^T |D| |B| dV = \int_F |B|^T |D| |B| t \cdot dF, \\ dV = dF \cdot t.$$

Оскільки під знаком інтеграла маємо $|B|$ і $|D|$, які не залежать від координат x і y , то одержимо остаточний вираз матриці жорсткості для трикутних кінцевих елементів:

$$|k|^{\mathbf{1}} = |B|^T |D| |B| t \cdot \Delta, \quad (10.18)$$

де t – товщина кінцевого елемента.

10.9 Функція переміщення для прямокутного кінцевого елемента

У багатьох випадках зручно конструкцію розбивати на плоскі прямокутні елементи (рис. 10.6).

Для таких елементів (рис. 10.7) функції u і v переміщень довільної точки задають у вигляді

$$u(x, y) = \left| \frac{1}{ab}(a-x)(b-y); \frac{1}{ab}x \cdot (b-y); \frac{1}{ab}xy; \frac{1}{ab}y \cdot (a-x) \right| \begin{Bmatrix} u_i \\ u_j \\ u_k \\ u_m \end{Bmatrix}, \quad (10.19)$$

$$v(x, y) = \left| \frac{1}{ab}(a-x)(b-y); \frac{1}{ab}x \cdot (b-y); \frac{1}{ab}xy; \frac{1}{ab}y \cdot (a-x) \right| \begin{Bmatrix} v_i \\ v_j \\ v_k \\ v_m \end{Bmatrix}. \quad (10.20)$$

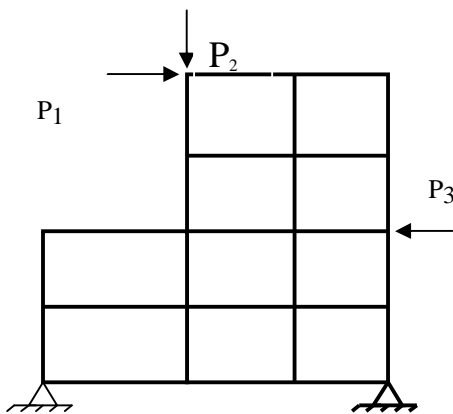


Рисунок 10.6 – Схема конструкції, яка розбита на прямокутні елементи

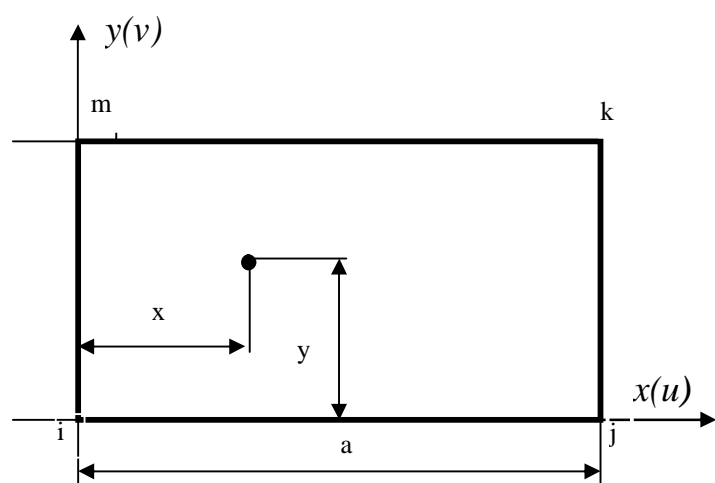


Рисунок 10.7 - Кінцевий прямокутний елемент

Підставивши ці функції у вирази деформацій (10.6), одержимо

$\{e\} = [B]\{d\}^1$, де $\{d\}^1$ – вектор деформацій вузлів.

$$[B] = \frac{1}{ab} \begin{vmatrix} y-b & 0 & b-y & 0 & y & 0 & -y & 0 \\ 0 & x-a & 0 & -x & 0 & x & 0 & a-x \\ x-a & y-b & -x & b-y & x & y & a-x & -y \end{vmatrix}. \quad (10.21)$$

Матриця жорсткості прямокутного елемента являє собою інтеграл

$$|k|^1 = \int_F |B|^T |D| |B| dV = t \int_F \frac{1}{ab} \begin{vmatrix} y-b & 0 & x-a \\ 0 & x-a & y-b \\ b-y & 0 & -x \\ 0 & -x & b-y \\ y & 0 & x \\ 0 & x & y \\ -y & 0 & a-x \\ 0 & a-x & -y \end{vmatrix} |D| \times$$

$$\times \begin{vmatrix} y-b & 0 & b-y & 0 & y & 0 & -y & 0 \\ 0 & x-a & 0 & -x & 0 & x & 0 & a-x \\ x-a & y-b & -x & b-y & x & y & a-x & -y \end{vmatrix} dx dy. \quad (10.22)$$

Як видно з виразу (10.22), коефіцієнти матриці жорсткості є інтегралами, залежними від x і y . Розмір матриці жорсткості 8×8 , її коефіцієнти можна узяти з підручника [9].

10.10 Приклад

Для пластини, показаної на рис 10.8, необхідно визначити еквівалентні напруження за третьою теорією міцності, якщо $E=2 \cdot 10^5$ МПа, $\nu=0,3$.

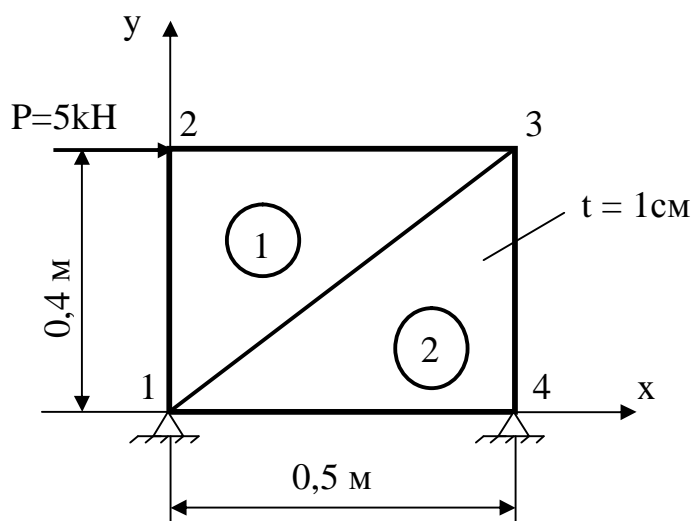


Рисунок 10.8 - Схема пластини

Розв'язання. Розбиваємо пластину на елементи. Для ілюстрації МКЕ розбиваємо всього на 2 елементи (природньо, таке розбиття дуже приблизне і не забезпечує достатньої точності розрахунку). Нумеруємо вузли і елементи.

Переходимо до обчислення матриць жорсткості елементів.

1-й елемент: його зв'язки 1-3-2 (починаємо з меншого номера і називаємо вузли, обходячи їх проти ходу годинникової стрілки).

Значить, і-й вузол –1, j-й – 3, k-й –2.

Обчислюємо коефіцієнти b і c :

$$b_i = y_j - y_k = 0,4 - 0,4 = 0, \quad c_i = x_k - x_j = 0 - 0,5 = -0,5, \quad ,$$

$$b_j = y_k - y_i = 0,4 - 0 = 0,4, \quad c_j = x_i - x_k = 0 - 0 = 0,$$

$$b_k = y_i - y_j = 0 - 0,4 = -0,4, \quad c_k = x_j - x_i = 0,5 - 0 = 0,5, \quad ,$$

$$\Delta = \frac{1}{2} \cdot 0,4 \cdot 0,5 = 0,1.$$

Матриця $|B|$ дорівнює

$$|B| = \frac{1}{0,2} \begin{vmatrix} 0 & 0 & 0,4 & 0 & -0,4 & 0 \\ 0 & -0,5 & 0 & 0 & 0 & 0,5 \\ -0,5 & 0 & 0 & 0,4 & 0,5 & -0,4 \end{vmatrix}.$$

Матриця жорсткості першого елемента:

$$|k|_I^1 = \frac{1}{0,2} \begin{vmatrix} 0 & 0 & -0,5 \\ 0 & -0,5 & 0 \\ 0,4 & 0 & 0 \\ 0 & 0 & 0,4 \\ -0,4 & 0 & 0,5 \\ 0 & 0,5 & -0,4 \end{vmatrix} \frac{2 \cdot 10^5 (1-0,3)}{(1-2 \cdot 0,3)(1+0,3)} \begin{vmatrix} 0 & \frac{0,3}{1-0,3} & 0 \\ 0,3 & 1 & 0 \\ 0 & 0 & \frac{1-2 \cdot 0,3}{2 \cdot (1-0,3)} \end{vmatrix} \times$$

$$\times \frac{1}{0,2} \begin{vmatrix} 0 & 0 & 0,4 & 0 & -0,4 & 0 \\ 0 & -0,5 & 0 & 0 & 0 & 0,5 \\ -0,5 & 0 & 0 & 0,4 & 0,5 & -0,4 \end{vmatrix} \cdot 0,01 \cdot 0,1;$$

$$|k|_I^1 = \begin{vmatrix} 481 & 0 & 0 & -385 & -481 & 385 \\ 0 & 1688 & -577 & 0 & 577 & -1682 \\ 0 & -577 & 1077 & 0 & -1077 & 577 \\ -385 & 0 & 0 & 308 & 385 & -308 \\ -481 & 577 & -1077 & 385 & 1558 & -962 \\ 385 & -1682 & 577 & -308 & -962 & 1990 \end{vmatrix}.$$

2-й елемент: його зв'язки 1-4-3 . Аналогічно визначаємо:

$$|k|_{II}^1 = \begin{vmatrix} 1077 & 0 & -1077 & 577 & 0 & -577 \\ 0 & 308 & 385 & -308 & -385 & 0 \\ -1077 & 385 & 1558 & -962 & -481 & 577 \\ 577 & -308 & -962 & 1990 & 385 & -1682 \\ 0 & -385 & -481 & 385 & 481 & 0 \\ -577 & 0 & 577 & -1682 & 0 & 1682 \end{vmatrix}.$$

Формуємо матрицю жорсткості конструкції. Оскільки пластина має чотири вузли, а кожен вузол має два степені вільності, то розмірність матриці жорсткості конструкції має розмірність 8x8. При занесенні коефіцієнтів до матриці жорсткості конструкції слід уважно стежити, які вузли включають елементи, і яка їх послідовність.

До правої частини заносимо навантаження, прикладені у вузлах. Оскільки навантаження прикладене тільки в другому вузлі в горизонтальному напрямку, то заносимо до третього рядка навантаження $5 \cdot 10^{-3}$ (переводиться в Меганьютони (МН)). У решті рядків ставимо нулі.

481	0	-481	385	0	-385	1077	577	$\left. \begin{matrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{matrix} \right\} \begin{matrix} 0 \\ 0 \\ 5 \cdot 10^3 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}$
1077	0			0	-577			
0	1082	577	-	-577	0	385	-308	
0	308		1682	-385	0			
-481	577	1558	-962	-1077	385	0	0	
385	-1682	-962	1990	577	-308	0	0	
0	-577	-1077	577	1077	0	-481	385	
0	-385			481	0			
-385	0	385	-308	0	308	577	-1682	
-577	0			0	1682			
-1077	385	0	0	-481	577	1558	-962	
577	-308	0	0	385	-1682	-962	1990	

Враховуємо граничні умови:

$$u_1 = 0, \quad v_1 = 0, \quad u_4 = 0, \quad v_4 = 0.$$

Після їх врахування одержимо:

$$\begin{vmatrix} 1558 & -962 & -1077 & 385 \\ -962 & 1990 & 577 & -308 \\ -1077 & 577 & 1558 & 0 \\ 385 & -308 & 0 & 1990 \end{vmatrix} \begin{Bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} = \begin{Bmatrix} 5 \cdot 10^{-3} \\ 0 \\ 0 \\ 0 \end{Bmatrix}.$$

Вирішуючи систему рівнянь, знайдемо:

$$u_2 = 0,0084м, v_2 = 0,0024м, u_3 = 0,0049м, v_3 = -0,00125м.$$

Визначаємо напруження в елементах.

1-й елемент:

$$\{s\} = \begin{Bmatrix} s_x \\ s_y \\ t_{xy} \end{Bmatrix} = [D][B] \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix} = \frac{2 \cdot 10^5 (1-0,3)}{(1-2 \cdot 0,3)(1+0,3)} \begin{vmatrix} 1 & \frac{0,3}{1-0,3} & 0 \\ 0,3 & 1 & 0 \\ 0 & 0 & \frac{1-2 \cdot 0,3}{2 \cdot (1-0,3)} \end{vmatrix} \times.$$

$$\times \frac{1}{2 \cdot 0,1} \begin{vmatrix} 0 & 0 & 0,4 & 0 & -0,4 & 0 \\ 0 & -0,5 & 0 & 0 & 0 & 0,5 \\ -0,5 & 0 & 0 & 0,4 & 0,5 & -0,4 \end{vmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0,049 \\ -0,00125 \\ 0,0084 \\ 0,0024 \end{Bmatrix} = \begin{Bmatrix} -118,4 \\ 84,2 \\ 105,3 \end{Bmatrix}.$$

2-й елемент:

$$\times \begin{vmatrix} -0,4 & 0 & 0,4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0,5 & 0 & 0,5 \\ 0 & -0,4 & -0,5 & 0,4 & 0,5 & 0 \end{vmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0,00401 \\ -0,00125 \end{Bmatrix} = \begin{Bmatrix} -36,1 \\ -84,2 \\ 94,7 \end{Bmatrix}.$$

$$\{s\} = \frac{2 \cdot 10^5 (1 - 0,3)}{(1 - 2 \cdot 0,3)(1 + 0,3)} \begin{vmatrix} 1 & \frac{0,3}{1 - 0,3} & 0 \\ 0,3 & 1 & 0 \\ 0 & 0 & \frac{1 - 2 \cdot 0,3}{2(1 - 0,3)} \end{vmatrix} \cdot \frac{1}{2 \cdot 0,1} \times$$

Як бачимо, більш навантаженим є 1-й елемент. Для нього визначаємо головні нормальні напруження за формулами:

$$s_{1,3} = \frac{1}{2} \left[s_x + s_y \pm \sqrt{(s_x - s_y)^2 + 4t_{xy}^2} \right],$$

$$s_1 = \frac{1}{2} \left[-118 + 84,2 + \sqrt{(-118 - 84,2)^2 + 4 \cdot 105,3^2} \right] = 123 \text{ МПа},$$

$$s_3 = \frac{1}{2} \left[-118 + 84,2 - \sqrt{(-118 - 84,2)^2 + 4 \cdot 105,3^2} \right] = -157 \text{ МПа}.$$

Оскільки маємо плоский деформований стан, то

$$s_2 = -m(s_1 + s_3) = -0,3(123 - 157) = 10,2 \text{ МПа}.$$

Відповідно до третьої теорії міцності

$$\sigma_{екв} = s_1 - s_3 \leq [s], \quad \sigma_{екв} = 123 + 157 = 280 \text{ МПа}.$$

10.11 Реалізація напружено-деформованого стану конструкції за допомогою ПЕОМ

10.11.1 Введення початкових даних

Текст програми наведений у додатку Б.

Розглянемо підготовку початкових даних, необхідних для введення у комп'ютер математичної моделі конструкції.

Прослідкуємо рішення задачі МКЕ на ЕОМ на прикладі плоскої задачі (див. рис. 10.9).

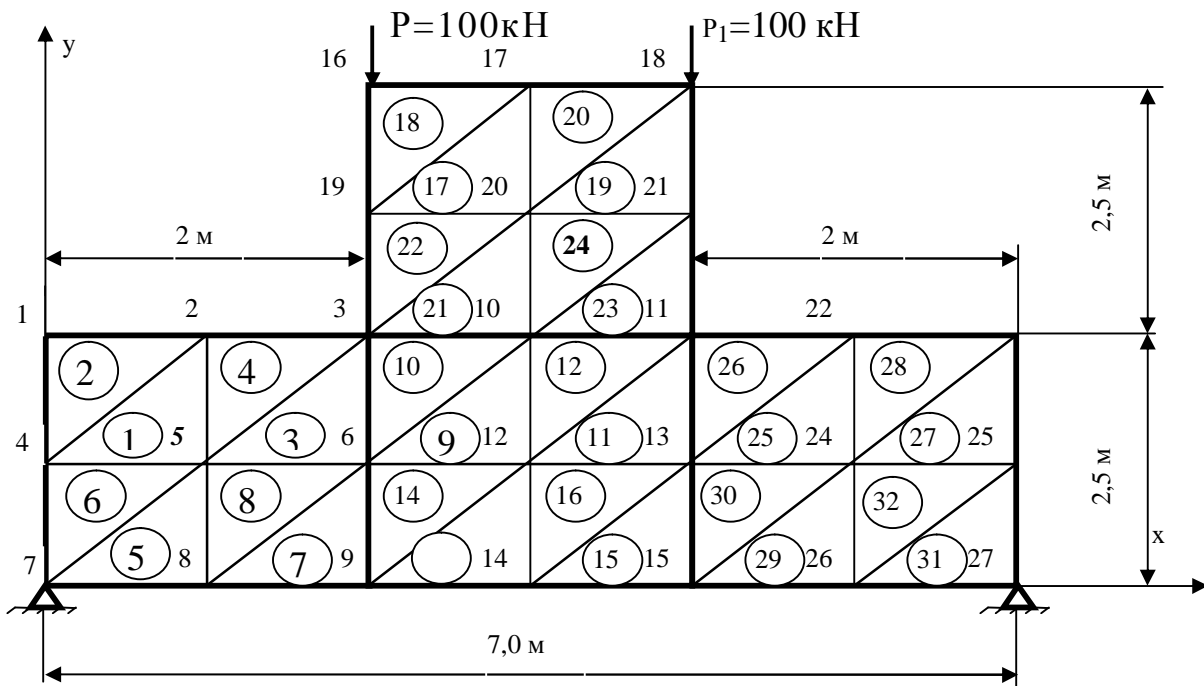


Рисунок 10.9 - Пластина з розбивкою на елементи

Найбільш складним зі всіх етапів рішення задачі МКЕ на ЕОМ є введення початкових даних. При введенні необхідно ввести кількість вузлів, елементів, їх зв'язки, навантаження, граничні умови, властивості матеріалу. Простий спосіб введення наведений в табл. 10.1 і 10.2.

Таблиця 10.1

Номер вузла	Координати		Граничні умови	Навантаження	
	x	y		P_x	P_y
1	0	2,5	00	0	0
2	1,0	2,5	00	0	0
.
7	0	0	11	0	0
.
16	2	5,0	00	0	-100
.
27	7,0	0	11	0	0

Таблиця 10.2

Номер елемента	Зв'язки			E	m
	4	2	5		
1	4	2	5	200 000	0,3
2	1	4	2	200 000	0,3
.
32	24	26	25	200 000	0,3

Найбільш трудомістке ручне введення, коли конструкція уручну розбивається на елементи.

Рекомендується введення даних виконувати за допомогою програми, яка дозволяє автоматично розбивати конструкції на елементи, формувати зв'язки елементів, визначати координати вузлів. Її текст наведений у додатку Г.

Укряй необхідно введені початкові дані заносити у файл, щоб при їх коректуванні вводити не всі дані, а тільки ті, що міняються. Бажано задавати графічне введення даних, які потім трансформуються в числові та заносяться у файл. Граничні умови (умови закріплення) задають у вигляді масиву чисел з нулями і одиницями. Одиниці відповідають закріпленням вузлів в певному напрямку, а нулі – їх відсутності.

10.11.2 Виведення початкових даних на екран й у файл

Залежно від того, як було організовано введення даних, вони повинні відразу ж виводитися на екран комп'ютера і заноситися у файл. Це необхідно, щоб контролювати, чи вірно введені дані, і при необхідності повторного введення викликати відповідний файл.

10.11.3 Визначення коефіцієнтів матриці жорсткості елемента

Залежно від даного завдання коефіцієнти матриці жорсткості елемента повинні визначатися за відповідними залежностями. При відладці програми бажано коефіцієнти матриці жорсткості виводити на екран або у файл. Матриця жорсткості елемента повинна бути симетричною, а коефіцієнти її головної діагоналі повинні бути додатними.

10.11.4 Формування матриці жорсткості конструкції

Матриця жорсткості конструкції є таблицею коефіцієнтів, що складається з відповідних коефіцієнтів матриці жорсткості елементів. Наприклад, для елемента №14 (див. рис. 10.9), що включає вузли 6 – 9 – 12, коефіцієнти його матриці жорсткості мають бути занесені в рядки і стовпці 11, 12, 17, 18, 23 і 24. Оскільки матриця симетрична, то з метою економії оперативної пам'яті немає сенсу заносити симетричну частину.

При розбитті конструкції на елементи необхідно прагнути до того, щоб різниця номерів вузлів одного елемента була б якомога менше. У цьому випадку багато коефіцієнтів матриці жорсткості конструкції будуть нульовими, і замість квадратної матриці можна зберігати прямокутну матрицю. Її ширина рівна добутку максимальної різниці номерів вузлів, що входять в один елемент, плюс один, помноженому на ступінь вільності вузла.

Після того, як матриця жорсткості конструкції сформована, формується вектор навантажень – навантаження заносяться у відповідні рядки, а потім враховуються граничні умови. Щоб не порушувати ансамбль таблиці, рекомендується не викреслювати рядки і стовпці, відповідні номеру вузла, де є закріплення, а дорівнювати нулю всі коефіцієнти рядків і стовпців, окрім головного коефіцієнта, який дорівнюється одиниці.

10.11.5 Рішення системи рівнянь

У задачах МКЕ порядок системи рівнянь може бути дуже великим, близько 100 000 рівнянь і більше. Це залежить від даної конструкції і від її розбиття на елементи. Залежно від даного завдання можуть бути запропоновані різні методи рішення системи рівнянь. Дуже ефективним і достатньо точним є метод рішення системи рівнянь за Гаусом. Його ідея полягає в тому, що при прямому ході всі рівняння приводяться до вигляду, коли перший коефіцієнт кожного рівняння дорівнює одиниці, а наступні коефіцієнти виходять у вигляді різниці між коефіцієнтами рівнянь. В останньому рівнянні залишається один коефіцієнт, який дорівнює одиниці, звідки одержуємо величину останнього невідомого. Потім йдемо в

зворотному порядку, підставляємо знайдені рішення і знаходимо попереднє невідоме.

10.11.6 Визначення зусиль і напружень

Вирішивши систему рівнянь, одержуємо переміщення вузлів. Залежно від даної конструкції і виду кінцевих елементів за допомогою формул для векторів і зусиль визначаємо складові напружень і зусиль.

10.11.7 Виведення результатів розрахунків на екран і у файл

Особливість завдань МКЕ полягає в тому, що в результаті їх рішення одержуємо безліч чисел. У зв'язку з цим бажано виведення результатів виконувати у вигляді епюр або рисунків, на яких зафарбовувати елементи кольором, відповідним рівню напруження. Крім того, всі результати необхідно заносити у файл, який при необхідності можна роздрукувати.

11 РОЗРАХУНОК ВІСЕСИМЕТРИЧНИХ ЗАВДАЧ ТЕОРІЇ ПРУЖНОСТІ

11.1 Особливості вісесиметричної задачі теорії пружності

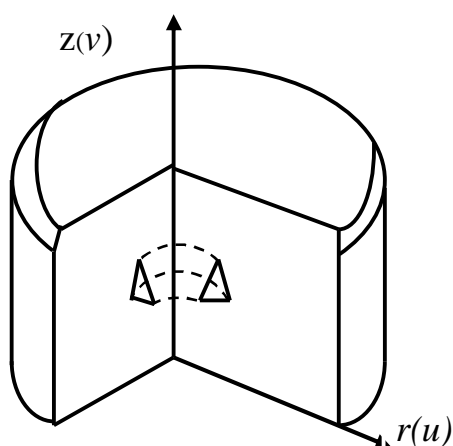


Рисунок 11.1 - Вісесиметричне тіло та торовидний кінцевий елемент

Дослідження розподілу напруження в тілах обертання (вісесиметричних тілах) при вісесиметричному вантаженні є великим практичним інтересом. Оскільки ці завдання теж тривимірні, з практичної точки зору вони аналогічні задачам про плоский напружений стан. Унаслідок симетрії деформований, а, отже, і напружений стан в будь-якому перерізі за віссю симетрії тіла

повністю визначається двома компонентами переміщення.

У вісесиметричному випадку будь-яке радіальне переміщення викликає деформацію в коловому напрямку, і так як напруження в цьому напрямку не дорівнює нулю, треба розглянути четверту компоненту деформації за відповідним напрямком. У цьому полягає відмінність вісесиметричного випадку від плоскої задачі теорії пружності.

Вектор деформації тепер має вигляд

$$\{\mathbf{e}\} = \begin{Bmatrix} \mathbf{e}_z \\ \mathbf{e}_r \\ \mathbf{e}_q \\ \mathbf{g}_{rz} \end{Bmatrix}, \quad (11.1)$$

де, як і для плоскої задачі $\mathbf{e}_z = \frac{\partial v}{\partial z}$, $\mathbf{e}_r = \frac{\partial u}{\partial r}$, $\mathbf{g}_{rz} = \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r}$.

Визначимо відносну деформацію \mathbf{e}_q в окружному напрямку (рис 11.2):

$$\mathbf{e}_q = \frac{B_1 A_1 - BA}{BA},$$

$$BA = r dq, \quad B_1 A_1 = (r + u) dq,$$

$$\mathbf{e}_q = \frac{(r + u) dq + r dq}{r dq} = \frac{u}{r}.$$

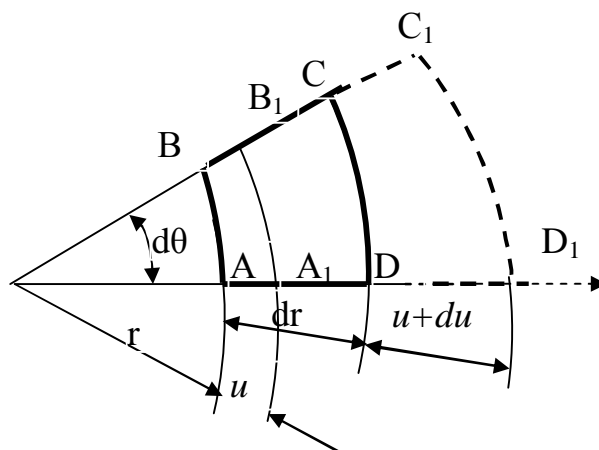
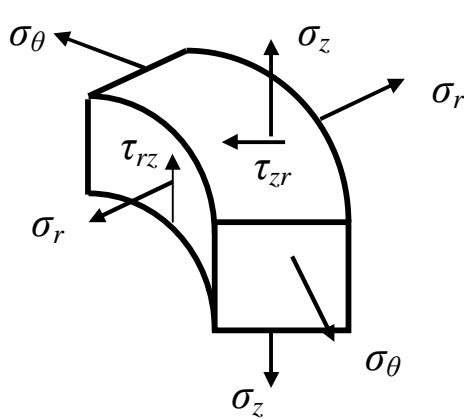


Рисунок 11.2 – Елемент осесиметричного тіла до і після деформації

Таким чином, вектор деформацій у цьому випадку має вигляд



$$\{e\} = \left\{ \begin{array}{l} \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial r} \\ u \\ r \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \end{array} \right\}. \quad (11.2)$$

Рисунок 11.3 – Напружений стан для вісесиметричної задачі

Вектор напружень (рис. 11.3) тепер набуде вигляду

$$\{s\} = \begin{Bmatrix} s_z \\ s_r \\ s_q \\ t_{rz} \end{Bmatrix} = |D| \begin{Bmatrix} e_z \\ e_r \\ e_q \\ g_{rz} \end{Bmatrix}. \quad (11.3)$$

11.2 Функція переміщення

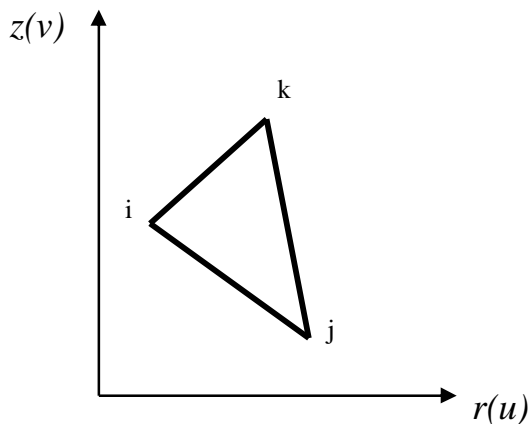


Рисунок 11.4 – Переріз кінцевого елемента

Залежно від виду даних кінцевих елементів необхідно задавати відповідну функцію.

Як і для плоскої задачі теорії пружності, найбільш простими є трикутні елементи (точніше, тори, поперечними перерізами яких є трикутники). Порівнюючи рис. 10.5 і 11.4, бачимо, що вони однакові, тільки замість осей x і y маємо осі r і z . Тому

скористуємося результатами для функції переміщень плоскої задачі теорії пружності. Таким чином, маємо

$$u = \left| \frac{a_i + b_i r + c_i z}{2\Delta}; \frac{a_j + b_j r + c_j z}{2\Delta}; \frac{a_k + b_k r + c_k z}{2\Delta} \right| \begin{Bmatrix} u_i \\ u_j \\ u_k \end{Bmatrix} \quad (11.4)$$

$$v = \left| \frac{a_i + b_i r + c_i z}{2\Delta}; \frac{a_j + b_j r + c_j z}{2\Delta}; \frac{a_k + b_k r + c_k z}{2\Delta} \right| \begin{Bmatrix} v_i \\ v_j \\ v_k \end{Bmatrix}. \quad (11.5)$$

11.3 Вектори деформацій і напружень

Вектор деформацій тепер має вигляд

$$\{e\} = \begin{Bmatrix} e_z \\ e_r \\ e_q \\ g_{rz} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial r} \\ \frac{u}{r} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \end{Bmatrix} = |B| \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix}, \quad (11.6)$$

$$|B| = \frac{1}{2D} \begin{vmatrix} 0 & c_i & 0 & c_j & 0 & c_k \\ b_i & 0 & b_j & 0 & b_k & 0 \\ \frac{a_i + b_i r + c_i z}{r} & 0 & \frac{a_j + b_j r + c_j z}{r} & 0 & \frac{a_k + b_k r + c_k z}{r} & 0 \\ c_i & b_i & c_j & b_j & c_k & b_k \end{vmatrix}. \quad (11.7)$$

Вектор напружень запишеться так:

$$\{S\} = \begin{Bmatrix} S_z \\ S_r \\ S_q \\ t_{zr} \end{Bmatrix} = \frac{E(1-m)}{(1-2m)(1+m)} \begin{vmatrix} 1 & \frac{m}{1-m} & \frac{m}{1-m} & 0 \\ \frac{m}{1-m} & 1 & \frac{m}{1-m} & 0 \\ \frac{m}{1-m} & \frac{m}{1-m} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2m}{2(1-m)} \end{vmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_k \\ v_k \end{Bmatrix} \quad (11.8)$$

11.4 Матриця жорсткості кінцевого елемента

Використовуючи загальний вираз матриці жорсткості для кінцевого елемента (1.6), одержимо:

$$|k|^{\mathbf{1}} = \int_V |B|^T |D| |B| dv = \int_F |B|^T |D| |B| 2p dFr. \quad (11.9)$$

До виразу (11.9) входить матриця $|B|$, коефіцієнти третього рядка якої (відповідний коловій деформації) залежать від r і z . Це не дозволяє виносити з-під інтеграла матрицю $|B|$. З метою спрощення ці коефіцієнти замінимо наближеними значеннями \bar{r} і \bar{z} , відповідними середнім значенням:

$$\bar{r} = \frac{r_i + r_j + r_k}{3}, \quad \bar{z} = \frac{z_i + z_j + z_k}{3}.$$

Тепер матриця $|B|$ замінюється на $|\bar{B}|$, що дозволить набути наближеного значення коефіцієнтів матриці жорсткості елемента:

$$|\bar{k}|^{\mathbf{1}} = |\bar{B}|^T |\bar{D}| |\bar{B}| 2p \bar{r} \bar{D}, \quad (11.10)$$

де

$$|B| = \frac{1}{2D} \begin{vmatrix} 0 & c_i & 0 & c_j & 0 & c_k \\ b_i & 0 & b_j & 0 & b_k & 0 \\ \frac{a_i + b_i \bar{r} + c_i \bar{z}}{\bar{r}} & 0 & \frac{a_j + b_j \bar{r} + c_j \bar{z}}{\bar{r}} & 0 & \frac{a_k + b_k \bar{r} + c_k \bar{z}}{\bar{r}} & 0 \\ c_i & b_i & c_j & b_j & c_k & b_k \end{vmatrix}. \quad (11.11)$$

Тут $a_i = r_j z_k - r_k z_j$, $b_i = z_j - z_k$, $c_i = r_k - r_j$, а решта коефіцієнтів можна одержати круговою підстановкою індексів $i \rightarrow j \rightarrow k \rightarrow i$.

При формуванні матриці жорсткості конструкції у вектор навантажень повинні записуватися значення сумарних навантажень, які рівні:

$$P_{2(n-1)+m} = 2pr_n q_{n,m}, \quad (11.12)$$

де n – номер вузла, в якому діє розподілене колом навантаження;

$m = 1$ - коли навантаження діє в напрямі r ;

$m = 2$ - коли навантаження діє в напрямі z ;

r_n – радіус вузла n .

11.5 Приклад

За допомогою МКЕ оцінити міцність втулки, показаної на рис. 11.5, за IV теорією міцності, якщо $[\sigma] = 100$ МПа, $E = 2 \cdot 10^5$ МПа, $\mu = 0,3$.

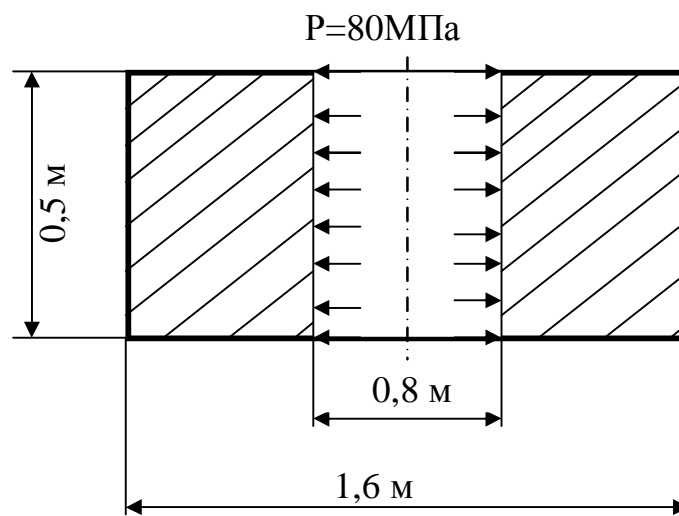


Рисунок 11.5- Розрахункова схема втулки

Розв'язання. Як кінцеві елементи беремо тори з поперечними перерізами у вигляді трикутників.

Переріз втулки розбиваємо на елементи (рис. 11.6), нумеруємо вузли і елементи.

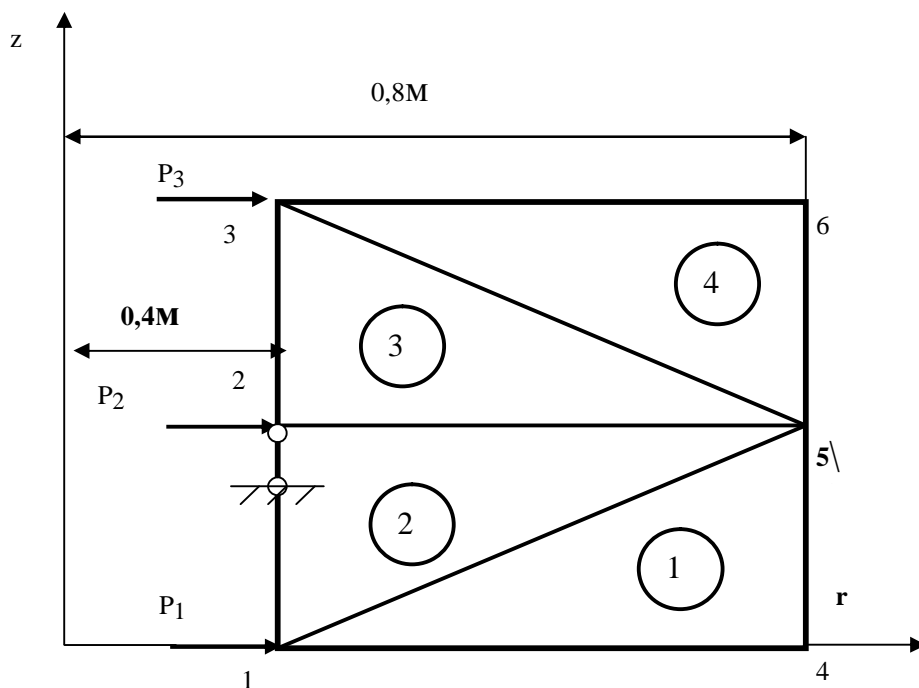


Рисунок 11.6 – Розбивання втулки на елементи

Матрицю жорсткості кінцевих елементів обчислюємо за формулою (11.10).

1-й елемент. Його зв'язки 1 – 4 – 5:

$$a_i = r_j \cdot z_k - r_k \cdot z_j = 0,8 \cdot 0,25 - 0,8 \cdot 0 = 0,2 ,$$

$$a_j = r_k \cdot z_i - r_i \cdot z_k = 0,8 \cdot 0 - 0,4 \cdot 0,25 = -0,1 ,$$

$$a_k = r_i \cdot z_j - r_j \cdot z_i = 0,4 \cdot 0 - 0,8 \cdot 0 = 0 ,$$

$$b_i = z_j - z_k = 0 - 0,25 = -0,25 ,$$

$$b_j = z_k - z_i = 0,25 - 0 = 0,25 ,$$

$$b_k = z_i - z_j = 0 - 0 = 0 ,$$

$$c_i = r_k - r_j = 0,8 - 0,8 = 0 ,$$

$$c_j = r_i - r_k = 0,4 - 0,8 = -0,4 ,$$

$$c_k = r_j - r_i = 0,8 - 0,4 = 0,4 ,$$

$$\frac{a_i + b_i \bar{r} + c_i \bar{z}}{\bar{r}} = \frac{0,2 - 0,25 \cdot 0,67 + 0 \cdot 0,08}{0,67} = 0,05 ,$$

$$\frac{a_j + b_j \bar{r} + c_j \bar{z}}{\bar{r}} = \frac{-0,1 - 0,25 \cdot 0,67 - 0,4 \cdot 0,08}{0,67} = 0,15 ,$$

$$\frac{a_k + b_k \bar{r} + c_k \bar{z}}{\bar{r}} = \frac{0 + 0 \cdot 0,67 + 0,4 \cdot 0,08}{0,67} = 0,048 ,$$

$$\bar{r} = \frac{r_i + r_j + r_k}{3} = \frac{0,4 + 0,8 + 0,8}{3} = 0,67 ,$$

$$\bar{z} = \frac{z_i + z_j + z_k}{3} = \frac{0 + 0 + 0,25}{3} = 0,08 ,$$

$$\Delta = \frac{1}{2} \cdot 0,4 \cdot 0,25 = 0,05 .$$

Матриця $|B|$ для першого елемента дорівнює

$$|B|_I = \begin{vmatrix} 0 & 0 & 0 & -0,4 & 0 & 0,4 \\ -0,25 & 0 & 0,25 & 0 & 0 & 0 \\ 0,05 & 0 & 0,15 & 0 & 0,048 & 0 \\ 0 & -0,25 & -0,4 & 0,25 & 0,4 & 0 \end{vmatrix} \frac{1}{2 \cdot 0,05} .$$

Матриця жорсткості першого елемента запишеться так:

$$|k|_I^{\mathbf{1}} = \frac{1}{2 \cdot 0,05} \begin{vmatrix} 0 & -0,25 & 0,05 & 0 \\ 0 & 0 & 0 & -0,25 \\ 0 & 0,25 & 0,15 & -0,4 \\ -0,4 & 0 & 0 & 0,25 \\ 0 & 0 & 0,048 & 0,4 \\ 0,4 & 0 & 0 & 0 \end{vmatrix} \cdot \frac{2 \cdot 10^5 (1 - 0,3)}{(1 - 2 \cdot 0,3)(1 + 0,3)} \times$$

$$\times \left| \begin{array}{cccccc} 1 & \frac{0,3}{0-0,3} & \frac{0,3}{0-0,3} & 0 & 0 & 0 \\ \frac{0,3}{0-0,3} & 1 & \frac{0,3}{0-0,3} & 0 & 0 & 0 \\ \frac{0,3}{0-0,3} & \frac{0,3}{0-0,3} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2 \cdot 0,3}{2(1-0,3)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2 \cdot 0,3}{2(1-0,3)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2 \cdot 0,3}{2(1-0,3)} \end{array} \right| \cdot \frac{1}{2 \cdot 0,05} \times$$

$$\times \left| \begin{array}{cccccc} 0 & 0 & 0 & -0,4 & 0 & 0,4 \\ -0,25 & 0 & 0,25 & 0 & 0 & 0 \\ 0,05 & 0 & 0,15 & 0 & 0,048 & 0 \\ 0 & -0,25 & -0,4 & 0,25 & 0,4 & 0 \end{array} \right| \cdot 2 \cdot 3,14 \cdot 0,05 .$$

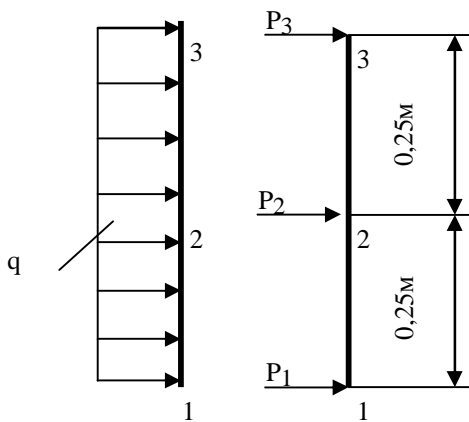


Рисунок 11.7 – Заміна інтенсивного навантаження вузловими силами

Аналогічно обчислюються матриці жорсткості для решти елементів. Після чого, як і для попередніх задач, формується матриця жорсткості конструкції.

Тут слід звернути увагу на граничні умови і на вектори навантажень.

Для того, щоб втулка не переміщалася як єдине ціле у напрямку осі z (із-за наближеності обчислень сума сил уздовж осі не буде дорівнювати нулю), необхідно її в цьому напрямку в одній з точок

закріпити, наприклад $v_2 = 0$.

При формуванні вектора навантажень слід врахувати, що тиск, розподілений уздовж лінії 1–2–3 (рис.11.7), необхідно прикласти у вузлах 1-2-3.

$$P_1 = P_3 = \frac{1}{2} p \cdot 0,25 \cdot 2p \cdot 0,4 = 80 \cdot 0,1p = 25,2MN,$$

$$P_2 = p \cdot 0,25 \cdot 2p \cdot 0,4 = 50,4MN.$$

Значить

$$R_1 = 25,2MN, \quad R_3 = 50,4MN,$$

$$R_5 = 25,2MN.$$

Розв'язавши систему рівнянь, знайдемо:

$$u_1 = 30,2 \cdot 10^{-5}, \quad v_1 = 0, \quad u_2 = 31,8 \cdot 10^{-5} \text{ м}, \quad v_2 = -1,9 \cdot 10^{-5} \text{ м},$$

$$u_3 = 30,2 \cdot 10^{-5}, \quad v_3 = -3,9 \cdot 10^{-5}, \quad u_4 = 22,1 \cdot 10^{-5} \text{ м}, \quad v_4 = 0,1 \cdot 10^{-5} \text{ м},$$

$$u_5 = 20,1 \cdot 10^{-5}, \quad v_5 = -1,9 \cdot 10^{-5}, \quad u_6 = 22,1 \cdot 10^{-5} \text{ м}, \quad v_6 = -0,1 \cdot 10^{-5} \text{ м}.$$

Визначимо вектор напружень за формулою

$$\{\mathbf{S}\} = \begin{Bmatrix} \mathbf{S}_z \\ \mathbf{S}_r \\ \mathbf{S}_q \\ \mathbf{t}_{rz} \end{Bmatrix} = |D| \overline{|B|} \{d\}^1.$$

Підставивши матриці $|D|$, $\overline{|B|}$ і вектор переміщень для першого елемента, знайдемо вектор напружень:

$$\{\mathbf{S}\}_I = \begin{Bmatrix} \mathbf{S}_z \\ \mathbf{S}_r \\ \mathbf{S}_q \\ \mathbf{t}_{rz} \end{Bmatrix} = \begin{Bmatrix} -3,7 \\ -22,3 \\ 64,4 \\ -5,8 \end{Bmatrix}.$$

Тепер визначаємо головні нормальні напруження за формулою

$$\begin{aligned} \mathbf{s}_{2,3} &= \frac{1}{2} \left[\mathbf{S}_r + \mathbf{S}_z \pm \sqrt{(\mathbf{S}_r - \mathbf{S}_z)^2 + 4\mathbf{t}_{rz}^2} \right] = \\ &= \frac{1}{2} \left[-22,3 - 3,7 \pm \sqrt{(-22,3 + 3,7)^2 + 4 \cdot 5,8^2} \right]. \end{aligned}$$

$$\mathbf{S}_2 = -1MPa, \quad \mathbf{S}_3 = -25MPa, \quad \mathbf{S}_1 = \mathbf{S}_q = 64,4MPa.$$

Еквівалентні напруження за IV теорії міцності дорівнюють

$$s^{IV} = \sqrt{\frac{1}{2}[(s_1 - s_2)^2 + (s_1 - s_3)^2 + (s_3 - s_2)^2]} =$$

$$\sqrt{\frac{1}{2}[(64,4 + 1)^2 + (-1 + 25)^2 + (64,4 + 25)^2]} = 72 < [s] = 100 \text{ МПа.}$$

Умова міцності виконана.

11.6 Тестове завдання

Як контрольний приклад розглянута задача про вантаження товстостінної труби внутрішнім тиском (задача Ламе) (рис.11.8). На рис.11.9 зображена схема розбиття цієї конструкції на торовидні кінцеві елементи. У таблиці 11.1 наведені результати розрахунків за методом кінцевих елементів у зіставленні з результатами розрахунків за формулою Ламе.

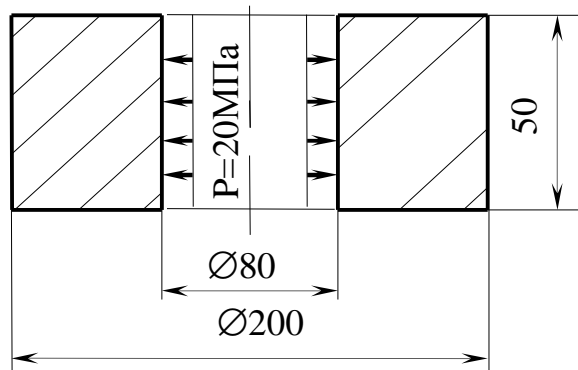


Рисунок 11.8 - Розрахункова схема тестової задачі

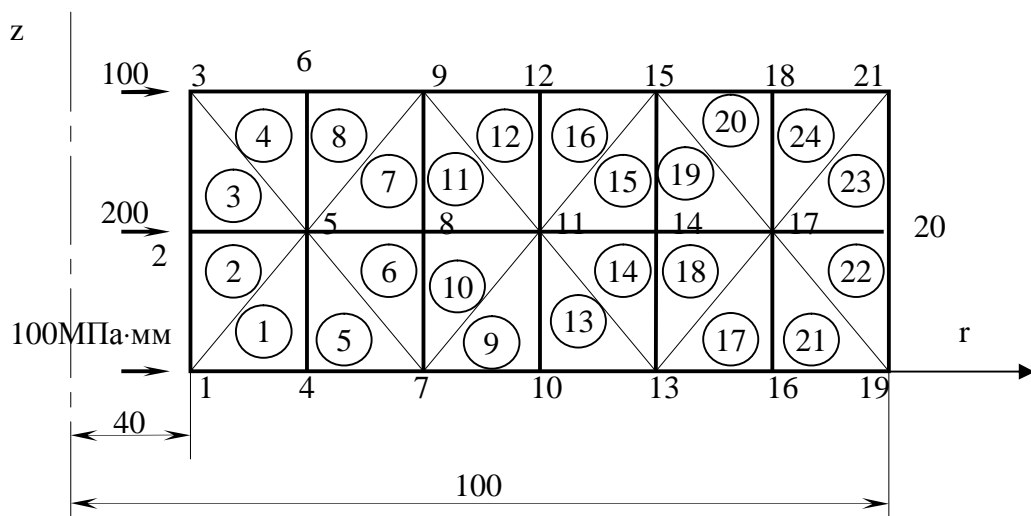


Рисунок 11.9 - Розбивка на кінцеві елементи тестової втулки

Таблиця 11.1 – Зіставлення результатів рішення тестової задачі за методом кінцевих елементів і за формулою Ламе

Величина	r, мм	За МКЕ	За Ламе	Погрішність
U, мм	40	0,00385	0,00395	2,6%
V, мм	100	0,00675	0,00672	0,5%
S_q МПа	43,3	24,6	24,0	2,5%
S_q МПа	96,7	7,86	7,86	0%
S_r МПа	43,3	-15,97	-16,1	0,9%
S_r МПа	96,7	-0,33	-0,32	3,0%

Текст програми наведено у додатку В.

11.7 Блок-схеми програми для ПЕОМ вісесиметричної задачі теорії пружності

Блок-схеми для ПОВМ рішення вісесиметричної задачі теорії пружності подані на рис. 11.10...11.13.

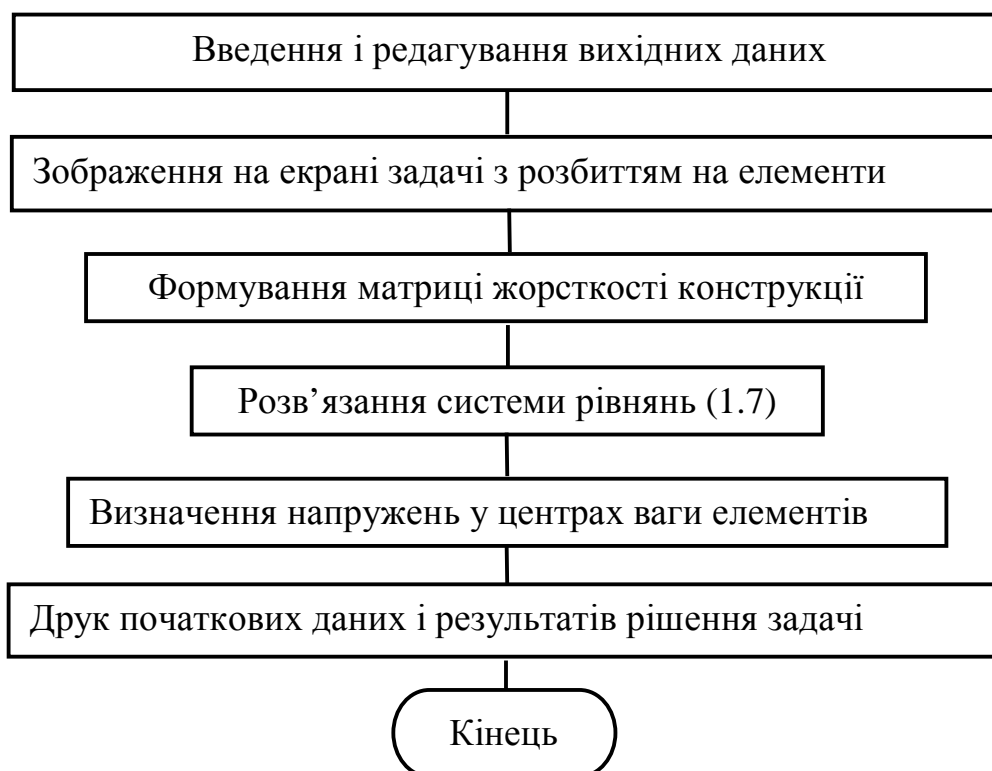


Рисунок 11.10 - Блок-схема основної програми

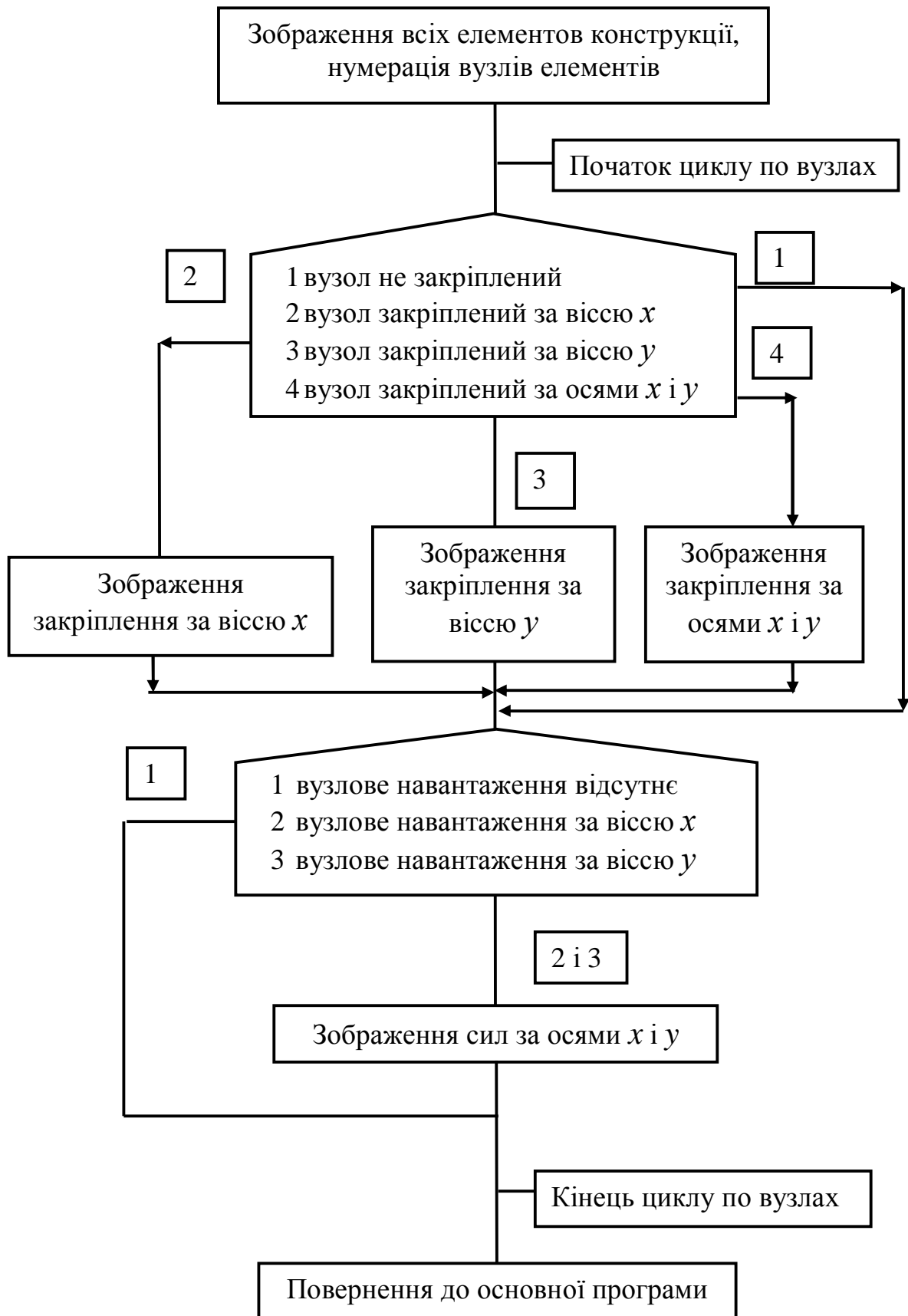


Рисунок 11.11 - Блок- схема зображення задачі на екрані

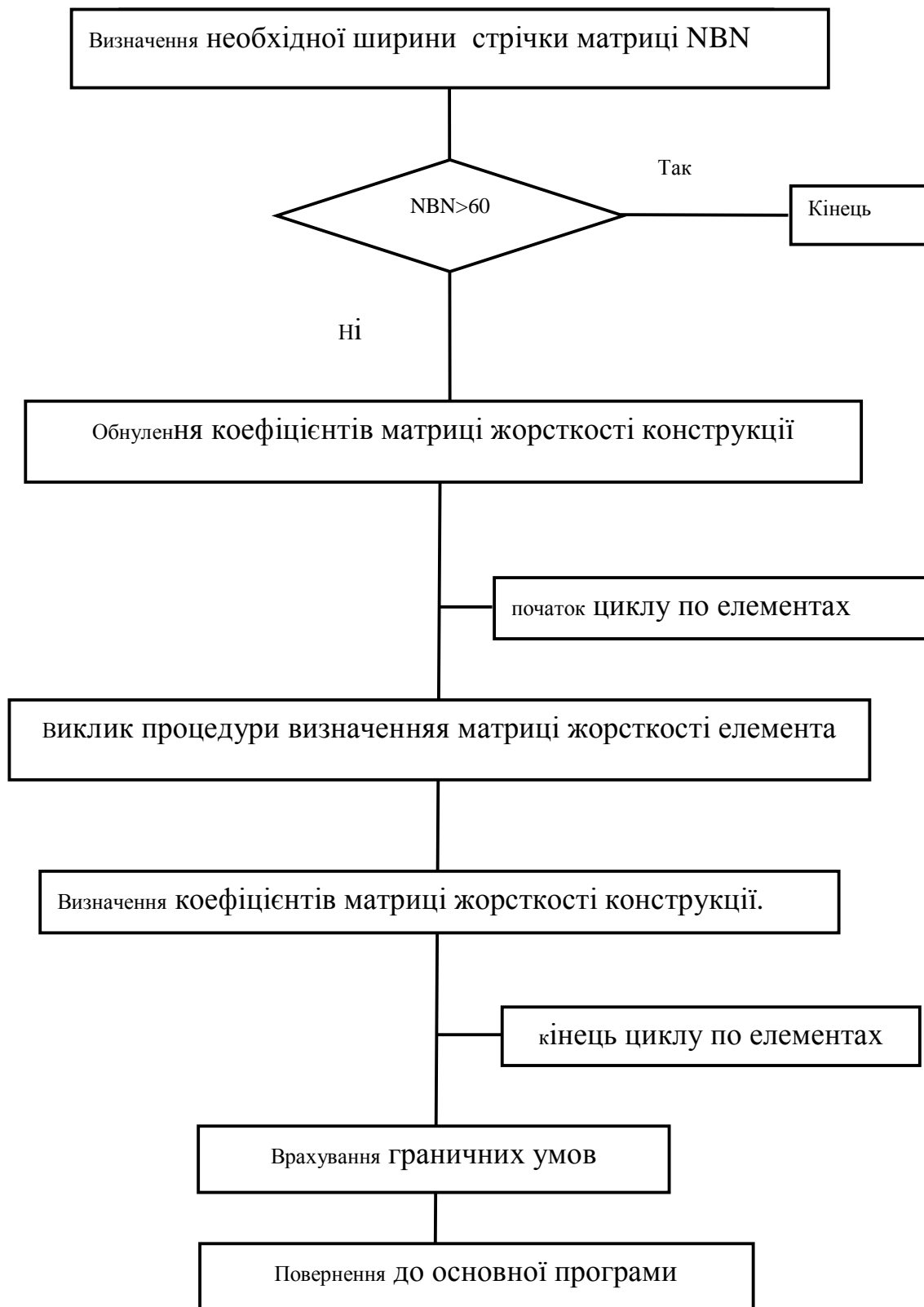


Рисунок 11.12 - Блок-схема процедури матриці жорсткості конструкції

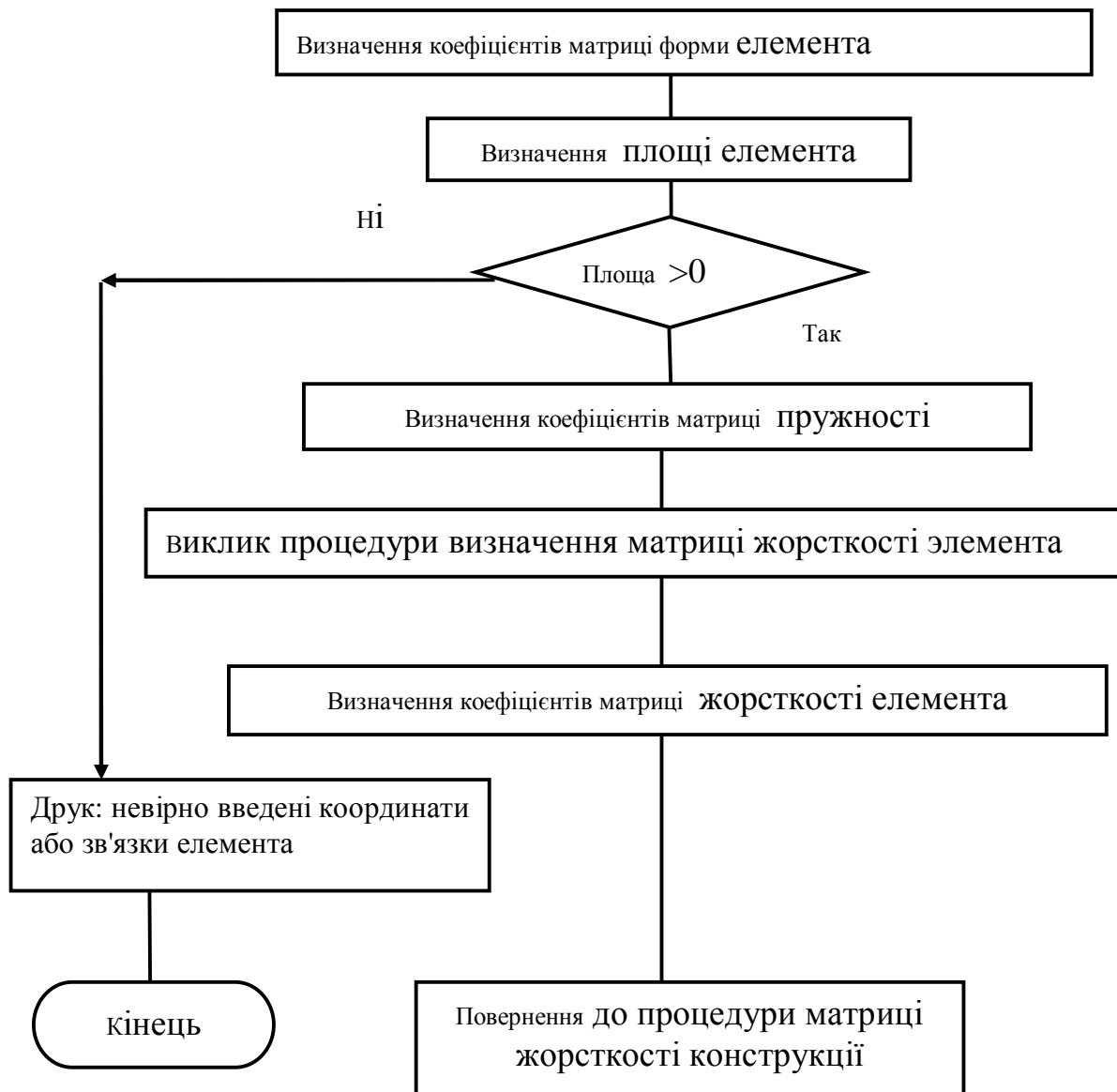


Рисунок 11.13 - Блок-схема матриці жорсткості елемента

11.8 Розрахунок пресових посадок

Дослідженню розподілу напружень у пресових посадках присвячений цілий ряд праць (наприклад, [15], С. 478), але скрізь, як правило, розглядається класичний випадок (рис. 11.14, а) – втулка постійної товщини насаджена на гладкий вал. У більшості випадків втулка має складну конфігурацію (рис. 11.14, б). У цьому випадку напружений стан пресових посадок при складних контурах деталей з'єднання доцільно досліджувати за допомогою МКЕ. Особливість задачі полягає в тому, що

необхідно враховувати початкові деформації однієї з деталей з'єднання (зовнішньої – додатні, а внутрішньої – від'ємні).

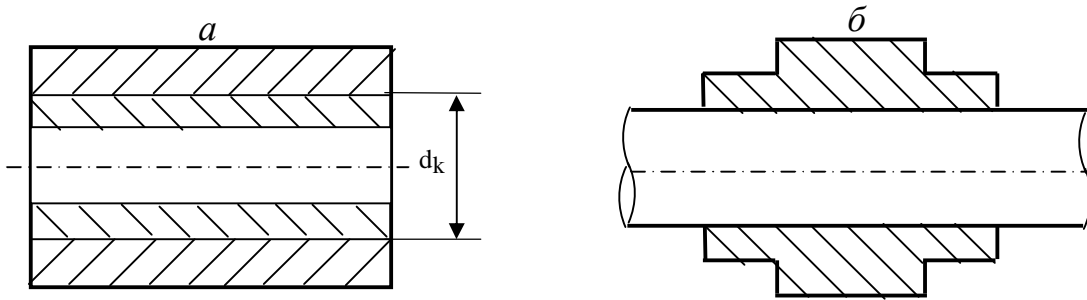


Рисунок 11.14 - Схеми пресових з'єднань

Тоді, окрім деформацій викликаних зовнішнім навантаженням, виникають деформації від посадки. Вони характеризуються додатковим вектором

$$\{\mathbf{e}_0\} = \begin{Bmatrix} 0 \\ \mathbf{e}_{r0} \\ \mathbf{e}_{q0} \\ 0 \end{Bmatrix}, \quad (11.13)$$

де \mathbf{e}_{r0} і \mathbf{e}_{q0} – відносні радіальна і колова деформації, які беруться постійними у межах елемента і дорівнюють $\frac{d}{d_k}$ (d – натяг, залежний від виду пресової посадки; d_k – діаметр поверхні контакту деталей, що сполучаються).

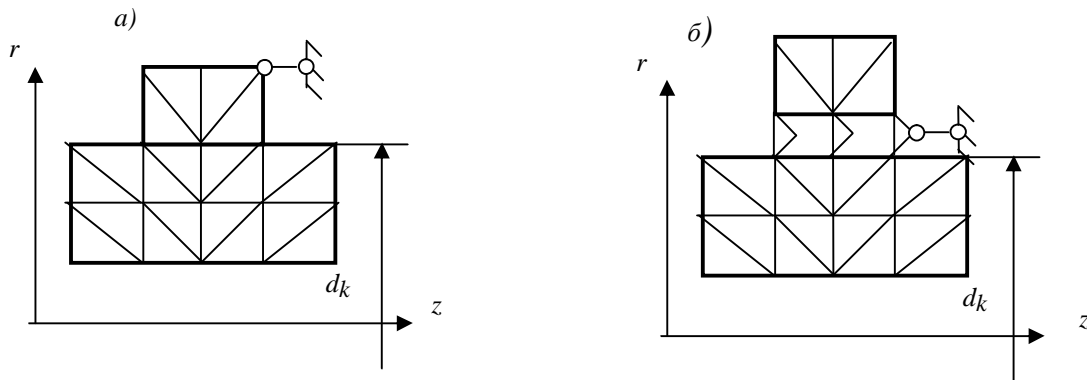
Тепер вектор напружень запишеться так

$$\{\bar{\mathbf{s}}\}^1 = |\bar{D}| |\bar{B}| \{\mathbf{d}\}^1 - |\bar{D}| \{\mathbf{e}_0\}. \quad (11.14)$$

А матриця жорсткості елемента (11.10) набуде вигляду

$$|\bar{k}|^1 = |\bar{B}|^T |\bar{D}| |\bar{B}| 2p \bar{r} \Delta - 2p |\bar{B}|^T |\bar{D}| \{\mathbf{e}_0\} \bar{r} \Delta. \quad (11.15)$$

Розглядаємо пресове з'єднання як одне пружне тіло, частина елементів якого випробовують початкову деформацію. При такій розрахунковій схемі діють сили зчеплення, від яких виникають осьові нормальні напруження σ_z . Щоб виключити ці сили, необхідно на стику деталей вводити дуже малі елементи, які мають велику жорсткість (див. рис. 11.15,б).



а – наявність сил зчеплення між деталями;

б – відсутність сил зчеплення

Рисунок 11.15 - Розбивання пресового з'єднання на елементи

12 РОЗРАХУНОК ВІСЕСИМЕТРИЧНИХ ТІЛ ПРИ ДОВІЛЬНОМУ НАВАНТАЖЕННІ

У розрахунковій практиці часто доводиться мати справу з вузлами, що є з'єднаннями з натягом тіл обертання, які сприймають у загальному випадку навантаження крутні та згинальні моменти. До них відносяться: з'єднання піввісь – ступиця барабана, цапфа – корпус стінки рудорозмельного млина, труба жорсткості – муфта шахтної підйомної машини і т.п.

В існуючій розрахунковій практиці немає рекомендації щодо розрахунку подібних з'єднань навіть у разі простих їх конфігурацій, не кажучи про існуючі насправді складні контури елементів таких з'єднань. У даному розділі наводяться методики розрахунку з'єднань тіл обертання при симетричному (від вивертаючого моменту) і кососиметричному (від крутного моменту) навантаженнях (рис.12.1б).

12.1 Симетричне навантаження

Перехід від двомірних до тримірних завдань збільшує необхідний для розрахунку час, . Ілюстрацією такого положення є розрахунок ступиці (див. рис.12.6), що є товстостінною оболонкою. Оскільки згинальне навантаження, яке сприймає ступиця, не є вісесиметричним, використовуємо напіваналітичний метод рішення. У цьому випадку задача зведена до двовимірної (без координати, уздовж якої властивості не змінюються), навантаження в осьовому напрямку задається точково, а в коловому розкладається у ряд Фур'є. [1]

Вектор зовнішнього навантаження $\{P_i\}$ визначається через його проекції P_q . Вираз (1.7) містить $2n$ канонічних рівнянь (n – кількість вузлів).

Воно справедливе для кожної i -гармоніки. При симетричному і антисиметричному навантаженнях величини зусиль для кожної гармоніки подані на рис. 12.1.

$$\left\{ P_i^l \right\} = pR \begin{Bmatrix} q_t^n \\ q_z^n \\ q_q^n \end{Bmatrix} \text{ при } n=1, 2, \dots \quad \left\{ P_i^l \right\} = 2pR \begin{Bmatrix} q_t^0 \\ q_z^0 \\ 0 \end{Bmatrix} \text{ при } n=0; \quad (12.1)$$

де q_t^0, q_z^n, q_q^n – амплітудні значення радіального, осьового і колового навантажень, що є коефіцієнтами ряду Фур'є n -ї гармоніки.

При згинанні навантаження симетричне щодо осі, функція $f(\theta)$ – парна і може бути розкладена в ряд Фур'є.

$$f(q) = \frac{1}{2} d_0 + \sum_{l=1}^{\infty} d_l \cos nq,$$

$$d_l = \frac{1}{p} \int_{-p}^p f(q) \cos nq dq \quad (12.2)$$

$$n = 0, 1, 2, \dots$$

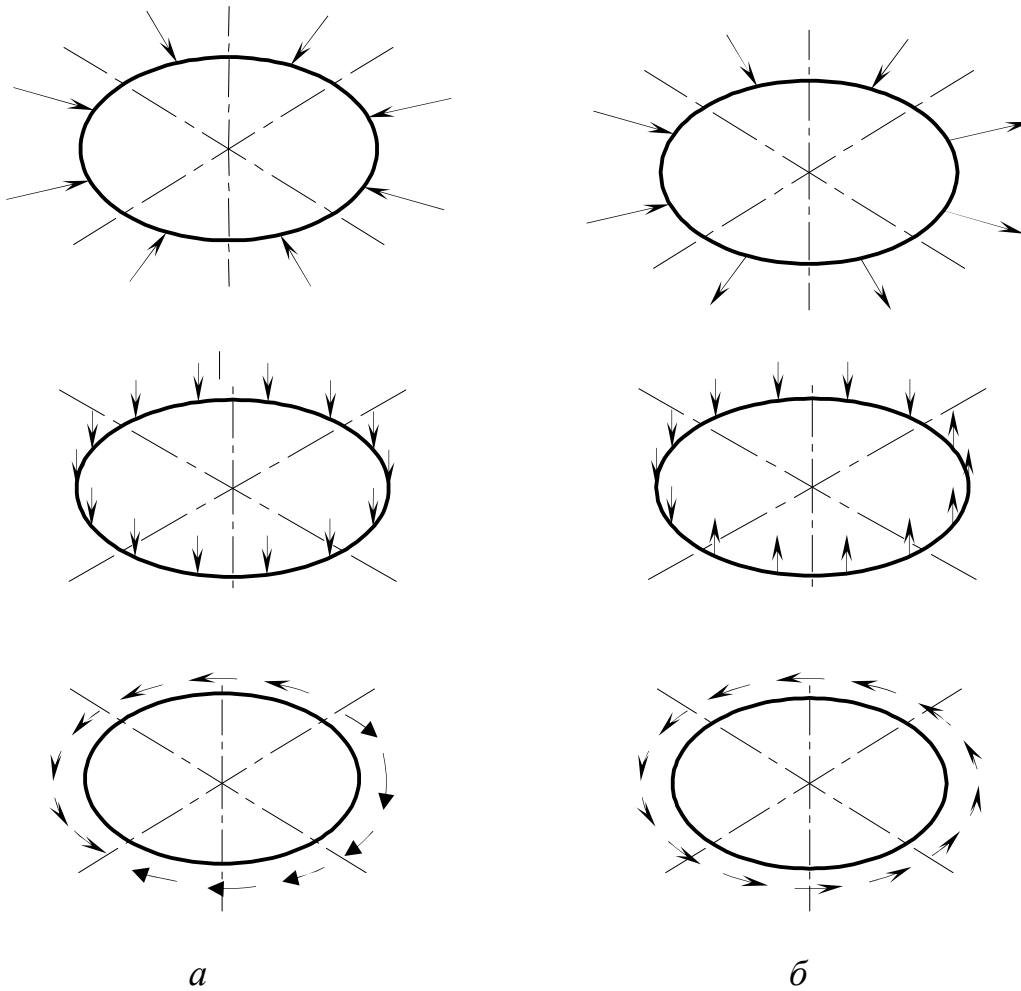


Рисунок 12.1- Симетричні (а) і антисиметричні (б) компоненти переміщень і навантажень у вісесиметричному тілі

Для навантаження, рівномірно розподіленого у межах кута 2β :

$$a_0 = \frac{2p}{p} \int_0^b f(q) dq = \frac{2}{p} \int_0^b f(q) dq + \frac{2p}{p} \int_b^b f(q) dq = \frac{2}{p} \int_0^b q dq = \frac{2qb}{p};$$

$$a_l = \frac{2p}{p} \int_0^b f(q) \cos nq dq = \frac{2}{p} \int_0^b f(q) \cos nq dq + \frac{2p}{p} \int_b^b f(q) \cos nq dq +$$

$$+ \frac{2}{p} \int_0^b f(q) q \cos nq dq = \frac{2q}{p} \sin nb.$$

При цьому

$$f(q) = \frac{qb}{p} + \frac{2q}{p} \sum_{l=1}^{\infty} \frac{1}{n} \sin nb \cos nq.$$

Для радіального рівномірно розподіленого в межах кута 2β (β задається умовами задачі) амплітудне значення навантаження q_t^0 запишеться так:

$$q_t^0 = \frac{2q_t b}{p}, \quad q_t^n = \frac{2q_t}{pn} \sin nb,$$

де

$$q_t = \frac{P_t}{2R \sin b}.$$

Вираз q_z^n виходить з q_t^n шляхом заміни індексів r на z , а q_q^n має вигляд

$$q_q^n = \frac{2q_q}{p} \cdot \frac{1 - \cos nb}{n},$$

$$q_r = \frac{P_r}{2R \sin b}.$$

Компоненти переміщень описуються двомірними функціями форми, відповідними трикутному елементі при вісесиметричному напруженому стані. Вони підкоряються залежностям, відповідним законам кожної гармоніки, і для симетричного навантаження рівні:

$$u^n = [N'_1, N'_2, N'_3] \cos nq \{u^n\}^e, \quad v^n = [N'_1, N'_2, N'_3] \cos nq \{v^n\}^e,$$

$$w^n = [N'_1, N'_2, N'_3] \sin nq \{w^n\}^e. \quad (12.3)$$

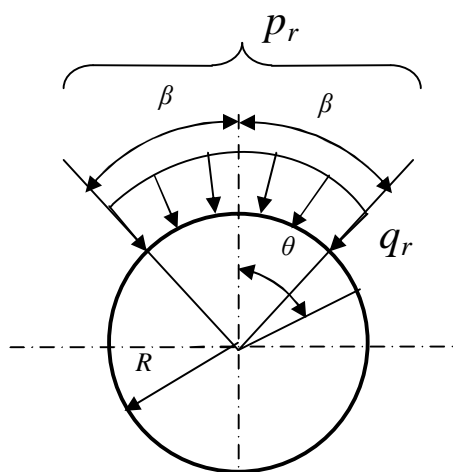


Рисунок 12.2 – Схема навантаження вала в коловому напрямку

Вираз деформацій через переміщення в циліндрових координатах для тривимірного випадку має вигляд

$$\{e\} = \begin{Bmatrix} e_t \\ e_z \\ e_q \\ g_{tz} \\ g_{tq} \\ g_{zq} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial v}{\partial z} \\ \frac{u}{r} + \frac{1}{r} \cdot \frac{\partial w}{\partial q} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \\ \frac{1}{r} \cdot \frac{\partial u}{\partial q} + \frac{\partial w}{\partial r} \cdot \frac{w}{r} \\ \frac{1}{r} \cdot \frac{\partial v}{\partial q} + \frac{\partial w}{\partial z} \end{Bmatrix} = [B^l]\{d\}. \quad (12.4)$$

Від n -ої гармоніки матриця

$$[B^n] = [B_i^n B_j^n B_m^n], \text{ де}$$

$$[B_i^n] = \frac{1}{2\Delta} \begin{Bmatrix} z_{jm} \cos nq & 0 & 0 \\ 0 & r_{mj} \cos nq & 0 \\ \frac{1}{r}(r_j z_m - r_m z_j) + \\ + z_{jm} r + r_{mj} z) \cos nq & 0 & \frac{1}{r}(r_j z_m - r_m z_j) + \\ r_{mj} \cos nq & z_{jm} \cos nq & + z_{jm} r + r_{mj} z) \cos nq \\ -\frac{1}{r}(r_j z_m - r_m z_j) + \\ + z_{jm} r + r_{mj} z) \sin nq & 0 & -\frac{1}{r}(r_j z_m - r_m z_j) + \\ 0 & -\frac{1}{r}(r_j z_m - r_m z_j) + \\ + z_{jm} r + r_{mj} z) \sin nq & r_{mj} \sin nq \end{Bmatrix} \quad (12.5)$$

Тут $r_i, z_i, r_j, z_j, r_m, z_m$ – координати вузлів елемента, а

$$r_{mj} = r_m - r_j, \quad z_{jm} = z_j - z_m.$$

Вираз $[B_j^n], [B_m^n]$ одержимо круговою підстановкою індексів в матриці. Якщо визначити матрицю $[B^l]$ для центра ваги елемента з координатами

$$\bar{r} = \frac{r_i + r_j + r_m}{3}, \quad \bar{z} = \frac{z_i + z_j + z_m}{3},$$

то наближений вираз матриці жорсткості елемента від l -гармоніки набуде вигляду:

$$[K] = 2p[\bar{B}^n]^T [D][\bar{B}^n] \bar{r} \Delta. \quad (12.6)$$

Вирішуючи рівняння (1.7), знаходимо переміщення u, v, w вузлових точок елементів, а через них визначаємо деформації $\{e^n\}$ і напруження, що цікавить нас, в центрі ваги елементів від кожної гармоніки окремо:

$$\{s^n\} = [D]\{e^n\} = [D][B^n]\{d^n\}. \quad (12.7)$$

Підсумовуючи їх, одержимо повні значення напружень.

12.2 Кососиметричне навантаження

Описана методика придатна також і при дії антисиметричного навантаження. Для цього у виразі (12.17) симетричні складові q_t^n, q_z^n необхідно поміняти місцями з обернено симетричними, а у виразі (12.2) замінити косинус на синус і навпаки, після чого матриця $[B_i^n]$ набуде вигляду (12.8).

Вирази матриці жорсткості $[K]$ і вектора напруження $\{s\}$ одержимо, використовуючи (12.6) і (12.7).

Методика розрахунку тіл обертання при довільному навантаженні, викладена в попередньому розділі, була реалізована на ПЕОМ.

У результаті рішення задачі з використанням розробленої програми одержимо значення вузлових переміщень, радіальних, осьових, колових s_q , дотичних (s_{tz}, s_{tq}, s_{zq}) і еквівалентних за третьою теорією міцності напружень, а також напружень в коловому напрямку через 15 градусів.

$$[B_i^n] = \frac{1}{2\Delta} \cdot \left\{ \begin{array}{ccc} z_j m \sin nq & 0 & 0 \\ 0 & r_{mj} \sin nq & 0 \\ \frac{1}{r}(r_j z_m - r_m z_j + & 0 & -\frac{l}{r}(r_j z_m - r_m z_j + \\ + z_j m^{r+r_{mj}z}) \sin nq & & + z_j m^{r+r_{mj}z}) \sin nq \\ r_{mj} \sin nq & z_j m \sin nq & 0 \\ \frac{l}{r}(r_j z_m - r_m z_j + & 0 & -\frac{1}{r}(r_j z_m - r_m z_j + \\ + z_j m^{r+r_{mj}z}) \cos nq & & + r_{mj} z) \sin nq \\ 0 & \frac{l}{r}(r_j z_m - r_m z_j + & r_{mj} \cos nq \\ & + z_j m^{r+r_{mj}z}) \cos nq & \end{array} \right\} \quad (12.8)$$

Програма дозволяє досліджувати напружено-деформований стан тіл обертання самого довільного конструктивного оформлення при будь-якому навантаженні, яке має бути розкладене на симетричну і кососиметричну складові. Сумарні переміщення і напруження визначаються шляхом їх підсумовування від кожної гармоніки.

Викладена методика розрахунку тіл обертання при довільному навантаженні була апробована на відомих рішеннях:

1 Розраховано вал $D=50$ мм, жорстко затиснений з одного боку і навантажений на іншому кінці крутним моментом $M_{kp} = 0.2$ кН·м (див. рис.12.3).

2 Консольна балка $D=50$ мм, навантажена зосередженою силою $P=2$ кН (див. рис.12.4).

3 Циліндрова оболонка $D=830$ мм, затиснена по торцях і навантажена зовнішнім тиском $q = 0.8$ кН/м² (див. рис. 12.5).

Результати теоретичних обчислень і обчислень по МКЕ наведені в таблиці 12.1.

Таблиця 12.1

Об'єкт	Напруження, S_t , МПа		Розбіжність, %	Радіальні переміщення, мм		Розбіжність, %
	Теоретич.	МКЕ		Теоретич.	МКЕ	
Вал	7,46	7,29	2,3	0,03	0,0285	5,0
Балка	37,2	36,0	3,2	0,26	0,255	1,9
Оболонка	13,1	13,56	3,5	4,87	4,97	2,8

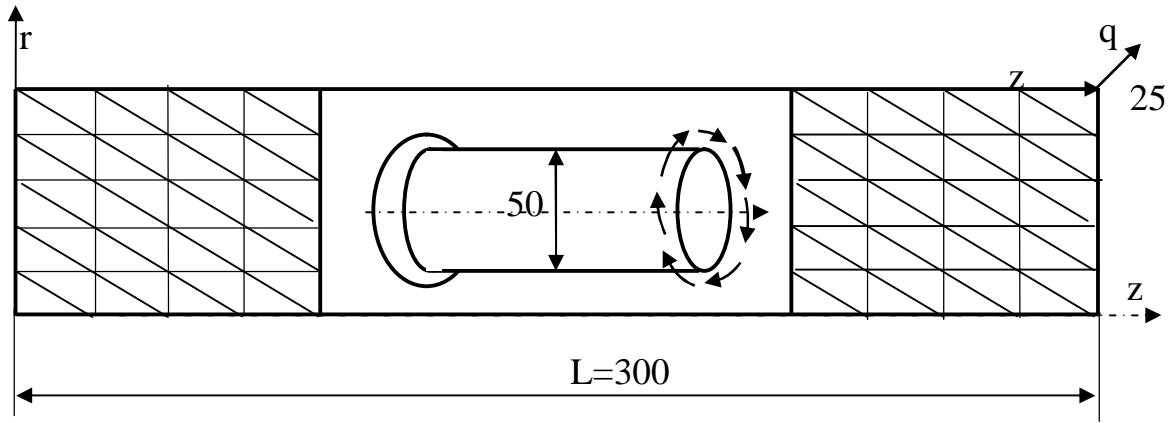


Рисунок 12.3 - Вал, навантажений на кінці крутним моментом $M_{кр}=80 \text{ кН} \cdot \text{см}$

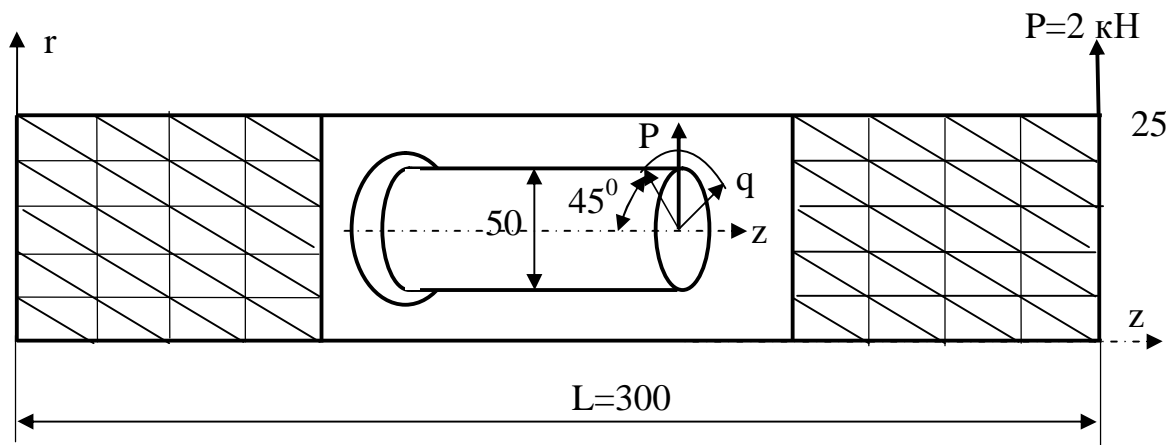


Рисунок 12.4 – Балка, навантажена на кінці силою $P=2 \text{ кН}$.

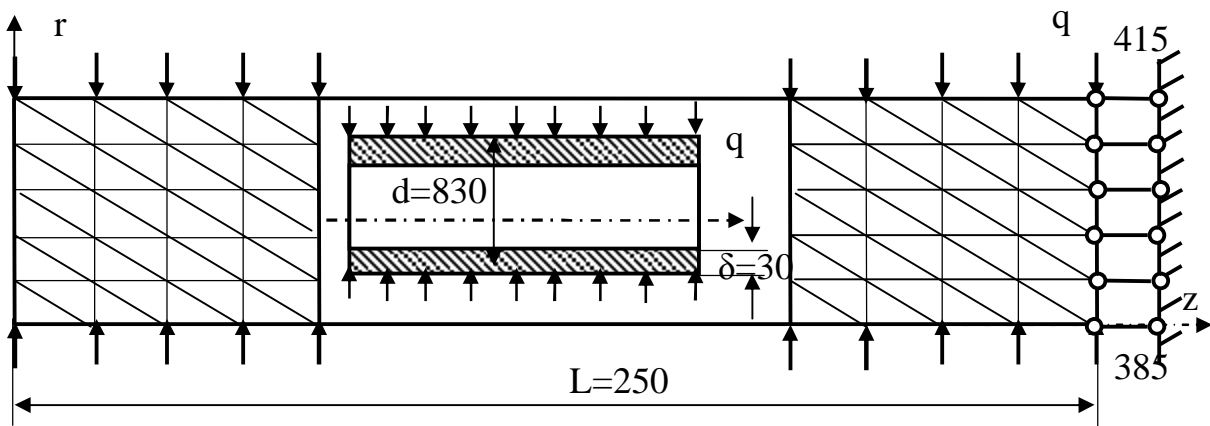
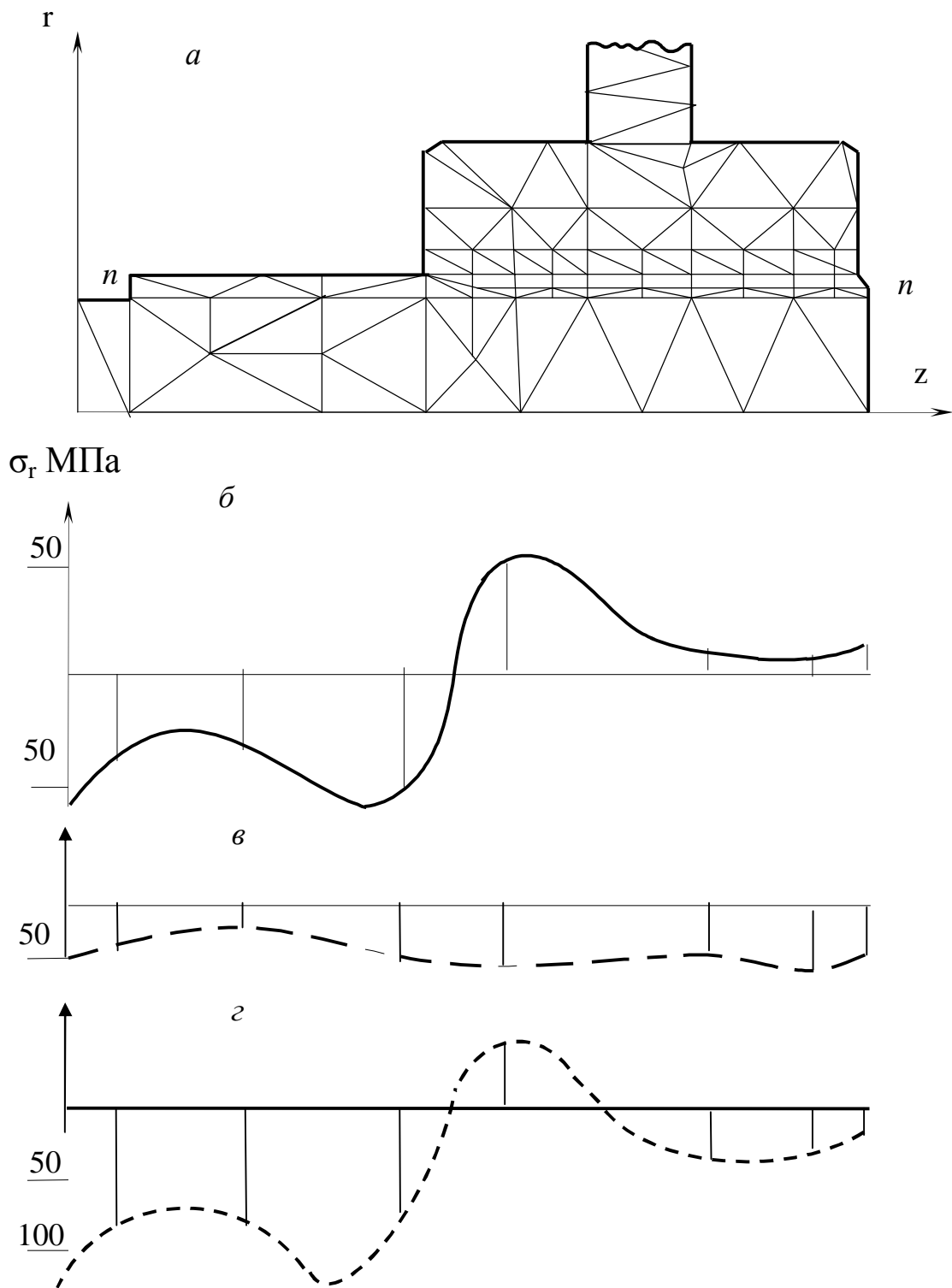


Рисунок 12.5 - Тонкостінна оболонка, навантажена зовнішнім тиском



а - розбивання з'єднання на елементи; б - епюра радіальних напружень від згинання; в- епюра радіальних напружень від посадки; г- сумарна епюра радіальних напружень
 Рисунок 12.6 – З'єднання «вал - ступиця» кранового барабана

У даний час багато з'єднань підйомно-транспортних машин, що працюють на згинання і кручення, здійснюються за рахунок натягу, отже, в розрахунку таких з'єднань треба враховувати напруження від обох навантажень. Поєднуючи обидві методики розрахунку, викладені в 12.1 і 12.2, можна досліджувати напружено-деформований стан тіл обертання при довільному навантаженні.

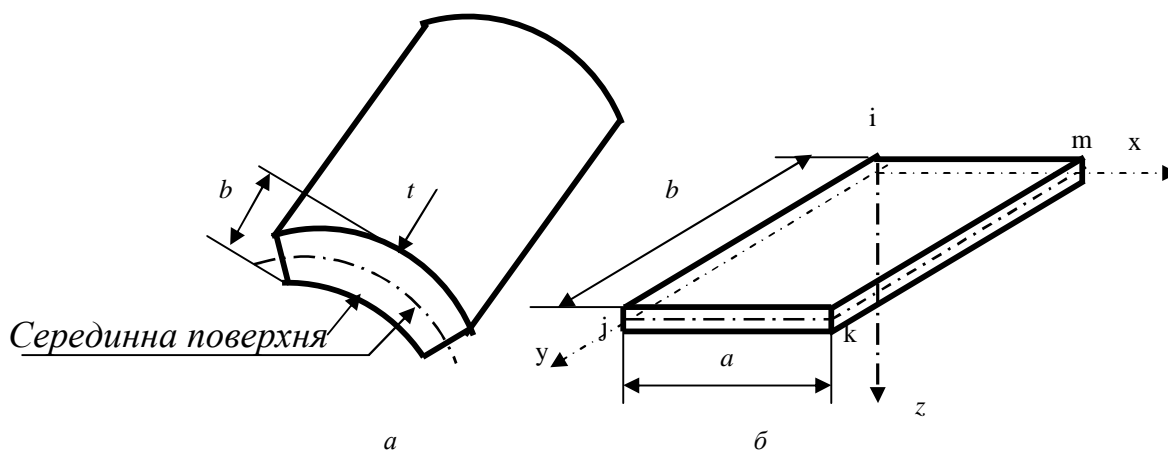
На рис. 12.6 наведені епюри напружень S_t від згинання, посадки і сумарні, такі, що мають максимальне значення по посадочній поверхні т-п з'єднання «вал – ступиця» барабана крана.

13 РОЗРАХУНОК ПЛАСТИН

13.1 Поняття про пластини. Гіпотеза Кірхгофа

Оболонкою називають тіло, у якого один розмір – товщина – значно

менше інших (рис.13.1, а): $\frac{t}{b} \leq \frac{1}{5}$.



a - оболонка; б - пластина

Рисунок 13.1 – Тонкостінні конструкції

Поверхню, що проходить через середину товщини, називають серединною поверхнею. Якщо серединна поверхня переходить в площину, то оболонка переходить в пластину (рис.13.1, б).

Пластиною називають тіло, обмежене двома площинами, відстань між якими h (товщина пластини) мала в порівнянні з розмірами цих площин.

Лінія перетину серединної площини з обмежуючими пластину бічними поверхнями утворює контур пластини.

Пластини, у яких товщина не перевищує однієї п'ятої меншого розміру, називають тонкими.

У всіх задачах, які були розглянуті вище, основні залежності між напруженнями і деформаціями наведені в точній формі, хоча в багатьох задачах рішення знаходилося приблизно. У класичній теорії пластин, щоб спростити задачу і звести її до двовимірної, із самого початку вводяться деякі гіпотези, які називаються гіпотезами Кирхгофа. Вони полягають в наступному:

- 1 Точки пластини, розташовані до її згинання на прямій, нормальній до серединної площини, при згинанні залишаються на цій прямій. Ця пряма повертається, залишаючись нормальною до зігнутої серединної площини. Дана гіпотеза аналогічна гіпотезі плоских перерізів в теорії згинання балок, і її часто називають гіпотезою прямих нормалей.
- 2 На основі гіпотези прямих нормалей встановлений лінійний закон зміни за товщиною нормальних напружень згинання і дотичних напружень кручення і одержані формули для кутів повороту і прогинів.
- 3 Шари пластини, паралельні серединній площині, не тиснуть один на одного. З цієї гіпотези виходить, що в будь-якому перерізі пластини, паралельному серединній площині, нормальні напруження дорівнюють нулю, і кожен елемент шару пластини, паралельного серединній площині, в загальному випадку згинання знаходиться в плоскому напруженому стані.
- 4 Крім того, приймають, що матеріал пластини однорідний, ізотропний, а виникаючі напруження нижче межі пропорційності, і тому напруження і деформації зв'язані між собою законом Гука.

- 5 Інтенсивності згинальних моментів уздовж осей x і y в пластині прийнято позначати через M_x і M_y (рис. 13.2), а інтенсивність крутних моментів – через M_{xy} .

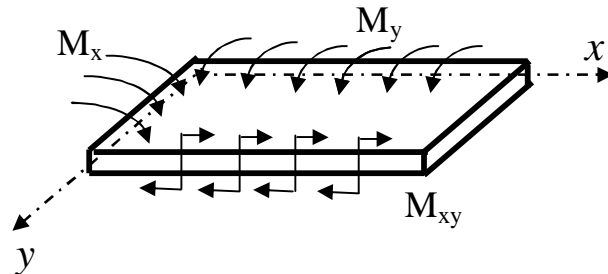


Рисунок 13.2 - Розподілені моменти M_x , M_y , M_{xy}

13.2 Зв'язок між переміщеннями, деформаціями і зусиллями

При згинанні пластини всі точки переміщуються перпендикулярно її площині. Ці переміщення (рис.13.3) називають прогинами. Усі елементи деформації виражають через прогини.

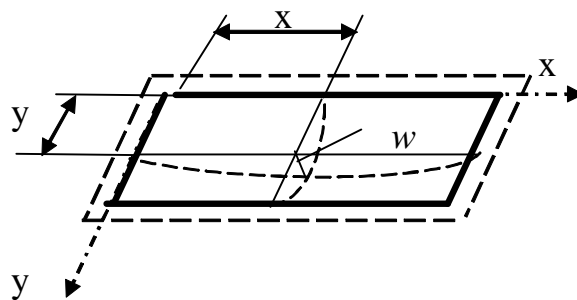


Рисунок 13.3 - Схема згинання пластини

Нормалі до серединної поверхні при згинанні повертаються відносно осей x і y на кути φ_x і φ_y відповідно:

$$q_x = \frac{\partial w}{\partial y}, q_y = \frac{\partial w}{\partial x}. \quad (13.1)$$

Аналогічно з рисунка 13.4 одержуємо

$$v = q_x z = z \frac{\partial w}{\partial y}.$$

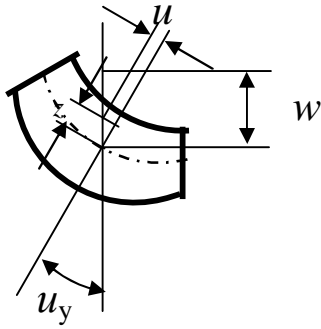


Рисунок 13.4 – Зв'язок переміщень u і w

Оскільки при згинанні маємо плоский напружений стан, то справедливі залежності Коші:

$$\{e\} = \begin{Bmatrix} e_x \\ e_y \\ g_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix},$$

куди підставляємо переміщення u і v :

$$\{e\} = \begin{Bmatrix} z \frac{\partial^2 w}{\partial x^2} \\ z \frac{\partial^2 w}{\partial y^2} \\ 2z \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix}. \quad (13.2)$$

Вектор напружень

$$\begin{Bmatrix} s_x \\ s_y \\ t_{xy} \end{Bmatrix} = \frac{E}{1-m^2} \begin{vmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1-m}{2} \end{vmatrix} \begin{Bmatrix} e_x \\ e_y \\ g_{xy} \end{Bmatrix} = \frac{Ez}{1-m^2} \begin{vmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1-m}{2} \end{vmatrix} \begin{Bmatrix} z \frac{\partial^2 w}{\partial x^2} \\ z \frac{\partial^2 w}{\partial y^2} \\ 2z \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix}. \quad (13.3)$$

Розподіл напружень за товщиною пластини показаний на рис.13.5.

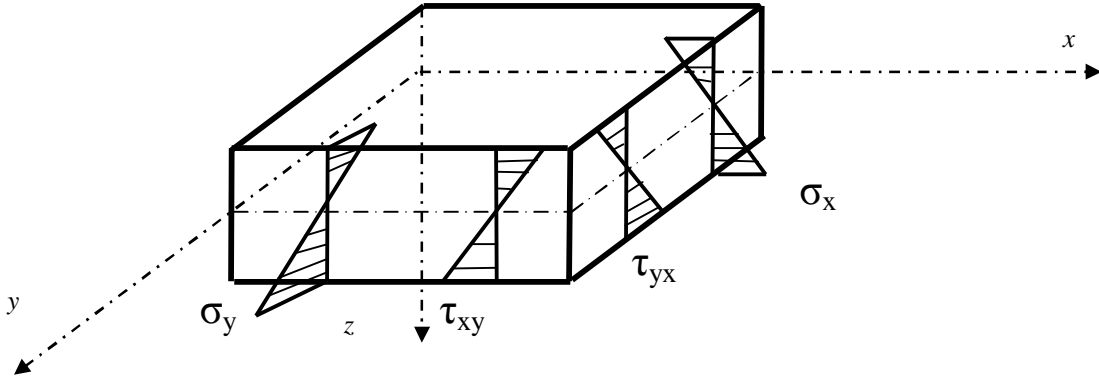


Рисунок 13.5 - Розподіл напружень при згинанні пластини

Визначимо моменти M_x , M_y і M_{xy} з умови рівноваги внутрішніх зусиль:

$$M_x = \int_{-\frac{t}{2}}^{\frac{t}{2}} s_x z dz = \int_{-\frac{t}{2}}^{\frac{t}{2}} \frac{Ez^2}{1-m^2} \left(\frac{\partial^2 w}{\partial x^2} + m \frac{\partial^2 w}{\partial y^2} \right) dz = \frac{Et^3}{12(1-m^2)} \left(\frac{\partial^2 w}{\partial x^2} + m \frac{\partial^2 w}{\partial y^2} \right).$$

$$M_y = \int_{-\frac{t}{2}}^{\frac{t}{2}} s_y z dz = \int_{-\frac{t}{2}}^{\frac{t}{2}} \frac{Ez^2}{1-m^2} \left(m \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right) dz = \frac{Et^3}{12(1-m^2)} \left(m \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right).$$

$$M_{xy} = \int_{-\frac{t}{2}}^{\frac{t}{2}} t_{xy} z dz = \int_{-\frac{t}{2}}^{\frac{t}{2}} \frac{Ez^2}{1+m} \frac{\partial^2 w}{\partial x \partial y} dz = \frac{Et^3}{12(1+m)} \frac{\partial^2 w}{\partial x \partial y}.$$

Вираз $\frac{Et^3}{12(1-m^2)}$, який позначимо через D , називається

циліндровою жорсткістю. З урахуванням цього одержимо вектор зусиль:

$$\{M\} = |D| \begin{vmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1+m}{2} \end{vmatrix} \begin{Bmatrix} z \frac{\partial^2 w}{\partial x^2} \\ z \frac{\partial^2 w}{\partial y^2} \\ 2z \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix}. \quad (13.4)$$

13.3 Функція переміщень для прямокутного елемента

Для прямокутної пластини (див. рис.13.1,б) в кожній вузловій точці вводяться переміщення, які мають по три компоненти: переміщення W_i у напрямку z , кут повороту θ_{x_i} навколо осі x і кут повороту θ_{y_i} навколо осі y .

Для визначення функції форми за дванадцятьма параметрами зручно використовувати поліноми. При цьому в повному поліномі четвертого степеня необхідно опустити частину членів.

Вираз

$$w = a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 + a_7x^3 + a_8x^2y + a_9xy^2 + a_{10}y^3 + a_{11}x^3y + a_{12}xy^3 \quad (13.5)$$

має певні переваги. Зокрема, уздовж будь-якої лінії $x = \text{const}$ або $y = \text{const}$ переміщення W змінюватиметься за кубічним законом. Усі зовнішні межі і межі між елементами складаються саме з таких ліній. Оскільки поліном третього степеня єдиним чином визначається чотирма постійними, переміщення уздовж межі однозначно визначаються значеннями переміщень і кутів нахилу у вузлових точках на кінцях цієї межі. А оскільки для суміжних елементів значення на кінцях межі однакові, уздовж будь-якої межі між елементами функція W буде безперервною.

Постійні a_1, \dots, a_{12} визначаються з системи дванадцяти рівнянь, що зв'язують значення W кутів нахилу у вузлових точках, які виходять в результаті підстановки координат цих точок.

Наприклад:

$$w = a_1 + a_2x_i + a_3y_i + a_4x_i^2 + a_5x_iy_i + a_6y_i^2 + a_7x_i^3 + a_8x_i^2y_i + a_9x_iy_i^2 + a_{10}y_i^3 + a_{11}x_i^3y_i + a_{12}x_iy_i^3,$$

$$\left(\frac{\partial w}{\partial y}\right)_i = q_{x_i} = a_3 + a_5x_i + 2a_6y_i + a_8x_i^2 + 2a_9x_iy_i + 3a_{10}y_i + a_{11}x_i^3 + 3a_{12}x_iy_i^2,$$

$$\left(\frac{\partial w}{\partial x}\right)_i = q_{y_i} = a_2 + 2a_4x_i + a_5y_i + 3a_7x_i^2 + 2a_8x_iy_i + a_9y_i^2 + 3a_{11}x_i^2y_i + a_{11}y_i^3.$$

Ці дванадцять рівнянь можна записати в матричній формі:

$$\{d\}^e = |C|\{a\} \quad (13.6)$$

де $|C|$ – матриця розмірності 12x12, залежна від вузлових координат, а $\{a\}$ – вектор, що містить 12 невідомих постійних.

Обертаючи систему (13.6), одержуємо

$$\{a\} = |C|^{-1}\{d\}^e. \quad (13.7)$$

Вираз для переміщень усередині елемента тепер можна записати в стандартній формі:

$$\{w\} = |N|\{d\}^e. \quad (13.8)$$

де $|N| = |N_i, N_j, N_k, N_e|$ – функції форми.

$$|N| = l \left[\frac{1}{2}(\xi_0 + 1)(\eta_0 + 1)(1 + \xi_0 + \eta_0 - \xi^2 - \eta^2), a\xi_l(\xi_0 + 1)^2(\xi_0 - 1)(\eta_0 + 1), b\eta_l(\xi_0 + 1)(\eta_0 + 1)^2(\eta_0 - 1) \right], \quad (13.9)$$

де $x = \frac{x - x_c}{a}$; $\eta = \frac{y - y_c}{b}$; $x_0 = x_1x$; $h = hh_e$; x_c і y_c – координати центра ваги елемента.

Вираз для матриці $|B|$ виходить безпосередньо із співвідношень (13.9) і (13.2).

Матриця жорсткості елемента визначається з інтеграла

$$|K| = \iint |B|^T |D| |B| dx dy. \quad (13.10)$$

Оскільки в матриці $|B|$ маємо коефіцієнти, залежні від x і y , то при отриманні матриці жорсткості необхідно виконати інтегрування, внаслідок чого одержимо матрицю 12×12 , коефіцієнти якої можна знайти в одному з джерел, наприклад у праці [9].

14 НАПРУЖЕНО-ДЕФОРМОВАНИЙ СТАН ТОНКОСТІННИХ КОНСТРУКЦІЙ

У даному розділі виконано дослідження напружень і деформацій тонкостінних просторових систем типу стрижнів (рис. 14.1), довільних складок (рис. 14.2, а) і оболонок обертання (рис. 14.2, б) при зовнішньому вантаженні.

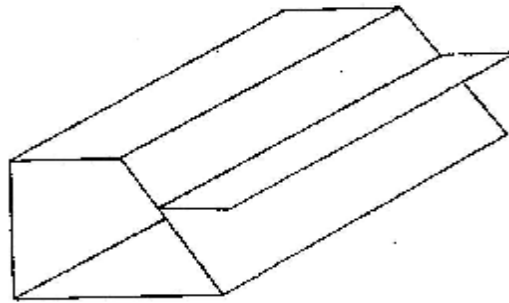
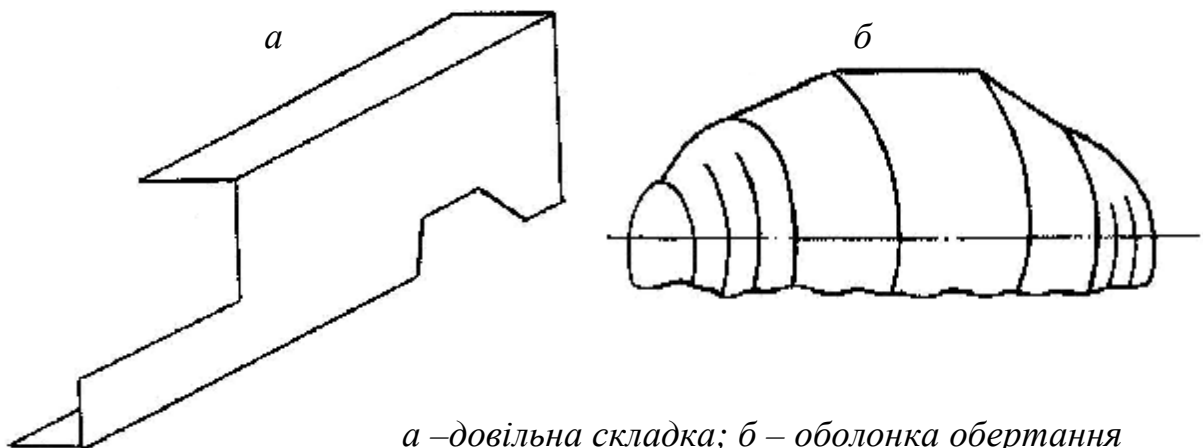


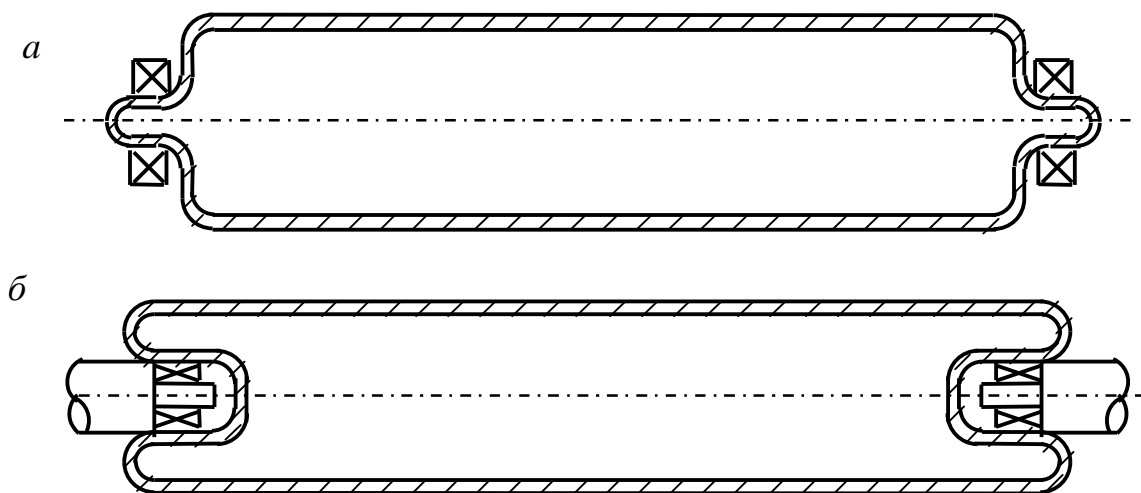
Рисунок 14.1- Тонкостінна просторова система типу призматичного стрижня



а – довільна складка; б – оболонка обертання

Рисунок 14.2

Методика, математичною основою якої є метод кінцевих елементів, розроблялася стосовно розрахунку роликів конвеєрів нової уніфікованої конструкції [13] (див. рис. 14.3).



*а - ролики з цапфами; б - ролики з вбудованими підшипниками
Рисунок - 14.3 Ролики нової уніфікованої конструкції*

14.1 Визначення параметрів кінцевого елемента

Як кінцеві використовуємо елементи пластин трикутної форми. Їх матриця жорсткості є комбінацією матриць жорсткості для плоскої задачі теорії пружності, яка розглянута в десятому розділі (див. 10.18), і матриці жорсткості для пластин, що випробовують згинання. Розглянемо мембранні і згинальні сили і переміщення елемента (рис. 14.4 і 14.5).

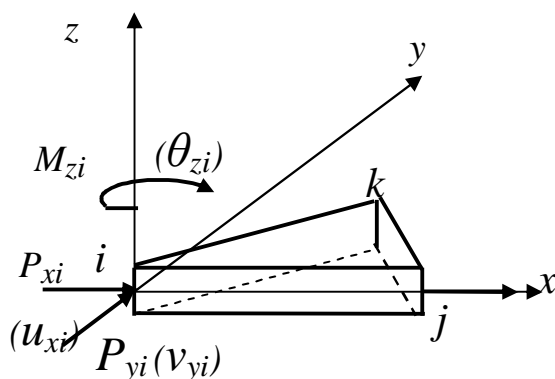


Рисунок 14.4 - Мембранні сили і переміщення

Тепер розглянемо дію згинальних сил (див. рис. 14.5).

Вираз для прогину пластинки, через який однозначно визначаються переміщення і напруження, в загальному випадку має вигляд

$$\{w\} = [N]\{d\}^e, \quad (14.1)$$

де $[N]$ – функції форми, які залежать від параметрів елемента у вибраній системі координат;

$\{d\}^e$ – вектор-стовпець вузлових переміщень.

$$\{d\}^e = \begin{Bmatrix} d_i \\ d_j \\ d_m \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix}, \quad (14.2)$$

$$\{d_i\} = \begin{Bmatrix} w_i \\ q_{xi} \\ q_{yi} \end{Bmatrix} = \begin{Bmatrix} w_i \\ -\left(\frac{\partial w}{\partial y}\right)_i \\ \left(\frac{\partial w}{\partial x}\right)_i \end{Bmatrix}. \quad (14.3)$$

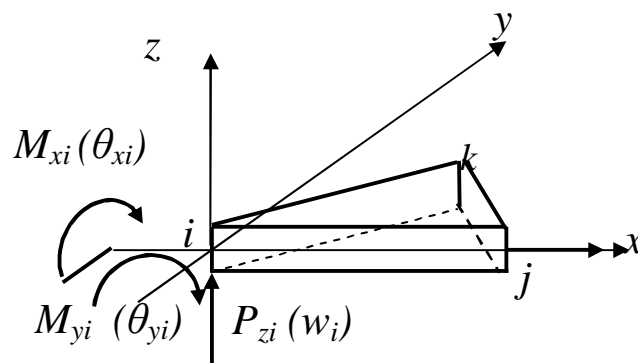


Рисунок 14.5 - Згинаючі сили і переміщення

Оскільки як кінцевий прийнято трикутний елемент, то повинні бути задані дев'ять (по троє на кожен вузол) незалежних параметрів. Якщо ж

задавати переміщення у вигляді полінома, то необхідно задавати десять параметрів, оскільки повний поліном третього степеня має десять коефіцієнтів. У зв'язку з цим один коефіцієнт слід опустити. Цього можна уникнути, якщо скористатися L – координатами. Положення будь-якої точки елемента визначається координатами L_1 , L_2 і L_3 . Вони пов'язані з декартовими координатами таким чином:

$$\begin{aligned} x &= L_1 X_1 + L_2 X_2 + L_3 X_3 \\ y &= L_1 Y_1 + L_2 Y_2 + L_3 Y_3 \\ 1 &= L_1 + L_2 + L_3 \end{aligned} \quad (14.4)$$

На рис.14.6 наведені L -координати і їх зв'язок з декартовими координатами. Легко відмітити, що координата L_1 точки P показує відношення площі заштрихованого трикутника до площі всього трикутника.

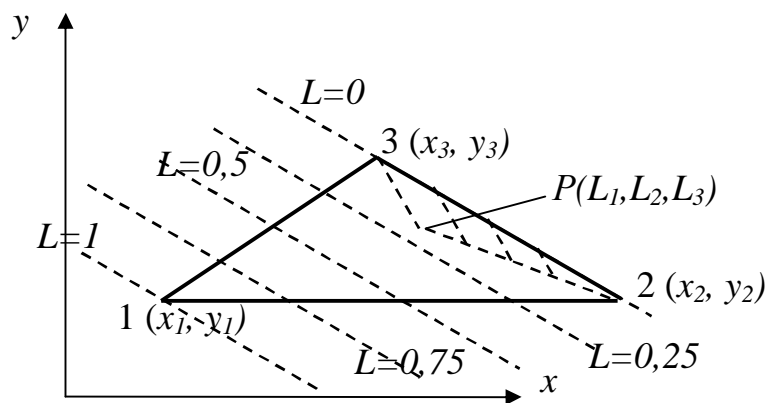


Рисунок 14.6 - L -координати і їх зв'язок з декартовою системою координат

$$L_1 = \frac{\text{площа} P23}{\text{площа} 123}. \quad (14.5)$$

Вирішивши (14.4) відносно L_1 , L_2 і L_3 , одержимо:

$$\left. \begin{aligned} L_1 &= \frac{a_1 + e_1 x + c_1 y}{2\Delta} \\ L_2 &= \frac{a_2 + e_2 x + c_2 y}{2\Delta} \\ L_3 &= \frac{a_3 + e_3 x + c_3 y}{2\Delta} \end{aligned} \right\} \quad (14.6)$$

Значення a_i , b_i , c_i і 2Δ визначаються виразами (10.15). Переміщення пластинки в L координатах можна подати у вигляді

$$w = b_1 L_1 + b_2 L_2 + b_3 L_3 + b_4 (L_2^2 L_1 + \frac{1}{2} L_1 L_2 L_3) + \dots + b_9 (L_1^2 L_3 + \frac{1}{2} L_1 L_2 L_3). \quad (14.7)$$

Після підстановки вузлових значень w_i ;

$$q_{xi} = -\left(\frac{\partial w}{\partial y}\right)_i; \quad q_{yi} = \left(\frac{\partial w}{\partial x}\right)_i. \quad (14.8)$$

Визначаємо коефіцієнти $\beta_1, \beta_2, \dots, \beta_9$, а, отже, і функцію форми.

Функція форми для першого вузла набуває вигляду

$$[N_1]^T = \left\{ \begin{array}{l} L_1 + L_1^2 L_2 + L_1^2 L_3 - L_1 L_2^2 - L_1 L_3^2 \\ \epsilon_3 (L_1^2 L_3 + \frac{1}{2} L_1 L_2 L_3) - \epsilon_2 (L_3 L_1^2 + \frac{1}{2} L_1 L_2 L_3) \\ C_3 (L_1^2 L_2 + \frac{1}{2} L_1 L_2 L_3) - C_2 (L_3 L_1^2 + \frac{1}{2} L_1 L_2 L_3) \end{array} \right\}, \quad (14.9)$$

де $\epsilon_2, \epsilon_3, C_2, C_3$ – коефіцієнти, визначувані виразом (10.15).

Функції форми для вузлів 2 і 3 одержуємо циклічною перестановкою індексів 1-2-3 і т.п.

Деформації в пластині, визначувані через переміщення, дорівнюють

$$\{e\} = \left\{ \begin{array}{l} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{array} \right\}. \quad (14.10)$$

Оскільки, прогин є функцією L_1, L_2 і L_3 , які у свою чергу залежать від x і y , то похідні вирази (14.10) мають бути узяті як для складної функції, зокрема

$$\frac{\partial w}{\partial x} = \frac{\partial w}{\partial L_1} \cdot \frac{\partial L_1}{\partial x} + \frac{\partial w}{\partial L_2} \cdot \frac{\partial L_2}{\partial x} + \frac{\partial w}{\partial L_3} \cdot \frac{\partial L_3}{\partial x}, \quad (14.11)$$

але

$$\frac{\partial L_1}{\partial x} = \epsilon_1 \frac{1}{2\Delta}; \quad \frac{\partial L_2}{\partial x} = \epsilon_2 \frac{1}{2\Delta}; \quad \frac{\partial L_3}{\partial x} = \epsilon_3 \frac{1}{2\Delta},$$

а значить

$$\frac{\partial w}{\partial x} = \frac{1}{2\Delta} \left(\epsilon_1 \frac{\partial w}{\partial a_1} + \epsilon_2 \frac{\partial w}{\partial a_2} + \epsilon_3 \frac{\partial w}{\partial a_3} \right) \quad (14.12)$$

Маючи це на увазі, одержимо:

$$\begin{aligned} -\frac{\partial^2 w}{\partial x^2} = & \frac{-1}{4\Delta^2} \{ 2[(2\epsilon_1\epsilon_2 + 2\epsilon_1\epsilon_3 - \epsilon_2^2 - \epsilon_3^2)L_1 + (\epsilon_1^2 - 2\epsilon_1\epsilon_2)L_2 + (\epsilon_1^2 - 2\epsilon_1\epsilon_3)L_3]w_1 + \\ & + [\epsilon_2\epsilon_3(\epsilon_3 - \epsilon_2)L_1 + \epsilon_3\epsilon_1(\epsilon_3 - \epsilon_2 + 2\epsilon_1)L_2 + \epsilon_2\epsilon_1(\epsilon_3 - \epsilon_2 - \epsilon_1)L_3]qx_1 + [(4\epsilon_1(\epsilon_2c_3 - \epsilon_3c_2) + \\ & + \epsilon_2\epsilon_3(c_3 - c_2))L_1 + (2\epsilon_1^2c_3 + \epsilon_1\epsilon_3(c_3 - c_2))L_2 + (\epsilon_1\epsilon_2(c_3 - c_2) - 2\epsilon_1^2c_2)L_3]qy_1 + \\ & + 2[(2\epsilon_2\epsilon_3 + 2\epsilon_2\epsilon_1 - \epsilon_3^2 - \epsilon_1^2)L_2 + (\epsilon_2^2 - 2\epsilon_2\epsilon_3)L_3 + (\epsilon_2^2 - 2\epsilon_2\epsilon_1)L_1 + [\epsilon_3\epsilon_1(\epsilon_1 - \epsilon_3)L_2 + \\ & + \epsilon_1\epsilon_2(\epsilon_1 - \epsilon_3 + 2\epsilon_2)L_3 + \epsilon_3\epsilon_2(\epsilon_1 - \epsilon_3 - 2\epsilon_2)L_1]qx_2 + [(4\epsilon_2(\epsilon_3c_1 - \epsilon_1c_3) + \\ & + \epsilon_3\epsilon_1(c_1 - c_3))L_2 + \epsilon_2\epsilon_1(c_1 - c_3) - 2\epsilon_2^2c_3)L_1]qy_2 + 2[2\epsilon_3\epsilon_1 + 2\epsilon_3\epsilon_2 - \epsilon_1^2 - \epsilon_2^2)L_3 + (\epsilon_3^2 - \\ & - 2\epsilon_3\epsilon_1)L_1 + (\epsilon_3^2 - 2\epsilon_3\epsilon_2)L_2]w_3 + [\epsilon_1\epsilon_2(\epsilon_2 - \epsilon_1)L_3 + \epsilon_2\epsilon_3(\epsilon_2 - \epsilon_1 + 2\epsilon_3)L_1 - \\ & - \epsilon_1\epsilon_3(\epsilon_2 - \epsilon_1 - 2\epsilon_3)L_2]qx_3 + +4\epsilon_3(\epsilon_1c_2 - \epsilon_2c_1) - \epsilon_1\epsilon_2(c_2 - c_1))L_3 + \\ & + (2\epsilon_3^2c_2 + \epsilon_3\epsilon_2(c_2 - c_1))L_1 - (\epsilon_3\epsilon_1(c_2 - c_1) - 2\epsilon_3^2c_1^2)L_2]qy_3 \} \quad (14.13) \end{aligned}$$

$$-\frac{\partial^2 w}{\partial y^2} = \frac{-1}{4\Delta^2} \{ 2[(2c_1c_2 + 2c_1c_3 - c_2^2 - c_3^2)L_1 + (c_1^2 - 2c_1c_2)L_2 + (c_1^2 - 2c_1c_3)L_3]w_1 +$$

$$\begin{aligned}
& + [c_2c_3(c_3 - c_2)L_1 + c_3c_1(c_3 - c_2 + 2c_1)L_2 + c_2c_1(c_3 - c_2 - 2c_1)L_3]qy_1 + [(4c_1(c_2\theta_3 - \\
& - c_3\theta_2) + c_2c_3(\theta_3 - \theta_2))L_1 + (2c_1^2c_3 + c_1c_3(\theta_3 - \theta_2))L_2 + (c_1c_2(\theta_3 - \theta_2) - \\
& - 2c_1^2\theta_2)L_3]qx_1 + 2[(2c_2c_3 + 2c_2c_1 - c_3^2 - c_1^2)L_2 + (c_2^2 - 2c_2c_3)L_3 + (c_2^2 - \\
& - 2c_2c_1)L_1] + [c_3c_1(c_1 - c_3)L_2 + \\
& + c_1c_2(c_1 - c_3 + 2c_2)L_3 + c_3c_2(c_1 - c_3 - 2c_2)L_1]qy_2 + [(4c_2(c_3\theta_1 - c_1\theta_3) + c_3c_1(\theta_1 - \theta_3))L_2 + \\
& + (2c_2^2\theta_1 + c_2c_1(\theta_1 - \theta_3))L_3 + (c_2c_3(\theta_1 - \theta_3) - 2c_2^2\theta_3)L_1]qx_2 + 2[2c_3c_1 + \\
& + 2c_3c_3 - c_1^2 - c_2^2)L_3 + (c_3^2 - 2c_3c_1)L_1 + \\
& + (c_3^2 - 2c_3c_2)L_2]w_3 + [c_1c_2(c_2 - c_1)L_3 + c_2c_3(c_2 - c_1 + 2c_3)L_1 + \\
& + c_1c_3(c_2 - c_1 - 2c_3)L_2]qy_3 + [(4c_3(c_1\theta_2 - c_2\theta_1) + c_1c_2(\theta_2 - \theta_1))L_3 + \\
& + (2c_3^2\theta_2 + c_3c_2(\theta_2 - \theta_1))L_1 + (c_3c_1(\theta_2 - \theta_1) - 2c_3^2\theta_1^2)L_2]qx_3 \} \tag{14.14}
\end{aligned}$$

$$\begin{aligned}
2 \frac{\partial^2 w}{\partial x \partial y} = \frac{1}{2\Delta^2} \{ & [2(c_1\theta_2 + c_2\theta_1 + c_1\theta_3 + c_3\theta_1 - c_2\theta_2 - c_3\theta_3)L_1 + 2(c_1\theta_1 - c_1\theta_2 - c_2\theta_1)L_2 + \\
& + 2(c_1\theta_1 - c_1\theta_3 - c_3\theta_1)L_3]w_1 + [(0,5(c_3\theta_2 + c_2\theta_3)(\theta_3 - \theta_2) + 2\theta_1(\theta_3c_2 - \theta_2c_3))L_1 + \\
& + (0,5(\theta_3 - \theta_2)(c_1\theta_3 + c_3\theta_1) + 2\theta_1c_1\theta_3)L_2 + (0,5(\theta_3 - \theta_2)(\theta_2c_1 + \theta_1c_2) - 2\theta_1c_1\theta_2)L_3]qx_1 + \\
& + [(0,5(c_3 - c_2)(\theta_3c_2 + \theta_2c_3) + 2c_1(\theta_2c_3 - \theta_3c_2))L_1 + (2\theta_1c_1c_3 + 0,5(c_3 - \\
& - c_2)(c_1\theta_3 + c_3\theta_1)L_2 + (0,5(c_3 - c_2)(c_1\theta_2 + c_2\theta_1) - 2\theta_1c_2c_1)L_3]qy_1 + [2(c_2\theta_3 + c_3\theta_2 + \\
& + c_3\theta_2 + c_2\theta_1 + c_1\theta_2 - c_3\theta_3 - c_1\theta_1)L_2 + 2(c_2\theta_2 - c_2\theta_3 - c_3\theta_2)L_3 - \\
& - 2(c_2\theta_2 - c_2\theta_1 - c_1\theta_2)L_1]w_2 + [(0,5(c_1\theta_3 + c_3\theta_1)(\theta_1 - \theta_3) + 2\theta_2(\theta_1c_3 -
\end{aligned}$$

$$\begin{aligned}
& -\vartheta_3 c_1))L_2 + (0,5(\vartheta_1 - \vartheta_3)(c_2 \vartheta_1 - \vartheta_2 c_1) + 2\vartheta_2 c_2 \vartheta_1)L_3 + (0,5(\vartheta_1 - \vartheta_3)(c_2 \vartheta_3 + \vartheta_2 c_3) - \\
& - 2\vartheta_2 c_2 \vartheta_3)L_1]q_{x_2} + [(0,5(c_1 - c_3)(\vartheta_1 c_3 + \vartheta_3 c_1) + 2c_2(\vartheta_3 c_1 - \vartheta_1 c_3)L_2 - \\
& - (2\vartheta_2 c_2 c_1 - 0,5(c_1 - c_3)(c_2 \vartheta_1 + c_1 \vartheta_2))L_3 + (0,5(c_1 - c_3)(\vartheta_3 c_2 + \vartheta_2 c_3) - \\
& - 2\vartheta_2 c_3 c_2)L_1]q_{y_2} + [2(c_2 \vartheta_1 + c_1 \vartheta_3 + c_3 \vartheta_2 + c_2 \vartheta_3 - c_1 \vartheta_1 - c_2 \vartheta_2)L_3 + \\
& + 2(c_3 \vartheta_3 - c_3 \vartheta_1 - c_1 \vartheta_3)L_1 + 2(c_3 \vartheta_3 - c_3 \vartheta_2 - c_2 \vartheta_3)L_2]w_3 + \\
& (0,5(c_2 \vartheta_1 + c_1 \vartheta_2)(\vartheta_2 - \vartheta_1) + 2\vartheta_3(\vartheta_2 c_1 - \vartheta_1 c_2))L_3 + \\
& + (0,5(\vartheta_2 - \vartheta_1)(c_3 \vartheta_2 + \vartheta_3 c_2) + 2\vartheta_3 c_3 \vartheta_2)L_1 - (0,5(\vartheta_2 - \vartheta_1)(c_3 \vartheta_1 + \vartheta_3 c_1) - 2\vartheta_3 c_3 \vartheta_1)L_2]q_{x_3} + \\
& (0,5(c_2 - c_1)(\vartheta_2 c_1 + \vartheta_1 c_2) + 2c_3(\vartheta_1 c_2 - \vartheta_2 c_1))L_3 + (2\vartheta_3 c_3 c_2 + \\
& + 0,5(c_2 - c_1)(\vartheta_2 c_3 + \vartheta_3 c_2))L_1 + 0,5(c_2 - c_1)(\vartheta_1 c_3 + \vartheta_3 c_1) - 2\vartheta_3 c_3 c_1)L_2]q_{y_3}. \quad (14.15)
\end{aligned}$$

Вирази (14.13), (14.14) і (14.15) в розгорненій формі є деформації пластинки трикутної форми. Відокремивши від них вектор переміщень $\{w_1, \theta_{x1}, \theta_{y1}, w_2, \theta_{x2}, \theta_{y2}, w_3, \theta_{x3}, \theta_{y3}\}^T$, одержуємо матрицю [B]:

$$[B] = \frac{1}{4\Delta^2} [\Phi_i, \Phi_j, \Phi_m], \quad (14.16)$$

де

$$\{\Phi_i\} = \begin{bmatrix} j_i(\vartheta, \vartheta); & y_i(\vartheta, \vartheta); & c_i(\vartheta, c) \\ j_i(c, c); & c_i(c\vartheta\vartheta); & y_i(c, c) \\ -2j_i(\vartheta, c); & -2y_i(\vartheta, c); & -2y_i(c, \vartheta) \end{bmatrix} [L]_{ijm}. \quad (14.17)$$

Матриці Φ_j і Φ_m виходять з Φ_i шляхом циклічної підстановки індексів.

$$[L]_{ijm} = [L_i, L_j, L_m]^T. \quad (14.18)$$

$$j_i(d, h) = [h_i d_i + h_j d_j + h_i d_m + h_m d_i - h_m d_m,$$

$$h_i d_i - h_j d_j - h_i d_j, h_i d_i - h_i d_m - h_m d_i] \quad (14.19)$$

$$y_i(d, h) = [0,5(h_m d_j + h_j d_m)(d_m - d_j) + 2d_i(d_m h_j - d_j h_m),$$

$$0,5(h_i d_m + h_m d_i)(d_m - d_j) + 2d_i h_i d_m,$$

$$0,5(h_i d_j + h_j d_i)(d_m - d_j) - 2d_i h_i d_j]. \quad (14.20)$$

$$c_i(d, h) = [4d_i(d_j h_n - d_m h_j) + d_j d_m (h_m - h_j),$$

$$2d_i^2 h_m + d_m d_i (h_m - h_j), d_j d_i (h_m - h_j) - 2d_i^2 h_j]. \quad (14.21)$$

Відповідно напруженнями є згинальні і крутні моменти на одиницю довжини у напрямках x і y :

Функції $\varphi(b, b)$, $\psi_i(b, b)$, $\chi_i(b, c)$.. виходять з виразів (14.19)...(14.21) шляхом транспонування і підстановки у них замість формальних параметрів d і h , фактичних b і c .

$$\{S\} = \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = [D]\{e\}, \quad (14.22)$$

де

$$[D] = \frac{Et^3}{12(1-m^2)} \begin{bmatrix} 1 & m & 0 \\ m & 1 & 0 \\ 0 & 0 & \frac{1-m}{2} \end{bmatrix}.$$

Згідно з виразом (1.6) матриця жорсткості елемента, що пов'язує вузлові сили (поперечну силу і два моменти в кожному вузлі) з відповідними вузловими переміщеннями, дорівнює

$$[K]^e = \iint [B]^T [D][B] dx dy . \quad (14.23)$$

Оскільки до матриці [B] входять a_1, a_2 і a_3 , то інтегрування виразу (14.23) пов'язане з великими труднощами. Зазвичай у таких випадках виконують чисельне інтегрування. У зв'язку з тим, що до матриці входять тільки квадратичні члени, то інтегрування за трикутником буде точним при використанні всього лише трьох точок (див. [1], таб. 8.3), а час чисельного інтегрування майже не відрізняється від часу розрахунку при точному інтегруванні. Виходячи зі сказаного,

$$[K]^e = \sum_{n=1}^3 [B]^T [D][B_n] 2W\Delta , \quad (14.24)$$

де $n=1$; $2W$ - вагові коефіцієнти;

$2W=1/3$ – (див. [1], табл. 8.3, порядок елемента – другий);

[B_n] – матриця [B]; L -координати, які відповідно до тієї ж таблиці дорівнюють:

при $n=1$	$L_1=0.5;$	$L_2= 0.5;$	$L_3=0;$
при $n=2$	$L_1=0;$	$L_2=0.5;$	$L_3=0.5;$
при $n=3$	$L_1=0.5;$	$L_2=0;$	$L_3=0.5.$

Одержавши матриці жорсткості для плоского напруженого стану і для пластини при згинанні, можна записати матрицю жорсткості при одночасній дії мембранних і згинальних сил. Вона складається з наступних підматриць:

$$[K_{rs}]^e = \begin{bmatrix} [K_{rs}]^p & & & & & \\ & 0 & 0 & 0 & 0 & \\ & & 0 & 0 & 0 & \\ 0 & 0 & [K_{rs}]^g & & & \\ 0 & 0 & & & & 0 \\ 0 & 0 & & & & 0 \\ 0 & 0 & & & & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} . \quad (14.25)$$

Тут $[K_{rs}]^p$ - підматриця жорсткості при дії мембранних сил;

$[K_{rs}]^g$ - підматриця жорсткості при дії згинальних сил;

r і s змінюються від 1 до 3 (оскільки елемент має три вузлові точки).

Повна матриця жорсткості елемента має розмірність 18x18 і складається з дев'яти підматриць $[K_{rs}]$:

$$[K]^P = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}. \quad (14.26)$$

У підматриці $[K_{rs}]$ коефіцієнт у шостому рядку і шостому стовпці дорівнює нулю. Він повинен відображати жорсткість елемента повороту навколо осі, перпендикулярної площині елемента.

Для того, щоб останнє з шести рівнянь рівноваги (відповідне напрямку Q_z) в місцевій системі координат не оберталось в рівність $0=0$ і в глобальній системі – в рівність $A*0=0$, користуємося системою фіктивних коефіцієнтів.

$$\begin{Bmatrix} M_{zi} \\ M_{zj} \\ M_{zk} \end{Bmatrix} = aEt\Delta \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{bmatrix} \cdot \begin{Bmatrix} q_{zi} \\ q_{zj} \\ q_{zk} \end{Bmatrix} \quad (14.27)$$

При такій матриці рівновага в місцевій системі координат не порушується.

У виразі (14.27) α – коефіцієнт, який слід задавати, зазвичай рекомендують $\alpha \leq 0,03$. У даній роботі, згідно з рекомендаціями [1], стр.239, взяли $\alpha = 0,0001$.

14.2 Жорсткість плоского трикутного елемента в глобальній системі координат

Щоб одержати жорсткість трикутного елемента в глобальній системі координат, необхідно, перш за все, визначити локальні осі і їх напрямні косинуси, які простіше за все розв'язуються з використанням векторної алгебри.

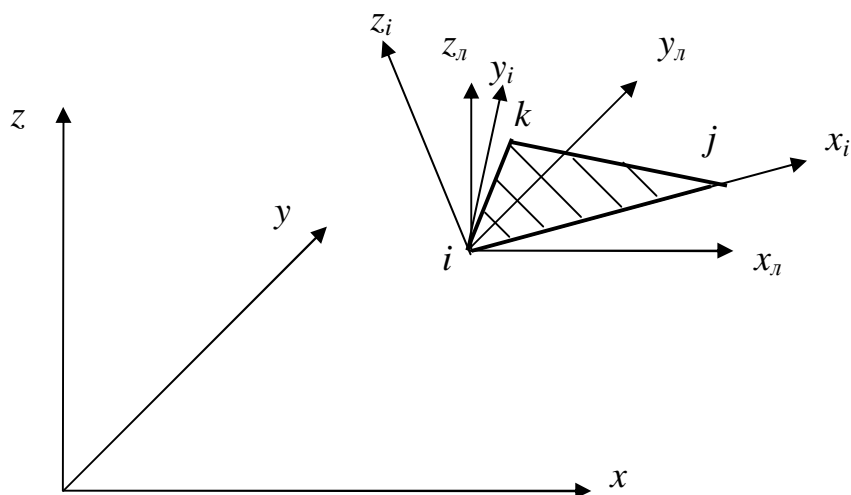


Рисунок 14.7– Системи координат: глобальна - x, y, z , паралельна глобальній – x_l, y_l, z_l і локальна – x_i, y_i, z_i

Вважаємо, що вісь x_l проходить через сторону ij , вісь z_i направимо перпендикулярно площині трикутника, а вісь y_l перпендикулярна площині, що проходить через осі x_l і z_l .

Вісь x_l визначається вектором v_{ij} з глобальними координатами:

$$v_{ij} = \begin{Bmatrix} x_j - x_i \\ y_j - y_i \\ z_j - z_i \end{Bmatrix} = \begin{Bmatrix} x_{lj} \\ y_{lj} \\ z_{lj} \end{Bmatrix}. \quad (14.28)$$

Тут x_{lj} , y_{lj} і z_{lj} – координати точки j в системі осей x_l , y_l і z_l , які паралельні глобальним осям і проходять через точку i .

Її напрямні косинуси виходять діленням проекції вектора на осі на довжину:

$$v_x = \begin{Bmatrix} l_{x1x} \\ l_{x1y} \\ l_{x1z} \end{Bmatrix} = \frac{1}{l_{ij}} \begin{Bmatrix} x_{lj} \\ y_{lj} \\ z_{lj} \end{Bmatrix}, \quad (14.29)$$

де $l_{ij} = \sqrt{x_{lj}^2 + y_{lj}^2 + z_{lj}^2}$

Напрямок осі z_1 визначається як векторний добуток сторін трикутника ij і im .

$$v_z = v_{ij} \cdot v_{im} = \begin{Bmatrix} y_{lj} z_{lm} & - & z_{lj} y_m \\ -x_j z_{lj} & - & x_{lj} z_{lm} \\ -x_{lj} y_{lj} & - & x_{lm} y_{lj} \end{Bmatrix}. \quad (14.30)$$

Довжина вектора

$$l_z = \sqrt{(y_{nj} z_{nm} - z_{nj} y_{nm})^2 + (-x_{nm} y_{nj} - x_{nj} y_{nm})^2 + (-x_{nj} z_{nm} - z_{nj} x_{nm})^2},$$

а напрямні косинуси:

$$v_{z_1} = \begin{Bmatrix} \lambda_{z_1 x} \\ \lambda_{z_1 y} \\ \lambda_{z_1 z} \end{Bmatrix} = \frac{1}{2\Delta} \begin{Bmatrix} y_{lj} z_{lm} & - & z_{lj} y_{lm} \\ -x_{lm} z_{lj} & - & x_{lj} z_{lm} \\ -x_{lj} y_{lm} & - & x_{lm} y_{lj} \end{Bmatrix}. \quad (14.31)$$

Напрямні косинуси осі y_1 визначаються як напрямні косинуси вектора, перпендикулярного одночасно до осей x_1 і z_1 .

$$v_{y_1} = \begin{Bmatrix} I_{y_1 x} \\ I_{y_1 y} \\ I_{y_1 z} \end{Bmatrix} = \begin{Bmatrix} I_{z_1 y} I_{x_1 z} & - & I_{z_1 z} I_{x_1 y} \\ I_{z_1 z} I_{x_1 x} & - & I_{z_1 x} I_{x_1 z} \\ I_{z_1 x} I_{x_1 y} & - & I_{z_1 y} I_{x_1 x} \end{Bmatrix} \quad (14.32)$$

Тепер матриця жорсткості в глобальних координатах набуває вигляду

$$[K] = [T]^T [K]^n [T], \quad (14.33)$$

де $[T]$ – матриця напрямних косинусів.

$$[T] = \begin{Bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{Bmatrix}. \quad (14.34)$$

$$L = \begin{Bmatrix} I & 0 \\ 0 & I \end{Bmatrix}. \quad (14.35)$$

$$[I] = \begin{Bmatrix} I_{x_1x} & I_{x_1y} & I_{x_1z} \\ I_{y_1x} & I_{y_1y} & I_{y_1z} \\ I_{z_1x} & I_{z_1y} & I_{z_1z} \end{Bmatrix}. \quad (14.36)$$

Вище наведені всі вирази, необхідні для формування системи канонічних рівнянь (1.7). Вирішивши цю систему, знаходимо переміщення вузлів у глобальній системі координат.

Визначення напружень через переміщення проводимо наступним чином:

а) у площині пластини:

$$[S] = [D][B][T_1]\{d\}, \quad (14.37)$$

де $[B]$ - матриця, визначувана виразом (10.17);

$\{d\}$ – вектор переміщень в глобальній системі координат.

$$\{d\} = \begin{Bmatrix} d_i \\ d_j \\ d_m \end{Bmatrix} \quad (14.38)$$

або

$$\{d\} = \begin{Bmatrix} U_i \\ J_i \\ w_i \\ q_{xi} \\ q_{yi} \\ q_{zi} \end{Bmatrix}. \quad (14.39)$$

$$[T_1] = \begin{bmatrix} L_1 & 0 & 0 \\ 0 & L_1 & 0 \\ 0 & 0 & L_1 \end{bmatrix}. \quad (14.40)$$

$$[L_1] = \begin{bmatrix} I_{x_1x} & I_{x_1y} & I_{x_1z} \\ I_{y_1x} & I_{y_1y} & I_{y_1z} \end{bmatrix}. \quad (14.41)$$

б) перпендикулярно площині пластини:

$$[S] = [D][B][T_2]\{d\}, \quad (14.42)$$

де $[B]$ – матриця, яка визначається виразом (14.16).

$$[T_2] = \begin{bmatrix} L_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & L_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_1 \end{bmatrix}, \quad (14.43)$$

$$[L_2] = [I_{z_1x} \quad I_{z_1y} \quad I_{z_1z}]. \quad (14.44)$$

14.3 Програма визначення деформацій і напружень в тонкостінних конструкціях

14.3.1 Сутність програми

Вище детально викладена теорія методу кінцевих елементів стосовно до оболонкових конструкцій. На підставі отриманих формул складена програма розрахунку напружень і деформацій оболонок і складок. Програма (її текст наведено в додатку Д) складається з шести підпрограм: GDATA – читання початкових даних; MATRIC – формування матриці жорсткості конструкції; STIFT – матриці жорсткості кінцевого елемента; SISTEM – рішення системи рівнянь; DEFORM – визначення переміщень; SORCE – визначення напружень.

14.3.2 Апробація програми

Результати, одержані за допомогою програми, з метою перевірки їх достовірності необхідно порівняти з відомими аналітичними

рішеннями задач опору матеріалів і теорії пружності. Автори зіставили наступні задачі: розтягання стрижня, згинання криволінійного стрижня, згинання пластини, згинання оболонки.

Розтягання стрижня (рис.14.8...14.12).

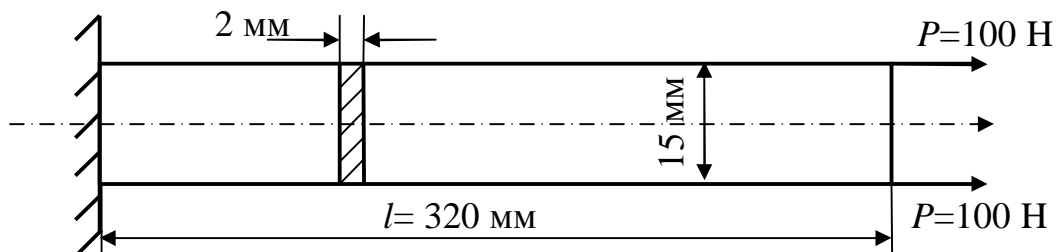


Рисунок 14.8 - Схема розтянутого стрижня

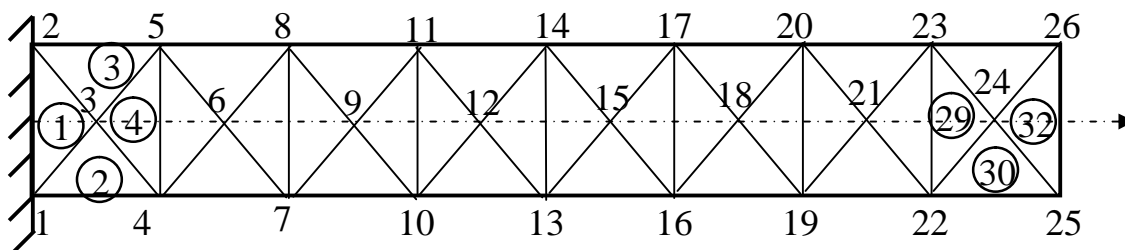
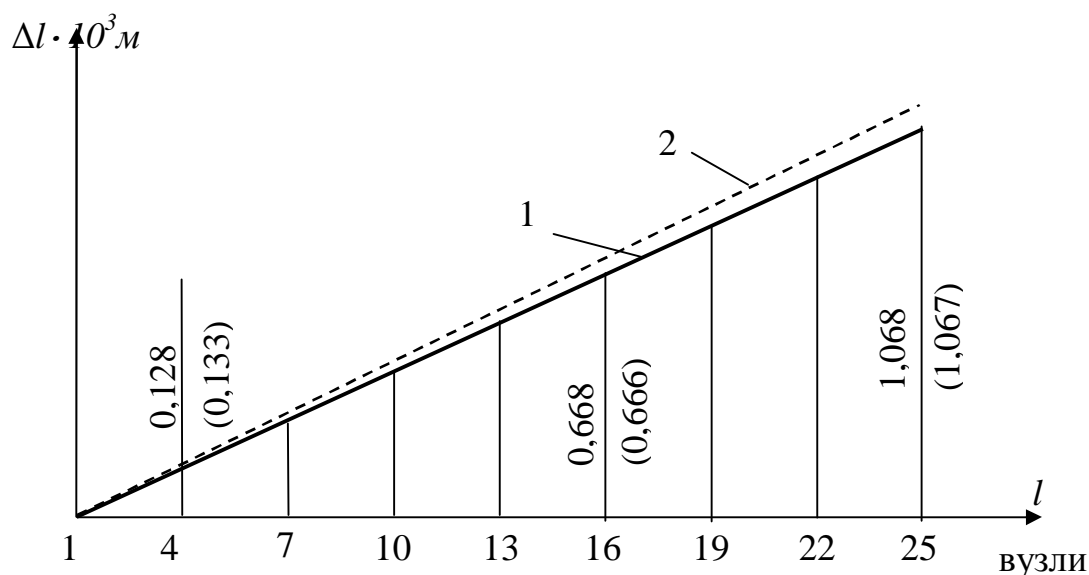
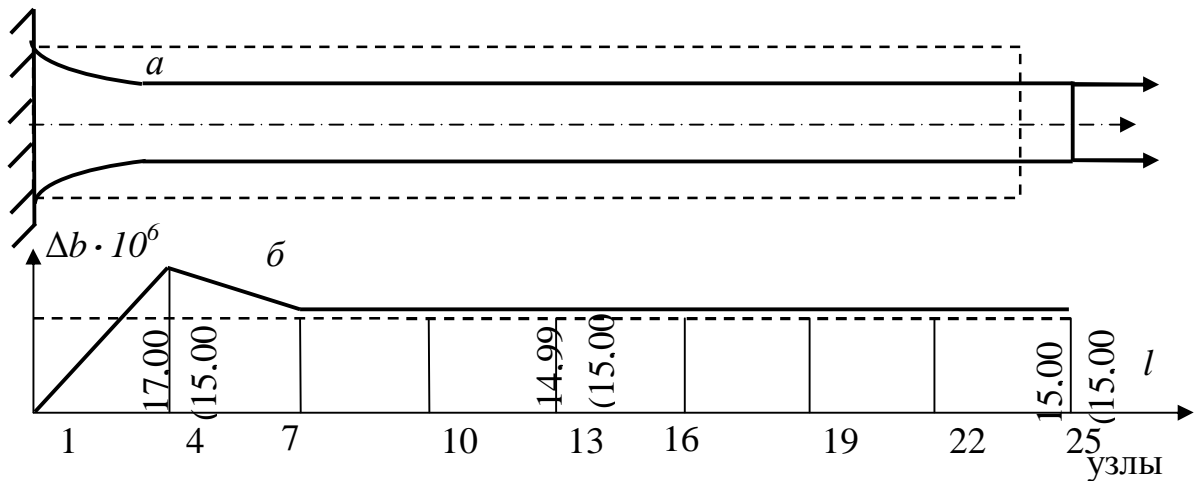


Рисунок 14.9 - Схема розбивки на елементи розтянутого стрижня (номери елементів в кружочках)



*крива 1 - за методом кінцевих елементів (за програмою);
крива 2 - за формулами опору матеріалів (значення у дужках)*

Рисунок 14.10 - Епюри подовжніх подовжень



a - схема стрижня до і після деформації; *б* - епюри поперечних подовжень: крива 1 - за програмою; крива 2 - за формулами опору матеріалів (значення у дужках).

Рисунок 14.11 – Поперечна деформація стрижня

На рис.14.10 і 14.11 наведені епюри подовжніх і поперечних подовжень. Подовжні подовження, пораховані за програмою, відрізняються від теоретичних всього на 0,5%. Поперечні подовження, обчислені за програмою, в більшості точок співпадають з теоретичними значеннями, але різко відрізняються в районі кріплення. Це пов'язано з тим, що формули опору матеріалів не враховують особливостей закріплення стрижня (див. рис. 14.11,б).

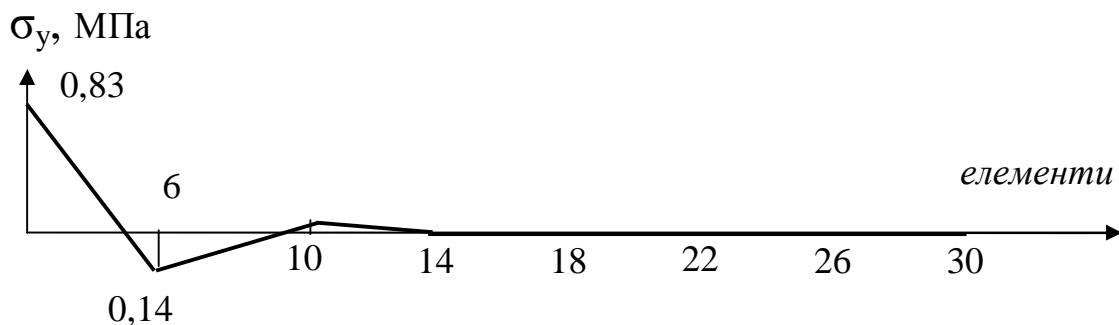


Рисунок - 14.12 Епюра напружень σ_y , які одержані за програмою

Цим же пояснюється і розбіжності у визначенні напружень σ_y . Як видно з рис. 14.12, напруження σ_y має сплеск у закладення, а в решті елементів, як і за елементарною теорією, дорівнює нулю. Напруження σ_x у всіх елементах за обома методиками повністю співпадає і дорівнює 6,67 МПа.

Згинання балки (рис. 14.13 ...14.15).

На рис. 14.13 зображена балка довжиною 0,32 м з поперечним

перерізом $3 \times 30 \text{ мм}^2$, яка жорстко закріплена по краях і навантажена силою $P=0,4 \text{ кН}$.

З рис. 14.14 видно, що значення прогинів, розраховані за програмою, відрізняються від теоретичних всього на 1,3%, а моменти – на 2 %.

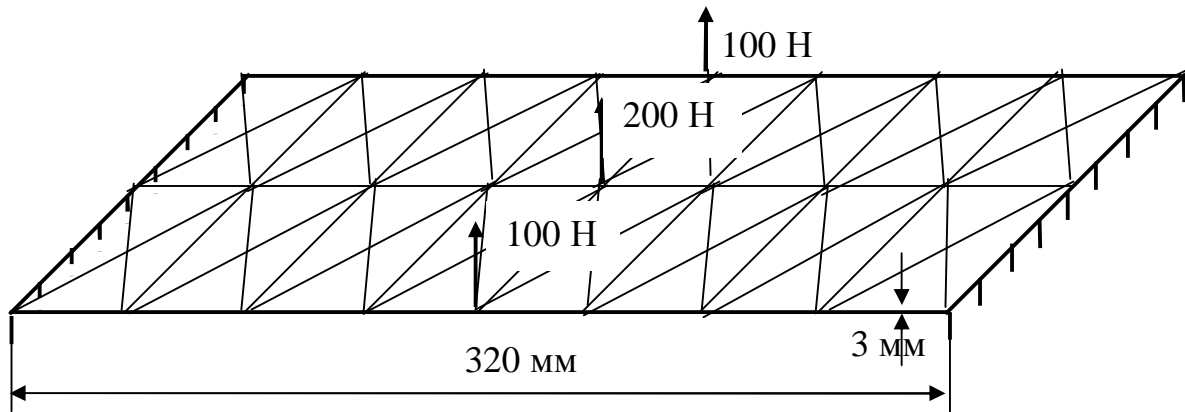
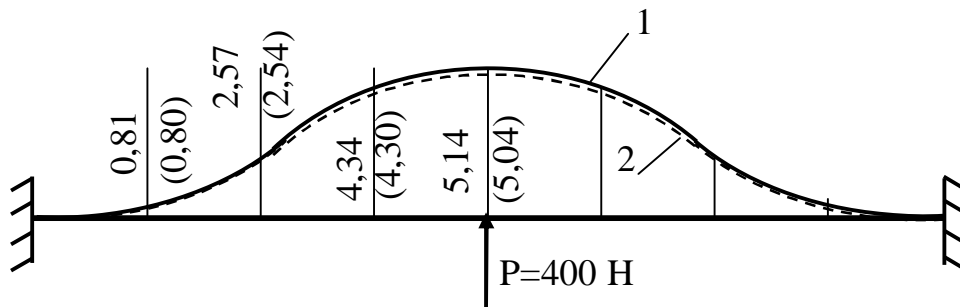
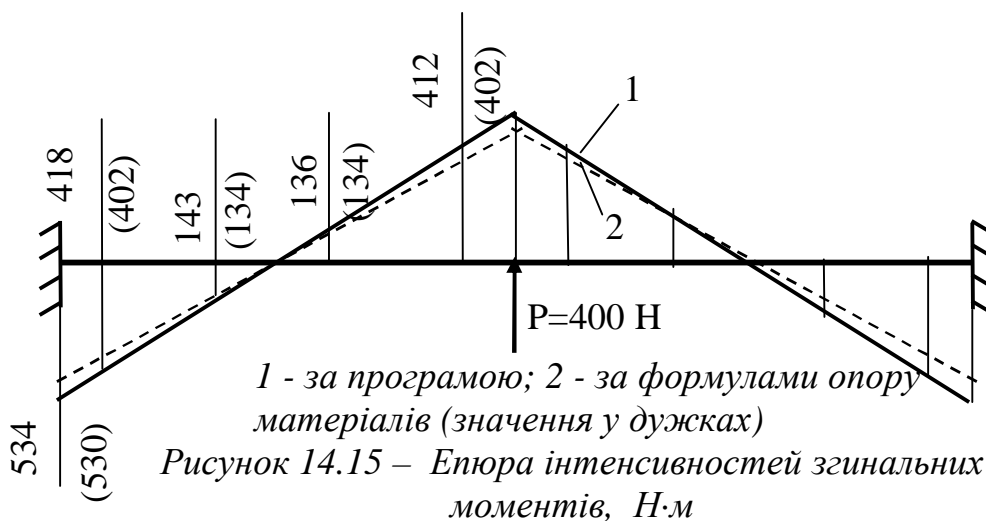


Рисунок 14.13– Схема і розбивання на елементи зацмленої балки



крива 1 – за програмою;
крива 2 (значення у дужках) - за формулами опору матеріалів
Рисунок 14.14 - Пружна лінія, мм.



1 - за програмою; 2 - за формулами опору матеріалів (значення у дужках)
Рисунок 14.15 – Епюра інтенсивностей згинальних моментів, Н·м

Згинання криволінійного стрижня(рис. 14.16 ...14.18).

На рис. 14.16,а показано криволінійний стрижень діаметром 200 мм, шириною 10 мм і товщиною 2 мм, який розтягується силою $P=40$ кН.

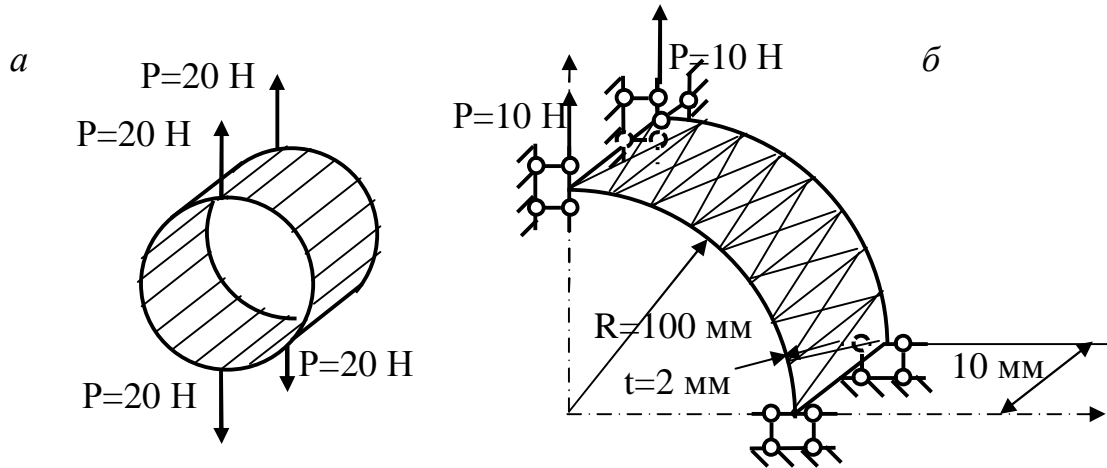
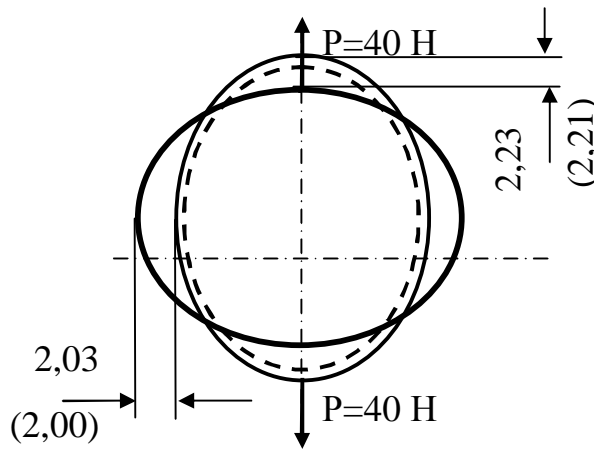
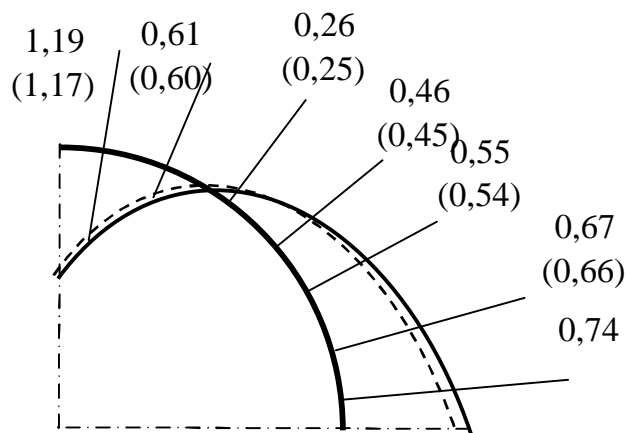


Рисунок 14.16 – Схема і розбивання на елементи криволінійного стрижня



крива 1 - за програмою;
крива 2 – за формулами опору матеріалів (значення у дужках, мм)

Рисунок 14.17 - Зігнута вісь стрижня



крива 1 - за програмою;
крива 2 – за формулами опору матеріалів (значення у дужках, Н·м)

Рисунок 14.18 – Епюри згинальних моментів

Як видно з рис. 14.17 і 14.18, переміщення, обчислені за програмою, відрізняються від теоретичних значень всього на 1%, а згинальні моменти – на 2%.

Згинання квадратної пластинки (рис. 14.19 ...14.21).

На рис. 14.19 зображена шарнірно закріплена по контуру квадратна пластинка $1000 \times 1000 \text{ мм}^2$ товщиною 1 мм з рівномірним навантаженням $p=80 \text{ кН/м}^2$.

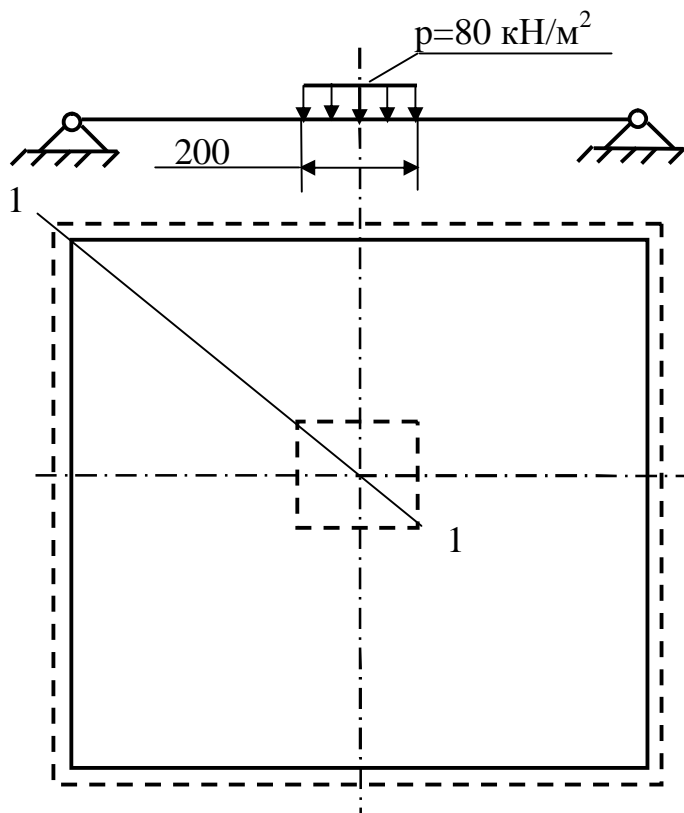
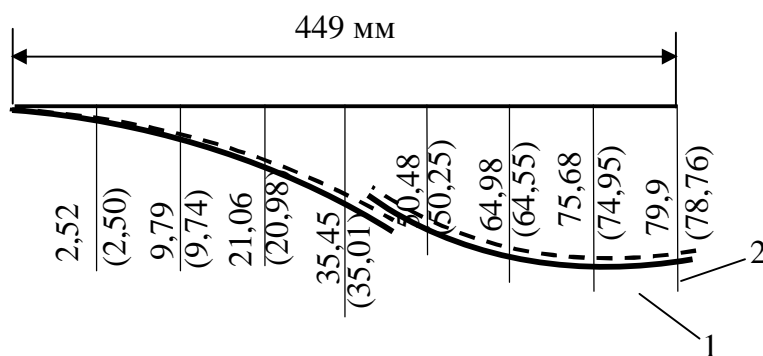


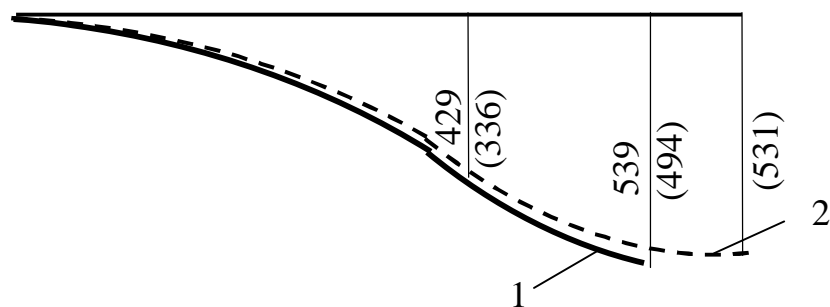
Рисунок 14.19 – Схема квадратної пластинки



крива 1 - за програмою;

крива 2 - за формулами теорії пружності (значення у дужках)

Рисунок 14.20 - Епюра прогинів квадратної пластини подовж перерізу



крива 1 - за програмою; крива 2 - за формулами теорії пружності
(значення у дужках)

Рисунок 14.21 - Епюра інтенсивностей згинальних моментів
у квадратній пластині подовж перерізу 1-1, Н

Циліндрична оболонка (рис. 14.22 ...14.24).

На циліндричну оболонку діаметром 250мм і довжиною 1000мм, яка шарнірно закріплена по краях, діє по центру сила $P=1,25$ кН (рис. 14.22).
Необхідно знайти напруження в перерізі 1-1.

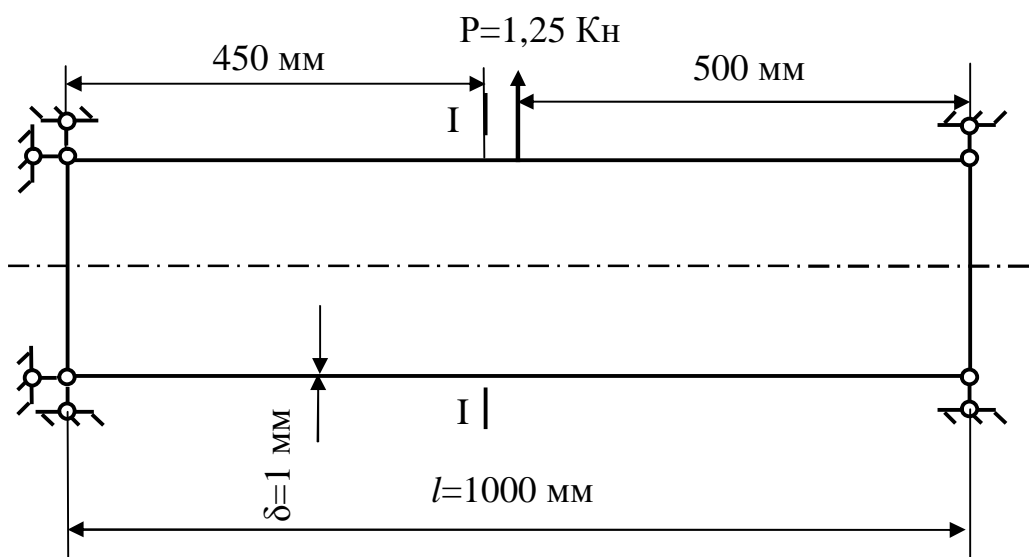


Рисунок 14.22 - Схема навантаження циліндричної оболонки

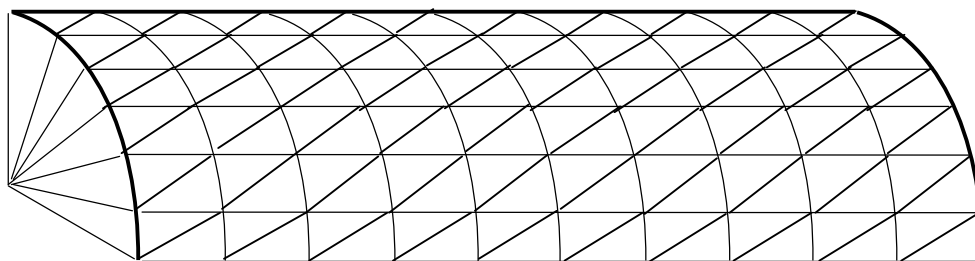
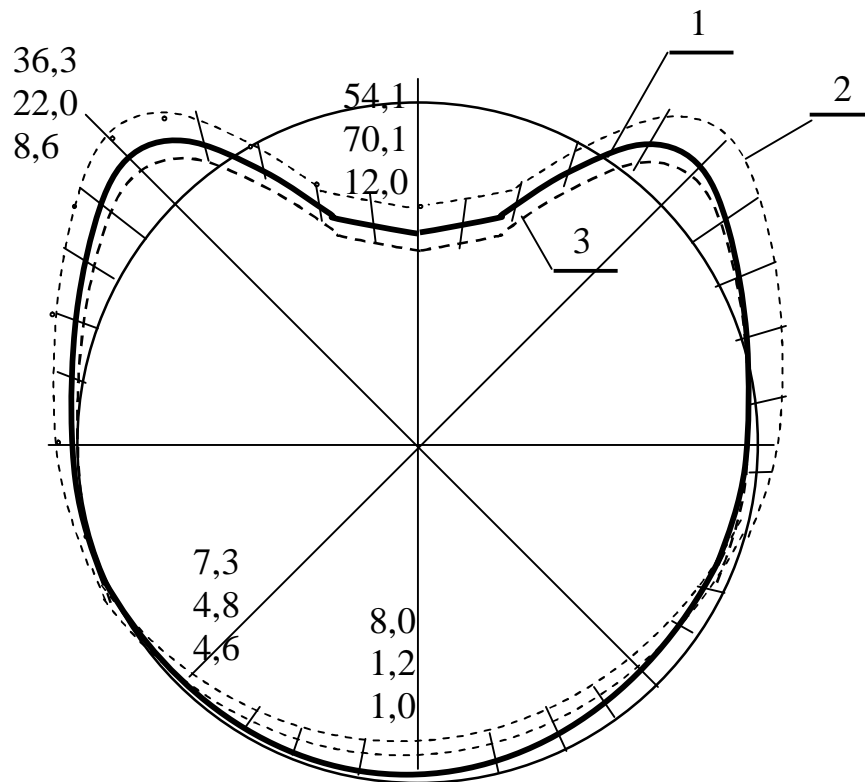


Рисунок - 14.23 Схема розбивання на елементи циліндричної
оболонки



крива 1 - за програмою; крива 2 - рішення В.З.Власова [15];
 крива 3 -експериментальна крива В.З.Власова [15]
 Рисунок 14.24 -Епюри подовжніх нормальних напружень для перерізу 1-1

Як видно з рис. 14.24, характер епюр напружень за програмою, за рішенням і експериментом В.З. Власова однаковий. Але є незначні чисельні розбіжності з рішенням В.З. Власова (див. [15] стр.295). Це пов'язано з тим, що він вів розрахунок за напівмоментною теорією, тоді як за програмою враховувався моментний стан у подовжньому і поперечному напрямках.

Наше рішення ближче підходить до експериментальних значень В.З.Власова (погрішність за максимальним значенням складає всього 2,8%), ніж його теоретичне рішення.

Контрольний приклад підготовки вихідних даних

Необхідно визначити напруження і переміщення циліндрової оболонки, показаної на рис.14.25, яка навантажена двома зосередженими силами. Матеріал: Ст. 3 $E = 2 \cdot 10^5$ МПа .

Розв'язання. Оскільки навантаження кососиметричне щодо горизонтальної площини і симетричне щодо вертикальної площини, то досить розрахувати восьму частину оболонки (рис. 14.26).

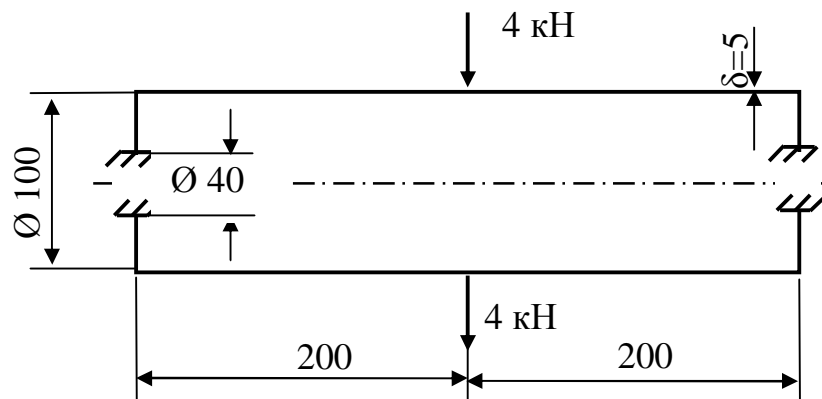


Рисунок - 14.25 Циліндрова оболонка

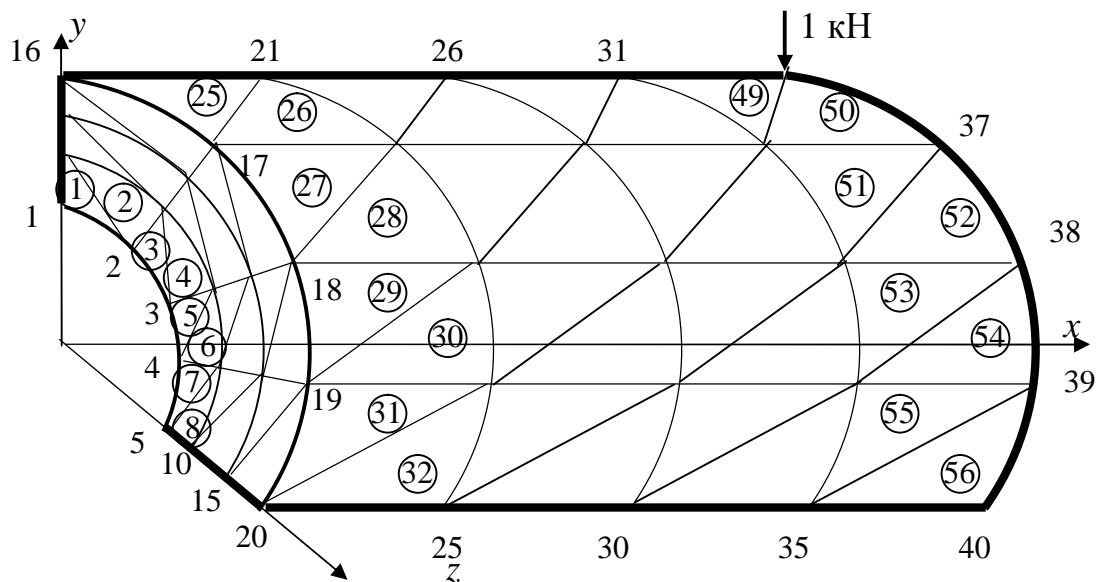


Рисунок 14.26 - Схема розбивання оболонки на елементи

Розіб'ємо її в окружному напрямку на чотири кроки, а уздовж утворюючої першу ділянку (круглу пластину) – на три кроки, а циліндрову частину – на чотири кроки.

У результаті такого розбиття кількість:

- вузлів - $NP = (4 + 1)(3 + 4 + 1) = 40$;

- елементів - $NE = 2 \cdot 4(3 + 4) = 56$;

- закріплених вузлів - $NB = 2[4 + 3 + 4] = 22$.

Ввівши ці дані, одержимо розбивку заданої оболонки на елементи, яка представлена на рис. 14.26.

14.3.3 Висновки з апробації програми

Наведені зіставлення результатів розрахунку напружень і переміщень за програмою і за теоретичними формулами дозволяють стверджувати наступне.

Рішення задач МКЕ за розробленою в роботі програмою дає практично ті ж результати, що і теорія. Але можливість розрахунку при різних конструктивних схемах і навантаженнях за єдиним алгоритмом дає безперечний пріоритет розробленій програмі в порівнянні з різними теоретичними рішеннями.

15 НАПРУЖЕНО-ДЕФОРМОВАНИЙ СТАН ТОНКОСТІННИХ ТІЛ ОБЕРТАННЯ ПРИ ДОВІЛЬНОМУ НАВАНТАЖЕННІ

У даному розділі викладена методика визначення напружено-деформованого стану тонкостінних тіл обертання при довільному навантаженні. В основу методики покладений напіваналітичний метод кінцевих елементів, що дозволяє розмірність задачі визначати тільки геометрією і властивостями матеріалу елементів конструкцій, що розраховуються, незалежно від характеру зміни навантаження в тих напрямках, в яких вказані геометрія і властивості постійні. У пропонованій методиці навантаження в осьовому напрямку, в якому в загальному випадку можуть змінюватися геометрія і властивості матеріалів тіл обертання, задається точково, а в коловому напрямку, в якому геометрія і властивості постійні, – розкладається у ряди Фур'є. Це дозволяє розбиття на кінцеві елементи проводити тільки в осьовому напрямку (кінцеві елементи в загальному випадку є усіченими конусами), тобто звести двомірне завдання розрахунку тіл обертання до одновимірного, що вимагає меншої оперативної пам'яті ЕОМ і різко скорочує час розрахунку. Методика забезпечує високу точність розрахунків завдяки точному врахуванню конструктивного оформлення і навантаження елементів, що розраховуються.

У розділі наведені алгоритм і блок-схеми програми реалізації методики на ПЕОМ. Програма розроблена стосовно розрахунку уніфікованої конструкції барабанів стрічкових конвеєрів, проте, може бути використана для розрахунку будь-яких елементів конструкцій, що є тонкостінними тілами обертання.

У кінці розділу з метою апробації програми наводяться порівняння результатів рішення декількох задач за пропонованою методикою і за існуючими класичними методиками.

15.1 Представлення зовнішнього навантаження

На рис. 15.1 наведено приклад тонкостінної оболонки, що містить сферичні, конічні і циліндрові ділянки, а також вертикальні торцеві стінки, напружено-деформований стан якої може бути досліджений за пропонованою методикою. При цьому навантаження оболонки може бути довільним.

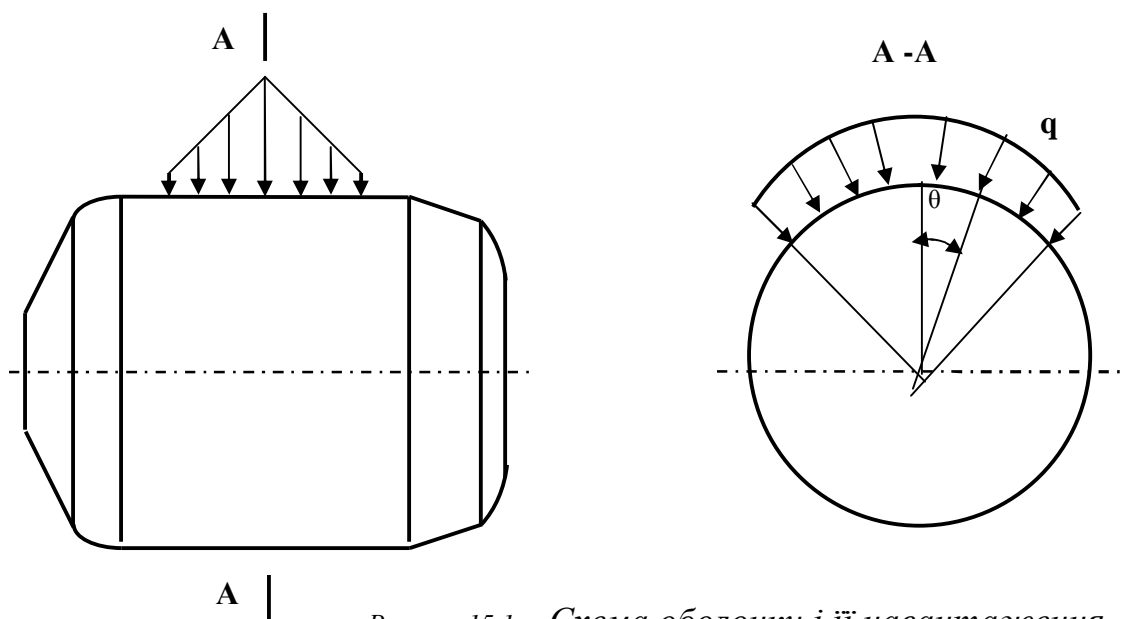


Рисунок 15.1 – Схема оболонки і її навантаження

Навантаження в осьовому напрямку задається точково, а в коловому розкладається у ряд Фур'є.

При дії симетричного навантаження функція $f(q)$ - парна, оскільки $f(q) = f(-q)$. Отже, вона може бути розкладена в неповний ряд Фур'є, складений тільки з одних косинусів [16].

$$f(q) = \frac{1}{2}a_0 + \sum_{k=1}^{\infty} a_k \cos kq,$$

де

$$a_k = \frac{1}{p} \int_{-p}^p f(q) \cos kq dq \quad (k = 0, 1, 2, \dots).$$

При дії кососиметричного навантаження функція $f(q)$ - непарна, отже, вона розкладається у ряд Фур'є за синусами [16].

$$f(q) = \sum_{k=1}^{\infty} b_k \sin kq,$$

де

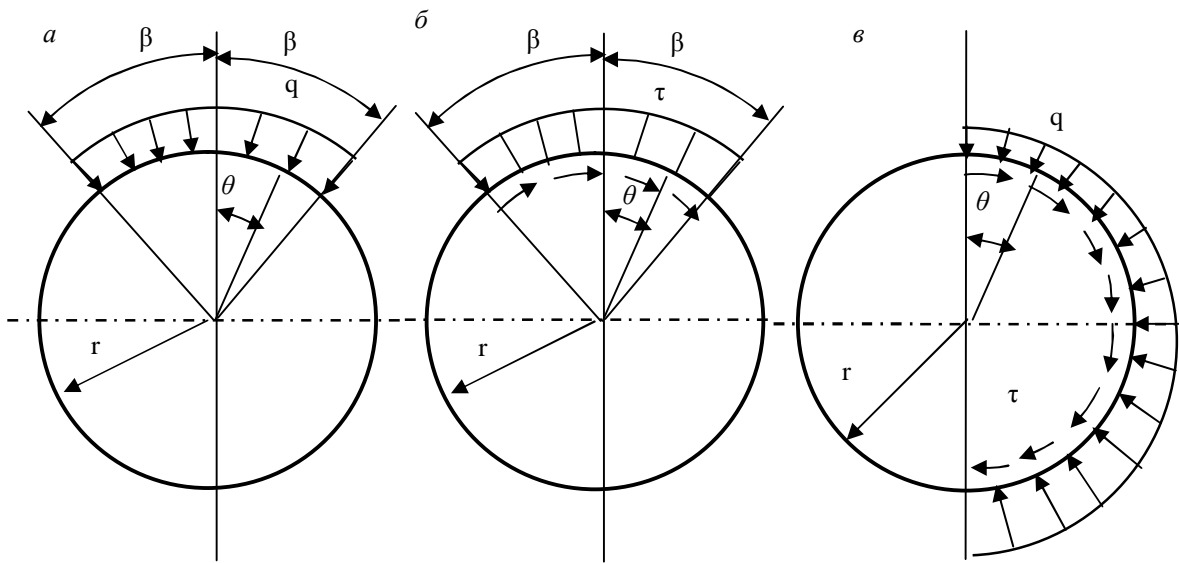
$$b_k = \frac{2}{p} \int_0^p f(q) \sin kq dq \quad (k = 1, 2, \dots).$$

У цьому розділі розглядаються три найбільш характерні варіанти навантаження оболонок барабанів конвеєрів: рівномірно розподілене в коловому напрямку симетричне радіальне навантаження, рівномірно розподілене в коловому напрямку кососиметричне тангенціальне навантаження і навантаження ведучого барабана конвеєрів за законом Ейлера (рис. 15.2).

Для рівномірно розподіленого в межах кута $2b$ симетричного радіального навантаження коефіцієнти ряду Фур'є визначаються залежностями:

$$a_0 = \frac{2}{p} \int_0^p f(q) dq = \frac{2}{p} \int_0^b f(q) dq + \frac{2}{p} \int_b^p f(q) dq = \frac{2}{p} \int_0^b q dq = \frac{2qb}{p};$$

$$a_k = \frac{2}{p} \int_0^p f(q) \cos kq \, dq = \frac{2}{p} \int_0^b f(q) \cos kq \, dq + \frac{2}{p} \int_b^p f(q) \cos kq \, dq = \frac{2q}{pk} \sin kb. \quad (15.1)$$



а - рівномірне розподілене в коловому напрямку симетричне радіальне навантаження; б - рівномірне розподілене в коловому напрямку косиметричне тангенціальне навантаження; в - навантаження ведучого барабана за законом Ейлера

Рисунок 15.2 - Схема навантаження оболонок барабанів конвеєрів

При цьому для даного навантаження

$$f(q) = \frac{qb}{p} + \frac{2q}{pk} \sum_{k=1}^{\infty} \sin kb \cos kq. \quad (15.2)$$

Для отримання коефіцієнтів ряду Фур'є від косиметричного тангенціального навантаження в залежностях (15.1) інтенсивність радіального навантаження q необхідно замінити на τ . При цьому для вказаного тангенціального навантаження

$$f(q) = \frac{tb}{p} + \frac{2t}{p} \sum_{k=1}^{\infty} \frac{1}{k} \sin kb \cos kq. \quad (15.3)$$

Аналогічно в неповний ряд Фур'є, складений з одних косинусів, може бути розкладене будь-яке інше симетричне радіальне або

кососиметричне тангенціальне навантаження.

Розглянемо навантаження барабана конвеєра за законом Ейлера (рис. 15.3).

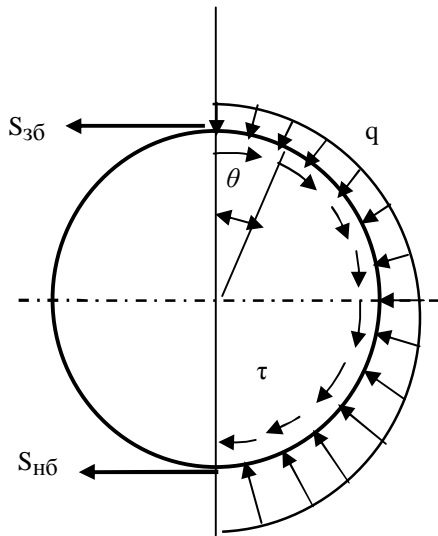


Рисунок 15.3— Схема навантаження барабана конвеєра за законом Ейлера

$$S = S_{зб} e^{q\theta};$$

$$q = \frac{S}{r} = \frac{S_{зб} e^{q\theta}}{r}; \quad t = qf = \frac{f \cdot S_{зб} e^{q\theta}}{r}.$$

Тут $S_{зб}$, $S_{нб}$ і S – відповідно зусилля збігаюче, набігаюче і при довільному куті θ .

Як правило, як початкові параметри задають $S_{\max} = S_{нб} + S_{зб}$ і крутний момент

$$M_{кр} = (S_{нб} - S_{зб})r.$$

Виразимо радіальне і тангенціальне погонні навантаження, а також коефіцієнт тертя f через S_{\max} і

$M_{кр}$.

$$q = q_0 e^{qf}; \quad t = q_0 f e^{qf};$$

$$q_0 = \frac{S_{\max} r - M_{кр}}{2r^2 B}; \quad f = \frac{1}{p} \ln C. \quad (15.4)$$

Тут
$$C = \frac{S_{\max} r + M_{кр}}{S_{\max} r - M_{кр}};$$

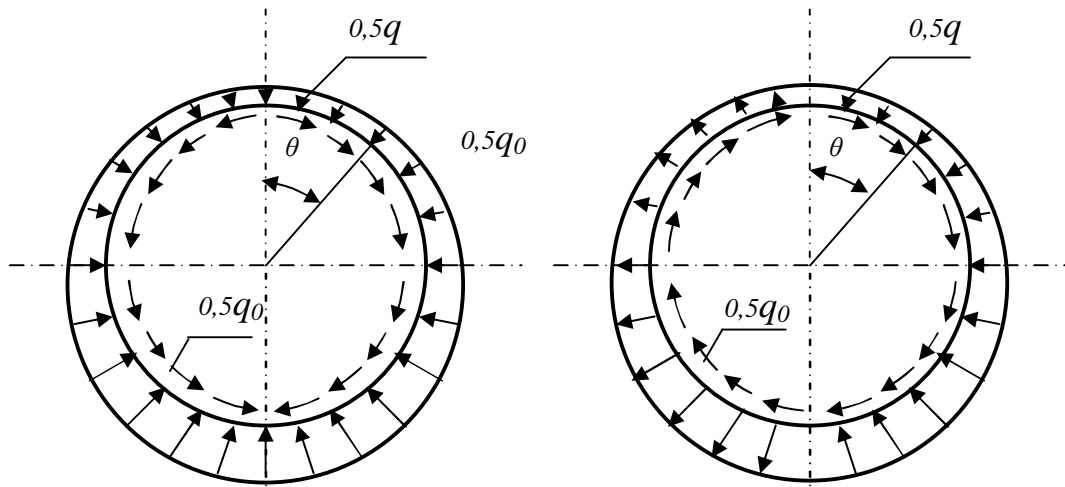
B - ширина стрічки конвеєра.

Вантаження барабана за законом Ейлера (див. рис. 15.2,в) розіб'ємо на симетричне і кососиметричне навантаження (рис. 15.4).

Коефіцієнти ряду Фур'є для симетричного вантаження:

а) радіальне навантаження -

$$a_0 = \frac{2}{p} \int_0^p f(q) dq = \frac{2}{p} \int_0^p \frac{1}{2} q dq = \frac{1}{p} \int_0^p q_0 e^{qf} dq = \frac{q_0}{\ln C} [C - 1];$$



a - симетричне навантаження; *б* - косиметричне навантаження
Рисунок 15.4 – Розкладання навантаження барабана за законом Ейлера на симетричне і косиметричне

$$a_k = \frac{2}{p} \int_0^p f(q) \cos kq dq = \frac{2}{p} \int_0^p \frac{q_0 e^{qf}}{2} \cos kq dq = \frac{q_0}{p} \frac{\ln C}{k^2 p \left(1 + \frac{f^2}{k^2}\right)} [C(-1)^k - 1].$$

При цьому

$$f(q) = \frac{q_0}{2 \ln C} (C - 1) + \frac{q_0}{p} \sum_{k=1}^{\infty} \frac{\ln C}{k^2 p \left(1 + \frac{f^2}{k^2}\right)} [C(-1)^k - 1] \cos kq. \quad (15.5)$$

б) колове навантаження -

$$a_k = 0;$$

$$b_k = \frac{2}{p} \int_0^p f(q) \sin kq dq = \frac{2}{p} \int_0^p \frac{q_0}{2} f e^{qf} \sin kq dq = -\frac{q_0 f}{p} \cdot \frac{[C(-1)^k - 1]}{k \left[1 + \frac{f^2}{k^2}\right]};$$

$$f(q) = -\frac{q_0 f}{p} \sum_{k=1}^{\infty} \frac{C(-1)^k - 1}{k \left[1 + \frac{f^2}{k^2} \right]} \sin kq. \quad (15.6)$$

Коефіцієнти ряду Фур'є для кососиметричного навантаження:

а) радіальне навантаження -

$$a_k = 0;$$

$$b_k = \frac{2}{p} \int_0^p f(q) \sin kq \, dq = \frac{2}{p} \int_0^p \frac{q_0}{2} e^{qf} \sin kq \, dq = -\frac{q_0}{p} \cdot \frac{[C(-1)^k - 1]}{k \left[1 + \frac{\ln^2 C}{k^2 p^2} \right]};$$

$$f(q) = -\frac{q_0}{p} \sum_{k=1}^{\infty} \frac{C(-1)^k - 1}{k \left[1 + \frac{\ln^2 C}{k^2 p^2} \right]} \sin kq. \quad (15.7)$$

б) колове навантаження -

$$a_0 = \frac{2}{p} \int_0^p f(q) \, dq = \frac{2}{p} \int_0^p \frac{1}{2} q_0 f e^{qf} \, dq = \frac{q_0 f}{\ln C} [C - 1];$$

$$a_k = \frac{2}{p} \int_0^p f(q) \cos kq \, dq = \frac{2}{p} \int_0^p q_0 f e^{qf} \cos kq \, dq = \frac{q_0 f \ln C}{p^2} \frac{[C(-1)^k - 1]}{k^2 \left[1 + \frac{\ln^2 C}{p^2 k^2} \right]};$$

$$f(q) = \frac{q_0 f}{2 \ln C} [C - 1] + \frac{q_0 f \ln C}{p^2} \sum_{k=1}^{\infty} \frac{C(-1)^k - 1}{k^2 \left[1 + \frac{\ln^2 C}{p^2 k^2} \right]} \cos kq. \quad (15.8)$$

15.2 Формування матриць жорсткості кінцевого елемента

Розбиття оболонки на кінцеві елементи проводитимемо тільки в осьовому напрямку. При такому розбитті кінцеві елементи у загальному випадку будуть усіченими конусами (рис.15.5). Вважатимемо, що виділені кінцеві елементи взаємодіють тільки у вузлах, які в даному випадку є

колами. Розрахунок вестимемо окремо для кожної з гармонік рядів Фур'є, а результати підсумовувати. При цьому переміщення точок серединної поверхні елемента в будь-якому його поперечному перерізі (зокрема вузловому) підкорятимуться залежностям, відповідним закону кожної гармоніки.

При симетричному навантаженні:

$$\begin{aligned} u(q) &= u^n \cos kq; \\ w(q) &= w^n \cos kq; \\ v(q) &= v^n \sin kq. \end{aligned} \tag{15.9}$$

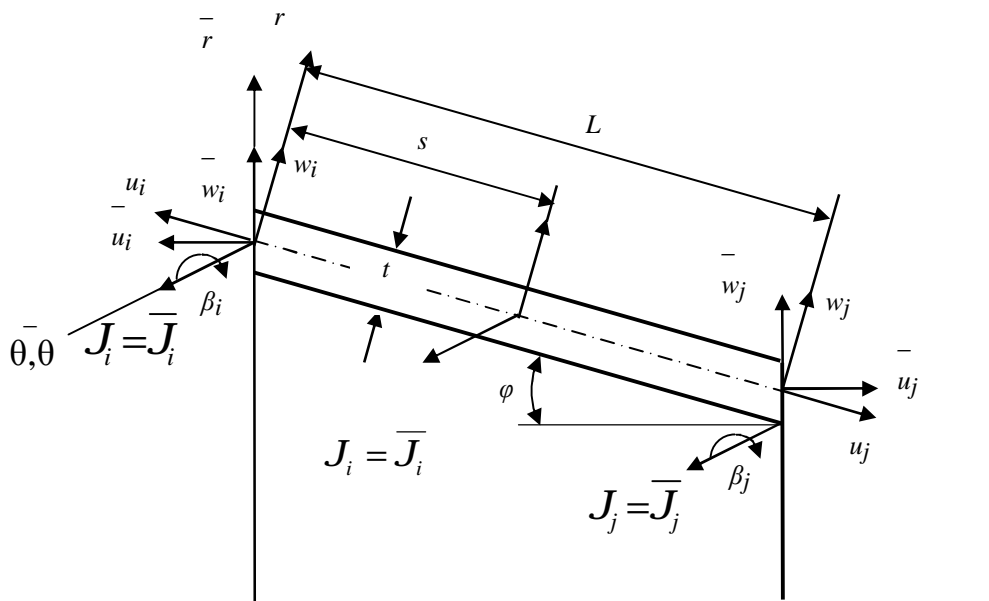


Рисунок 15.5 – Схема кінцевого елемента

При кососиметричному навантаженні:

$$\begin{aligned} u(q) &= u^n \sin kq; \\ w(q) &= w^n \sin kq; \\ v(q) &= v^n \cos kq, \end{aligned} \tag{15.10}$$

де u^n , w^n і v^n – амплітудні значення переміщень для даного поперечного перерізу оболонки.

Переміщення, що входять до залежностей (15.9) і (15.10), визначені щодо місцевої (локальної) системи координат, пов'язаної з даним кінцевим елементом (див. рис. 15.5). При цьому вісь s направлена уздовж утворюючої конуса, осі r і q - уздовж нормалі і уздовж дотичної до його серединної поверхні. Можливі зсуви вузлів i і j даного кінцевого елемента подамо трьома лінійними переміщеннями - u , w , v і одним кутовим - кутом повороту навколо осі q :

$$b = \frac{dw}{ds}.$$

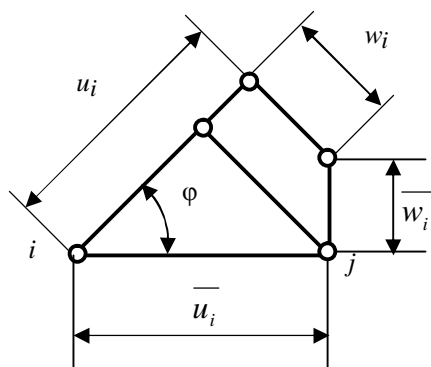
Амплітудні значення цих переміщень назовемо вузловими переміщеннями і подамо у вигляді матриці-стовпця $\{d^a\}$.

$$\{d^n\} = \{u_i^n, w_i^n, b_i^n, v_i^n, u_j^n, w_j^n, b_j^n, v_j^n\}^T \quad (15.11)$$

- вузлові переміщення кінцевого елемента в локальній системі координат, де

$$b^n = \frac{dw^n}{ds}. \quad (15.12)$$

При сумісному розгляді системи кінцевих елементів зручно використовувати переміщення не щодо локальної, а щодо глобальної системи координат, пов'язаної з оболонкою в цілому. Вісь \bar{s} глобальної системи направимо паралельно осі оболонки, вісь \bar{r} - їй перпендикулярно, а вісь \bar{q} глобальної системи сумістимо з віссю q локальної. Переміщення щодо глобальної системи позначимо тими ж



буквами, що і до локальної, але з верхньою рискою (див. рис. 15.5).

Виразимо переміщення в локальній системі через переміщення в глобальній системі (рис. 15.6).

$$u_i = \bar{u}_i \cos j + \bar{w}_i \sin j ;$$

$$w_i = -\bar{u}_i \sin j + \bar{w}_i \cos j .$$

Рисунок 15.6 – Схема перетворення переміщень з глобальної системи

в локальну

Аналогічно:

$$\begin{aligned} u_j &= \bar{u}_j \cos j + \bar{w}_j \sin j ; \\ w_j &= -\bar{u}_j \sin j + \bar{w}_j \cos j . \end{aligned}$$

Відповідний зв'язок вузлових переміщень щодо локальної і глобальної систем координат в матричній формі можна подати у вигляді

$$\{d^n\} = [I] \{\bar{d}^n\}, \quad (15.13)$$

де $\{\bar{d}^n\}$ – матриця-стовпець вузлових переміщень кінцевого елемента в глобальній системі координат.

$$\{\bar{d}^n\}^T = \{\bar{u}_i, \bar{w}_i, \bar{b}_i, \bar{v}_i, \bar{u}_j, \bar{w}_j, \bar{b}_j, \bar{v}_j\}. \quad (15.14)$$

$$[I] = \begin{pmatrix} \cos j & \sin j & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sin j & \cos j & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos j & \sin j & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sin j & \cos j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (15.15)$$

Амплітудні переміщення точок довільного поперечного перерізу даного кінцевого елемента виразимо через його вузлові переміщення за допомогою спеціальних функцій положення, які візьмемо у вигляді [6]:

$$\begin{aligned} u^a &= a_1 + a_2 s; \\ w^a &= a_3 + a_4 s + a_5 s^2 + a_6 s^3; \\ v^a &= a_7 + a_8 s. \end{aligned} \quad (15.16)$$

де s – координата даного поперечного перерізу елемента в локальній системі координат (див. рис. 15.5).

Підставивши у вирази (15.16) і (15.12) координати вузлів $s_i = 0$ і

$s_i = L$, одержимо вузлові переміщення $\{d^n\}$.

Звідси коефіцієнти, виражені через вузлові переміщення:

$$a_1 = u_i^n; \quad a_2 = \frac{u_j^n - u_i^n}{L};$$

$$a_3 = w_i^n; \quad a_4 = \left(\frac{dw^n}{ds} \right)_i;$$

$$a_5 = \frac{3}{L^2} (w_i^n - w_j^n) - \frac{1}{L} \left[\left(\frac{dw^n}{ds} \right)_j + 2 \left(\frac{dw^n}{ds} \right)_i \right];$$

$$a_6 = \frac{2}{L^3} (w_i^n - w_j^n) + \frac{1}{L^2} \left[\left(\frac{dw^n}{ds} \right)_j + \left(\frac{dw^n}{ds} \right)_i \right];$$

$$a_7 = v_i^n; \quad a_8 = \frac{v_j^{an} - v_i^n}{L}.$$

При цьому залежність амплітудних переміщень в довільному поперечному перерізі елемента від вузлових переміщень в матричній формі має вигляд:

$$\{f^a\} = \begin{Bmatrix} u^n \\ w^n \\ v^n \end{Bmatrix} = [N][I]\{d^n\} \quad (15.17)$$

де $[N]$ – матриця функцій положення.

Матриця $[N]$ може бути подана у вигляді:

$$[N] = \begin{vmatrix} 1-s' & 0 & 0 & 0 & s' & 0 & 0 & 0 \\ 0 & 1-3s'^2 + 2s'^3 & L(s' - 2s'^2 + s'^3) & 0 & 0 & 3s'^2 - 2s'^3 & L(-s'^2 + s'^3) & 0 \\ 0 & 0 & 0 & 1-s' & 0 & 0 & 0 & s' \end{vmatrix} \quad (15.18)$$

де s' – безрозмірна координата даного поперечного перерізу елемента $s' = s/L$.

Для тонкостінного тіла обертання при довільному вантаженні компоненти деформації серединної поверхні можуть бути визначені через переміщення за наступними залежностями теорії пружності [6]:

$$\{e\} = \begin{Bmatrix} e_s \\ e_q \\ g_{sq} \\ C_s \\ C_q \\ C_{sq} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial s} \\ \frac{1}{r} \cdot \frac{\partial v}{\partial q} + \frac{1}{r} (w \cos j + u \sin j) \\ \frac{1}{r} \cdot \frac{\partial u}{\partial q} + \frac{\partial v}{\partial s} - v \sin j \frac{1}{r} \\ -\frac{\partial^2 w}{\partial s^2} \\ -\frac{1}{r^2} \cdot \frac{\partial^2 w}{\partial q^2} + \frac{\partial v}{\partial q} \cdot \frac{\cos j}{r^2} - \frac{\sin j}{r} \cdot \frac{\partial w}{\partial s} \\ 2 \left(-\frac{1}{r} \cdot \frac{\partial^2 w}{\partial s \partial q} + \frac{\sin j}{r^2} \cdot \frac{\partial w}{\partial q} + \frac{\cos j}{r} \cdot \frac{\partial v}{\partial s} - \frac{\sin j \cdot \cos j}{r^2} v \right) \end{Bmatrix}, \quad (15.19)$$

де e_s, e_q – лінійні відносні деформації уздовж осей s і q ;

g_{sq} – кут зсуву в площині sq ;

C_s – зміна кривизни елемента у напрямку осі s ;

C_q – зміна кривизни елемента у напрямку осі q ;

C_{sq} – відносне кручення серединної поверхні елемента.

e_s, e_q і g_{sq} є мембранними деформаціями серединної поверхні елемента, а деформації, C_q і C_{sq} пов'язані з її згинанням і крученням.

Підставивши в (15.19) значення переміщень з (15.17) в матричному вигляді одержимо:

$$\{e\} = [B] \{d^n\} = [B][I] \{\bar{d}^n\}. \quad (15.20)$$

Члени матриці $[B]$ для симетричного вантаження поміщені в табл.15.1.

За відомими деформаціями $\{e\}$ (вважаючи, що початкові деформації

відсутні) можна знайти напруження в будь-якій точці елемента:

$$\{d\} = [D]\{e\} = [D][B]\{d^n\} = [D][B][I]\{d^n\}, \quad (15.21)$$

де $[D]$ – матриця пружності елемента.

Якщо під $\{S\}$ розуміти не напруження, а внутрішні зусилля, що доводяться на одиницю довжини оболонки в осьовому або окружному напрямках, матриці $\{S\}$ і $[D]$ для тонкостінного тіла обертання набувають вигляду:

$$\{S\} = \begin{Bmatrix} N_s \\ N_q \\ N_{sq} \\ M_s \\ M_q \\ M_{sq} \end{Bmatrix}; \quad (15.22)$$

$$[D] = \frac{Et}{1-m^2} \begin{vmatrix} 1 & m & 0 & 0 & 0 & 0 \\ m & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1-m}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{t^2}{12} & \frac{mt^2}{12} & 0 \\ 0 & 0 & 0 & \frac{mt^2}{12} & \frac{t^2}{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{t^2(1-m)}{24} \end{vmatrix}, \quad (15.23)$$

де E і m – модуль пружності при розтяганні і коефіцієнт Пуассона матеріалу даного елемента;

t – товщина елемента.

Внутрішні зусилля, віднесені до локальної системи координат, показані на рис. 15.7.

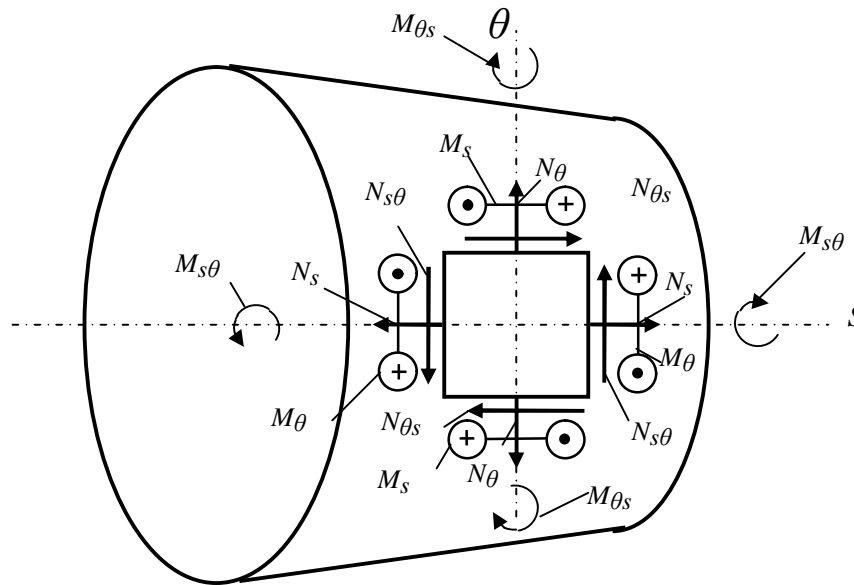


Рисунок 15.7 – Схема внутрішніх зусиль, що діють на елемент оболонки

N_s – нормальне зусилля уздовж осі S , що припадає на одиницю довжини кола елемента;

N_q – нормальне зусилля уздовж осі θ , що припадає на одиницю довжини утворюючої елемента;

N_{sq} – сила, яка лежить в площині, перпендикулярній осі s , і паралельна осі θ , така, що припадає на одиницю довжини кола елемента;

M_s – момент в площині, паралельній осі s (навколо осі q), що припадає на одиницю довжини кола елемента;

M_q – момент в площині, паралельній осі q (навколо осі s), що доводиться на одиницю довжини утворюючої елемента;

M_{sq} – крутний момент, в площині, перпендикулярній осі s і паралельній осі θ , що припадає на одиницю довжини кола елемента.

Зусилля N_q і N_{sq} є мембранними, а зусилля, M_q і M_{sq} – згинальними і крутними.

У вузлах кінцевих елементів введемо фіктивні зосереджені вузлові сили $\{F\}$, які діють в напрямках вузлових переміщень і еквівалентні напруженням, розподіленим за кільцевими межами елементів:

$$\{F\} = \begin{Bmatrix} F_{ui} \\ F_{wi} \\ M_{bi} \\ F_{vi} \\ F_{uj} \\ F_{wj} \\ M_{bj} \\ F_{vj} \end{Bmatrix}. \quad (15.24)$$

З метою зробити фіктивні вузлові сили статично еквівалентними граничним напруженням задамо нескінченно малі віртуальні (довільні) вузлові переміщення і прирівняємо зовнішню роботу фіктивних сил на даних переміщеннях і внутрішню роботу напружень, які виникають в даному елементі в результаті цих переміщень.

Хай $d\{\bar{d}^n\}$ – матриця віртуальних вузлових переміщень даного елемента. Викликані ними прирости амплітудних переміщень і деформацій у середині елемента згідно з формулами (15.17) і (15.20):

$$\begin{aligned} d\{f^n\} &= [N][I]d\{\bar{d}^n\}; \\ d\{e\} &= [B][I]d\{\bar{d}^n\}. \end{aligned}$$

Робота, що здійснюється вузловими силами, дорівнює сумі добутків кожної сили на відповідне переміщення, тобто в матричному вигляді:

$$dA_F = (d\{\bar{d}^n\})^T \{F\},$$

де $(d\{\bar{d}^n\})^T$ – матриця, транспонована матриці $d\{\bar{d}^a\}$, тобто матриця-стовпець, записана у вигляді матриці-рядка.

Внутрішня робота напруження

$$\begin{aligned} dA_s &= \int_F (d\{e\})^T \{s\} dF = \int_F ([B][I]d\{\bar{d}^n\})^T \{s\} dF = \int_F ([I]d\{\bar{d}^n\})^T [B]^T \{s\} dF = \\ &= (d\{\bar{d}^n\})^T \int_F [I]^T [B]^T \{s\} dF = (d\{\bar{d}^n\})^T \int_F [I]^T [B]^T [D][B][I]d\{\bar{d}^n\} dF. \end{aligned}$$

Прирівняємо одержані роботи

$$dA_F = dA_S.$$

Звідси

$$\{F\} = [k]\{\bar{d}^n\}, \quad (15.25)$$

де

$$[k] = \int_F [I]^T [B]^T [D][B][I] dF \quad \text{– матриця жорсткості елемента.} \quad (15.26)$$

$$[k] = \int_0^{2p} \int_0^L [I]^T [B]^T [D][B][I] r dq dS. \quad (15.27)$$

Визначимо співмножники матриці, які дає інтегрування за q . Для цього з добутку матриць $[B]^T [D][B]$ випишемо члени, що містять параметр θ , заздалегідь виписавши аналогічні члени з матриць $[B][D]$ і $[B]$.

Для симетричного навантаження вказані матриці мають вигляд:

$$[B] = \begin{bmatrix} \cos kq & 0 & 0 & 0 & \cos kq & 0 & 0 & 0 \\ \cos kq & \cos kq & \cos kq & \cos kq & \cos kq & \cos kq & \cos kq & \cos kq \\ \sin kq & 0 & 0 & \sin kq & \sin kq & 0 & 0 & \sin kq \\ 0 & \cos kq & \cos kq & 0 & 0 & \cos kq & \cos kq & 0 \\ 0 & \cos kq & \cos kq & \cos kq & 0 & \cos kq & \cos kq & \cos kq \\ 0 & \sin kq & \sin kq & \sin kq & 0 & \sin kq & \sin kq & \sin kq \end{bmatrix};$$

$$\begin{aligned}
[B]^T &\Rightarrow \begin{vmatrix} \cos kq & \cos kq & \sin kq & 0 & 0 & 0 \\ 0 & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ 0 & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ 0 & \cos kq & \sin kq & 0 & \cos kq & \sin kq \\ \cos kq & \cos kq & \sin kq & 0 & 0 & 0 \\ 0 & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ 0 & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ 0 & \cos kq & \sin kq & 0 & \cos kq & \sin kq \end{vmatrix} ; \\
|B^T| |D| &\Rightarrow \begin{vmatrix} \cos kq & \cos kq & \sin kq & 0 & 0 & 0 \\ \cos kq & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ \cos kq & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ \cos kq & \cos kq & \sin kq & \cos kq & \cos kq & \sin kq \\ \cos kq & \cos kq & \sin kq & 0 & 0 & 0 \\ \cos kq & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ \cos kq & \cos kq & 0 & \cos kq & \cos kq & \sin kq \\ \cos kq & \cos kq & \sin kq & \cos kq & \cos kq & \sin kq \end{vmatrix} .
\end{aligned}$$

Таким чином, усі члени добутку матриць $[B]^T [D][B]$ для симетричного навантаження (див. стор. 188) є сумою додатків, що містять $\sin^2 kq$ або $\cos^2 kq$.

При $k = 0$

$$\int_0^{2p} \cos^2 kq dq = 2p ; \quad \int_0^{2p} \sin^2 kq dq = 0 .$$

При $k = 1, 2, 3, \dots$

$$\int_0^{2p} \cos^2 kq dq = p ; \quad \int_0^{2p} \sin^2 kq dq = p .$$

На основі даних результатів при визначенні членів матриці $[k]$ за формулою (15.27) можна проводити інтегрування не за площею оболонки, а за довжиною її утворюючої, а потім - набуті значення членів матриці $[k]$ для нульової гармоніки домножити на $2p$, а для решти гармонік на p .

При цьому:

для $k = 0$

$$[k] = 2p \int_0^L [I]^T [B]^T [D] [B] [I] r ds; \quad (15.28)$$

для $k = 1, 2, 3, \dots$

$$[k] = p \int_0^L [I]^T [B]^T [D] [B] [I] r ds, \quad (15.29)$$

де в матрицях $[B]^T$ і $[B]$ опущені множники $\sin kq$ і $\cos kq$. При цьому в матриці $[B]$ з опущеними $\sin kq$ і $\cos kq$ члени, які їх містили, для нульової гармоніки ($k = 0$) мають бути прийнятими нулю.

Для отримання за формулою (15.28) або (15.29) матриці жорсткості $[k]$ елемента використовуємо чисельне інтегрування за формулою Сімпсона [8]

$$\int_0^L f(s) ds = \frac{L}{3n} \left[f(s_0) + 4f(s_1) + 2f(s_2) + 4f(s_3) + \dots + 2f(s_{n-2}) + 4f(s_{n-1}) + f(s_n) \right], \quad (15.30)$$

де

$$f(S) = [I]^T [B]^T [D] [B] [I] r.$$

Оскільки функції положення $[N]$ – (15.18), що входять до $[B]^T$ і $[B]$, виражені через безрозмірну ординату, інтегрування проводитимемо в ординатах S' . Кількість інтервалів розбиття візьмемо $n = 10$. При цьому вираз (15.30) набуде вигляду:

$$\int_0^1 f(s') L ds' = \frac{L}{3 \cdot 10} \left[f(0) + 4f(0,1) + 2f(0,2) + 4f(0,3) + \dots + 2f(0,8) + 4f(0,9) + f(1) \right]. \quad (15.31)$$

Для середнього перерізу елемента (при $s' = 0,5$) знаходимо допоміжну матрицю

$$[A1] = [D] [B] [I]. \quad (15.32)$$

Матриця $[k]$ – це квадратна матриця, члени якої є фіктивними внутрішніми зусиллями у вузлах даного елемента у напрямку кожного з його амплітудних переміщень від кожного з цих переміщень, яке дорівнює одиниці.

Для кососиметричного навантаження в матрицях $[B]$, $[B]^T [D]$ і $[B]^T [D] [B]$ необхідно замінити $\cos kq$ на $\sin kq$, $\sin kq$ на $\cos kq$ і в матриці $[B]$ змінити на протилежні знаки коефіцієнтів, що містять перші добутки за θ . Члени матриці $[B]$ для кососиметричного вантаження поміщені в табл. 15.2.

15.3 Формування матриці жорсткості конструкції і системи рівнянь рівноваги вузлів

Розв'язуючою для даної задачі за визначенням напружено-деформованого стану тонкостінних тіл обертання буде система рівнянь рівноваги всіх вузлів конструкції, яка записана у вигляді рівності внутрішніх фіктивних вузлових сил, що діють у напрямку вузлових переміщень умовним вузловим силам $\{R\}$ від зовнішнього навантаження, що діють в тих напрямках:

$$\left\{ \begin{array}{c} P_{u1} \\ P_{w1} \\ P_{b1} \\ P_{v1} \\ \\ P_{um} \\ P_{wm} \\ P_{bm} \\ P_{vm} \end{array} \right\} = \left\{ \begin{array}{c} R_{u1} \\ R_{w1} \\ R_{b1} \\ R_{v1} \\ \\ R_{um} \\ R_{wm} \\ R_{bm} \\ R_{vm} \end{array} \right\}, \quad (15.33)$$

де m – кількість вузлів.

Вираз (15.25) дає внутрішні фіктивні вузлові сили для кожного з елементів, які розглядаються окремо. Сили $\{P\}$ для конструкції в цілому є сумарними фіктивними вузловими силами. При цьому сили $\{P\}$ для вузла, що входить до двох сусідніх елементів повинні дорівнювати сумі відповідних сил $\{F\}$ кожного з цих елементів. Сили $\{P\}$ можуть бути одержані як добутки матриці жорсткості конструкції в цілому $[A]$ на матрицю - стовпець вузлових переміщень всіх вузлів.

$$\{P\} = [A] \{d^n\}^k, \quad (15.34)$$

де

$$\{d_a\}^k = \{\bar{u}_i, \bar{w}_i, \bar{b}_i, \bar{v}_i, \dots, \bar{u}_n, \bar{w}_n, \bar{b}_n, \bar{v}_n\}^T. \quad (15.35)$$

Матриця жорсткості конструкції – це квадратна матриця розміром $4n \times 4n$, члени якої є фіктивними внутрішніми зусиллями в кожному вузлі i у напрямку кожного з його амплітудних переміщень від кожного з $4n$ одиничних амплітудних переміщень всіх вузлів. При цьому ненульовими будуть тільки коефіцієнти, що є фіктивними силами в кожному з вузлів i (рис. 15.8) від одиничних переміщень цього ж вузла і двох інших вузлів $(i-1)$ і $(i+1)$ що належать елементам, які з'єднуються вузлом i .

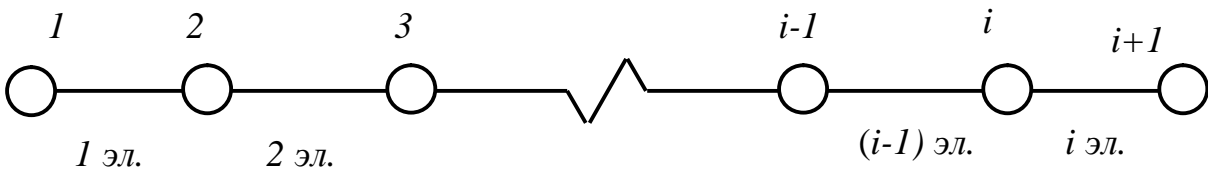


Рисунок 15.8 – Схема до визначення коефіцієнтів матриці жорсткості конструкції

Фіктивні сили у вузлі i від переміщень двох сусідніх вузлів $(i-1)$ і $(i+1)$ будуть дорівнювати коефіцієнтам матриць жорсткості $[k]$ елементів, які з'єднуються, $(i-1)$ і $(i+1)$, що є відповідними фіктивними силами для кожного з цих елементів, які розглядаються окремо. Фіктивні сили у вузлі i від одиничних переміщень самого вузла i будуть дорівнювати сумі відповідних коефіцієнтів матриць $[k]$ елементів $(i-1)$ і i , що є фіктивними силами у вузлі i від одиничних переміщень для кожного з цих елементів. Принцип утворення матриці жорсткості конструкції $[A]$ за матрицями жорсткості елементів $[k]$ показаний нижче на прикладі конструкції, що складається з двох елементів.

У програмі рішення задачі на ПЕОМ організований цикл за елементами. У процесі його формуються матриці жорсткості всіх

елементів з одночасним засиланням коефіцієнтів цих матриць в елементи пам'яті машини, відведені під відповідні коефіцієнти матриці жорсткості конструкції в цілому $[A]$. При цьому через симетрію матриці $[A]$ визначаються тільки коефіцієнти, розташовані на головній діагоналі матриці і над нею.

З метою зменшення ширини машинної матриці, відповідній матриці жорсткості конструкції, матриця виходить з верхньої кутової половини $[A]$ зсувом всіх рядків, окрім верхнього, вліво так, що клітки головної діагоналі потрапляють в клітки першого стовпця.

Умовні вузлові сили $\{R\}$ від зовнішнього навантаження для кожної даної гармоніки еквівалентні дійсному навантаженню, розподіленому колом вузла за законом даної гармоніки. Визначимо їх з умови рівності елементарних робіт умовного і дійсного навантажень на відповідних віртуальних переміщеннях.

Для симетричного радіального навантаження, рівномірно розподіленого в межах кута $2b$ (див. рис.15.2,а), $R_{ui} = R_{bi} = R_{vi} = 0$ ($i = 1, 2, \dots, n$), а R_{wi} для кожної гармоніки повинна бути еквівалентна навантаженню, що є відповідним членом ряду (15.1)

$$\text{При } k = 0 \quad R_{wi} dw_i = \int_0^{2p} \frac{qb}{p} dw_i \cos kqr dq,$$

де dw_i – віртуальне вузлове (амплітудне) переміщення у напрямку дії умовної сили R_{wi} , r – радіус кола вузла i .

Звідси

$$R_{wi} = \int_0^{2p} \frac{qb}{p} rdq = \frac{qb}{p} 2p r. \quad (15.36)$$

При $k = 1, 2, \dots$

$$R_{wi} = \int_0^{2p} \frac{2q}{pk} \sin kb \cos kq \cdot dw_i \cdot \cos kq \cdot rdq.$$

Звідси

$$R_{wi} = \frac{2q}{pk} \sin kb \cdot r \int_0^{2p} \cos^2 kq dq = \frac{2q}{pk} \sin kb \cdot pr. \quad (15.37)$$

Таким чином, умовні вузлові сили від вказаного зовнішнього радіального навантаження для кожної гармоніки рівні відповідним членам ряду (1.2), помноженим на $2p r$ при $k = 0$ і на $p r$ при $k = 1, 2, \dots$

Для тангенціального кососиметричного навантаження, рівномірно розподіленого в межах кута $2b$ (див. рис.15.2,б) $R_{ui} = R_{bi} = R_{wi} = 0$, а

значення R_{vi} визначається залежностями:

-при $k = 0$

$$R_{vi} = \frac{tb}{p} 2p r ; \quad (15.38)$$

$$\text{при } k = 1, 2, \dots \quad R_{vi} = \frac{2t}{pk} \sin kb \cdot pr , \quad (15.39)$$

тобто умовні вузлові сили від даного тангенціального навантаження для кожної гармоніки рівні відповідним членам ряду (15.3), який домножений на $2p r$ при $k = 0$ і на $p r$ при $k = 1, 2, \dots$

Аналогічно можуть бути одержані значень R_{wi} і R_{vi} для радіального і колового навантажень барабана конвейера від вантаження її стрічкою за законом Ейлера (див. рис.15.2,в). Для даного вантаження $R_{ui} = R_{bi} = 0$.

а) Симетричне вантаження (див. рис.15.4,а):

-при $k = 0$

$$\left. \begin{aligned} R_{wi} &= \frac{q_0 p r}{\ln C} (C - 1) \\ R_{vi} &= 0 \end{aligned} \right\}; \quad (15.40)$$

-при $k = 1, 2, \dots$

$$\left. \begin{aligned} R_{wi} &= q_0 r \frac{\ln C}{k^2 p \left(1 + \frac{f^2}{k^2}\right)} [C(-1)^k - 1] \\ R_{vi} &= -q_0 r f \frac{C(-1)^k - 1}{k \left[1 + \frac{f^2}{k^2}\right]} \end{aligned} \right\}. \quad (15.41)$$

б) Кососиметричне вантаження:

при $k = 0$

$$R_{wi} = 0;$$

$$R_{vi} = \frac{q_0 f p r}{\ln C} [C - 1], \quad (15.42)$$

-при $k = 1, 2, \dots$

$$\left. \begin{aligned} R_{wi} &= -q_0 r \frac{C(-1)^k - 1}{k \left[1 + \frac{f^2}{k^2} \right]}, \\ R_{vi} &= \frac{q_0 f r \ln C}{p} \cdot \frac{C(-1)^k - 1}{k^2 \left[1 + \frac{f^2}{k^2} \right]}. \end{aligned} \right\} \quad (15.43)$$

З використанням матриці жорсткості конструкції $[A]$ система рівнянь, що дозволяє визначити амплітудні переміщення всіх вузлів (15.35), набуде вигляду

$$[A] \{d^n\}^k = \{R\}. \quad (15.44)$$

15.4 Врахування граничних умов

Якщо в якомусь з вузлів якісь з можливих його переміщень $\overline{u}_i, \overline{w}_i, \overline{v}_i, \overline{b}_i$ (див. рис. 15.5) відсутні, то при рішенні системи (15.44) ці переміщення повинні виходити нульовими. Досягається це таким чином: члени матриці $[R]$, що є умовними вузловими силами від зовнішнього навантаження в даних вузлах і в даних напрямках, призначаються як нуль. Для цього в цих вузлах зовнішнє навантаження задається в початкових даних прийнятим нулю. Спеціальними операторами програми рішення задачі коефіцієнти матриці, що є фіктивними вузловими силами у вказаних вузлах і вказаних напрямках від одиничних вузлових переміщень в цих же вузлах і в цих же напрямках, беруться як одиниця. Решта всіх членів

матриці, що знаходяться в рядках і стовпцях, які містять дані одиничні коефіцієнти, беруться як нуль.

15.5 Визначення переміщень і напружень

Рішення системи (15.44) з урахуванням граничних умов для кожної гармоніки рядів Фур'є дає амплітудні переміщення $\{d^n\}$ всіх вузлів конструкції від навантаження даної гармоніки.

За амплітудними переміщеннями за формулами (15.9) і (15.10) визначаються вузлові переміщення для різних значень q (див. рис.15.1), для чого в програмі передбачений цикл за q від $q = 0^0$ до $q = 180^0$ з довільним кроком, який задається в початкових даних.

За матрицею $[A1]$ для кожного з елементів, прочитаною зі стрічки, і його вузловим переміщенням $\{d^n\}$ за залежністю (15.21) визначаються внутрішні зусилля (15.22) для серединного поперечного перерізу елемента. При цьому, оскільки матриця $[A1]$ знаходиться за матрицею $[B]$ з опущеними множниками $\sin kq$ і $\cos kq$ (див. стр. 181), залежність (15.21) дає амплітудні значення внутрішніх зусиль. При цьому, залежності, що визначають внутрішні зусилля для довільного значення кута, мають вигляд:

– для симетричного навантаження:

$$N_s(q) = N_s^n \cos kq; \quad N_q(q) = N_q^n \cos kq;$$

$$N_{sq}(q) = N_{sq}^n \sin kq; \quad M_{sq} = M_{sq}^n \sin kq;$$

$$M_q(q) = M_q^n \cos kq; \quad M_{sq}(q) = M_{sq}^n \sin kq, \quad (15.45)$$

– для кососиметричного навантаження можуть також бути використані залежності (15.45), але із заміною $\sin kq$ на $\cos kq$ і навпаки.

За внутрішніми силовими чинниками (15.22) визначаються напруження в серединних поперечних перерізах елементів:

$$\begin{aligned}
s_s(q) &= \frac{N_s(q)}{t} + \frac{6M_s(q)}{t^2}; \\
s_q(q) &= \frac{N_q(q)}{t} + \frac{6M_q(q)}{t^2}; \\
t_{sq}(q) &= \frac{N_{sq}(q)}{t} + \frac{6M_{sq}(q)}{t^2}.
\end{aligned}
\tag{15.46}$$

де $s_q(q)$ – нормальні напруження, паралельні осі q ;

$s_s(q)$ – нормальні напруження, паралельні осі S ;

$t_{sq}(q)$ – дотичні напруження, що діють паралельно осі q на площадці, перпендикулярній осі S ;

t – товщина даного елемента.

За напруженнями (15.46) можуть бути одержані головні і еквівалентні напруження, а також кути нахилу головних площадок у відповідних точках даних поперечних перерізів:

$$\begin{aligned}
s_1 &= \frac{1}{2} \left[s_s(q) + s_q(q) + \sqrt{(s_s(q) - s_q(q))^2 + 4(t_{sq}(q))^2} \right]; \\
s_3 &= \frac{1}{2} \left[s_s(q) + s_q(q) - \sqrt{(s_s(q) - s_q(q))^2 + 4(t_{sq}(q))^2} \right]; \\
s_{екв} &= s_1 - s_3; \quad a = \arctg \frac{t_{sq}}{s_{sq} - s_3}.
\end{aligned}
\tag{15.47}$$

У програмі рішення задачі організовано цикл за k з підсумовуванням переміщень і напружень від всіх розглянутих гармонік. Крім того, передбачено порівняння нормальних амплітудних і дотичних напружень від кожної чергової гармоніки з відповідним сумарним амплітудним напруженням $(k-1)$ гармонік. Якщо відносні прирости

вказаних напружень від кожної з якихось трьох послідовних гармонік виявляються нижче заданої точності

$$\frac{(s_s^n)_k}{\sum_{i=1}^{i=k-1} (s_s^n)} < E; \quad \frac{(s_q^n)_k}{\sum_{i=1}^{i=k-1} (s_{sq}^n)} < E; \quad \frac{(t_{sq}^n)_k}{\sum_{i=1}^{i=k-1} (t_{sq}^n)} < E;$$

цикл за k припиняється, і видаються до друку наступні вихідні параметри: переміщення точок вузлових зусиль для заданих значень Q , внутрішні зусилля (15.45), напруження (15.46), головні і еквівалентні напруження (15.47), кути нахилу головних площадок (15.47) для серединних перерізів всіх елементів і заданих значень Q .

15.6 Блок-схеми програми

Блок-схеми програми для ПЕОМ подані на рис. 15.9...15.15.

У програмі заведені наступні позначення:

K_1 – номер гармоніки;

K_2 – лічильник гармонік;

K_3 – перевірка типу конструкції;

K_4 – коефіцієнт, що визначає вид розбиття оболонки на елементи ($K_4 = 1$ - ручне розбиття, $K_4 \neq 1$ - автоматичне розбиття);

K_5 – перевірка вірності зв'язків елементів;

K_6 – перевірка умови друку напружень ($K_6 \neq 0$);

K_7 – перевірка умови друку переміщень ($K_7 \neq 0$);

M – перевірка умови обчислення навантаження для крайнього лівого вузла;

L – перевірка умови обчислення навантаження для крайнього правого вузла;

SK(600, 24) - масив матриці жорсткості конструкції;

NSW(180,2) – масив зв'язків елементів;

SMAХ – максимальне зусилля, що діє на барабан;

MKR – крутний момент, що діє на барабан;

R – радіус барабана;

$P(600)$ – матриця амплітудних вузлових напружень від вантаження k -ю гармонікою;

$DIS(4,150)$ – матриця амплітудних вузлових переміщень від вантаження k -ю гармонікою.

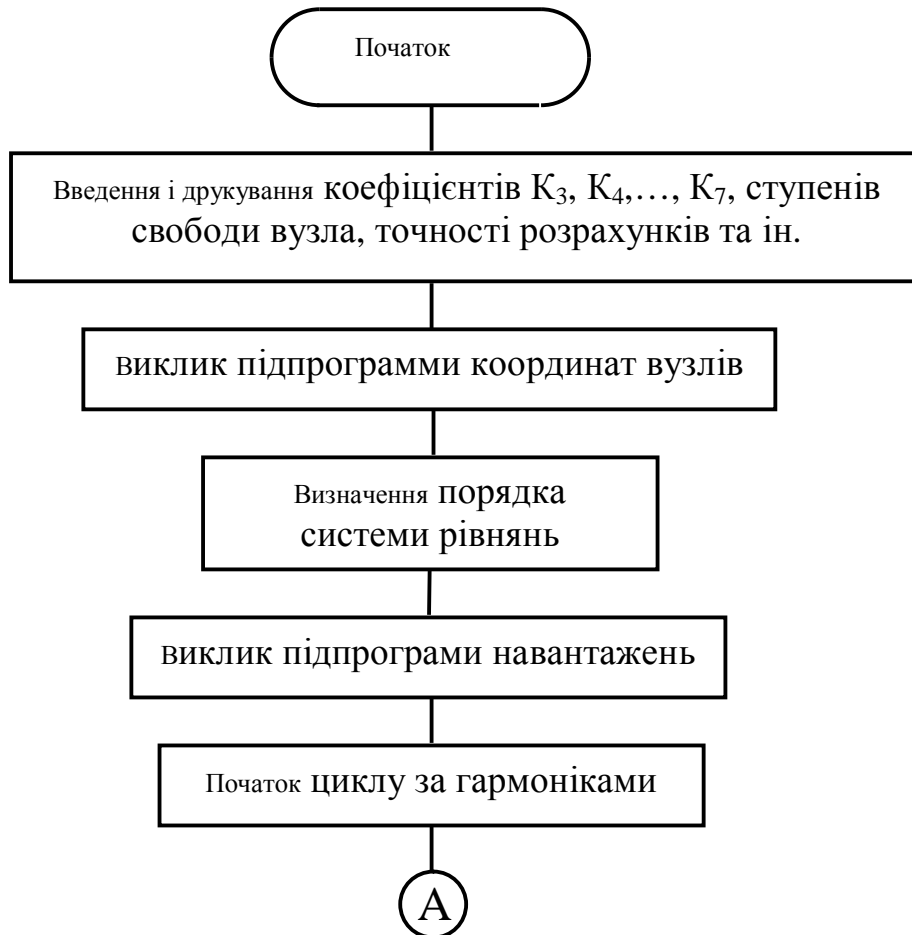


Рисунок 15.9 - Блок-схема основної програми

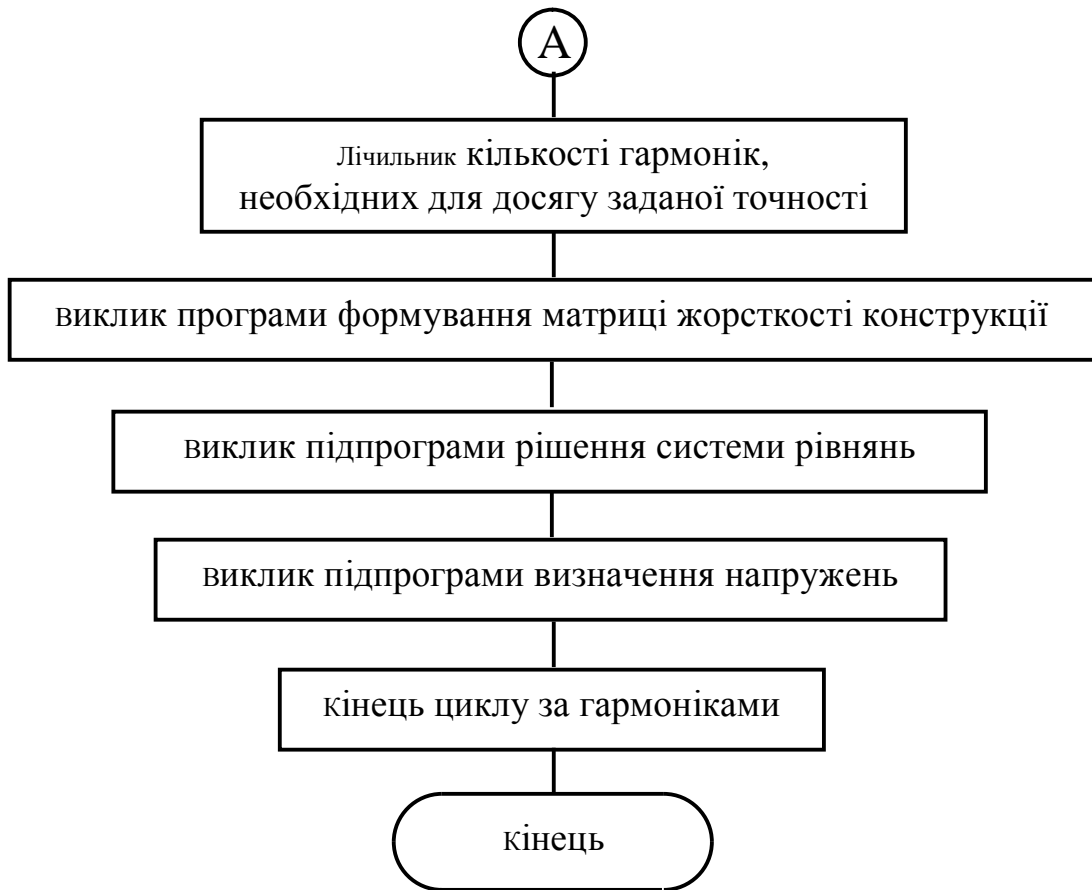


Рисунок 15.9, аркуш 2

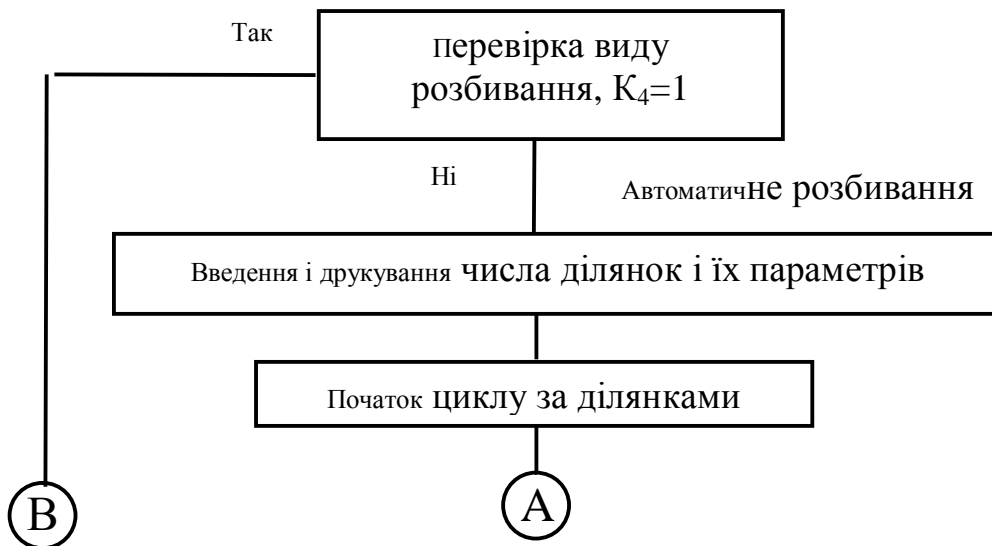


Рисунок 15.10 - Блок-схема підпрограми координат вузлів

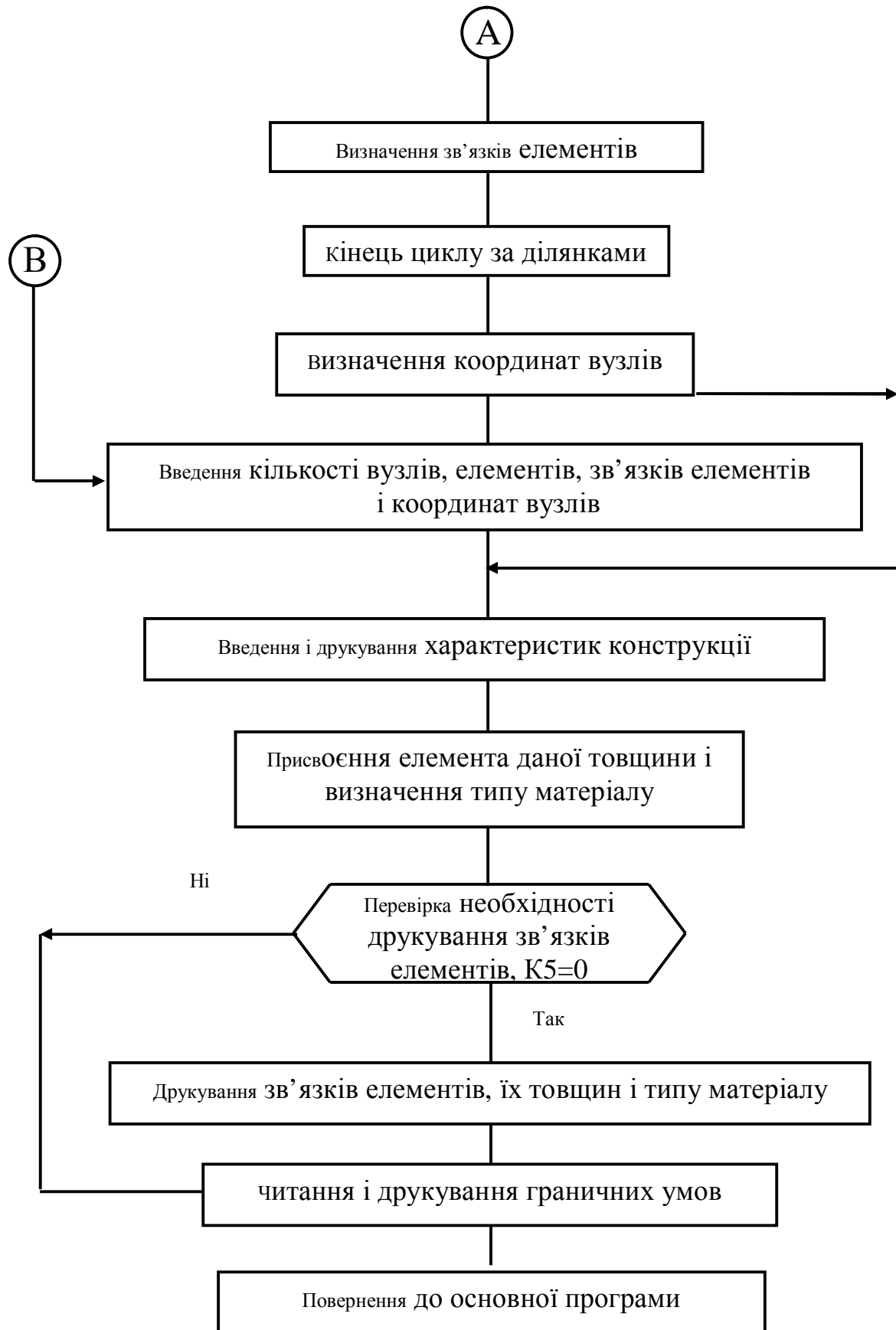


Рисунок 15.10, аркуш 2

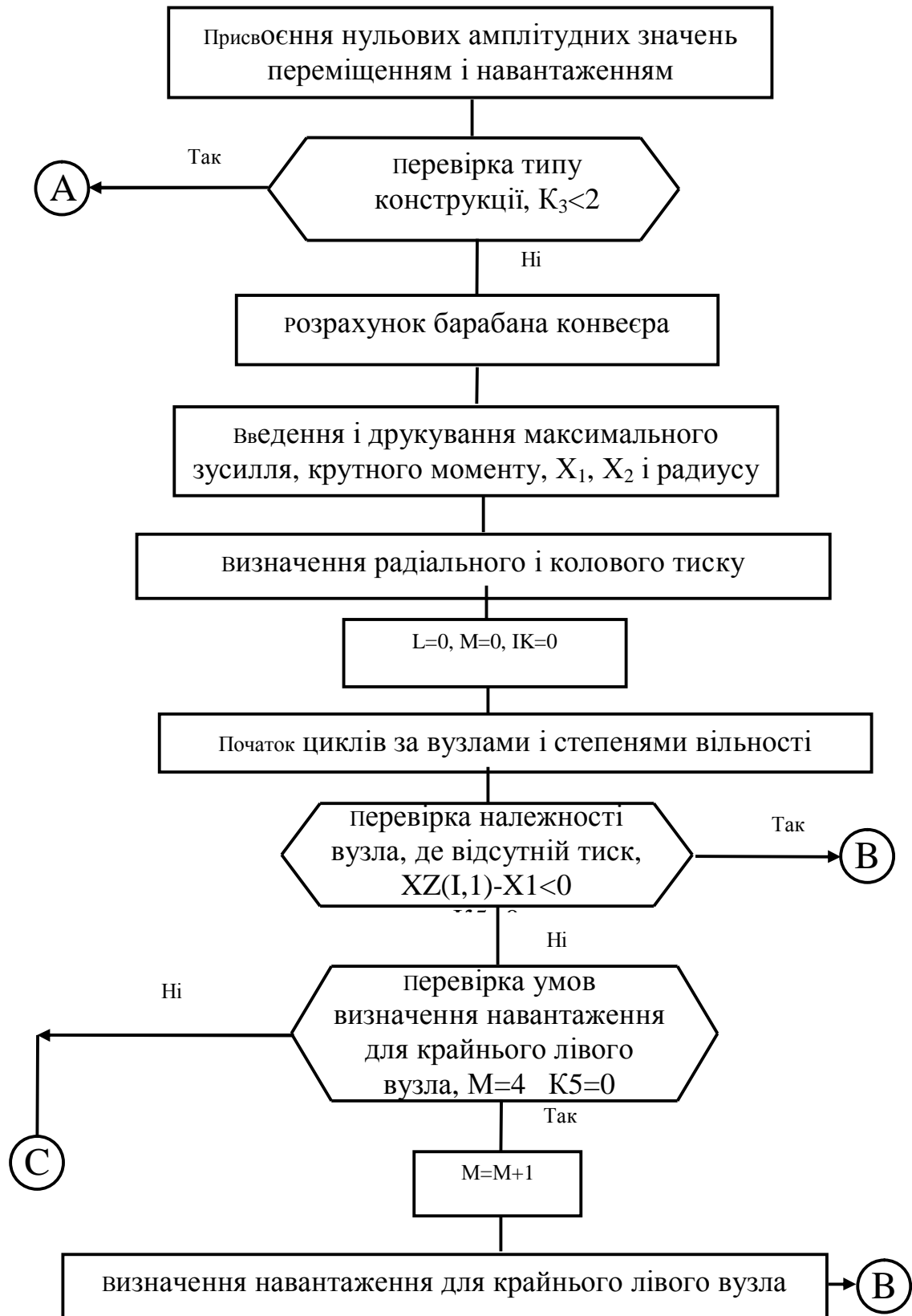


Рисунок 15.11 - Блок-схема підпрограми навантаження

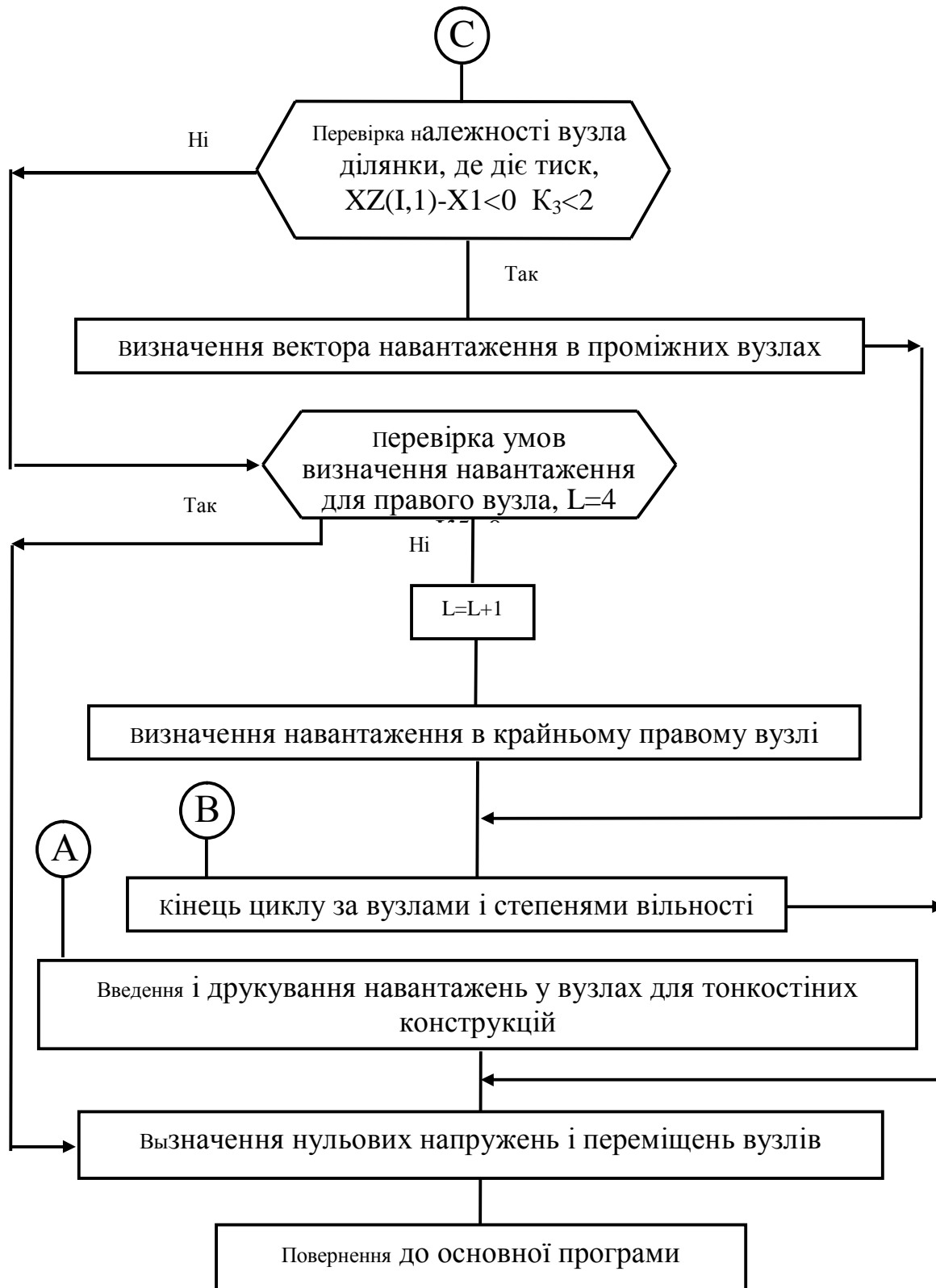


Рисунок 15.11, аркуш 2



Рисунок 15.12 - Блок-схема підпрограми матриці жорсткості конструкції

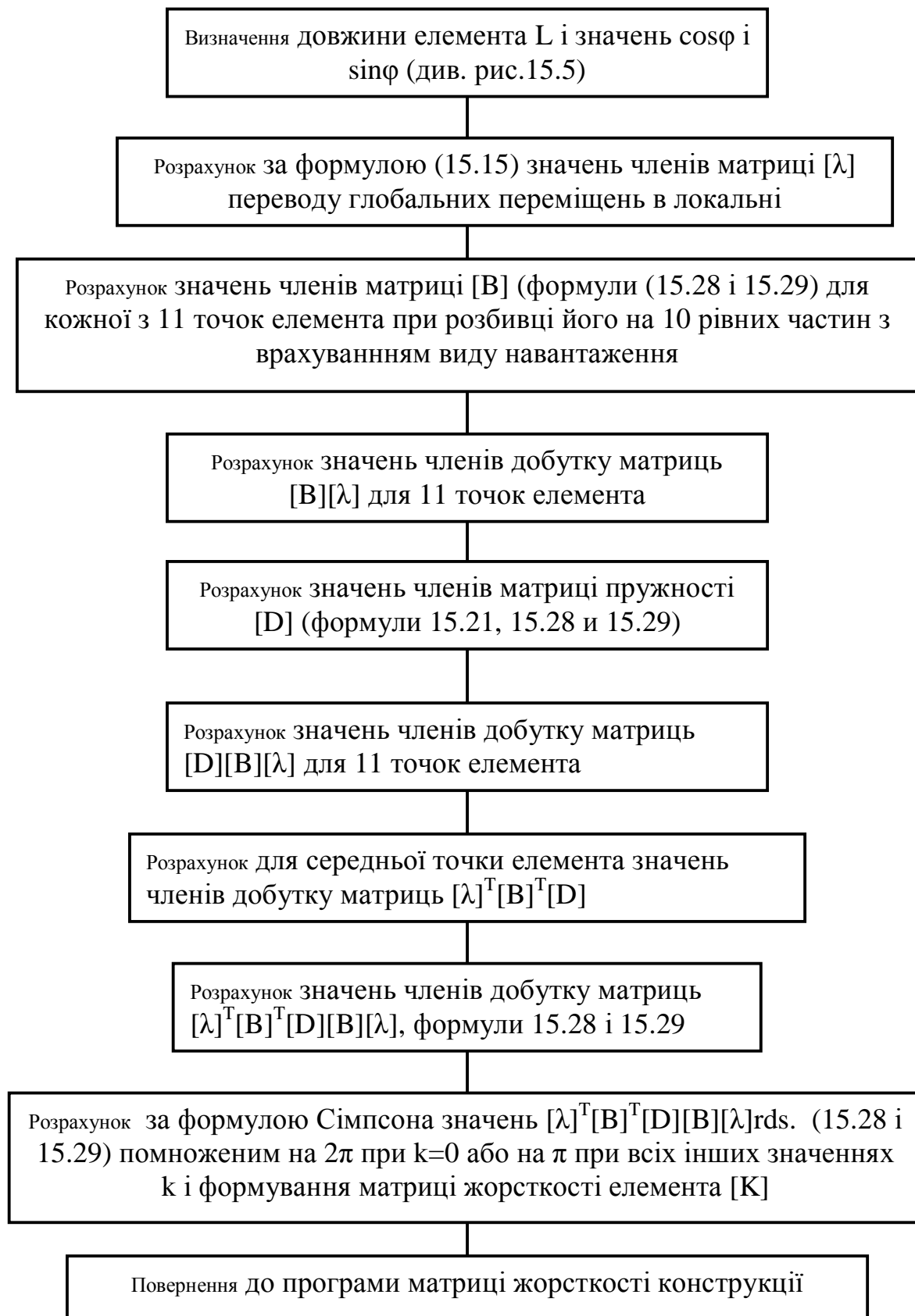


Рисунок 15.13 – Блок-схема підпрограми матриці жорсткості елемента

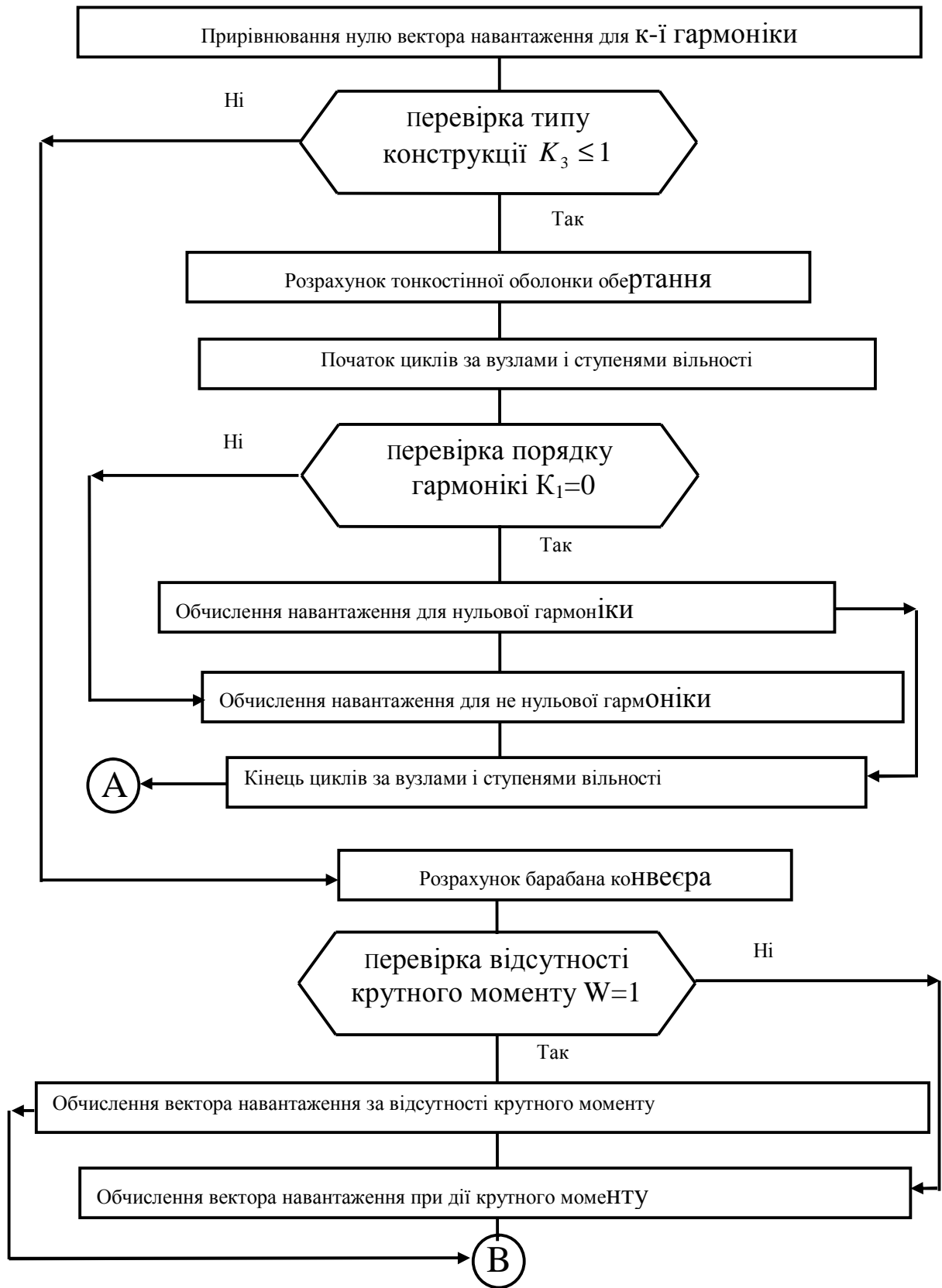


Рисунок 15.14 - Блок-схема підпрограми рішення системи рівнянь

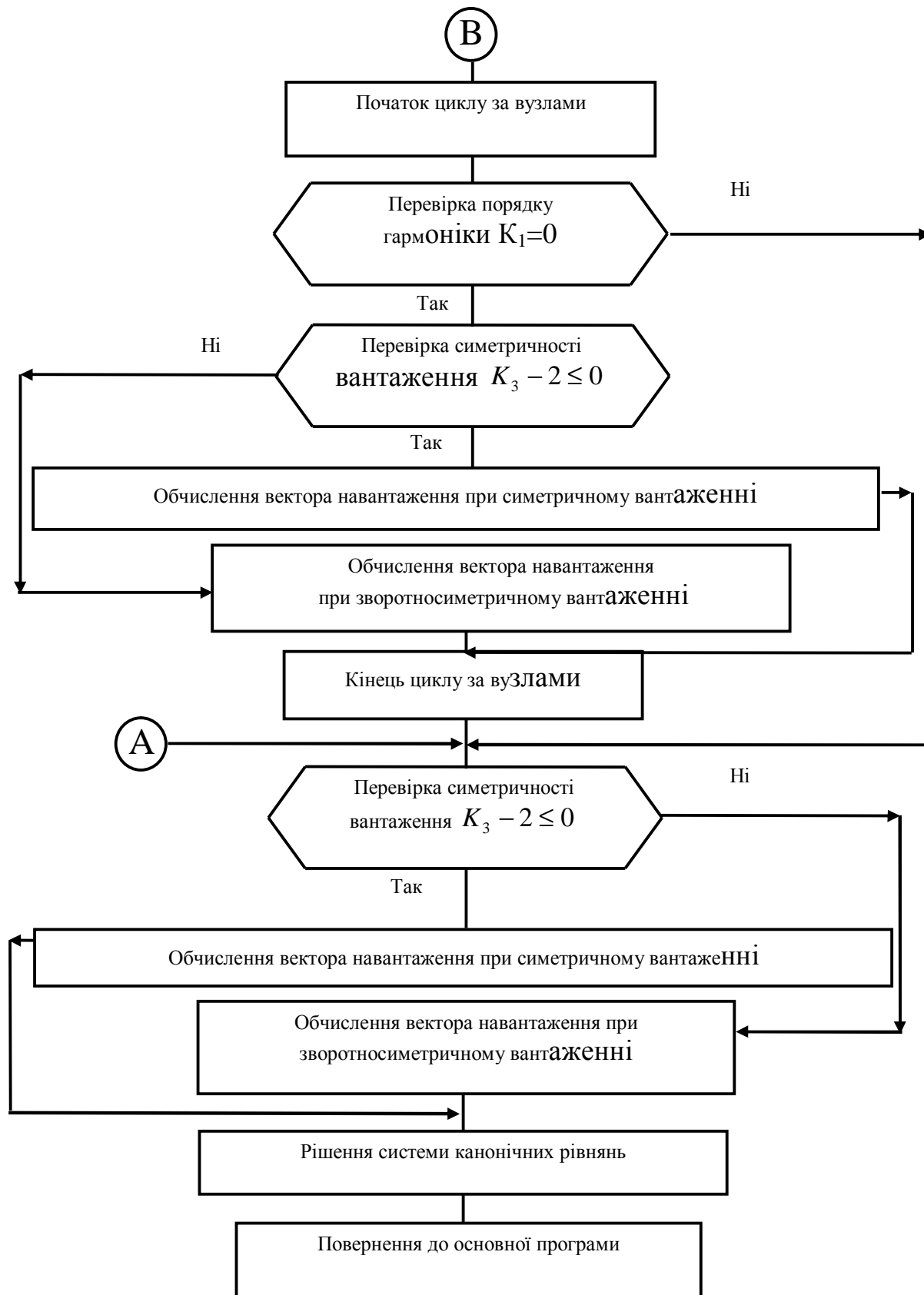


Рисунок 15.14, аркуш 2

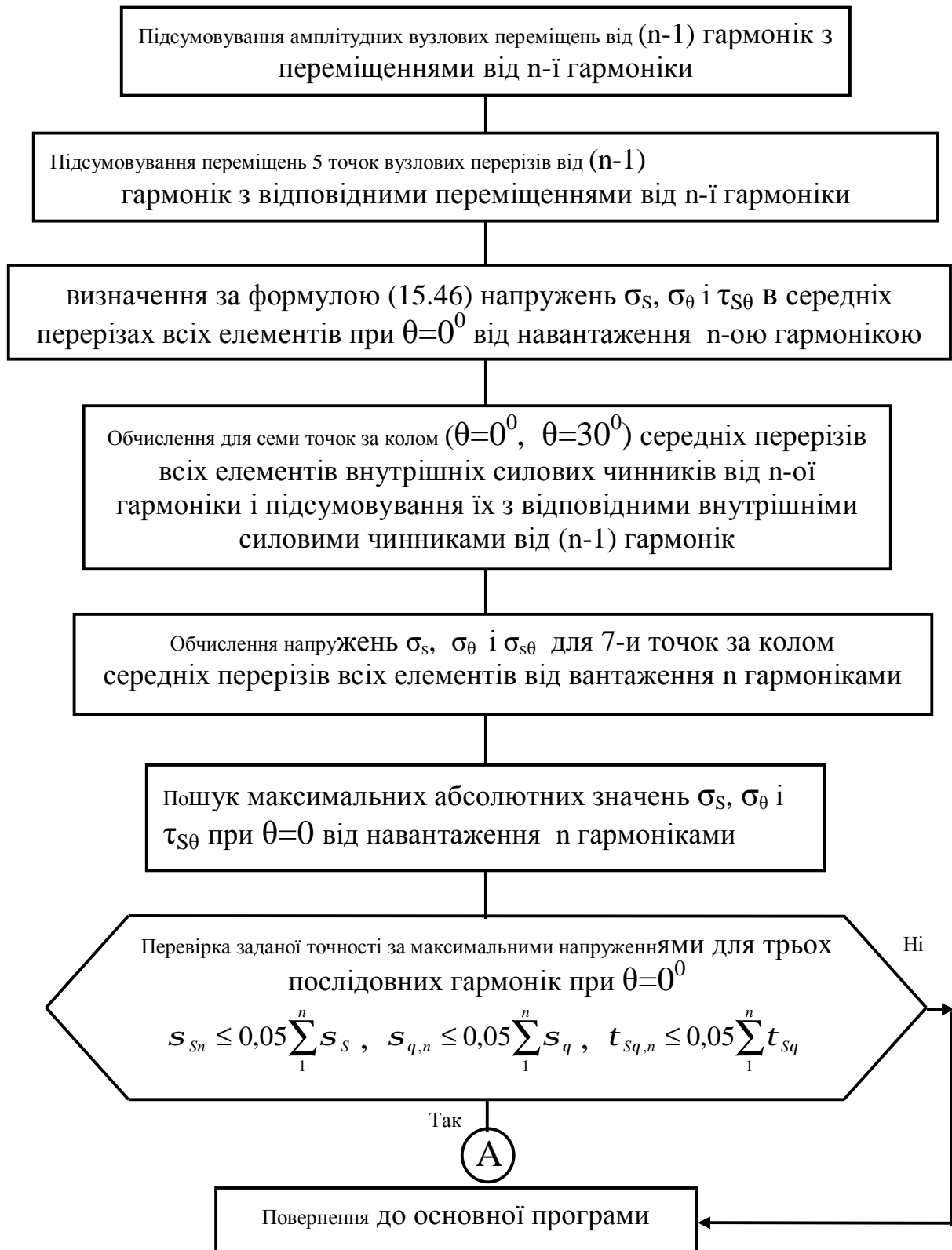


Рисунок 15.15 - Блок-схема підпрограми обчислення переміщень і напружень

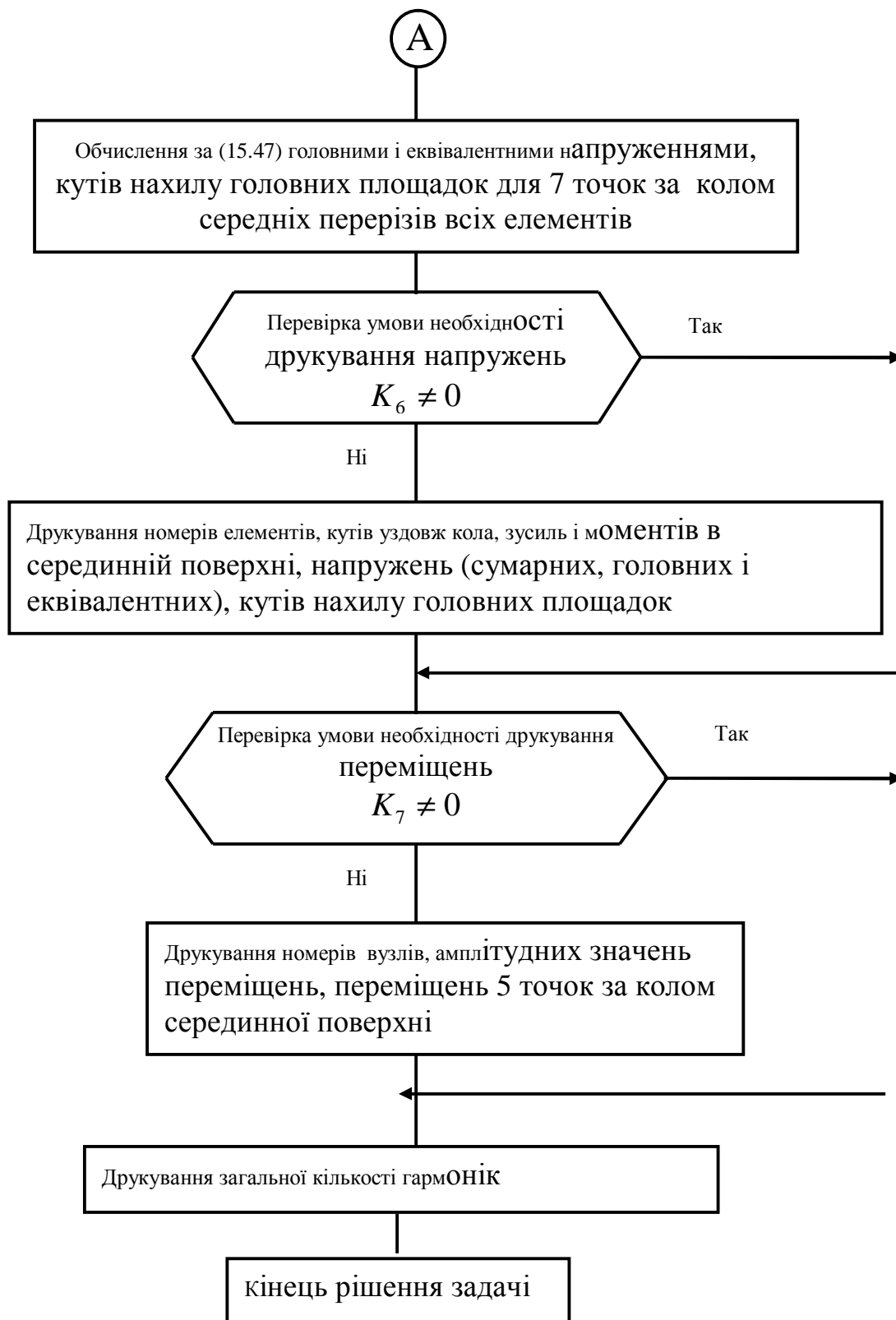


Рисунок 15.15, аркуш 2

15.7 Апробації методики на класичних прикладах розрахунків

З метою апробації розробленої методики і програми реалізації її на ПЕОМ були виконані розрахунки напружено-деформованого стану кільцевої пластини, навантаженої кільцевим розподіленим навантаженням (рис.15.16), а також конічної (рис. 15.18) і ступінчастої циліндрової (рис. 15.20) кришок, які знаходяться під дією рівномірного тиску. Отримані результати зіставлені з результатами аналітичних розрахунків вказаних елементів, наведених у праці [11].

15.7.1 Розрахунок кільцевої пластини під дією кільцевого розподіленого навантаження

Схема вантаження пластини показана на рис. 15.16. Початкові дані: $a=100\text{мм}$, $h=20\text{мм}$, $p=1\text{ Н/мм}^2$, матеріал пластини –Ст.3, $E=2\cdot 10^5\text{ МПа}$, $\mu=0,3$.

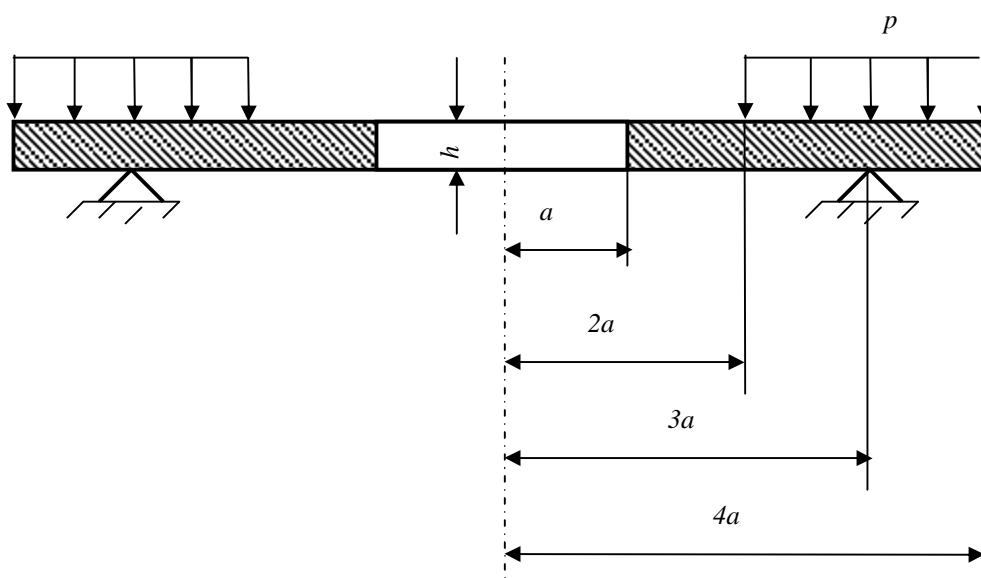
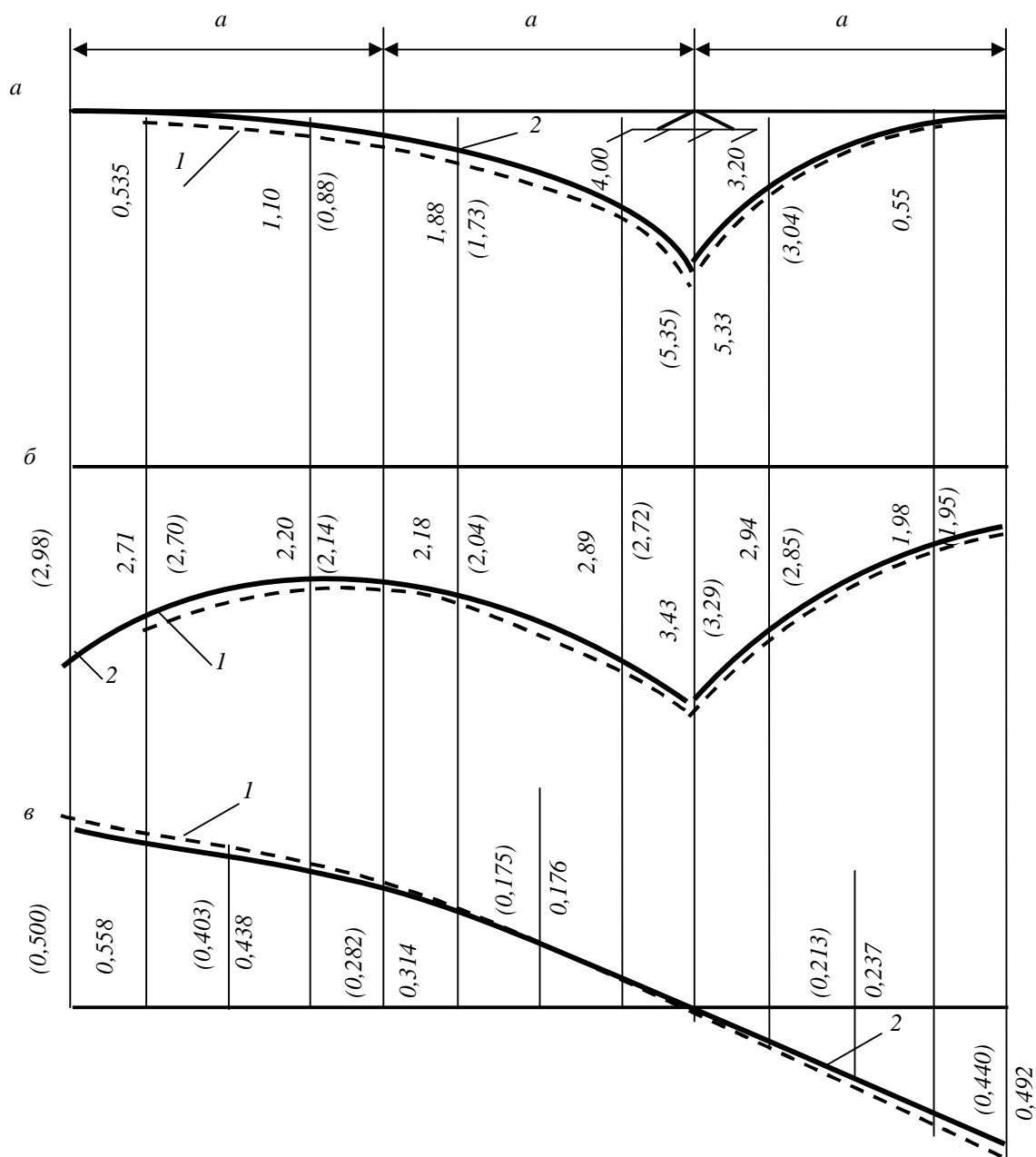


Рисунок 15.16 - Схема навантаження пластини

На рис. 15.17 наведені епюри згинальних моментів і прогинів, побудовані для даної пластини за наслідками розрахунків за пропонованою методикою (значення без дужок) і за аналітичними залежностями [11] (значення у дужках).

З рис. 15.17 видно, що максимальні згинальні моменти, обчислені за різними методиками відрізняються:

M_s – на 3,6%, а M_θ – на 4,3%. Прогини, обчислені методом кінцевих елементів, на 11,6% більше, ніж прогини, обчислені за аналітичними залежностями.



a – еюра радіальних згинальних моментів M_s , кН; $б$ – еюра тангенціальних згинальних моментів M_θ , кН; $в$ – еюра прогинів u , мм (1 – за пропонованою методикою, 2 – за аналітичними залежностями)
Рисунок 15.17 - Еюри згинальних моментів і прогинів кільцевої пластини

Великі погрішності по прогинах пояснюються тим, що при розрахунку за пропонованою методикою пластина розбивалася на незначну кількість елементів. Природньо, при збільшенні їх кількості, точність зростає.

15.7.2 Розрахунок конічної кришки під дією розподіленого навантаження

Схема навантаження кришки показана на рис. 15.18. Початкові дані: $s_1=15,5\text{мм}$, $s_2=129,4\text{ мм}$, $h=4\text{мм}$, $p=1\text{ Н/мм}^2$, матеріал пластини –Ст.3, $E=2\cdot 10^5\text{ МПа}$, $\mu=0,3$.

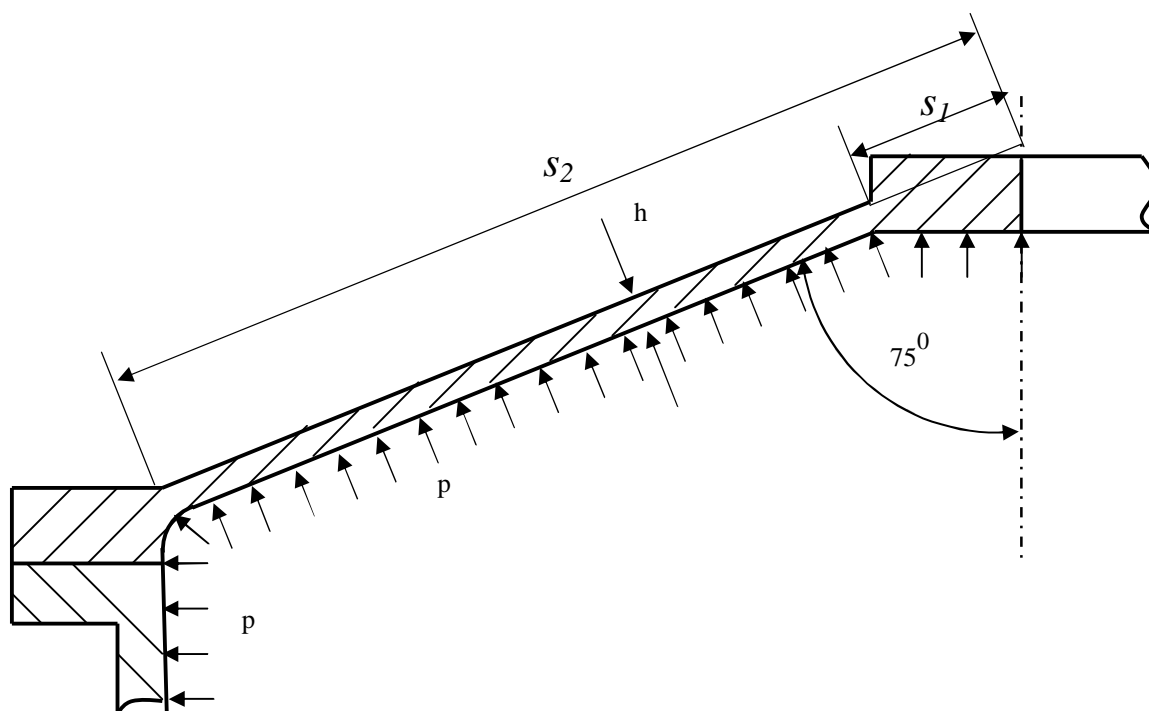
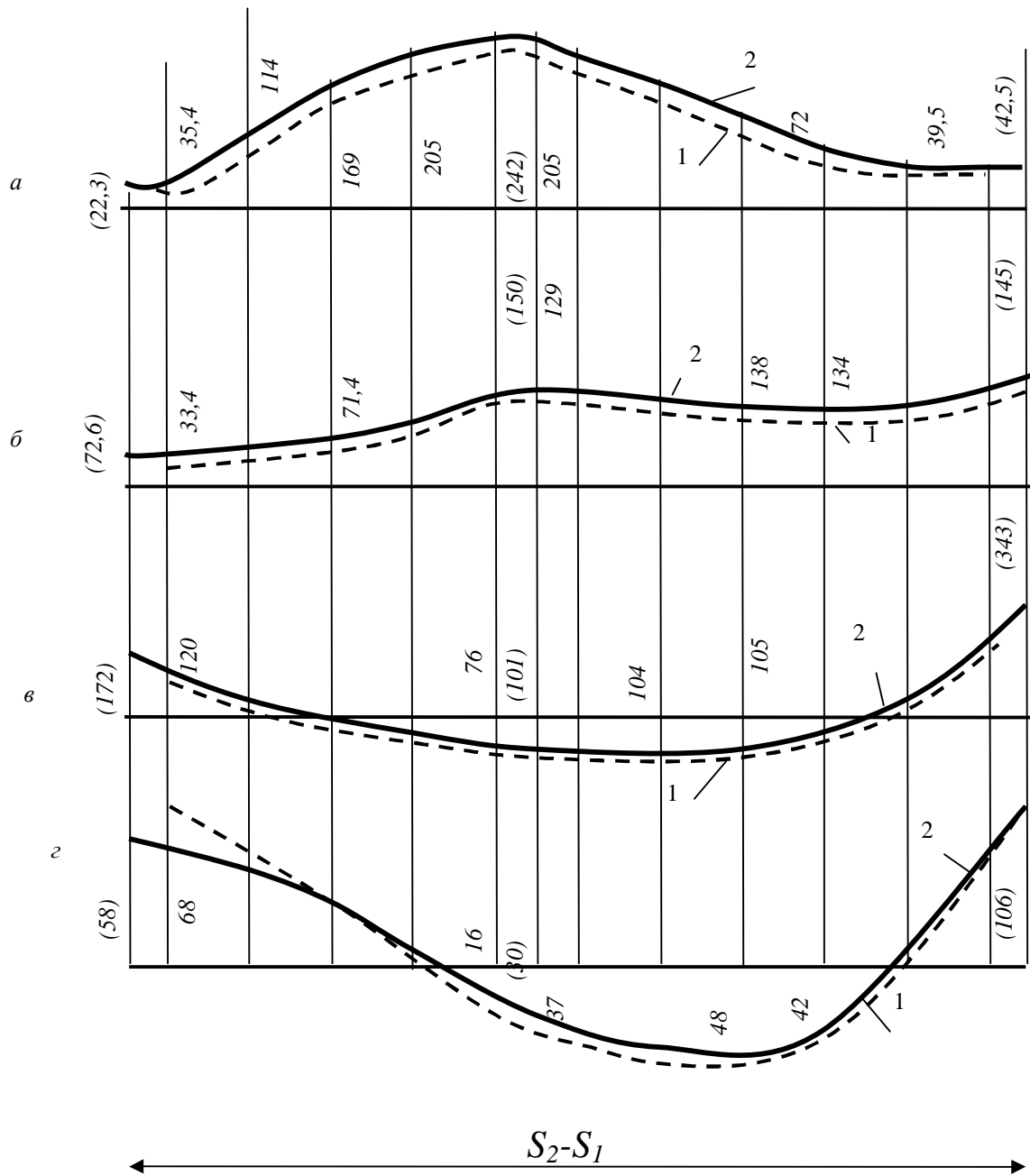


Рисунок 15.18 - Схема навантаження кришки

На рис. 15.19 наведені епюри розподілу радіальних подовжніх сил N_s , тангенціальних подовжніх сил N_θ і радіальних M_s і тангенціальних M_θ згинальних моментів уздовж кришки твірної, побудовані за наслідками розрахунків за пропонованою методикою (значення без дужок) і за аналітичними залежностями [11] (значення у дужках).

Аналізуючи епюри, подані на рис. 15.19, можна зробити наступні

висновки. Подовжні сили, визначені за двома методиками відрізняються на 14%, а згинальні моменти – на 1,7%.



а - еюра радіальних подовжніх сил N_s , Н/мм;
 б - еюра тангенціальних подовжніх сил N_θ , Н/мм;
 в - еюра радіальних згинальних моментів M_s , Н;
 г - еюра тангенціальних згинальних моментів M_θ , Н
 (1 - за запропонованою методикою,
 2 - за аналітичними залежностями)

Рисунок 15.19 - Еюри подовжніх сил і згинальних моментів у конічній кришці

15.7.3 Розрахунок ступінчастої циліндрової кришки під дією рівномірного тиску

Схема вантаження кришки показана на рис.15.20.

Початкові дані: $a=100$ мм, $l=31$ мм, $h=4$ мм, $p=1$ Н/мм².

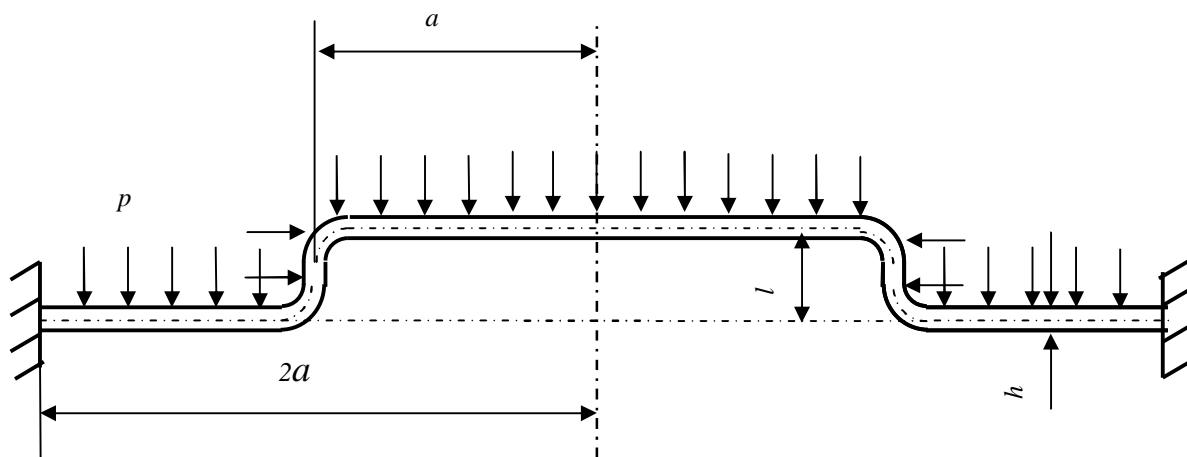


Рисунок 15.20 - Схема навантаження кришки

На рис.15.21 наведені епюри згинальних моментів для даної кришки, побудовані за наслідками розрахунків за пропонованою методикою (значення без дужок) і за аналітичними залежностями [11] (значення у дужках).

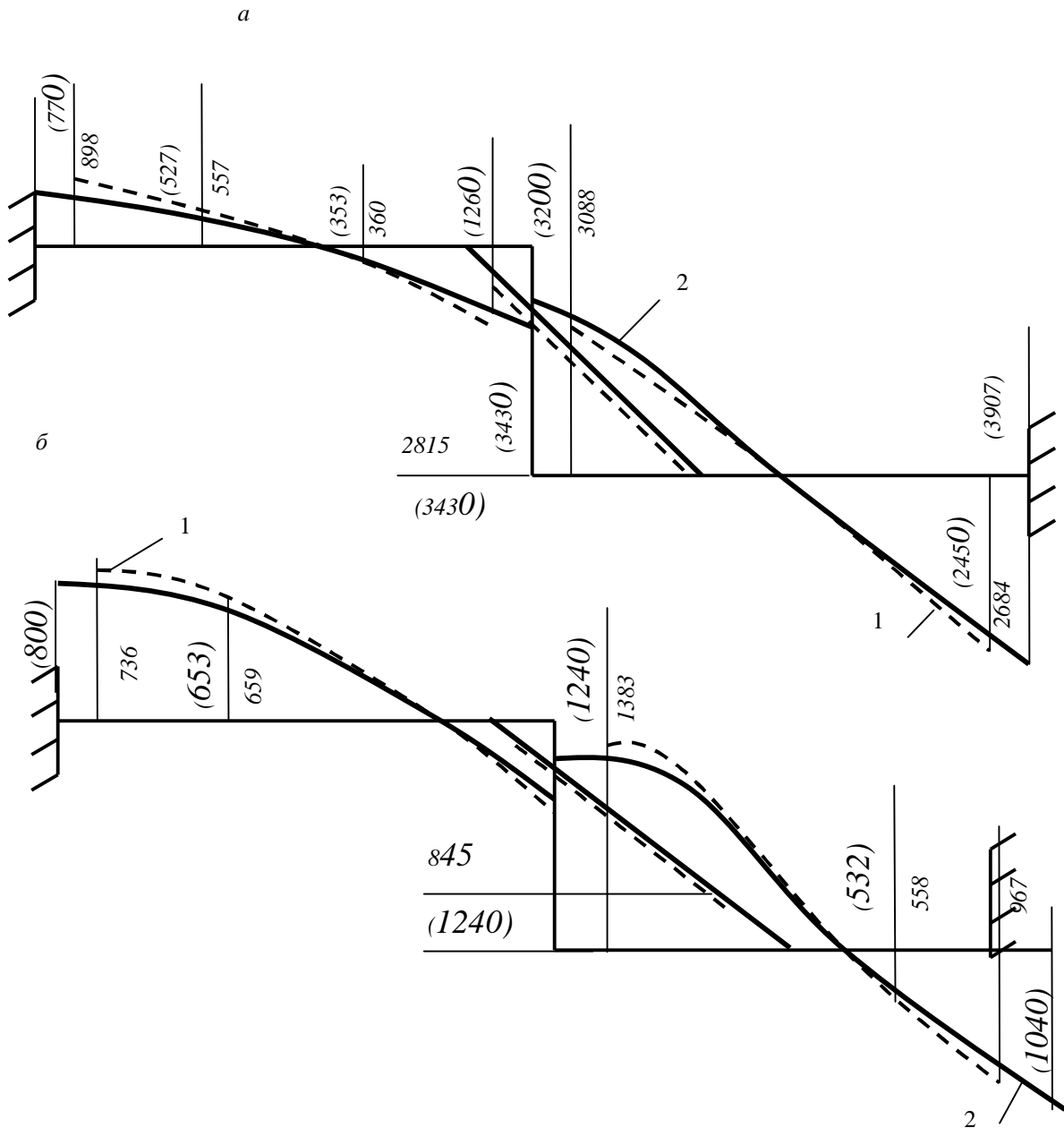
Зіставлення результатів, обчислених за двома методиками (див. рис. 15.21), показує відмінність максимальних радіальних моментів на 3,9%, а максимальних тангенціальних моментів на 10%.

15.7.4 Висновки з апробації методики

Наведені зіставлення результатів розрахунку внутрішніх силових фактів і переміщень для розглянутих елементів конструкцій за розробленою методикою і програмою, реалізації її на ПЕОМ і за класичними аналітичними залежностями дозволяє стверджувати наступне.

Розроблена методика визначення напружено-деформованого стану тонкостінних тіл обертання напіваналітичним методом кінцевих елементів і класичними залежностями дають практично однакові результати. Проте, аналітичні залежності можна ефективно використовувати тільки для тіл нескладної конфігурації і лише для певних видів вантаження, оскільки кожен

конкретний випадок конструктивного оформлення і вантаження вимагають індивідуального підходу і складних математичних викладень. Безперечно перевага розробленої методики полягає в тому, що вона дозволяє за єдиним алгоритмом оцінювати напружено-деформований стан тонкостінних тіл обертання довільної конструкції при довільному вантаженні.



а – еюра радіальних згинальних моментів M_s, N ;
 б – еюра тангенціальних згинальних моментів M_θ, H
 (1 – за пропонованою методикою, 2 – за аналітичними залежностями)

Рисунок 15.21 – Еюри згинальних моментів ступінчастої циліндрової кришці

16 РОЗРАХУНОК ОБ'ЄМНОЇ ЗАДАЧІ ТЕОРІЇ ПРУЖНОСТІ

16.1 Поняття про об'ємну задачу теорії пружності

Тіла, звані масивами, для яких всі розміри одного порядку, випробовують об'ємний напружений стан (рис. 16.1). У цьому випадку в довільній точці тіла виникає шість компонентів напруження:

$$s_x, s_y, s_z, t_{xy}, t_{xz}, t_{zy}.$$

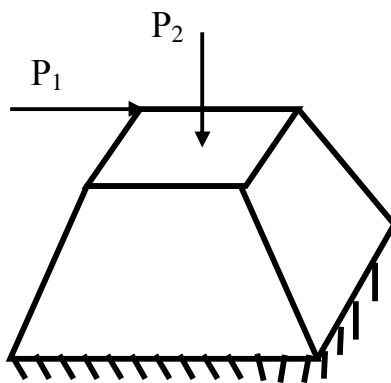


Рисунок 16.1- Масив

Узагальнений закон Гука для об'ємного напруженого стану має вигляд:

$$\varepsilon_x = \frac{1}{E} [\sigma_x - \mu(\sigma_y + \sigma_z)],$$

$$\varepsilon_y = \frac{1}{E} [\sigma_y - \mu(\sigma_x + \sigma_z)],$$

$$\varepsilon_z = \frac{1}{E} [\sigma_z - \mu(\sigma_x + \sigma_y)],$$

$$g_{xy} = \frac{t_{xy}}{G}, \quad g_{yz} = \frac{t_{yz}}{G}, \quad g_{xz} = \frac{t_{xz}}{G}. \quad (16.1)$$

Виразивши напруження через деформації, одержимо:

$$\begin{Bmatrix} s_x \\ s_y \\ s_z \\ t_{xz} \\ t_{yz} \\ t_{zy} \end{Bmatrix} = \frac{E(1-2m)}{(1-2m)(1+m)} \begin{bmatrix} 1 & \frac{m}{1-m} & \frac{m}{1-m} & 0 & 0 & 0 \\ \frac{m}{1-m} & 1 & \frac{m}{1-m} & 0 & 0 & 0 \\ \frac{m}{1-m} & \frac{m}{1-m} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2m}{2(1-m)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2m}{2(1-m)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2m}{2(1-m)} \end{bmatrix} \begin{Bmatrix} e_x \\ e_y \\ e_z \\ g_{xy} \\ g_{xz} \\ g_{yz} \end{Bmatrix} \quad (16.2)$$

Вектор деформацій для об'ємного напруженого стану (рис. 16.2) через переміщення виражається за допомогою залежностей Коші:

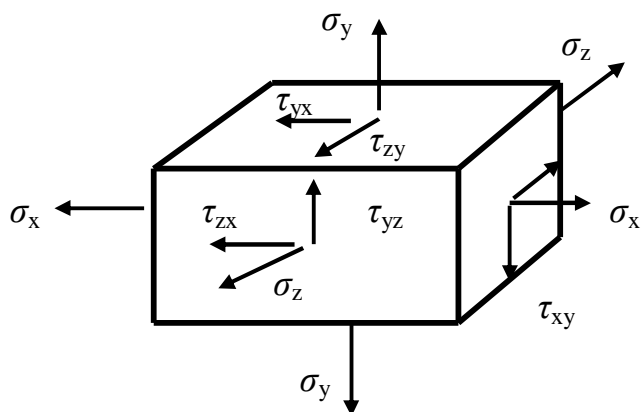


Рисунок 16.2 - Компоненти напружень в точці

$$\{e\} = \begin{Bmatrix} e_x \\ e_y \\ e_z \\ g_{xy} \\ g_{xz} \\ g_{yz} \end{Bmatrix} = \begin{Bmatrix} \frac{du}{dx} \\ \frac{dv}{dy} \\ \frac{dw}{dz} \\ \frac{du}{dy} + \frac{dv}{dx} \\ \frac{du}{dz} + \frac{dw}{dx} \\ \frac{dv}{dz} + \frac{dw}{dy} \end{Bmatrix} \quad (16.3)$$

16.2 Функція переміщення для тетраедра

При розгляді об'ємних задач МКЕ найбільш простим кінцевим елементом є тетраедр (рис. 13.3). Тепер вектор переміщення вузла має три складові:

$$\{d\}_i = \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}. \quad (16.4)$$

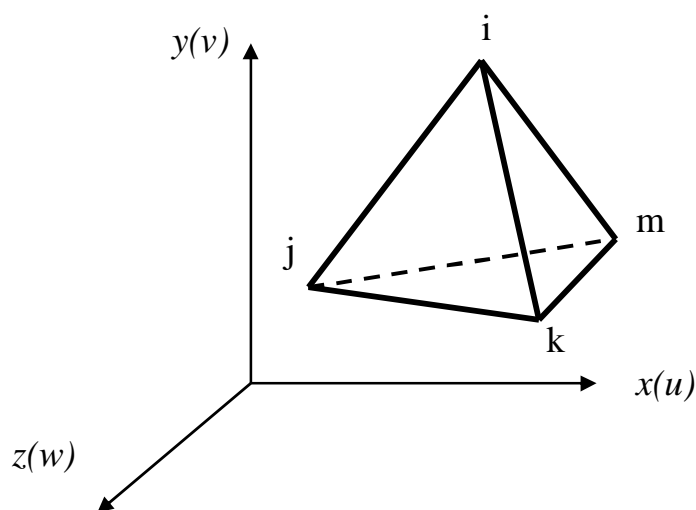


Рисунок 16.3 - Кінцевий елемент у вигляді тетраедра

Функція переміщення u , v або w тепер представляє поліном, що складається з чотирьох складових, оскільки тетраедр має чотири вузли:

$$u = a_0 + a_1x + a_2y + a_3z. \quad (16.5)$$

Коефіцієнти a_0 , a_1 , a_2 і a_3 виражаємо через переміщення вузлів:

$$\begin{aligned} u_i &= a_0 + a_1x_i + a_2y_i + a_3z_i, \\ u_j &= a_0 + a_1x_j + a_2y_j + a_3z_j \end{aligned} \quad (16.6)$$

$$u_k = a_0 + a_1x_k + a_2y_k + a_3z_k,$$

$$u_m = a_0 + a_1x_m + a_2y_m + a_3z_m.$$

Вирішуючи систему рівнянь за допомогою визначників, знайдемо a_0 , a_1 , a_2, a_3 :

$$a_0 = \frac{\begin{vmatrix} u_i & x_i & y_i & z_i \\ u_j & x_j & y_j & z_j \\ u_k & x_k & y_k & z_k \\ u_m & x_m & y_m & z_m \end{vmatrix}}{\begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_m & y_m & z_m \end{vmatrix}}, \quad a_1 = \frac{\begin{vmatrix} 1 & u_i & y_i & z_i \\ 1 & u_j & y_j & z_j \\ 1 & u_k & y_k & z_k \\ 1 & u_m & y_m & z_m \end{vmatrix}}{\begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_m & y_m & z_m \end{vmatrix}},$$

$$a_2 = \frac{\begin{vmatrix} 1 & x_i & u_i & z_i \\ 1 & x_j & u_j & z_j \\ 1 & x_k & u_k & z_k \\ 1 & x_m & u_m & z_m \end{vmatrix}}{\begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_m & y_m & z_m \end{vmatrix}}, \quad a_3 = \frac{\begin{vmatrix} 1 & x_i & y_i & u_i \\ 1 & x_j & y_j & u_j \\ 1 & x_k & y_k & u_k \\ 1 & x_m & y_m & u_m \end{vmatrix}}{\begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_m & y_m & z_m \end{vmatrix}}. \quad (16.7)$$

Проаналізувавши знаменник цих коефіцієнтів, переконуємося, що він дорівнює $6V$:

$$6V = \begin{vmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_m & y_m & z_m \end{vmatrix}. \quad (16.8)$$

Підставивши значення цих коефіцієнтів у вираз (16.5) і виконавши перетворення, одержимо:

$$u = \frac{1}{6V} \begin{vmatrix} a_i + b_i x + c_i y + d_i z & a_j + b_j x + c_j y + d_j z & a_k + b_k x + c_k y + d_k z \\ a_m + b_m x + c_m y + d_m z & & \end{vmatrix} \begin{Bmatrix} u_i \\ u_j \\ u_k \\ u_m \end{Bmatrix}, \quad (16.9)$$

$$\text{де } a_i = \begin{vmatrix} x_j & y_j & z_j \\ x_k & y_k & z_k \\ x_m & y_m & z_m \end{vmatrix}; \quad b_i = - \begin{vmatrix} 1 & y_j & z_j \\ 1 & y_k & z_k \\ 1 & y_m & z_m \end{vmatrix};$$

$$c_i = \begin{vmatrix} 1 & x_j & z_j \\ 1 & x_k & z_k \\ 1 & x_m & z_m \end{vmatrix}; \quad d_i = - \begin{vmatrix} 1_j & x_j & y_j \\ 1_k & x_k & y_k \\ 1_m & x_m & y_m \end{vmatrix}. \quad (16.10)$$

Аналогічно одержимо функції v і w .

Коефіцієнти $a_j, b_j, c_j, d_j, \dots, a_m, b_m, c_m, d_m$ виходять круговою підстановкою індексів $i \rightarrow j \rightarrow k \rightarrow m \rightarrow i$.

16.3 Вектори деформацій і напружень

Підставляючи значення функції u, v і w з виразу (16.9) у вираз (16.3), одержимо вектор деформацій елемента.

$$\{e\} = \frac{1}{6V} \begin{vmatrix} b_i & 0 & 0 & b_j & 0 & 0 & b_k & 0 & 0 & b_m & 0 & 0 \\ 0 & c_i & 0 & 0 & c_j & 0 & 0 & c_k & 0 & 0 & c_m & 0 \\ 0 & 0 & d_i & 0 & 0 & d_j & 0 & 0 & d_k & 0 & 0 & d_m \\ c_i & b_i & 0 & c_j & b_j & 0 & c_k & b_k & 0 & c_m & b_m & 0 \\ d_i & 0 & b_i & d_j & 0 & b_j & d_k & 0 & b_k & d_m & 0 & b_m \\ 0 & d_i & c_i & 0 & d_j & c_j & 0 & d_k & c_k & 0 & d_m & c_m \end{vmatrix} \begin{Bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \\ u_k \\ v_k \\ w_k \\ u_m \\ v_m \\ w_m \end{Bmatrix}. \quad (16.11)$$

Або $\{e\} = |B|\{d\}^e$, де $\{d\}^e$ вектор переміщень вузлів елемента.

$$[B] = \frac{1}{6V} \begin{vmatrix} b_i & 0 & 0 & b_j & 0 & 0 & b_k & 0 & 0 & b_m & 0 & 0 \\ 0 & c_i & 0 & 0 & c_j & 0 & 0 & c_k & 0 & 0 & c_m & 0 \\ 0 & 0 & d_i & 0 & 0 & d_j & 0 & 0 & d_k & 0 & 0 & d_m \\ c_i & b_i & 0 & c_j & b_j & 0 & c_k & b_k & 0 & c_m & b_m & 0 \\ d_i & 0 & b_i & d_j & 0 & b_j & d_k & 0 & b_k & d_m & 0 & b_m \\ 0 & d_i & c_i & 0 & d_j & c_j & 0 & d_k & c_k & 0 & d_m & c_m \end{vmatrix}. \quad (16.12)$$

Помноживши вираз (16.11) на матрицю пружності, одержимо вектор напружень

$$\{\varepsilon\} = |D||B|\{\delta\}^e, \quad (16.13)$$

де $|D|$ - матриця пружності.

$$|D| = \frac{E(1-2\mu)}{(1-2\mu)(1+\mu)} \begin{vmatrix} 1 & \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 0 & 0 & 0 \\ \frac{\mu}{1-\mu} & 1 & \frac{\mu}{1-\mu} & 0 & 0 & 0 \\ \frac{\mu}{1-\mu} & \frac{\mu}{1-\mu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\mu}{2(1-\mu)} \end{vmatrix}. \quad (16.14)$$

16.4 Матриця жорсткості для тетраедра

Підставивши значення матриць $|D|$ і $|B|$ у вираз (1.6), одержимо матрицю жорсткості кінцевого елемента. Оскільки в матриці $|B|$ коефіцієнти a , b , c , і d не залежать від x , y і z , то маємо:

$$|K|^1 = |B|^T |D| |B| V. \quad (16.15)$$

16.5 Функція переміщень і матриця жорсткості паралелепіпеда

При виборі кінцевого елемента у вигляді тетраедра, хоча це і найбільш простий елемент, важко уявити розбиття об'ємного тіла у вигляді набору пірамід. Значно простіше зобразити розбиття масиву на паралелепіпеди (рис. 16.4). Початок координат поміщаємо в центр паралелепіпеда. Вектор переміщень вузла має три складових: u , v і w .

$$\{d\}_i = \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix}.$$

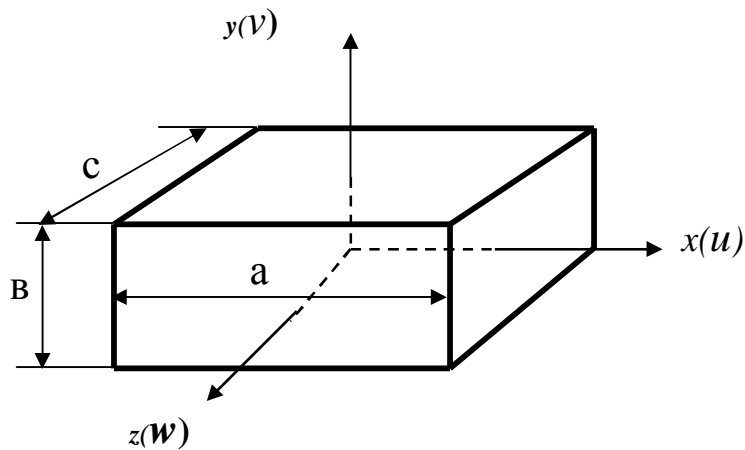


Рисунок 16.4– Кінцевий елемент у вигляді паралелепіпеда

Функцію переміщень представляємо у вигляді:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} N_1 & N_2 & \dots & N_8 \end{bmatrix} \{d\}^T \quad (16.16)$$

де N_1, N_2, \dots, N_8 – функції форми, що змінюються від 0 до 1:

$$N_1 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta), \quad N_2 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta),$$

$$N_3 = \frac{1}{8}(1 + \xi)(1 + \eta)(1 + \zeta), \quad N_4 = \frac{1}{8}(1 + \xi)(1 - \eta)(1 + \zeta),$$

$$N_5 = \frac{1}{8}(1 - \xi)(1 - \eta)(1 - \zeta), \quad N_6 = \frac{1}{8}(1 - \xi)(1 + \eta)(1 - \zeta),$$

$$N_7 = \frac{1}{8}(1 + \xi)(1 + \eta)(1 - \zeta), \quad N_8 = \frac{1}{8}(1 + \xi)(1 - \eta)(1 - \zeta),$$

де $\xi = \frac{2x}{a}$; $\eta = \frac{2y}{b}$; $\zeta = \frac{2z}{c}$; $\{d\}^1$ – вектор переміщень вузлів

елемента, що складається з 24 переміщень (8 вузлів \times 3 степені вільності).

Підставивши функції переміщень (16.16) у залежності Коші (16.3), визначимо вектор деформації, куди входить матриця $|B|$.

Як і для плоскої задачі теорії пружності при виборі прямокутних елементів, в підінтегральному виразі матриці жорсткості містяться змінні, залежні від x , y , і z . Коефіцієнти матриці жорсткості в паралелепіпеді можна знайти у книзі [9].

17 РОЗРАХУНОК ВІБРОІЗОЛЬОВАНОГО ФУНДАМЕНТУ ШАБОТНОГО МОЛОТА

Крім міцністних розрахунків можливі й інші області застосування МКЕ. У двох завершальних розділах наведені приклади застосування методу до рішення динамічних задач (розділ 17), зокрема, задач з області теорії коливань і приклад рішення комплексної задачі ущільнення дискретного матеріалу в умовах нелінійної, нестационарної теплопровідності (розділ 18).

17.1 Модель розрахунку віброізольованого фундаменту

При роботі звичайних штампувальних молотів значна частина енергії, яка не використовується на деформацію поковки, витрачається на коливання шабota і фундаменту молоту і надає шкідливої дії на навколишні споруди і здоров'я людей. Коливання фундаменту впливають

також на точність виконання технологічних процесів, здійснюваних на машинах, розташованих навіть на відносно великій відстані від молота. Для молотів з вагою падаючих частин більше 88,3 кН це відстань більше 80 м [16]. Для пом'якшення шкідливого впливу коливань ґрунту використовують спеціальні важкі жорсткі фундаменти, що закладаються на великі глибини, – до материкових порід. Наприклад, молот з вагою падаючих частин 196,2 кН, вага якого з шаботом складає 6690,4 кН, встановлений на фундамент вагою 32373 кН [16]. Проте споруда таких фундаментів, вартість яких у деяких випадках більше вартості самих молотів, не завжди усуває шкідливу дію вібрацій ґрунту.

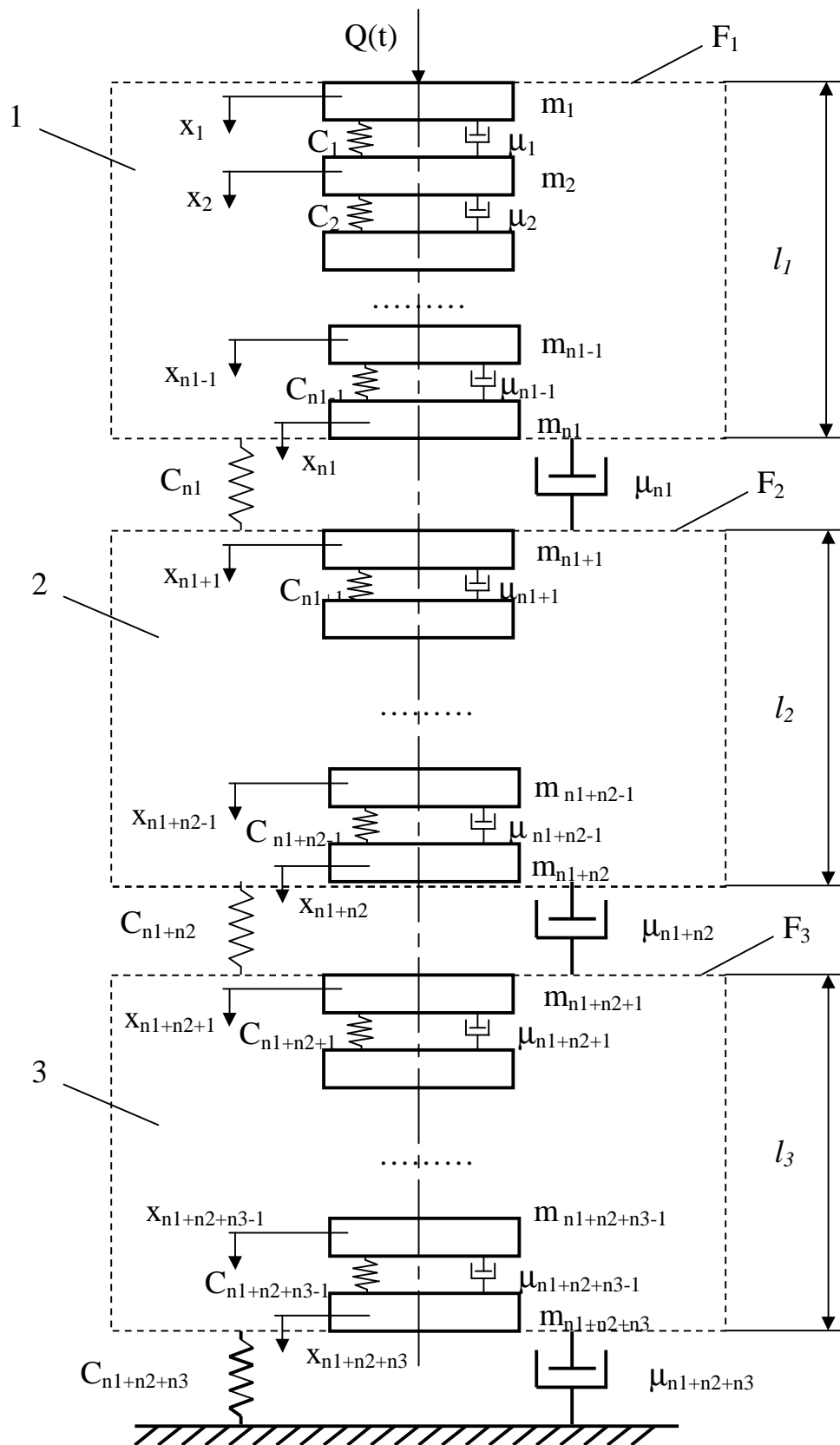
З метою зниження вібрацій застосовують віброізолювані фундаменти, в яких ізолюють масивний залізобетонний блок або безпосередньо шабот. Система віброізоляції може бути опорною, коли блок або шабот встановлені на віброізолювані опори, або підвісною, коли блок або шабот вивішені на віброізолюючих елементах.

Усі реальні механічні системи – це системи з розподіленими параметрами. Розрахунок таких систем складний. Тому, при складанні розрахункової схеми конструкції, її розподілену масу часто замінюють деякою кількістю зосереджених мас. Розрахунок системи із зосередженими параметрами виявляється простішим. Виходячи з цього, шабот, інерційний і фундаментний блоки замінимо їх дискретними аналогами. Схема такого дискретного аналога зображена на рис. 17.1. Надалі шабот, інерційний блок і фундамент умовно називатимемо блоками.

Заміна в розрахунковій моделі віброізоляції молота блоків матеріальними точками, як, наприклад, в статті [19], є дуже грубою. У даному розділі розглядається модель, в якій система з розподіленими параметрами розбита на чималу кількість дискретних елементів.

Кожен блок розбиваємо на n_i зосереджених мас, зв'язаних пружними і демпфуючими елементами, які моделюють пружні й демпфуючі властивості самого матеріалу. Коефіцієнти жорсткості пружних елементів між дискретними масами можна визначити за формулами:

$$\begin{aligned}
 C_1 = C_2 = \dots = C_{n_1-1} &= \frac{E_1 \cdot F_1}{l_1} (n_1 - 1); \\
 C_{n_1+1} = C_{n_1+2} = \dots = C_{n_1+n_2-1} &= \frac{E_2 \cdot F_2}{l_2} (n_2 - 1); \\
 C_{n_1+n_2+1} = C_{n_1+n_2+2} = \dots = C_{n_1+n_2+n_3-1} &= \frac{E_3 \cdot F_3}{l_3} (n_3 - 1).
 \end{aligned}
 \tag{17.1}$$



1 – шабот, 2 – інерційний блок, 3 – фундаментальний блок
 Рисунок 17.1 – Розрахункова схема: дискретна модель фундаменту

Тут E_j , F_j , l_j ($j=1,2,3$), – модуль пружності, площа поперечного перерізу, загальна висота шабета, інерційного блоку і фундаменту відповідно.

Унаслідок недосконалої пружності матеріалу відбувається природне поглинання енергії коливань в матеріалі. Ця енергія перетворюється в теплоту, яка потім розсівається. Демпфуюча здатність матеріалів залежить від багатьох чинників і розглядається як самостійна характеристика, що вимагає експериментального визначення з урахуванням реальних технологічних і експлуатаційних чинників. Демпфуюча здатність матеріалу зазвичай характеризується відносним розсіянням енергії, приблизно рівному подвоєному логарифмічному декременту коливань $\eta=\psi/2$ [20].

Використовуючи формули логарифмічного декременту коливань $h_i = b_i t_i$, коефіцієнта загасання $b_j = \mu_j / (2 * m_j)$ і умовного періоду затухаючих коливань $\tau_{1j} = 2p / \sqrt{c_j / m_j - b_j^2}$, встановимо залежність між коефіцієнтом опору μ і логарифмічним декрементом коливань.

Одержимо

$$m_i = \frac{h_i}{p} \sqrt{\frac{m_i c_i}{1 + \frac{h_i^2}{4p^2}}}. \quad (17.2)$$

Наведемо значення коефіцієнта відносного розсіяння енергії ψ для деяких матеріалів: сталь – 0,01...0,02; сірий чавун – 0,23; залізобетон – 0,25; пробка – 0,04; тканинна гума – 0,1..1,0 [20].

Таким чином, демпфуючу здатність матеріалу моделюватимемо внутрішнім лінійним тертям. Вважатимемо, що пружні елементи між блоками підкоряються закону Гука, а демпфери – закону лінійного опору. У цьому випадку матричне рівняння вимушених коливань системи має вигляд:

$$[m]\{\ddot{x}\} + [m]\{\dot{x}\} + [c]\{x\} = \{Q\}, \quad (17.3)$$

де $[m]$ – діагональна матриця мас точок системи;

$[m]$ – матриця узагальнених коефіцієнтів опору;

$[c]$ – матриця узагальнених коефіцієнтів жорсткості;

$\{x\}$ – вектор-стовпець узагальнених координат, відлічуваних від положення рівноваги;

$\{Q\}$ – вектор-стовпець вимушуючих сил.

У векторі $\{Q\}$ тільки перший елемент відмінний від нуля, інші – дорівнюють нулю, тобто

$$\{Q\} = \begin{pmatrix} F(t) \\ 0 \\ \dots \\ 0 \end{pmatrix}.$$

Для отримання матриці узагальнених коефіцієнтів опору скористаємося формулою

$$R_i = -m_{i-1} \ddot{x}_{i-1} + (m_{i-1} + m_i) \ddot{x}_i - m_i \ddot{x}_{i+1} \quad i = 1, 2, \dots, n, \quad (17.4)$$

де R_i – реакція в'язкого тертя, що діє на i -у точку,

n – загальна кількість матеріальних точок системи. У даній формулі для випадків $i=0$ або $i=n$ величини m_0 або \ddot{x}_{n+1} слід вважати нулями. Застосовуючи формулу (17.4) для всіх точок системи, одержимо матрицю узагальнених коефіцієнтів опору:

$$[m] = \begin{pmatrix} m_1 & -m_1 & 0 & \cdot & 0 & 0 \\ -m_1 & m_{1+2}m & -m_2 & \cdot & 0 & 0 \\ 0 & -m_2 & m_{2+3}m & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & m_{n-2+n-1}m & -m_{n-1} \\ 0 & 0 & 0 & \cdot & -m_{n-1} & m_{n-1+n} \end{pmatrix}. \quad (17.5)$$

Узагальнені коефіцієнти жорсткості знайдемо за формулою

$$F_{пружi} = -C_{i-1} x_{i-1} + (C_{i-1} + C_i) x_i - C_i x_{i+1}, \quad i = 1, 2, \dots, n, \quad (17.6)$$

де $F_{пруж.i}$ – реакція пружних елементів, що діє на i -ю матеріальну точку.

Для випадків $i=0$ або $i=n$ величини C_0 або x_{n+1} слід вважати нулями. Застосовуючи формулу (17.4) для всіх тіл системи, одержимо матрицю узагальнених коефіцієнтів жорсткості:

$$[c] = \begin{matrix} \begin{matrix} C_1 & -C_1 & 0 & \cdot & 0 & 0 \\ -C_1 & C_1+C_2 & -C_2 & \cdot & 0 & 0 \\ 0 & -C_2 & C_2+C_3 & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & 0 & 0 & \cdot & C_{n-2}+C_{n-1} & -C_{n-1} \\ 0 & 0 & 0 & \cdot & -C_{n-1} & C_{n-1}+C_n \end{matrix} & \cdot & \end{matrix} \quad (17.7)$$

Періодичну вимушуючу силу, яка прикладається до системи падаючими частинами молота, розрахуємо, виходячи з імпульсу, який вони передають. Під час розв’язання задачі були випробувані різні форми

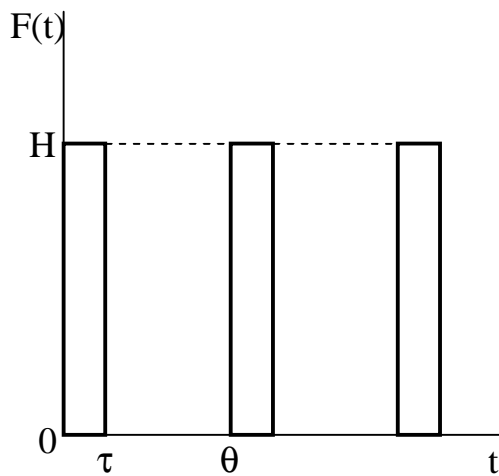


Рисунок 17.2 – Графік збудованої сили

надання ударного імпульсу (прямокутна, трикутна, пілкоподібна, «зрізана» синусоїда). З’ясувалося, що основними параметрами, які впливають на амплітуду і закон руху точок системи, є амплітуда, період і час дії збудовувальної сили. Форма імпульсу на результат рішення впливає трохи. Тому представимо силу у вигляді прямокутного імпульсу висотою Н

(рис. 17.2), час дії сили – τ , період – q .

Формула для обчислення імпульсу, який передається бабою молота фундаменту:

$$S = mv(1 + k), \quad (17.8)$$

де S – передаваний імпульс, m – маса падаючих частин молота, k – коефіцієнт відновлення при ударі.

Амплітуда зовнішньої сили одержана за формулою

$$H = \frac{S}{t}. \quad (17.9)$$

Аналітична формула, що описує зовнішню силу:

$$F(t) = \begin{cases} H, \text{ якщо } (t - \text{trunc}(\frac{t}{q}) * q) < t; \\ 0 \text{ інакше.} \end{cases} \quad (17.10)$$

Тут $\text{trunc}(\frac{t}{q})$ – функція узяття цілої частини числа.

На ґрунт передаватиметься сила

$$N = C_{n_1+n_2+n_3} \cdot x_{n_1+n_2+n_3} + m_{n_1+n_2+n_3} \cdot \ddot{x}_{n_1+n_2+n_3}. \quad (17.11)$$

Величину N_{max}/H називають коефіцієнтом передачі сили (КПС).

17.2 Приклад розрахунку і аналіз коливань

Для розрахунку реального фундаменту візьмемо шабот, інерційний блок і фундамент, що складаються з 2-х зосереджених мас кожен ($n_1 = 2$; $n_2 = 2$; $n_3=2$). Їх параметри: шабот – довжина 3,6 м, ширина 1,9 м, висота 2,135 м, матеріал – сірий чавун, маса 63 т, модуль пружності $E_1=1,35 \cdot 10^5$ МПа; інерційний блок – довжина 4,5 м, ширина 2,4 м, висота 4,5 м, матеріал – залізобетон, маса 102 т, модуль пружності $E_2=0,2 \cdot 10^5$ МПа; фундамент – довжина 5,4 м, ширина 2,8 м, висота 8,6 м, матеріал – залізобетон, маса 189 т, модуль пружності $E_3=0,2 \cdot 10^5$ МПа.

Маса падаючих частин молота складає 3000 кг, їх швидкість – 7 м/с, коефіцієнт відновлення при ударі – 0,25. Час дії сили $t = 0,1$ с, період сили $q = 10t$. За формулами (17.1) розрахуємо жорсткості пружних елементів шабота, інерційного блоку, фундаменту, за формулою (17.2) одержимо

коефіцієнти опору між елементами системи. Візьмемо: $C_4 = 1 \cdot 10^7$ Н/м; $C = 1 \cdot 10^8$ Н/м; $C_{12} = 4,5 \cdot 10^8$ Н/м; $\mu_4 = 4,2 \cdot 10^5$ Н·с/м; $\mu_{8=} 5,5 \cdot 10^5$ Н·с/м; $\mu_{12} = 4,5 \cdot 10^7$ Н·с/м (C_{12} і μ_{12} відповідають ґрунту середньої рихлості [16]). Вважаємо, що всі зосереджені маси мали нульові початкові умови: $x_i(0) = 0$; $\dot{x}_i(0) = 0$, $i=1,2,\dots,12$.

Підставивши одержані коефіцієнти у формули (17.5) і (17.7), одержимо матриці узагальнених коефіцієнтів інерції і жорсткості. За формулами (17.8), (17.9) і (17.10) одержимо величину імпульсу, амплітуду і аналітичний вираз для зовнішньої сили. Підставимо одержані матриці узагальнених коефіцієнтів у систему рівнянь (17.3) і одержану систему 12 диференціальних рівнянь. Її вирішимо чисельним методом Рунге-Кутта з автоматичним підбором кроку інтегрування для t в інтервалі $[0; 2q]$ с. Результати рішення наведені на рис. 17.3 і 17.4.

Аналізуючи отримані результати, дійдемо висновку, що з часом близько 0,5–1 с. Власні коливання системи затухають, і залишається тільки вимушена складова коливань.

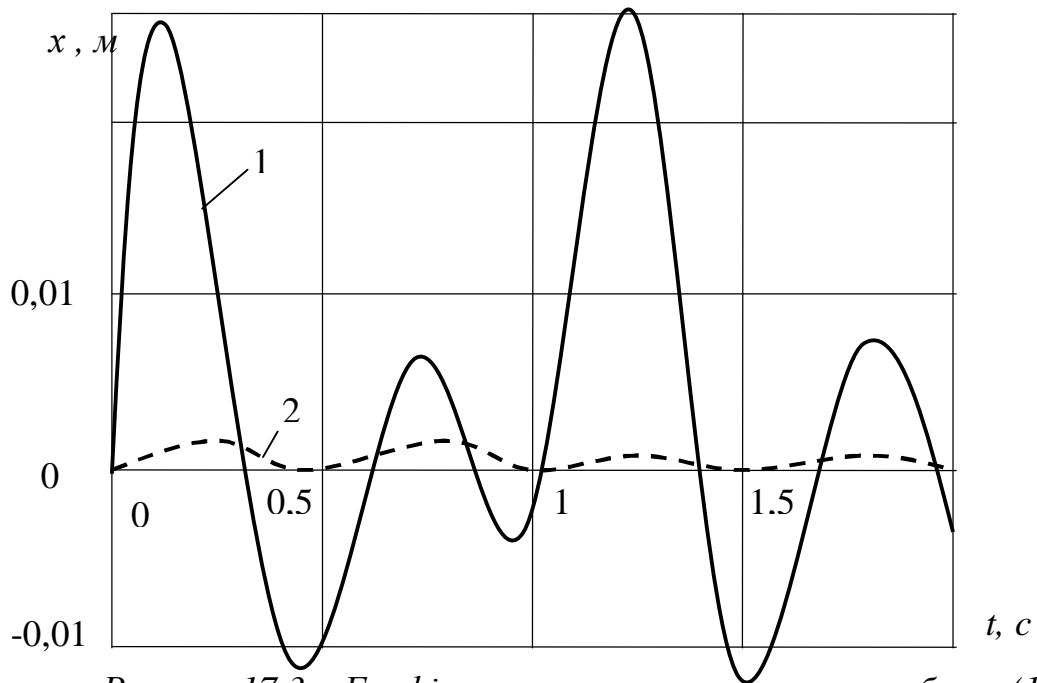


Рисунок 17.3 – Графік коливань верхнього краю шабота (1) і нижнього краю фундаменту (2)

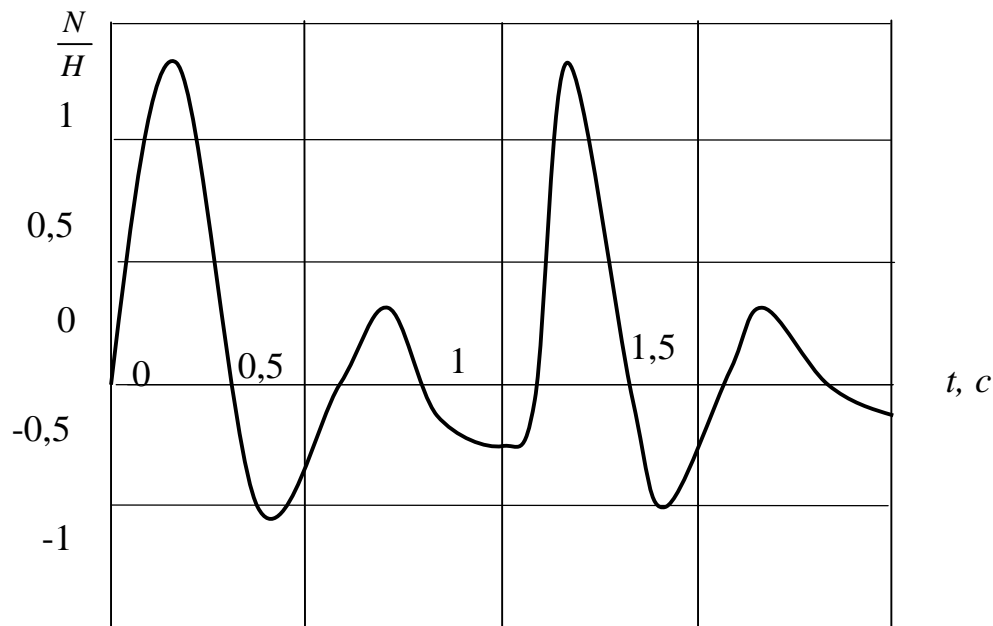


Рисунок 17.4 – Графік залежності відношення N/H від часу

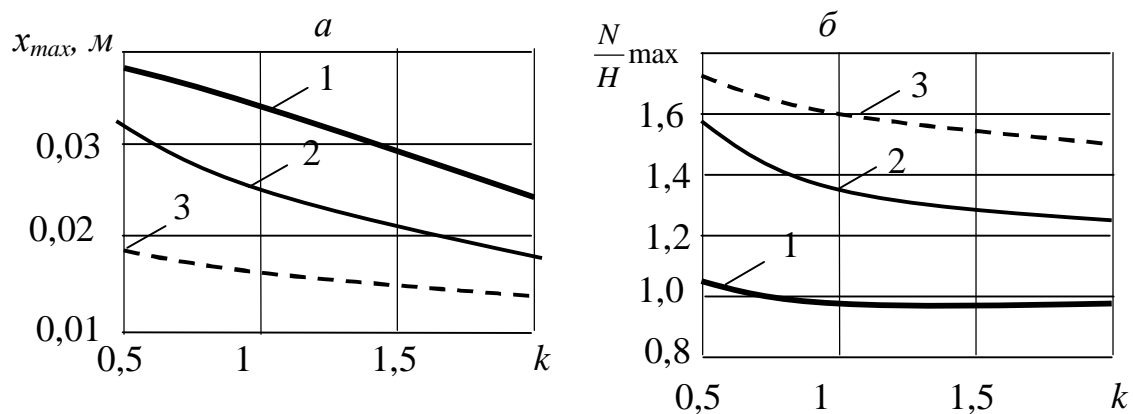
До моменту завдання наступного удару амплітуда коливань шабота зменшується у 10 разів. З рис. 17.4 витікає, що максимальна сила тиску на ґрунт трохи більше амплітудного значення зовнішньої збуджуючої сили. Проте, враховуючи велику площу підстави фундаменту, визначили, що додатковий тиск на ґрунт невеликий і складає приблизно $1/10$ тиску, що надається власною вагою фундаменту.

Результати розрахунку показали, що коливання кожного з елементів шаботу, інерційного блоку і підстави фундаменту практично співпадають. Це пояснюється великими коефіцієнтами жорсткості і опору між елементами блоків. Крім того, розбиття блоків на більшу кількість елементів (5, 6, 8) практично не впливає на результат. Під час проведеної роботи ми з'ясували, що при розрахунку віброізолювані фундаментів доцільно розбивати блоки на 2 зосереджені маси з в'язкопружним зв'язком.

У роботі як модель віброізолюваного фундаменту шаботного молота була розглянута система з 6 степенями вільності і вирішувалася система 6 диференціальних рівнянь другого порядку.

Проведено дослідження впливу коефіцієнтів опору між блоками на характер руху системи і КПС при різних коефіцієнтах жорсткості. Як номінальні прийняті значення коефіцієнтів опору і жорсткості, наведені

вище. Результати обчислень подані на рис. 17.5. На цих графіках криві 1, 2, 3 відповідають коефіцієнтам жорсткості при значеннях, які дорівнюють 0,5, 1 і 2 від номінальних. Буквою k позначено відношення коефіцієнтів жорсткості до номінальних. Значення коефіцієнта k варіювалися у межах від 0,5 до 2. Аналізуючи одержані графіки, дістанемо висновок, що збільшення коефіцієнтів опору викликає зменшення амплітуди коливань і КПС, тоді як збільшення коефіцієнтів жорсткості викликає зменшення амплітуди і зростання КПС.



1 – при $k=0.5$, 2 – при $k=1$, 3 – при $k=2$, де k – доля від номінальних значень коефіцієнтів жорсткості

Рисунок 17.5 – Графіки залежності амплітуди коливань (а) і максимального значення КПС (б) від коефіцієнтів опору (в долях від номінальних значень)

Розрахунок динамічної поведінки конструкції полягає у визначенні переміщень і напружень як функцій часу. Динамічний розрахунок може як попередній етап містити дослідження власних коливань, внаслідок чого визначаються частоти і форми власних коливань конструкції; в деяких випадках ця інформація викликає і самостійний інтерес. Інший підхід полягає в прямому інтегруванні матричного рівняння руху за допомогою тих або інших чисельних процедур.

Розрахунок власних коливань вимагає у разі систем великого розміру вельми великих витрат машинного часу. Тому рішення динамічних задач методом розкладання за власними формами доцільно виконувати у тому випадку, коли для отримання прийнятної точності результатів досить обмежитися врахуванням лише декількох основних тонів коливань. Проте у

багатьох випадках (наприклад, при розрахунку складних стрижневих або оболонкових конструкцій) потрібно враховувати велику кількість тонів власних коливань, і метод розкладання за власними формами стає неефективним. У цих випадках більш економічним виявляється пряме інтегрування диференціального рівняння

$$Mv' + Cv' + Kv = P \quad (17.12)$$

за допомогою тієї або іншої чисельної процедури. У рівнянні (17.12) вузлові сили, еквівалентні поза вузловому навантаженню, включені для спрощення запису до матриці P . У цьому рівнянні M – матриця мас конструкції, C – матриця демпфування, K – матриця жорсткості, а v – вектор переміщень.

Розглянемо стрижневу систему. Вільний стрижень (рис. 17.6), що має довжину l , площу перерізу F , погонну масу q_m , здійснює подовжні коливання. Розіб'ємо стрижень на n елементів (рис. 17.1) з довжиною l_i елемента l_i . При цьому кількість вузлів $n+1$. Оскільки розглядаються подовжні коливання стрижня, то кожен вузол зміщуватиметься лише в осьовому напрямку. Одержана модель має $n+1$ ступінь вільності. Переміщення кожного вузла v_i , а також швидкість і прискорення, і необхідно визначати у будь-який момент часу t .

Для використовуваних кінцевих елементів матриця жорсткості, згідно з

$$[17], \text{ має вигляд: } k^e = \frac{EF}{l^e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, k_i \text{ де } l^e - \text{довжина елемента.} \quad (17.13)$$

Матриця мас кінцевого елемента, згідно з формулою [17], має вигляд:

$$m^e = \frac{q_m l^e}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad (17.14)$$

де m^e – матриця мас кінцевого елемента, l^e – довжина елемента.

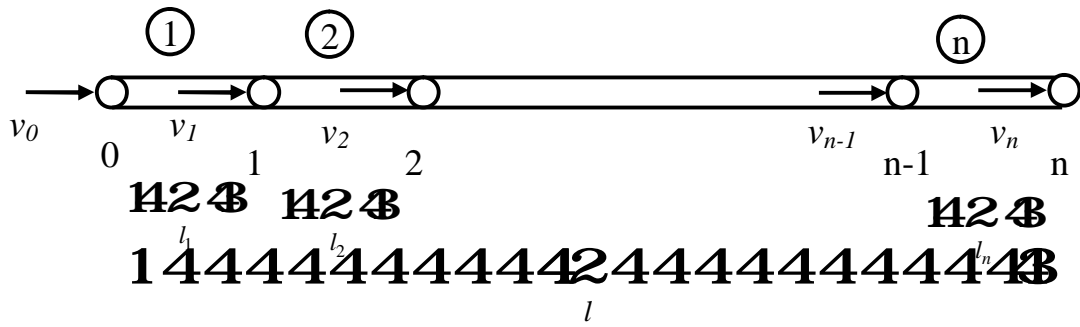


Рисунок 17.6 – Стрижнева система, що здійснює подовжні коливання

Добуток $q_m l^e$ є безпосередньо масою кінцевого елемента. Тоді матрицю мас можна подати у вигляді

$$m^e = \frac{m_i}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} \frac{m_i}{3} & \frac{m_i}{6} \\ \frac{m_i}{6} & \frac{m_i}{3} \end{bmatrix}, \quad (17.15)$$

де m_i – маса i кінцевого елемента.

Перетворюємо до аналогічного вигляду i матрицю жорсткості кінцевого елемента, виконавши заміну

$$k_i = \frac{EF}{l^e}, \quad (17.16)$$

де k_i – пружність (жорсткість) кінцевого елемента i .

Тоді матриця жорсткості кінцевого елемента набуде вигляду:

$$k^e = k_i \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} k_i & -k_i \\ -k_i & k_i \end{bmatrix}. \quad (17.17)$$

Варто відзначити, що матриця демпфування кінцевого елемента має структуру аналогічну структурі матриці жорсткості кінцевого елемента:

$$c^e = c_i \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} c_i & -c_i \\ -c_i & c_i \end{bmatrix}, \quad (17.18)$$

де c_i – в'язкість i кінцевого елемента. Сформуємо матрицю мас всієї системи, вона набуде вигляду:

$$M = \begin{bmatrix} \frac{m_1}{3} & \frac{m_1}{6} & & & & & \\ \frac{m_1}{6} & \frac{m_1}{3} + \frac{m_2}{3} & \frac{m_2}{6} & & & & \\ & \frac{m_2}{6} & \frac{m_2}{3} + \frac{m_3}{3} & \frac{m_3}{6} & & & \\ & & \frac{m_3}{6} & \frac{m_3}{3} + \frac{m_4}{3} & & & \\ & & & & & & \\ & & & & & \frac{m_{n-1}}{3} + \frac{m_n}{3} & \frac{m_n}{6} \\ & & & & & \frac{m_n}{6} & \frac{m_n}{3} \end{bmatrix} .$$

Сформуємо матрицю жорсткості всієї системи, вона набуде вигляду:

$$K = \begin{bmatrix} k_1 & -k_1 & & & & & \\ -k_1 & k_1 + k_2 & -k_2 & & & & \\ & -k_2 & k_2 + k_3 & -k_3 & & & \\ & & -k_3 & k_3 + k_4 & & & \\ & & & & & & \\ & & & & & k_{n-1} + k_n & -k_n \\ & & & & & -k_n & k_n \end{bmatrix} .$$

Сформуємо матрицю демпфіювання всієї системи, вона буде такою:

$$C = \begin{bmatrix} c_1 & -c_1 & & & & & \\ -c_1 & c_1 + c_2 & -c_2 & & & & \\ & -c_2 & c_2 + c_3 & -c_3 & & & \\ & & -c_3 & c_3 + c_4 & & & \\ & & & & & & \\ & & & & & & \\ & & & & & c_{n-1} + c_n & -c_n \\ & & & & & -c_n & c_n \end{bmatrix} \cdot$$

Підставивши одержані матриці в рівняння (17.12), ми повинні будемо знайти значення векторів v, v', v'' .

Оскільки коливання вільні, то вектор зовнішніх сил, прикладених до вузлів, дорівнює нулю, і рівняння (17.1) набуде вигляду:

$$Mv'' + Cv' + Kv = 0. \tag{17.19}$$

Розглядатимемо систему із закріпленим нульовим вузлом. Система виводиться з рівноваги, коли в початковий момент часу n вузла надана початкова швидкість V_0 .

Новий вигляд моделі поданий на рис. 17.7.

До внутрішніх параметрів об'єкта відносяться ті параметри, які описують стан або властивості самого об'єкта (табл. 17.1).

Таблиця 17.1 – Властивості кінцевого елемента

№	Позначення	Опис
1	m	Маса
2	k	Жорсткість
3	c	В'язкість

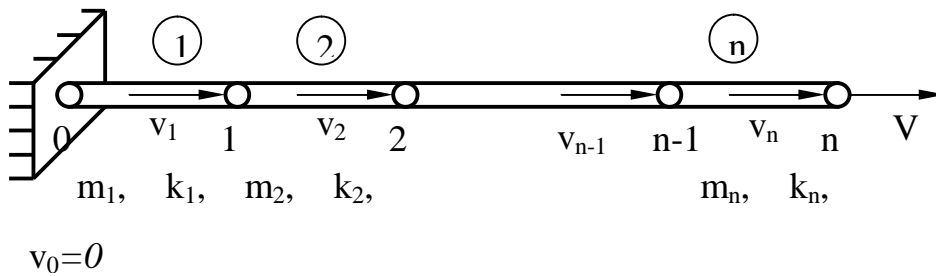


Рисунок 17.7 – Стрижньова система, що здійснює подовжні коливання, із закріпленням нульовим вузлом

До зовнішніх параметрів моделі можна віднести параметри, що управляють або змінюють стан об'єкта. До такого роду параметрам відносяться, наприклад, початкові і граничні умови, подані в таблицях 17.2 і 17.3 відповідно.

Таблиця 17.2 – Початкові умови

№	Позначення	Опис
1	V_0	Початкова швидкість n вузла
2	Δt	Шаг змінення часу

Таблиця 17.3 – Граничні умови

№	Позначення	Опис
1	$v_0(t)=0;$ $v'_0(t)=0; v''_0(t)=0$	Умови закріплення нульового вузла
2	FinishTime	Кінцевий час інтегрування

Розглянемо методи інтегрування крокового типу. У цих методах матриці V_{i+1} , v'_{i+1} , v''_{i+1} , що відносяться до деякого моменту часу $t_{i+1} = t_1 + \Delta t$ (Δt – крок інтегрування), обчислюються через їх значення, знайдені на попередніх кроках.

Чисельне інтегрування можна застосувати і в методі розкладання за власними формами. З іншого боку, всяка матриця v може бути єдиним чином подана у вигляді розкладання за формами власних коливань (якщо включити до цього розкладання всі форми). Процес чисельного інтегрування рівняння (17.19) можна розуміти як неявне покрокове

визначення коефіцієнтів цього розкладання, що є функціями часу. Таким чином, пряме інтегрування рівняння (17.19) можна трактувати як неявне і одночасне виконання тих же обчислень, які в методі розкладання за власними формами здійснюються роздільно для кожного тону коливань.

Є, проте, і принципові відмінності між двома вказаними підходами до розрахунку динамічної поведінки конструкції. По-перше, в методі прямого інтегрування, на відміну від методу розкладання за власними формами, враховуються всі без виключення тони коливань, зокрема коливання з найвищими частотами. По-друге, інтегрування виконується тут з однаковим кроком по всіх тонах.

Взагалі кажучи, крок Δt при прямому інтегруванні повинен бути таким, щоб з достатньою точністю відтворювалися ті коливання, які грають найбільш істотну роль в динамічній поведінці конструкції, тобто коливання з відносно низькими частотами. Але зазвичай в спектрі частот кінцевоелементної моделі містяться і вельми високі частоти, і може статися, що вибраний крок значно перевершує періоди відповідних коливань. Ясно, що внески коливань з найвищими частотами до динамічної поведінки конструкції будуть в цьому випадку абсолютно спотворені, але це допустимо, оскільки, як було раніше сказано, вони не грають в цілому скільки-небудь істотній ролі. Проте важливо, щоб використовувана процедура інтегрування забезпечувала стійкість процесу, коли крок Δt як завгодно перевершує період коливання. Методи інтегрування, що задовольняють цій вимозі, називаються, безумовно стійкими. Для них єдиним критерієм вибору кроку Δt є точність результатів.

На практиці використовуються іноді й умовно стійкі методи інтегрування, в яких стійкість процесу забезпечується лише у тому випадку, коли крок Δt не перевершує деякого критичного значення Δt^* . Якщо ж узяти $\Delta t > \Delta t^*$, то відбувається стрімке зростання чисельних значень компонент матриці $v(t)$, які вже після декількох перших кроків можуть досягати скільки завгодно великих величин. Найчастіше величина Δt^* виявляється значно менше за те значення, яке необхідне для отримання прийнятної точності в безумовно стійких процедурах. Проте, в окремих випадках умовно стійкі процедури можуть виявитися більш економічними, якщо на кожному кроці вони вимагають, як це часто буває, меншої кількості арифметичних

операцій, чим безумовно стійкі процедури. Існують широко відомі методи чисельної інтегрування звичайних диференціальних рівнянь типу Адамса, Руунге-Кутта та ін. Проте вони мало придатні для інтегрування систем високого порядку, оскільки, будучи умовно стійкими, вимагають, проте, виконання великої кількості арифметичних операцій на кожному кроці. У зв'язку з цим стосовно матричного рівняння вигляду (17.8) розроблено декілька спеціальних процедур, серед яких метод Ньюмарка.

Метод Ньюмарка заснований на наступних (незалежних) розкладаннях $v(t_i + t)$ і $v'(t_i + t)$ у ряди за степенями t :

$$\begin{aligned} v(t_i + t) &= v_i + tv'_i + \frac{t^2}{2}v''_i + at^3v'''_i; \\ v'(t_i + t) &= v'_i + tv''_i + bt^2v'''_i. \end{aligned} \quad (17.20)$$

Тут утримані члени, що містять матрицю v''' . Коефіцієнти α і β вибираються так, щоб забезпечити безумовну стійкість процесу інтегрування; про їх чисельні значення буде сказано нижче.

Вважаючи $t = \Delta t$ і замінюючи приблизно $v''' = (v''_{i+1} - v''_i) / \Delta t$, приходимо до рівності:

$$\begin{aligned} v_{i+1} &= v_i + \Delta tv'_i + \frac{\Delta t^2}{2}v''_i + a\Delta t^2(v''_{i+1} - v''_i); \\ v'_{i+1} &= v'_i + \Delta tv''_i + b\Delta t(v''_{i+1} - v''_i). \end{aligned} \quad (17.21)$$

Знайдемо з першої рівності v''_{i+1} :

$$v''_{i+1} = \frac{1}{a\Delta t^2}(v_{i+1} - v_i) - \frac{1}{a\Delta t}v'_i + \left(1 - \frac{1}{2a}\right)v''_i. \quad (17.22)$$

Підстановка цього результату у вираз для v'_{i+1} дає формулу

$$v_{i+1} = \frac{b}{a\Delta t}(v_{i+1} - v_i) + \left(1 - \frac{b}{a}\right)v'_i + \frac{\Delta t}{2}\left(2 - \frac{b}{a}\right)v''_i, \quad (17.23)$$

яка в сукупності з (17.11) складає рекурентні співвідношення методу Ньюмарка.

Для відшукування V_{i+1} рівняння руху записується для моменту $t = t_{i+1}$:

$$Mv''_{i+1} + Cv'_{i+1} + Kv_{i+1} = P_{i+1}. \quad (17.24)$$

Після підстановки в цю рівність співвідношень (17.22), (17.23) приходимо до рівняння вигляду

$$Av_{i+1} = R_{i+1}, \quad (17.25)$$

де
$$A = K + \frac{b}{a\Delta t}C + \frac{1}{a\Delta t^2}M; \quad (17.26)$$

$$R_{i+1} = M \left[\frac{1}{a\Delta t^2}v_i + \frac{1}{a\Delta t}v'_i + \left(\frac{1}{2a} - 1\right)v''_i \right] + C \left[\frac{b}{a\Delta t}v_i + \left(\frac{b}{a} - 1\right)v'_i + \frac{\Delta t}{2} \left(\frac{b}{a} - 2\right)v''_i \right]. \quad (17.27)$$

Відзначимо, що метод Ньюмарка вимагає завдання в початковий момент часу $t_0 = 0$ не тільки матриці V_0, V'_0 , але також і матриці V''_0 . Зазвичай вважають $v'' = 0$; можна також знайти Mv''_0 з рівняння руху, записаного для моменту часу $t_0 = 0$.

Метод інтегрування Ньюмарка буде безумовним стійким, якщо

$$b \geq 0.5; \quad a \geq 0.25(b + 0.5)^2. \quad (17.28)$$

Для того, щоб врахувати умови закріплення нульового вузла стрижня, в одержаній матриці A (17.26) обнуляють перший рядок і перший стовпець. Потім на перетині першого рядка і першого стовпця ставлять одиницю (рішення методом Гауса припускає відмінність головної діагоналі матриці від нуля).

Для розрахунку реального фундаменту візьмемо шабот, інерційний блок і фундамент що складаються з 2-х зосереджених мас кожен, як в прикладі розділу 17.1. Чисельні значення параметрів теж беремо з прикладу розділу 17.1. Схема розрахункової моделі зображені на рис. 17.8.

Таким чином, одержуємо початкові дані для розрахунку:

- кількість кінцевих елементів 9;
- кількість вузлів 10;
- значення мас, жорсткості, в'язкості кінцевих елементів зведені в таблицю 17.4;
- початкова швидкість шабота 0.33 м/с.

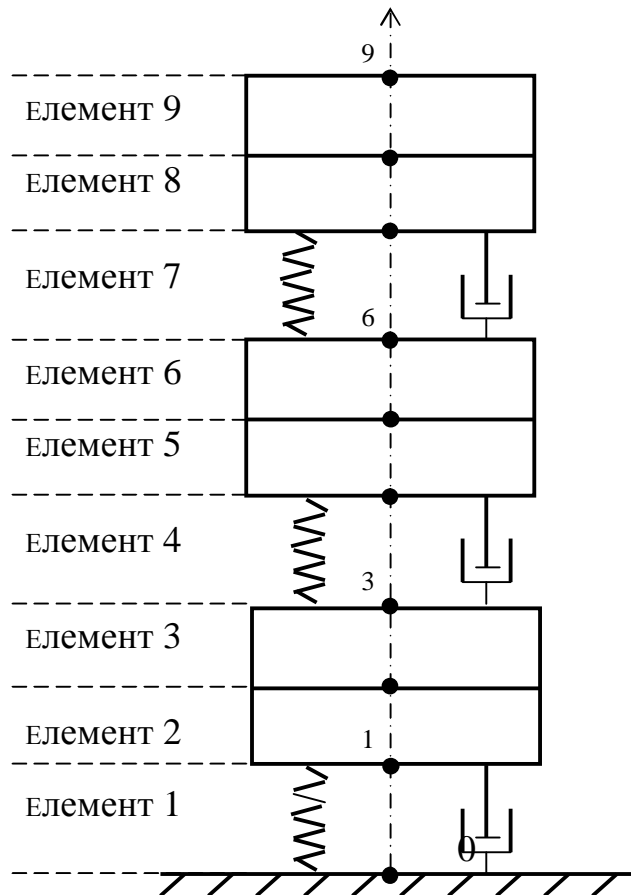


Рисунок 17.8 – Схема розрахункової моделі

Таблиця 17.4 – Початкові дані для розрахунку коливань кінцевих елементів

Параметр	Ел. 1	Ел. 2	Ел. 3	Ел. 4	Ел. 5	Ел. 6	Ел. 7	Ел. 8	Ел. 9
Маса, т	0	94,5	94,5	0	51	51	0	31,5	31,5
Жорсткість, $\times 10^7$ Н/м	45	7000	7000	10	9600	9600	1	86500	86500
В'язкість, $\times 10^5$ Н*с/м	450	0	0	5,5	0	0	4,2	0	0

Виконавши розрахунок, використовуючи метод Ньюмарка, одержимо наступні траєкторії коливань вузла 9 (верхня точка шабота) і вузла 1 (нижня точка фундаменту), як подано на рис. 17.8.

Колівання (див. рис. 17.9) повторюють коливання, одержані в прикладі розділу 17.2 (рис. 17.3), якщо врахувати, що вони вільні. Як видно, відбувається практично повне загасання до моменту часу $t=1c$, що сприятливо для завдання наступного удару (якщо брати вимушені коливання).

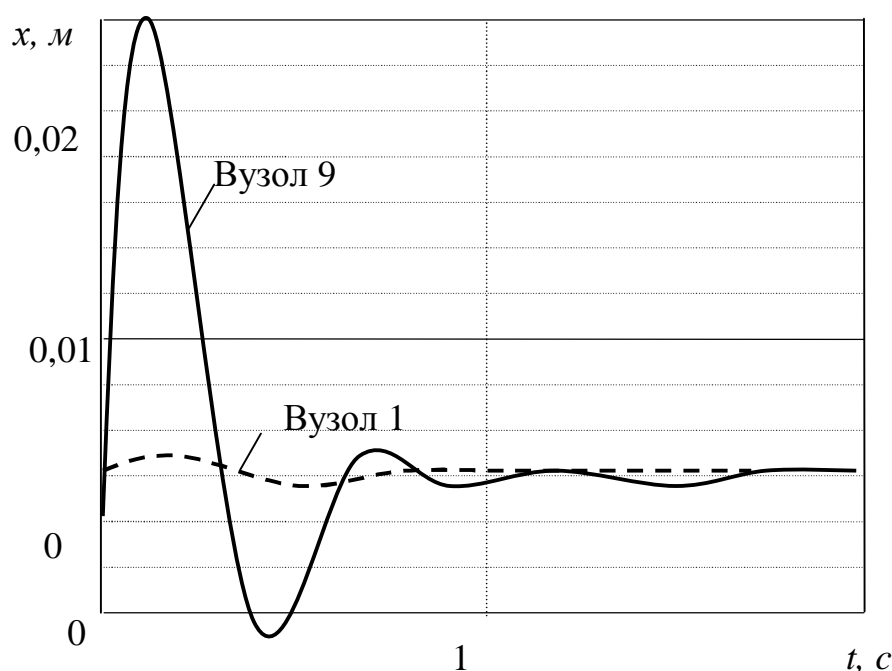


Рисунок 17.9 – Графік коливань верхнього краю шабота (вузол 9) і нижнього краю фундаменту (вузол 1)

Результати розрахунку показали, що коливання вузлів 1, 2, 3 (фундамент) практично співпадають, аналогічно і для 4, 5, 6 (інерційний блок), 7, 8, 9 (шабот). Це пояснюється великими коефіцієнтами жорсткості і опору між елементами блоків. Розбиття блоків на більшу кількість елементів (5, 6, 8) практично не впливає на результат. Під час проведеної роботи ми з'ясували, що при розрахунку віброізованих фундаментів доцільно розбивати блоки на 2 зосереджені маси з в'язкопружним зв'язком.

У роботі виведена трьохмасова модель шаботного молота, визначені ключові параметри впливу на вібрацію, проведений аналіз цих параметрів. Також запропоновані оптимальні поєднання параметрів, які значно зменшують вібрацію.

Зроблений висновок про те, що для побудови адекватної математичної моделі системи віброізоляції шаботних молотів потрібно враховувати нелінійності системи. Також розроблена модель розрахунку віброізолюваного фундаменту, що враховує пружнодемпфуючі властивості шабота, інерційного блоку і фундаменту.

18 РОЗРАХУНОК ПРОЦЕСУ УЩІЛЬНЕННЯ ДИСКРЕТНОГО МАТЕРІАЛУ В УМОВАХ НЕЛІНІЙНОЇ НЕСТАЦІОНАРНОЇ ТЕПЛОПРОВІДНОСТІ

Даний розділ є прикладом застосування МКЕ в задачах механіки суцільного середовища.

Практичні потреби багатьох галузей сучасного машинобудування (авіаційної, космічної, хімічної, нафтохімічної, гірничорудної, електротехнічної, атомної і т.д.) постійно висувають нові завдання щодо створення матеріалів із заданими фізико-хімічними і механічними властивостями. З рішенням цієї задачі нерозривно пов'язані проблеми інтенсифікації виробничих-технологічних процесів розвитку сучасної техніки і устаткування, що працюють при підвищених температурах, тиску, швидкостях і т.п. Іншим найважливішим завданням є підвищення коефіцієнта використання металу, зниження матеріаломісткості конструкцій, зменшення відходів при механічній обробці. Рішення цих питань вимагає освоєння нових прогресивних методів отримання напівфабрикатів і готових виробів, зокрема ширшого використання деталей з порошків і гранул.

Одним з перспективних напрямків виготовлення деталей з металевих порошків і гранул є процеси гарячого ізостатичного пресування (ГІП). Процес ГІП призначений для обробки тиском порошкових і пористих матеріалів при сумісній дії високого тиску і температур, що зазвичай перевищують відповідно 100 МПа і 1000⁰ С. У даний час технологію ГІП

використовують в основному для таких матеріалів, обробка яких традиційними методами вимагає великих додаткових витрат: жароміцних сплавів, тугоплавких металів, твердих сплавів, швидкорізальних сталей, титанових, берилієвих, магнієвих сплавів, феритів, керметов, кераміки і т.п.

18.1 Отримання математичної моделі об'єкта

Розглянемо ізотропне тіло, температурні деформації елементарного об'єму якого дуже малі в порівнянні з самим об'ємом і температурне поле нестационарне. В умовах теплопередачі під впливом зовнішнього тиску відбувається стискання аналізованого тіла. Для елементарного об'єму процес може бути описаний такими рівняннями:

$$\frac{\partial}{\partial t} (r i) = -divq - div(r \bar{u} i) - p div \bar{u} ; \quad (18.1)$$

$$\frac{\partial r}{\partial t} + div(r \bar{u}) = 0 ; \quad (18.2)$$

$$\frac{\partial r}{\partial t} = F(r, p, T, t). \quad (18.3)$$

Рівняння (18.1) є узагальненим диференціальним рівнянням енергетичного балансу, рівняння (18.2) є рівнянням нерозривності суцільного середовища і останнє рівняння – рівняння стану пресованого матеріалу.

У рівняннях (18.1) -(18.3): r – відносна щільність, тобто відношення щільності матеріалу до щільності його твердої фази; i – питома внутрішня енергія елементарного об'єму; q – питомий тепловий потік; \bar{u} – локальні швидкості переміщення; p – зовнішній тиск; t – час; T – абсолютна температура.

Відповідно до закону Фур'є:

$$q = -l grad T , \quad (18.4)$$

де λ – коефіцієнт теплопровідності.

Питома внутрішня енергія i пов'язана з температурою T залежністю

$$di = cdT, \quad (18.5)$$

де c – питома теплоємність речовини.

Вирішуватимемо одновимірну задачу теплопровідності, тобто всі фізичні величини, які входять до рівняння, залежать тільки від однієї просторової координати x . У такій постановці, рівняння (18.1)...(18.5) набувають виду:

$$c r \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left(l \frac{\partial T}{\partial x} \right) - c r u \frac{\partial T}{\partial x} - p \frac{\partial u}{\partial x}; \quad (18.6)$$

$$\frac{\partial r}{\partial t} + \frac{\partial}{\partial x} (r u) = 0; \quad (18.7)$$

$$\frac{\partial r}{\partial t} = F(r, p, T, t). \quad (18.8)$$

До внутрішніх параметрів об'єкта відносяться ті параметри, які описують стан або властивості самого об'єкта. До такого роду параметрів відносяться, наприклад, властивості матеріалу, подані в таблиці 18.1.

Таблиця 18.1 – Властивості матеріалу

№	Позначення	Опис	Один. вим.	Значення
1	2	3	4	5
1	r_T	Абсолютна щільність матеріалу	кг/м ³	8400
2	k_r	Коефіцієнти визначають залежність l від щільності		1,0
3	J			1,0
4	l_0	Коефіцієнт теплопровідності при $T=300$ К (18.9)		13,0
5	b_l	Коефіцієнт пропорційності (18.9)		0,012
6	c_0	Коефіцієнт теплоємності при $T=300$ К Коефіцієнт пропорційності (18.10)		500
7	f_c	(18.10)		0,2

Продовження таблиці 18.1

1	2	3	4	5
8	Ω	Атомний об'єм		1,1E-27
9	s_0	Напруження текучості при T=300 К		200
10	b_s	Коефіцієнт пропорційності напруження текучості		-0,45
11	$T_{пл}$	Температура плавлення		1600
12	m_0	Модуль зсуву при T=300 К		8,3E04
13	b_m	Коефіцієнт пропорційності модуля зсуву		-0,5
14	D_{0v}	Коефіцієнт пропорційності об'ємної дифузії		1,6E-02
15	D_{0b}	Коефіцієнт пропорційності граничної дифузії		1,8E-13
16	Q_v	Енергія активації об'ємної дифузії		1,85E2
17	Q_b	Енергія активації граничної дифузії		1,15E2
18	D_z	Діаметр зерна		4,0E-5
19	R_p	Початковий середній радіус частини порошку		1,0E-4
20	k	Показник повзучості		3,6
21	a	Коефіцієнт a		0,638
22	m	Коефіцієнт m		1,09
23	n	Коефіцієнт n		1,97
24	A_0	Постійна Дорна		1,22E5
25	B	Вектор Бюргерса		1,5E-10

До зовнішніх параметрів моделі можна віднести параметри, що управляють або змінюють стан об'єкту. До такого роду параметрів відносяться, наприклад, початкові і граничні умови, представлені в таблицях 18.2 і 18.3 відповідно.

Таблиця 18.2 – Початкові умови

№	Позначення	Опис
1	l_0	Початкова довжина стрижня
2	$T_0 = T_0(x)$	Розподіл початкової температури вузлів стрижня
3	$r_0 = r_0(x)$	Розподіл відносної щільності вузлів стрижня в початковий момент часу

Таблиця 18.3 – Граничні умови

№	Позначення	Опис
1	$T_L = T_L(t)$ і $T_R = T_R(t)$	Закон зміни температури лівої і правої меж стрижня за часом
2	$p = p(t)$	Закон зміни зовнішнього тиску за часом

Рішення задачі ущільнення дискретного матеріалу в умовах одновимірної нелінійної нестационарної теплопровідності можна виконувати згідно з блок-схемою, поданою на рис. 18.1. Як видно з блок-схеми, в розв'язанні задачі розрізняються чотири основні розрахункові блоки:

- 1) розрахунок теплофізичних параметрів пористого матеріалу на i -му кроці;
- 2) розрахунок координат вузлів на i -му кроці;
- 3) розрахунок поля температур на i -му кроці;
- 4) розрахунок поля відносної щільності на i -му кроці.

На першому етапі відбувається розрахунок теплофізичних параметрів пористого матеріалу таких, як теплопровідність і теплоємність.

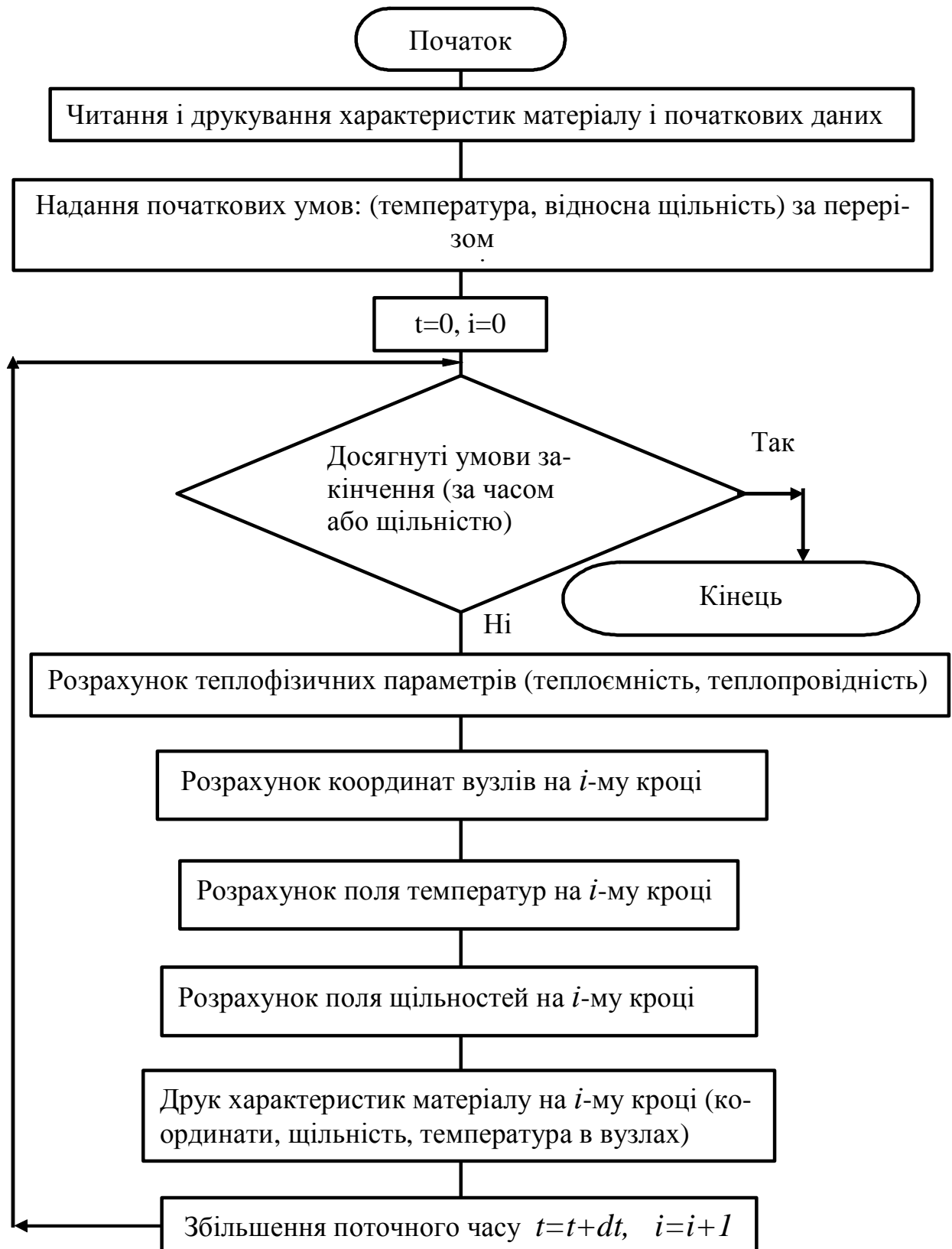


Рисунок 18.1 – Блок-схема алгоритму розв'язання задачі ущільнення дискретного матеріалу

На другому етапі відбувається перерахунок координат вузлів, унаслідок ущільнення дискретного матеріалу. Оскільки з кожним вузлом

асоційовано значення температури, щільність і т.д., то в процесі ущільнення координати вузлів міняються і їх необхідно перераховувати.

На третьому етапі розраховується поле температур в результаті рішення одновимірної нестационарної задачі теплопровідності. Завдання теплопровідності вирішується методом кінцевих елементів, який дозволяє визначити значення температури у вузлах перерізу за заданих граничних умов і теплофізичних параметрів матеріалу.

На четвертому етапі відбувається розрахунок поля відносної щільності. Ця задача вирішується за допомогою інтегрування диференціального рівняння стану (18.8) в межах тимчасового кроку.

Текст програми приведено у додатку Е.

18.2 Рішення задачі ущільнення дискретного матеріалу в умовах одновимірної нелінійної нестационарної теплопровідності

Для вирішення поставленого завдання необхідно уміти розраховувати теплофізичні параметри, такі як теплопровідність і теплоємність. Закони зміни цих параметрів для пористих матеріалів відрізняються від аналогічних законів для монолітних матеріалів. Від значень цих параметрів залежить характер передачі тепла усередині перерізу заготовки. При певних значеннях теплопровідності і теплоємності може виникнути великий градієнт температур усередині перерізу заготовки, що порушить структуру матеріалу. Тому, для отримання адекватних результатів дуже важливо уміти правильно враховувати вплив структури матеріалу на його теплофізичні властивості.

Для більшості використовуваних в техніці матеріалів, залежність коефіцієнта теплопровідності від температури можна апроксимувати залежністю [23]

$$l = l_0 n_1 n_2, \quad (18.9)$$

де l_0 – коефіцієнт теплопровідності при $T=300$ К;
 $n_1 = 1 + b(T - 300)$,

$n_2 = k_r r^j$ - коефіцієнти, що визначають залежність I від температури і відносної щільності (b і j - експериментальні коефіцієнти).

Питому теплоємність приблизно розраховуватимемо за формулою [24]

$$c = c_0 [1 + \psi (T - 300)], \quad (18.10)$$

де c_0 - коефіцієнт теплоємності при $T=300\text{K}$, ψ - експериментальний коефіцієнт.

У процесі ущільнення дискретного матеріалу відбувається зміна його щільності. Оскільки маса заготовки не змінюється, то відбувається зміна об'єму заготовки. Враховуючи також, що площа перерізу постійна, дістаємо висновку, що зміна щільності пресованого матеріалу викликає зміну довжини заготовки.

Для отримання адекватних результатів необхідно правильно враховувати вплив зміни довжини заготовки у процесі розрахунку МКЕ. Для врахування зміни заготовки існують два способи:

1) Використання кінцевих елементів постійної довжини. При цьому із зміною довжини заготовки відбувається зміна кількості кінцевих елементів.

2) Використання кінцевих елементів змінної довжини. При цьому із зміною довжини заготовки відбувається зміна довжин кінцевих елементів при постійній їх кількості.

Другий спосіб врахування зміни довжини виглядає переважніше, оскільки ідея зміни довжини кінцевого елемента краще відображає смисл процесу пресування. При розрахунку процесу кожному вузлу кінцевого елемента ставиться у відповідність декілька параметрів, таких як температура, щільність і інші. При ущільненні матеріалу заготовки відбувається зминання матеріалу, яке супроводжується переміщенням шарів матеріалу.

Розрахунок координат вузлів проводитимемо відповідно до наступних міркувань. Оскільки маса пресованого матеріалу залишається незмінна, то для i -го елемента стрижня, представленого на рис. 18.2, можна записати:

$$\overline{r}_i^0 \cdot h_i^0 = \overline{r}_i^1 \cdot h_i^1 \quad i = \overline{0, N-1}, \quad (18.11)$$

де $\overline{}$ – верхні індекси 0 і 1 означають попереднє і поточне значення;

\overline{r}_i – щільність і-го кінцевого елемента;

h_i - довжина і-го кінцевого елемента;

N - кількість кінцевих елементів.

Приймаючи, що щільність матеріалу змінюється по кінцевому елементу лінійно, одержимо наближений вираз для щільності кінцевого елемента:

$$\overline{r}_i = \frac{r_i + r_{i+1}}{2}. \quad (18.12)$$

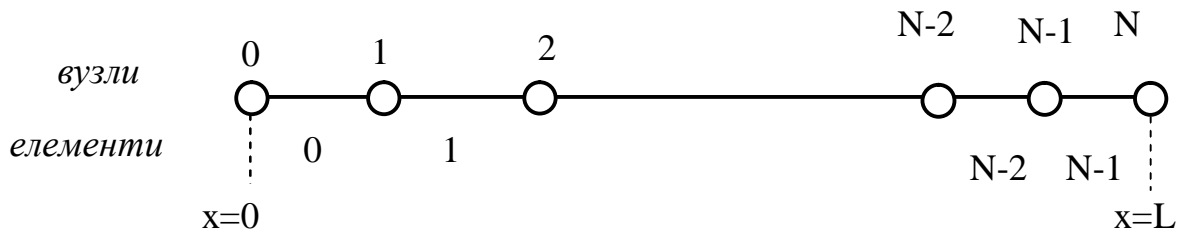


Рисунок 18.2 – Схема стрижня

Підставляючи вираз (18.11) в (18.12), можна одержати, що

$$h_i^1 = h_i^0 \cdot \frac{r_i^0 + r_{i+1}^0}{2} \cdot \frac{2}{r_i^1 + r_{i+1}^1} = h_i^0 \cdot \frac{r_i^0 + r_{i+1}^0}{r_i^1 + r_{i+1}^1} \quad i = \overline{0, N-1}. \quad (18.13)$$

Таким чином, за допомогою формули (18.13) можна визначити довжини кінцевих елементів на кожному подальшому тимчасовому кроці. Для знаходження координат вузлів кінцевих елементів, візьмемо що, координата 0-го вузла не змінюється і дорівнює 0. Таким чином, одержимо рекурентну формулу

$$x_0 = 0, \quad x_{i+1} = x_i + h_i \quad i = \overline{0, N-1}. \quad (18.14)$$

18.3 Розрахунок поля температур і рішення задачі теплопровідності методом кінцевих елементів

Як показано вище, математична модель об'єкта описується трьома рівняннями (18.6)...(18.8). Як базове рівняння доцільно вибрати рівняння (18.6), оскільки воно описує процес розповсюдження температури за перерізом заготовки. За відомими значеннями температури і фізичних параметрів у вузлах, з рівняння (18.8) знаходитимемо значення відносної щільності дискретного матеріалу.

Вирішуватимемо одновимірну нестационарну задачу теплопровідності на відрізку $0 \leq x \leq L$ [18]. Передбачається, що відомі:

- 1) залежності фізичних властивостей матеріалу, таких як теплопровідність і теплоємність;
- 2) початкові умови – розподіл температури в початковий момент часу;
- 3) граничні умови – закони зміни температури на правій і лівій межах відповідно.

Таким чином, потрібно вирішити одновимірне нестационарне рівняння теплопровідності

$$\frac{\partial}{\partial x} \left(l \frac{\partial j}{\partial x} \right) - cr \frac{\partial j}{\partial t} = 0 \quad (18.15)$$

з початковими даними

$$j_0 = j_0(x) \quad (18.16)$$

і граничними умовами

$$j_L = j_L(t) \text{ і } j_R = j_R(t) \text{ при } t \geq 0, \quad (18.17)$$

де j – функція температури; l – функція теплопровідності; c – функція теплоємності; r – функція відносної щільності.

Використовуємо МКЕ і розіб'ємо просторову область (відрізок $0 \leq x \leq L$) на N елементів з нумерацією вузлів і елементів, показаною на рис. 18.3. Застосовуючи метод часткової дискретизації запишемо апроксимацію у вигляді

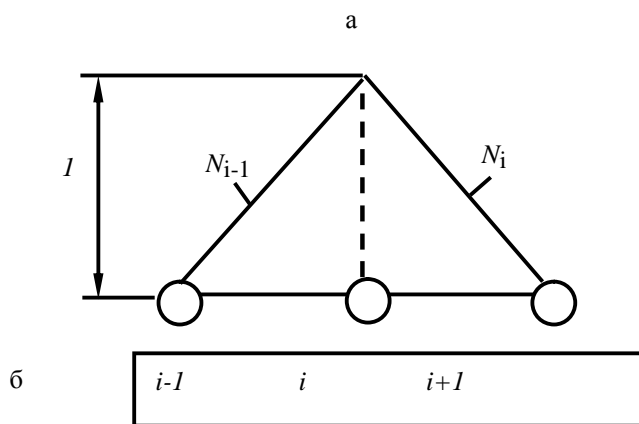
$$\hat{j} = \sum_{i=0}^N j_i(t) N_i(x), \quad (18.18)$$

де j_i – залежна від часу апроксимація температури у вузлі i ;

N_i – стандартна лінійна базисна функція, що асоціюється з i -м вузлом.

Рівняння методу зважених неув'язок з апроксимацією за Гальоркіним має вигляд:

$$\int_0^L \left[\frac{\partial}{\partial x} \left(l \frac{\partial \hat{j}}{\partial x} \right) - cr \frac{\partial \hat{j}}{\partial t} \right] N_i dx = 0, \quad l = 0, 1, 2, \dots, N. \quad (18.19)$$



а - нумерація елементів і вузлів, яка використана при рішенні задачі теплопровідності; б – глобальна кусочно-лінійна базисна функція, яка асоціюється з i -м вузлом

Рисунок 18.3 – Нумерація елементів і вузлів

Еквівалентна слабка форма, що одержується інтегруванням по частинах, має вигляд:

$$\int_0^L \frac{\partial}{\partial x} \left(I \frac{\partial \hat{j}}{\partial x} \right) N_l dx = 0, \quad \left\{ \begin{array}{l} u = N_l \quad du = \frac{\partial N_l}{\partial x} dx \\ dv = \frac{\partial}{\partial x} \left(I \frac{\partial \hat{j}}{\partial x} \right) dx \quad v = I \frac{\partial \hat{j}}{\partial x} \end{array} \right\}, \text{ звідси}$$

$$\int_0^L \frac{\partial}{\partial x} \left(I \frac{\partial \hat{j}}{\partial x} \right) N_l dx = \left[N_l I \frac{\partial \hat{j}}{\partial x} \right]_0^L - \int_0^L I \frac{\partial \hat{j}}{\partial x} \frac{\partial N_l}{\partial x} dx = 0.$$

Таким чином, одержуємо:

$$\int_0^L \left[\frac{\partial}{\partial x} \left(I \frac{\partial j}{\partial x} \right) - cr \frac{\partial j}{\partial t} \right] N_l dx = \left[N_l I \frac{\partial j}{\partial x} \right]_0^L - \int_0^L \left[I \frac{\partial j}{\partial x} \frac{\partial N_l}{\partial x} + cr N_l \frac{\partial j}{\partial t} \right] dx = 0. \quad (18.20)$$

Підставляючи сюди апроксимацію \hat{j} , дістаємо систему рівнянь:

$$\int_0^L \left[I \frac{\partial \hat{j}}{\partial x} \frac{\partial N_l}{\partial x} + cr N_l \frac{\partial \hat{j}}{\partial t} \right] dx = \left[N_l I \frac{\partial \hat{j}}{\partial x} \right]_0^L,$$

$$\int_0^L I \frac{\partial N_l}{\partial x} \sum_{m=0}^N j_m \frac{\partial N_m}{\partial x} dx + \int_0^L cr N_l \sum_{m=0}^N \frac{\partial j_m}{\partial t} N_m dx = \left[N_l I \frac{\partial \hat{j}}{\partial x} \right]_0^L. \quad (18.21)$$

Згрупуємо систему рівнянь (18.21) і подамо її у векторному вигляді:

$$C \frac{\partial j}{\partial t} + K j = f. \quad (18.22)$$

Типові компоненти матриць і правої частини мають вигляд:

$$C_{lm} = \int_0^L cr N_l N_m dx \quad f_l = \left[N_l I \frac{\partial \hat{j}}{\partial x} \right]_0^L. \quad (18.23)$$

Застосуємо до одержаного рівняння метод кінцевих різниць. Тоді вибираючи на осі t безліч дискретних точок, одержимо:

$$C \left. \frac{\partial j}{\partial t} \right|^{n+q} + K j^{n+q} = f^{n+q}, \quad (18.24)$$

де верхній індекс $n+q$ означає обчислення у момент часу $t = t_n + q\Delta t_n$, $0 \leq q \leq 1$. Використовуючи розкладання в ряд Тейлора, одержуємо:

$$j^n = j^{n+q} - q \left. \frac{\partial j}{\partial t} \right|^{n+q} + \frac{1}{2} q^2 \left. \frac{\partial^2 j}{\partial t^2} \right|^{n+q} + \dots \quad (18.25)$$

і

$$j^{n+1} = j^{n+q} + (1-q) \left. \frac{\partial j}{\partial t} \right|^{n+q} + \frac{1}{2} (1-q)^2 \left. \frac{\partial^2 j}{\partial t^2} \right|^{n+q} + \dots \quad (18.26)$$

Віднімаючи (18.24) з (18.25), одержимо апроксимацію за часом:

$$\left. \frac{\partial j}{\partial t} \right|^{n+q} \approx \frac{j^{n+1} - j^n}{\Delta t_n}. \quad (18.27)$$

Відзначимо, що погрішність такої апроксимації має порядок $O(\Delta t_n^2)$ при $q = \frac{1}{2}$ і $O(\Delta t_n)$ інакше. Помноживши рівність (18.25) на $1-q$, а (18.26) на θ , одержимо:

$$j^{n+q} \approx (1-q) j^n + q j^{n+1}. \quad (18.28)$$

Підставивши в рівняння (18.23) апроксимації (18.26) і (18.27) і застосувавши для f^{n+q} те ж уявлення, що і для j^{n+q} у (18.27), знаходимо:

$$\left\{ \frac{C}{\Delta t_n} + qK \right\} j^{n+1} + \left\{ -\frac{C}{\Delta t_n} + (1-q)K \right\} j^n = (1-q) f^n + q f^{n+1}. \quad (18.29)$$

$$\int_0^{h_i} l dx = \frac{l_1 + l_2}{2} h_i. \quad (18.32)$$

Таким чином, одержимо:

$$K_{ii} = K_{jj} = \frac{l_i + l_j}{2h_i}, \quad K_{ij} = K_{ji} = -\frac{l_i + l_j}{2h_i}. \quad (18.33)$$

Для матриці С формули для обчислення інтегралів від добутку функцій форми наведені в додатку Ж.

Приймаємо, що функція теплоємності c і відносної щільності ρ змінюється лінійно в межах одного кінцевого елемента, тобто

$$c = N_i c_i + N_j c_j = \begin{vmatrix} N_i & N_j \end{vmatrix} \cdot \begin{vmatrix} c_i \\ c_j \end{vmatrix} \quad i$$

$$r = N_i r_i + N_j r_j = \begin{vmatrix} N_i & N_j \end{vmatrix} \cdot \begin{vmatrix} r_i \\ r_j \end{vmatrix}, \quad (18.34)$$

$$\begin{aligned} C_{ii} &= \int_0^{h_i} c r N_i^2 dx = \int_0^{h_i} (N_i c_i + N_j c_j)(N_i r_i + N_j r_j) N_i^2 dx = \\ &= \int_0^{h_i} N_i^2 \left[N_i^2 c_i r_i + N_i N_j (c_i r_j + c_j r_i) + N_j^2 c_j r_j \right] dx = \\ &= \int_0^{h_i} N_i^4 c_i r_i dx + \int_0^{h_i} N_i^3 N_j (c_i r_j + c_j r_i) dx + \int_0^{h_i} N_i^2 N_j^2 c_j r_j dx = \\ &= c_i r_i \frac{h_i}{5} + (c_i r_j + c_j r_i) \frac{3! h_i}{5!} + c_j r_j \frac{2! 2!}{5!} h_i = \\ &= \frac{h_i}{5} \left[c_i r_i + \frac{1}{4} (c_i r_j + c_j r_i) + \frac{1}{6} c_j r_j \right] \end{aligned}, \quad (18.35)$$

$$\begin{aligned}
C_{ii} &= \int_0^{h_i} c r N_i N_j dx = \int_0^{h_i} (N_i c_i + N_j c_j)(N_i r_i + N_j r_j) N_i N_j = \\
&= \int_0^{h_i} N_i N_j [N_i^2 c_i r_i + N_i N_j (c_i r_j + c_j r_i) + N_j^2 c_j r_j] dx = \\
&= \int_0^{h_i} N_i^3 N_j c_i r_i dx + \int_0^{h_i} N_i^2 N_j^2 (c_i r_j + c_j r_i) dx + \int_0^{h_i} N_i N_j^3 c_j r_j dx = \\
&= c_i r_i \frac{3! h_i}{5!} + (c_i r_j + c_j r_i) \frac{2! 2!}{5!} h_i + c_j r_j \frac{3!}{5!} h_i = \\
&= \frac{h_i}{10} \left[\frac{1}{2} c_i r_i + \frac{1}{3} (c_i r_j + c_j r_i) + \frac{1}{2} c_j r_j \right], \tag{18.36}
\end{aligned}$$

$$\begin{aligned}
C_{jj} &= \int_0^{h_i} c r N_j^2 dx = \int_0^{h_i} (N_i c_i + N_j c_j)(N_i r_i + N_j r_j) N_j^2 = \\
&= \int_0^{h_i} N_j^2 [N_i^2 c_i r_i + N_i N_j (c_i r_j + c_j r_i) + N_j^2 c_j r_j] dx = \\
&= \int_0^{h_i} N_i^2 N_j^2 c_i r_i dx + \int_0^{h_i} N_i N_j^3 (c_i r_j + c_j r_i) dx + \int_0^{h_i} N_j^4 c_j r_j dx = \\
&= \frac{h_i}{5} \left[\frac{1}{6} c_i r_i + \frac{1}{4} (c_i r_j + c_j r_i) + c_j r_j \right]. \tag{18.37}
\end{aligned}$$

Для вектора f :

$$f_l = \left[N_l I \frac{\partial j}{\partial x} \right]_0^{h_i} = 0 \quad l = 1, 2, 3, \dots, N-1. \tag{18.38}$$

Таким чином, для будь-якого k -го кінцевого елемента, матриці мають вигляд:

$$|C|^k = \begin{vmatrix} C_{ii}^k & C_{ij}^k \\ C_{ji}^k & C_{jj}^k \end{vmatrix}, \tag{18.39}$$

де відповідні компоненти матриць обчислюються за формулами (18.33), (18.35) -(18.37).

Для матриці К.

При розгляді структури матриці К, легко відмітити, що вона співпадає із структурою матриці С для кожного кінцевого елемента. Звідси можна укласти, що і ансамблювана матриця К співпадає з ансамблюваною матрицею С, з точністю до назви компонент.

Для вектора f.

$$|C| = \begin{array}{c|cccccc} & 0 & 1 & 2 & & N-2 & N-1 \\ \hline 0 & C_{00}^0 & C_{01}^0 & & & & \\ \hline 1 & C_{10}^0 & C_{11}^0 & C_{12}^1 & & & \\ \hline 2 & & C_{21}^1 & C_{22}^1 & & & \\ \hline & & & C_{22}^2 & & & \\ \hline & & & & & & \\ \hline N-2 & & & & & C_{N-2,N-2}^{N-2} & C_{N-2,N}^{N-2} \\ \hline N-1 & & & & & C_{N-1,N-2}^{N-2} & C_{N-1,N}^{N-2} \\ \hline & & & & & & C_{N-1,N}^{N-1} \\ \hline N & & & & & & C_{N,N-}^{N-1} \end{array} \quad (18.40)$$

Враховуючи, що всі від'ємні компоненти вектора f рівні 0, одержимо:

$$|f|^T = (\bar{f}_1, 0, 0, \dots, 0, 0, \bar{f}_N). \quad (18.41)$$

Виводимо коефіцієнти для вирішення системи рівнянь. Враховуючи (18.41), перепишемо рівняння (18.28) для неграничних точок у вигляді:

$$\left\{ \frac{C}{\Delta t_n} + qK \right\} j^{n+1} = \left\{ \frac{C}{\Delta t_n} - (1-q)K \right\} j^n. \quad (18.42)$$

Опустимо для простоти верхні індекси і подамо одержану систему у вигляді:

$$A\mathbf{j} = D. \quad (18.43)$$

Оскільки матриці С і К мають явно виражену стрічкову три діагональну структуру, то систему рівнянь (18.43) доцільно вирішувати методом прогону.

Розпишемо рівняння (18.43) у по компонентному вигляді:

$$a_{1i}\mathbf{j}_{i-1} + a_{2i}\mathbf{j}_i + a_{3i}\mathbf{j}_{i+1} = d_i, \quad i = \overline{0, N-1}. \quad (18.44)$$

Використовуючи подання матриць С, К, запишемо вирази для компонент матриці А:

$$\begin{aligned} a_{1i+1} &= \frac{1}{\Delta t_n} C_{i+1,i}^i + qK_{i+1,i}^i, \quad a_{10} = 0, \\ a_{2i+1} &= \frac{1}{\Delta t_n} [C_{i,i}^{i-1} + C_{i,i}^i] + q[K_{i,i}^{i-1}, K_{i,i}^i], \\ a_{3i+1} &= \frac{1}{\Delta t_n} C_{i,i+1}^i + qK_{i,i+1}^i, \quad a_{3N} = 0. \end{aligned} \quad (18.45)$$

Компоненти матриці D мають вигляд:

$$d_i = (a_{1i} - K_{i,i-1}^{i-1})\mathbf{j}_{i-1} + (a_{2i} - (K_{i,i}^{i-1} + K_{i,i}^i))\mathbf{j}_i + (a_{3i} - K_{i,i+1}^i)\mathbf{j}_{i+1}. \quad (18.46)$$

Одержані вирази справедливі для неграничних точок ($i = \overline{0, N-1}$), оскільки в них вектор \mathbf{f} буде дорівнювати 0. Для граничних точок значення температури відомі (граничні умови). Для того, щоб не міняти структуру матриць А і D рівняння (18.43), включимо до них рівняння для граничних точок так, щоб при рішенні виконувалися граничні умови. Це можна зробити, доповнюючи компоненти матриць А і D наступними виразами:

$$a_{20} = 1, \quad a_{30} = 0, \quad d_0 = \overline{\mathbf{j}}_0,$$

$$a_{2N} = 1, a_{1N} = 0, d_N = \overline{j}_N, \quad (18.47)$$

де $\overline{j}_0, \overline{j}_N$ – задані значення температури на межах відрізка.

18.4 Розрахунок поля щільності

З прикладенням до порошкового матеріалу ізостатичного тиску p відбувається миттєва пластична деформація, яка здійснюється за рахунок різних видів ковзання дислокацій і двійникування в матеріалі твердої фази, а також за рахунок перерозподілу і нового укладання частинок матеріалу. Ступінь ущільнення в результаті пластичної деформації може бути визначений як [25]

$$3p\sqrt{a(1-r)^m} - r^n S_s = 0, \quad (18.48)$$

де ρ – відносна щільність; a, m, n – експериментально певні коефіцієнти; P – зовнішній тиск; S_s – напруження текучості.

Для швидкості ущільнення, здійснюваного за механізмом пластичності, можна записати:

$$\mathcal{R} = \begin{cases} \infty, & \text{при } p > \frac{S_s r^n}{3[a(1-r)^m]}; \\ 0, & \text{при } p \leq \frac{S_s r^n}{3[a(1-r)^m]}. \end{cases} \quad (18.49)$$

Для розрахунків за формулою (18.48) необхідно визначити напруження текучості S_s . Найбільш істотний вплив на величину S_s надає температура. Вплив температури на величину напруження текучості може бути виражено формулою

$$S_s = S_{s0} \left(1 - b \frac{T - 300}{T_{пл}} \right), \quad (18.50)$$

де S_{s0} – напруження текучості при 300 К; b – постійний коефіцієнт; T – абсолютна температура процесу; $T_{пл}$ – температура плавлення.

Оскільки до рівняння (18.48) не входить час або швидкість ущільнення, розв'язання цього рівняння при підвищених температурах не дозволяє врахувати характер процесу вантаження. Для врахування цих чинників доцільно скористатися методами теорії повзучості, яка добре зарекомендувала себе [25].

Повзучість починає істотно позначатися при підвищених температурах. У теорії повзучості заведено вважати, що швидкість деформацій є деякою функцією інтенсивності напруження. Експериментальні дослідження показали, що для опису процесів ущільнення, що протікають в пористих матеріалах, найбільш підходить степенева повзучість.

Степенева повзучість, як і пластична деформація, обумовлена переміщенням дислокацій. Проте, якщо при пластичній деформації ковзання дислокацій носить лавиноподібний характер при досягненні напруженнями критичного значення, то при повзучості ковзання дислокацій спостерігається при напруженнях нижче межі текучості.

Ущільнення за механізмом степеневої (дислокаційної) повзучості можна описати за допомогою феноменологічної теорії повзучості пористих тіл. Для випадку ізостатичного пресування пористого тіла, поведінка основного матеріалу якого описується степеневою залежністю, швидкість ущільнення $\dot{\epsilon}_t$ може бути визначена за формулою

$$\dot{\epsilon}_t = A \frac{3\sqrt{a(1-r)^m}}{r^{n-2}} \left[\frac{3p\sqrt{a(1-r)^m}}{r^n} \right]^k, \quad (18.51)$$

де A, k – експериментально визначені коефіцієнти.

В області високих температур і низьких напружень істотну роль у процесі ущільнення можуть відігравати механізми дифузійної повзучості.

Вважатимемо, що загальні рівняння повзучості пористих тіл справедливі також і для дифузійної повзучості. Тоді залежності для

дифузійної повзучості легко виходять з рівнянь степеневі повзучості (18.51) шляхом заміни функції температури A на коефіцієнт текучості C і підстановки значення $k = 1$. Таким чином, шляхом перетворення рівняння (18.51) з урахуванням прийнятих умов, одержуємо для дифузійної повзучості наступне рівняння

$$\dot{\epsilon}_n = \frac{9a(1-r)^m p}{r^{2(n-1)}} C. \quad (18.52)$$

У виразах, одержаних на основі феноменологічних теорій пластичності і повзучості пористих тіл, для визначення швидкості ущільнення за механізмами пластичності, дислокаційної і дифузійної повзучості передбачалося, що відбувається необоротне формозмінення в об'ємі твердої фази пористого тіла. Проте при ущільненні пористого тіла діє ще один механізм, а саме дифузія матеріалу від контактних поверхонь між частинками порошку. За деяких умов цей механізм може вносити істотний внесок до процесу ущільнення, особливо на завершальній стадії досягши теоретичної щільності.

Формули для визначення швидкості ущільнення пористого матеріалу за механізмом дифузії мають вигляд: [25]

$$\dot{\epsilon}_d = \begin{cases} \frac{43(1-r_0)^2 [dD_b + R_p(r-r_0)D_v]}{(r-r_0)^2 KTR_p^3} \Omega p, & \text{при } r < 0.9 \\ \frac{54\Omega \left[dD_b + R_p \left(\frac{1-r}{6} \right)^{\frac{1}{3}} D_v \right]}{KTR_p^3} \sqrt[5]{1-r} p, & \text{при } r \geq 0.9, \end{cases} \quad (18.53)$$

де r_0 – початкова щільність; D_b, D_v – коефіцієнти граничної і об'ємної дифузії; R_p – радіус частинки; Ω – атомний об'єм; d – ефективна ширина меж зерна; K – постійна Больцмана; R – універсальна газова постійна.

Одержані формули дозволяють врахувати вплив складу пресованого матеріалу на процес ущільнення через середній радіус частинок R_p . З формул видно, що чим менше розмір частинок, тим вище швидкість

ущільнення, що добре узгоджується з фактичними даними про гаряче пресування і спікання порошків.

Процес дифузії є процесом, що термічно активується. Залежності коефіцієнтів об'ємної і граничної дифузій від температури виражаються співвідношеннями:

$$D_v = D_{0v} e^{-\frac{Q_v}{RT}}, \quad (18.54)$$

$$dD_b = D_{0b} e^{-\frac{Q_b}{RT}}, \quad (18.55)$$

де D_{0b}, D_{0v} і δ – постійні матеріалу; Q_b, Q_v – енергії активації відповідно граничної і об'ємної дифузій.

З вищевикладеного видно, що процес ущільнення реалізується за різними механізмами, які діють одночасно.

Існують дві особливості розглянутих деформаційних механізмів. По-перше, до всіх математичних моделей механізмів, окрім механізму пластичності, входить час. По-друге, пластична деформація і дислокаційна повзучість реалізується в матеріалі, що деформується, за рахунок різних видів переміщень дислокацій. Тому, якщо розглядати два механізми, то їх дію можна розглядати за аналогією з короткочасною повзучістю, коли з додатком навантаження відбувається миттєва пластична деформація, якщо внутрішні напруження перевищують напруження текучості, а потім розвивається деформація повзучості. Якщо ж з прикладенням навантаження виникаючі в тілі напруження не перевищують напруження текучості, то пластична деформація зовсім відсутня, а відразу починається процес повзучості. Таким чином, процес пластичності є альтернативним у відношенню до інших механізмів. Якщо ізостатичний тиск p згідно з умовою (18.49), достатньо великий для переходу пресованого матеріалу в пластичний стан, то відбувається миттєве ущільнення до відносної щільності, визначуваної в результаті рішення трансцендентного рівняння (18.48), і подальше ущільнення за іншими механізмами йде від щільності - Γ_p . Інакше ущільнення за механізмом пластичності не відбувається і

деформації повзучості і дифузії в пористому тілі розвиваються від початкової щільності і R_0 .

Тепер необхідно встановити, як враховувати дію механізмів повзучості і дифузії. Механізми дислокаційної повзучості, дифузійної повзучості і поверхневої дифузії засновані на тому, що діє одночасно і незалежно один від одного різних фізичних процесах, які протікають в пресованому матеріалі. Отже, для врахування дії цих механізмів у процесі пресування можна взяти принцип адитивності, згідно з яким загальна швидкість ущільнення ρ пресованого матеріалу буде дорівнювати сумі швидкостей ущільнення за механізмами дислокаційної повзучості, дифузійної повзучості, дифузії на поверхні частинок, тобто

$$\mathcal{R} = \mathcal{R}_c + \mathcal{R}_n + \mathcal{R}_d. \quad (18.56)$$

З врахуванням (18.51), (18.52), (18.53), можна записати, що

$$\frac{\partial r}{\partial t} = Ap^k r^2 \left(\frac{3\sqrt{a(1-r)^m}}{r^n} \right)^{k+1} + \frac{9a(1-r)^m p}{r^{2(n-1)}} J + \frac{43(1-r_0)^2 [dD_b + R_p(r-r_0)D_v]}{(r-r_0)^2 KTR_p^3} \Omega p \quad \text{при } r < 0.9 \quad (18.57, a)$$

$$\frac{\partial r}{\partial t} = Ap^k r^2 \left(\frac{3\sqrt{a(1-r)^m}}{r^n} \right)^{k+1} + \frac{9a(1-r)^m p}{r^{2(n-1)}} J + \frac{54\Omega \left[dD_b + R_p \left(\frac{1-r}{6} \right)^{\frac{1}{3}} D_v \right]}{KTR_p^3} \sqrt[5]{1-r} p \quad \text{при } 0.9 < r < 1. \quad (18.57, b)$$

Таким чином, процес ущільнення описується трансцендентним рівнянням (18.48) і диференціальними рівняннями (18.57а) (18.57б).

Для знаходження коефіцієнта текучості, що входить до формули (18.52) можна скористатися формулою

$$c = C \frac{\Omega}{KTd^2} \left(D_v + \frac{pdD_b}{d} \right), \quad (18.58)$$

де C – експериментальна константа; d – діаметр зерна.

Для знаходження функції A , що входить до функції (18.51), можна скористатися залежністю

$$A = \frac{A_0 b D_v}{K T m^{k-1}}, \quad (18.59)$$

де A_0 – постійна Дорну; m – модуль зсуву, який визначаємо за формулою.

$$m = m_0 \left(1 + b \frac{T - 300}{T_{nl}}\right), \quad (18.60)$$

де m_0 – модуль зрушення при $T=300$ К.

18.5 Визначення оптимальних параметрів процесу пресування при охолодженні нагрітої заготовки

Використовуючи одержані математичні моделі, був розроблений програмний комплекс, що дозволяє проводити комплексний розрахунок процесу пресування дискретного матеріалу.

Як приклад використання одержаного програмного комплексу був змодульований реальний процес пресування дискретного матеріалу в умовах охолодження нагрітої заготовки.

Графік залежності зміни зовнішніх умов наведений на рис. 18.5. На першому етапі відбувається нагрів капсули з пресованим матеріалом в печі. Температура змінюється лінійно від T_0 до T_1 (початкова і робоча температура). Час досягнення матеріалом робочої температури можна визначити за формулою

$$t_I = \frac{T_1 - T_0}{V_T}, \quad (18.61)$$

де V_T – швидкість підйому температури, величина постійна, залежна від параметрів конкретного устаткування.

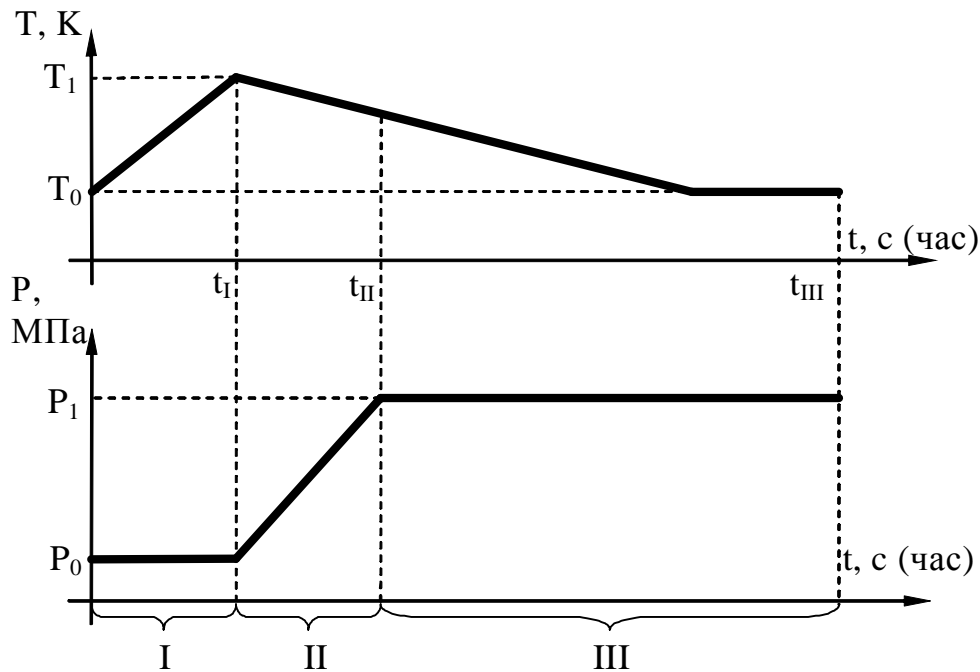


Рисунок 18.5 – Графік залежності зовнішніх умов

На матеріал, що нагрівається, не виявляється додаткового тиску, тому воно дорівнює атмосферному тиску P_0 .

На другому етапі капсулу з нагрітим матеріалом поміщають на гідравлічний прес і починають процес пресування. Тиск, що розвивається пресом, змінюється лінійно від P_0 до P_1 (початковий і робочий тиск). Час досягнення робочого тиску можна визначити за формулою

$$t_{II} = t_I + \frac{P_1 - P_0}{V_P}, \quad (18.62)$$

де V_P – швидкість підйому тиску, величина постійна, залежна від параметрів конкретного устаткування.

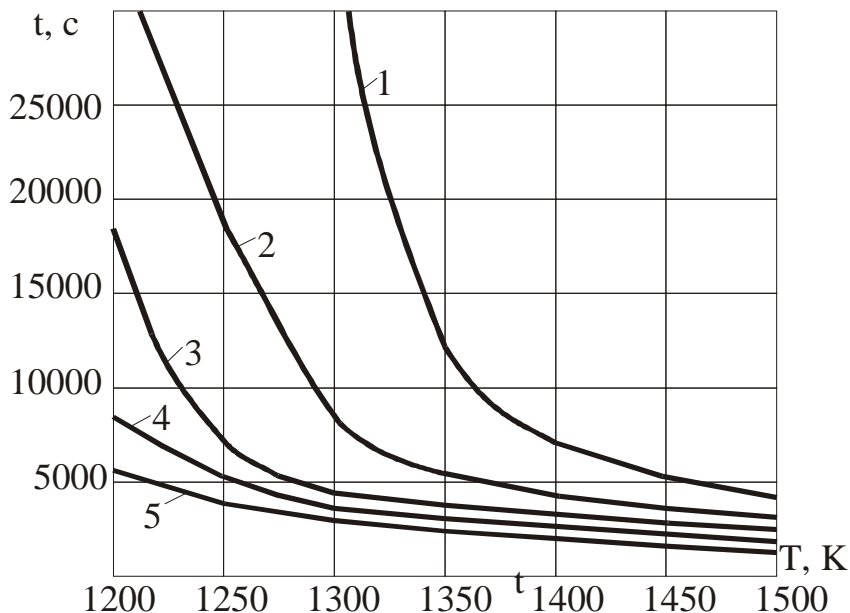
Температура капсули з матеріалом починає знижуватися згідно із законом

$$T(t) = \min(T_0; T_1 - b(t - t_I)) \quad (18.63)$$

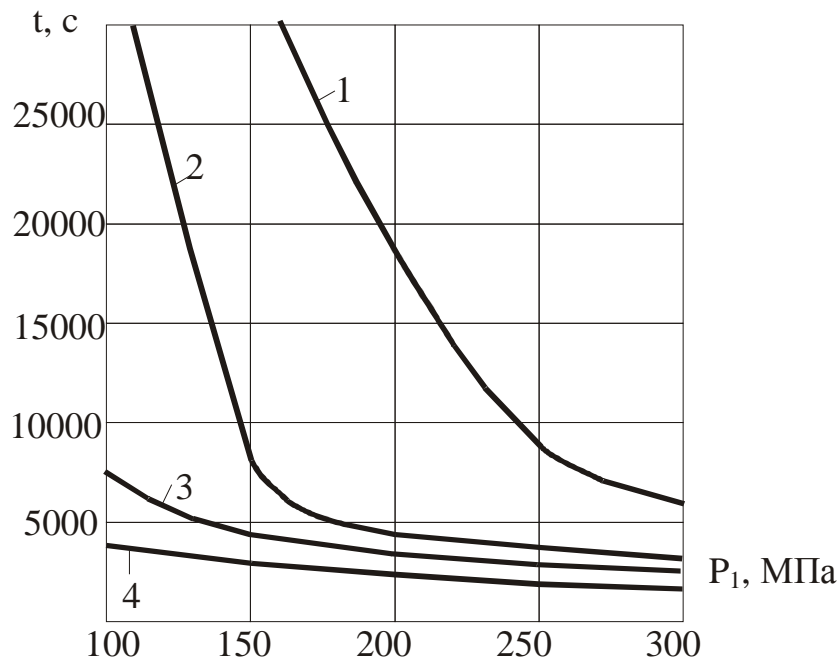
де b – швидкість охолодження заготовки.

На третьому етапі тиск, що розвивається пресом, утримується постійним і дорівнює робочому тиску P_1 . Умовами закінчення процесу ущільнення є: досягнення пресованим матеріалом відносної щільності 0.999 за всім перерізом, перевищення розрахунковим часом пресування допустимого часу t_{III} .

Як приклад розрахунку процесу ущільнення порошкового матеріалу був проведений розрахунок гарячого пресування порошку жароміцного сплаву на нікелевій основі. Початкові умови мали вигляд: початкова температура $T_0 = 300$ К, швидкість підйому температури $V_T = 0,03$ К/с, швидкість підйому тиску $V_P = 0,1$ МПа/с. Граничні умови варіювалися: робоча температура T_1 в діапазоні 1200..1500 К з кроком в 100 К, робочий тиск P_1 варіювався в діапазоні 100..300 МПа з кроком 50 МПа. Залежність часу досягнення заданої відносної щільності Γ_k від робочої температури і тиску наведені на рис. 18.6 і 18.7.



1-100 МПа, 2-150 МПа, 3-200 МПа, 4-250 МПа, 5-300 МПа
Рисунок 18.6 - Зміна часу ущільнення матеріалу від робочої температури T_1 при тиску P_1



1-1200 K, 2-1300 K, 3-1400 K, 4 -1500 K

Рисунок 18.7 - Зміна часу ущільнення матеріалу t від робочого тиску P_1 при температурі T_1

Аналізуючи отримані результати, можна зробити наступні висновки: керовані параметри необхідно вибирати так, щоб мінімізувати час досягнення пресованим матеріалом заданої щільності, не допускаючи утворення великого градієнта щільності за перерізом заготовки. Використання високого тиску і температур зменшує час ущільнення матеріалу, але для забезпечення однорідної структури заготовки, необхідно проводити додаткову витримку, що збільшує загальний час пресування. Таким чином, найбільш доцільним є застосування таких значень управляючих параметрів, при яких не потрібно проводити додаткову витримку заготовки. Додатковою перевагою подібної стратегії є зниження вимог до вживаного устаткування.

Таким чином, був досліджений процес ущільнення дискретного матеріалу в умовах нелінійної нестационарної теплопровідності. Для дослідження застосовувався МКЕ. Модель в одновимірній постановці реалізована у вигляді пакету прикладних програм. При розрахунку ущільнення враховувалася дія різних деформаційних механізмів: пластичності, повзучості, дифузії. Модель і програма розрахунку дозволяють визначати температуру і щільність матеріалу у будь-який момент часу за всім перерізом заготовки, а також визначати час досягнення дискретним матеріалом щільності твердої фази за всією товщиною заготовки при заданих параметрах.

ДОДАТКИ

ДОДАТОК А

Текст програми розрахунку просторових стрижневих систем STERGV

```
Program STERGV;
Uses TPcrt, TpWindow, Graph, User, BGI, Fonts, Printer, Dos, INITGRAF, Edt;
CONST
{ Blue = 0;} YellowOnBlue = $1E { $0E}; YellowOnRed = $4E { $40};
YellowOnCyan = $3E { $30}; GrayOnCyan = $38 { $30};
LtCyanOnBlue = $1B { $30}; WhiteOnBlue = $1F { $0F};
MagentaOnBlue = $1C { $0C}; BlackOnGray = $70 { $70};
WhiteOnGray = $7F { $70};
CONST Ext = 'DAT'; N_Max = 50; E_Max = 50;
Nodes_Pos : Array [1..10] of Byte = (5,13,21,29,38,45,52,59,67,74);
Elem_Pos : Array [1..9] of Byte = (5,12,19,27,35,43,52,61,70);
TYPE String3 = String[3];
{-----}
Var Nodes : Array [1..50, 1..9] of Real; { Координати, сили і моменти }
GU : Array [1..50] of Byte; { Граничні умови }
N_Nodes : Byte; { Число вузлів }
GDU : Array [1..50] of Real;
Elem : Array [1..50, 1..3] of Byte; { Зв'язки і тип перерізу }
Q_E : Array [1..50,1..2] of Real; { Розподілене навантаження }
F_E : Array [1..50, 1..5] of Real; { Розподілене навантаження }
{ Площа перерізу і момент інерції } { Модуль пржності }
N_Elem : Byte; { Кількість елементів }
i, j : Byte; FileName : String; Dr : Char;
Xmg, Ymg, gd, gm, t0, i1, KD : Integer; X1, X2, Y1, Y2, G1, q, Xa, Ya, Xk, Yk
:Real; NB, NSZ, NSN, K1, K2 : Integer; k, n, L, M, NBN, KF : Integer;
BE, BG, T : array [1..12, 1..12] of Real; R1 : array [1..150] of Real;
SK : array [1..150, 1..48] of Real; F : Text; S1, S2 : String;
Z1, Z2, T1, T2, E, GE : Real; quitter : Boolean;
Label 1, 3, 5, 6, 8;
(*{$I INITGRAF.PAS} { InitGraf - підключення графіки
```

```

PrintScr - друк екрана. }*) {$I PRINT.INC}
  {-----}
  Procedure Exit_Prog;
begin
(* RestoreCRTMode; *) Window(1,1,80,25); TextColor(LightGray);
TextBackGround(Black); ClrScr; TextMode(CO80);WriteLn('Good Luck!');
NormalCursor; Halt(1);
end;
{-----}
Procedure INSTRUK;
Var W      : WindowPtr; i, j      : Byte; W1, W2      : Integer; Dr  : Char;
CONST
N_Help = 27; S_Help : Array [1..N_Help] of String[58] = (
' Програма дозволяє розраховувати на міцність і жорст-'
'кість просторові стрижневі системи методом кінцевих'
'елементів. Максимальна кількість вузлів – 50, максимальна'
'кількість елементів – 50, різниця між вузлами одного'
'елемента не повинна перевищувати – 15 .',
' Нумерація вузлів і елементів довільна .',
' Координати вузлів задаються в довільній глобальній'
'системі координат, але так, щоб вся конструкція була в'
'першому квадранті, тобто всі коор-ти були додатними',
' закріпленню в якому-небудь напрямку відповідає 1,'
'відсутність закріплення – 0. Послідовність накладання'
'зв'язків: уздовж осей X,Y,Z і кутове. Наприклад: 101010- в'
' вузли відсутні, переміщення уздовж осей X і Z, кут пово,'
'роту щодо осі Y, але можливе переміщення упродовж Y ',
'і щодо осей X і Z. Додатні знаки зовнішніх ',
'навантажень наступні: сили – упродовж напрямків'
'осей координат, моменти – против ходу годинникової стрілки ',
'
'          ;"',
'
'          ',
' У результаті рішення одержуємо переміщення вузлів в'
'глобальній системі координат, значення подовжніх, попе-',
'речних сил, згинальних, крутних моментів, на початку і ',

```

'кінці кожного елемента. Додатні напрямки зусиль:',
 'подовжня сила – від першого до другого вузла елемента ',
 'поперечние сили – за ходом годинникової стрілки, а моменти – ',
 'проти ходу годинникової стрілки.');

```
begin
HiddenCursor;
If MakeWindow(W,10,4,70,21,True,True,False,$70,$7F,$7F, ' Help ') then;
If ActivateWindow(W) then; For i := 1 to 15 do FastTextWindow(S_Help[i],
1+i, 2); W1 := 1; W2 := 16;
Repeat
Dr := ReadKey; If (Dr = #0) and KeyPressed then
Begin
Dr := ReadKey; Case Dr of   #$50:   {Down}   If W2 < N_Help+1 then
Begin
ScrollWindow(True, 1); FastTextWindow(S_Help[W2], 16, 2); Inc(W2);
Inc(W1);
End;
    #$48:   {Up} If W1 > 1 then
        Begin
ScrollWindow(False, 1); Dec(W1);Dec(W2); If W1 > 1 then
FastTextWindow(S_Help[W1-1], 1, 2); End;
    #$51: {Page Down} If W2 < N_Help+1 then Begin ScrollWindow(True, 15);
Inc(W1,15);Inc(W2,15); For i := 1 to 15 do   If W1-1+i < N_Help+1 then
FastTextWindow(S_Help[W1-1+i], 1+i, 2); End;
    #$49:   {Page Up} If W1 > 1 then Begin
If W1 > 15 then j := 15 Else j := W1-1; ScrollWindow(False, j);
Dec(W1,j);Dec(W2,j); For i := 1 to j do If W1+i > 1 then
FastTextWindow(S_Help[W1-2+i], i, 2); End; End; end
Else If (Dr = #27) then begin NormalCursor; DisposeWindow(W); Exit; end;
Until False; end;
    {-----}
        Function Menu (i : Byte) : Byte;
Var   Dr : Char;
begin
```



```

Window(24,8,57,11); TextBackGround(Blue);  TextColor(Yellow);
ClrScr;  HiddenCursor;
SetFrameChars('|','=','-','\','L','r');  FrameWindow(23, 7, 58, 12, $1E, $1E,"");
WriteLn(' Введення початкових даних з файла');
WriteLn(' Введення початкових даних з дисплея');
WriteLn(' Інструкція'); Write(' Вихід з програми');
ChangeAttributeWindow(32, i, 2, $4E); Repeat
Dr := ReadKey;  KD:=i; If (Dr = #0) and KeyPressed then Begin Dr :=
ReadKey; Case Dr of  #50:  {Down} If i < 5 then Begin
ChangeAttributeWindow(32, i, 2, $1E);  Inc(i);
If i = 5 then i := 1; ChangeAttributeWindow(32, i, 2, $4E); End;
#$48:  {Up} If i > 0 then Begin
ChangeAttributeWindow(32, i, 2, $1E);  Dec(i);
If i = 0 then i := 4; ChangeAttributeWindow(32, i, 2, $4E); End; End; end
Else If (Dr = #27) or (Dr = #13) then begin
If Dr = #27 then Menu := 4 Else Menu := i;  NormalCursor;  Exit; end;
Until False; end;

    Procedure SERVIS;
    Label 0,1,2,3,4,5,6,7;
    Var W          : WindowPtr; (*{$I EDT.PAS} *)
    Var  Str_N  : String80;  Str_E  : String80;  Str1  : String;  SC   : Word;
    Zooming, Nodes_On : Boolean;
        Procedure Cur_Nodes(N, Attr : Byte);
        { N  - номер строки  Attr - атрибути курсора }
        Var i, j, k, s  : Byte; P          : ^Byte;  OffSet_P  : Word; begin
        N := N + 2; If Nodes_On then Begin s := 11;  OffSet_P := Ofs(Nodes_Pos); end
        Else begin  s := 10;  OffSet_P := Ofs(Elem_Pos);  end;
        For i := 1 to s do begin  If i = 1 then j := 2  Else begin
        P := Ptr(DSEG, OffSet_P+i-2);  j := P^ + 1; end;  If i = s then k := 80
        Else begin  P := Ptr(DSEG, OffSet_P+i-1); k := P^ - 1; end;
        k := k - j + 1; ChangeAttributeWindow(k, N, j, Attr);  end;  end;
        {-----}
    Function Menu_D (i :integer) : integer;
    Var Dr : Char; begin  Window(24,8,57,11);  TextBackGround(Blue);
    TextColor(Yellow);  ClrScr;  HiddenCursor; SetFrameChars('|','=','-','\','L','r');

```

```

FrameWindow(23, 7, 58, 12, YellowOnBlue, YellowOnBlue, '');
WriteLn(' Продовжити роботу'); WriteLn(' Данні записати у файл');
WriteLn(' Редагування даних'); WriteLn(' Закінчити роботу');
ChangeAttributeWindow(32, i, 2, YellowOnRed); Repeat
Dr := ReadKey; Case Dr of #50: {Down} If i < 5 then Begin
ChangeAttributeWindow(32, i, 2, YellowOnBlue); Inc(i);
If i = 5 then i := 1; ChangeAttributeWindow(32, i, 2, YellowOnRed); End;
#54: {Up} If i > 0 then Begin
ChangeAttributeWindow(32, i, 2, YellowOnBlue);
Dec(i); If i = 0 then i := 4; ChangeAttributeWindow(32, i, 2, YellowOnRed);
End; end; Until dr=#13; Menu_D:=i; end;
{-----}
Function Str_Nodes(N : Byte) : String80; { N - номер рядка }
Var Str1 : String[80]; Str2 : String[20]; Var i, j : Byte; begin
Str1 := Str_N; Str(N:2, Str2); Move(Str2[1], Str1[2], 2); For j := 1 to 9 do
begin
If j<4 then Str2 := Real_Right(Nodes[N,j],3,6,i)
else Str2 := Real_Right(Nodes[N,j],0,5,i);
If (j < 4) then Move(Str2[1], Str1[Nodes_Pos[j]+2], 6)
Else Move(Str2[1], Str1[Nodes_Pos[j+1]+2], 5); end;
Str2 := ByteB_StrR(GU[N],6); Move(Str2[1], Str1[Nodes_Pos[4]+2], 6);
Str_Nodes := Str1; end;
{-----}
Function Str_Elem(N : Byte) : String80; { N - номер рядка }
Var Str1 : String[80]; Str2 : String[20]; Var i, j, k : Byte; begin
k := 5; Str1 := Str_E; Str(N:2, Str2); Move(Str2[1], Str1[2], 2);
For j := 1 to 3 do begin If j = 3 then begin i := 5; k := 5; end
Else i := j; Str(Elem[N,j]:k, Str2); Move(Str2[1], Str1[Elem_Pos[i]+1], k);
end;
Str2 := Real_Right(Q_E[N,1], 3, 6, i); Str2 := Real_Right(Q_E[N,2], 3, 6, i);
Move(Str2[1], Str1[Elem_Pos[3]+2], 6); Move(Str2[1], Str1[Elem_Pos[4]+2],
6);
For j := 1 to 4 do Begin Str2 := Real_Right(F_E[Elem[N,3],j], 1, 6, i);
Move(Str2[1], Str1[Elem_Pos[j+5]+2], 6); end; Str_Elem := Str1; end;
{-----}

```

```

Var W_N1, W_N2, W_E1, W_E2 : Byte;
{-----}
    Procedure Init_Windows(N, M : Byte);
{ N - перший рядок у вікні N M - перший рядок у вікні E }
Var Str1, Str2, Str3 : String[80];
{ Str_N – вертикальні лінії Str1- з текстом Str2 - перетин ліній }
i, j, k : Byte; begin Window(1,W_N1+1,80,W_N2); TextColor(Yellow);
FillChar(Str2, SizeOf(Str2), '='); Str2[0] := #80;
For i := 1 to 10 do begin
If (i>5) and (i<>8) then Str2[Nodes_Pos[i]]:= 'T' else Str2[Nodes_Pos[i]]:= '+';
end;
Str1 := Str_N; Str1[2] := 'N'; Str1[3] := '%';
Str1[8] := 'X'; Str1[9] := ','; Str1[10] := 'M';
Str1[16] := 'Y'; Str1[17] := ','; Str1[18] := 'M';
Str1[24] := 'Z'; Str1[25] := ','; Str1[26] := 'M'; Str1[33] := 'T'; Str1[34] := 'Y';
Str1[41] := 'P'; Str1[42] := 'x'; Str1[43] := ',';
Str1[45] := 'P'; Str1[46] := 'y'; Str1[47] := ',';
Str1[49] := 'P'; Str1[50] := 'z'; Str1[51] := ',';
Str1[54] := 'b'; Str1[52] := ','; Str1[56] := 'k'; Str1[57] := 'H';
Str1[62] := 'M'; Str1[63] := 'x'; Str1[64] := ',';
Str1[66] := 'M'; Str1[67] := 'y'; Str1[68] := ',';
Str1[70] := 'M'; Str1[71] := 'z'; Str1[72] := ','; Str1[74] := 'b'; Str1[76] := 'k';
Str1[77] := 'H'; Str1[78] := '*'; Str1[79] := 'M';
FastWriteWindow(Str1, 1, 1, YellowOnBlue);
FastWriteWindow(Str2, 2, 1, YellowOnBlue);
For i := 3 to W_N2-W_N1 do begin
If (N <= N_Nodes) then Str3 := Str_Nodes(N) Else Str3 := Str_N; Inc(N);
FastWriteWindow(Str3, i, 1, YellowOnBlue); end;
{----- Init_Elem -----}
Window(1,W_E1+1,80,W_E2); TextColor(Yellow);
FillChar(Str2, SizeOf(Str2), '='); Str2[0] := #80;
For i := 1 to 9 do begin
If (i=4) or (i>7) then Str2[Elem_Pos[i]]:= 'T' else Str2[Elem_Pos[i]]:= '+';
If i=2 then Str2[Elem_Pos[i]]:= 'T'; end;
Str1 := Str_E; Str1[2] := 'N'; Str1[3] := '%';

```

```

Str1[10] := 'C'; Str1[11] := 'B'; Str1[12] := 'Я'; Str1[13] := 'З';
Str1[14] := 'И'; Str1[23] := 'K'; Str1[24] := 'O'; Str1[25] := 'C'; Str1[26] := 'И';
Str1[27] := 'Н'; Str1[28] := 'Y'; Str1[29] := 'C'; Str1[30] := 'И';
Str1[39] := 'T'; Str1[40] := 'И'; Str1[41] := 'П';
Str1[47] := 'F'; Str1[48] := ','; Str1[49] := 'C'; Str1[50] := 'M';
Str1[51] := '*'; Str1[52] := '2';
Str1[59] := 'T'; Str1[60] := 'X'; Str1[61] := ','; Str1[62] := ' ';
Str1[63] := 'T'; Str1[64] := 'Y'; Str1[65] := ',';
Str1[67] := 'T'; Str1[68] := 'K'; Str1[69] := 'P';
Str1[70] := ','; Str1[71] := ' '; Str1[72] := 'B'; Str1[74] := 'C';
Str1[75] := 'M'; Str1[76] := '*'; Str1[77] := '4';
FastWriteWindow(Str1, 1, 1, YellowOnBlue);
FastWriteWindow(Str2, 2, 1, YellowOnBlue); For i := 3 to W_E2-W_E1 do
begin
If (M <= N_Elem) then Str3 := Str_Elem(M) Else Str3 := Str_E; Inc(M);
FastWriteWindow(Str3, i, 1, YellowOnBlue); end;
Window(1, W_E1, 80, W_E2); end;
{-----}
Procedure Edit_Dat;
{ N, M - перший рядок у вікні (1,2) NC, MC - положення строкового
курсора
NCC, MCC - положення курсора (N% стовпця)
i - номер рядка поточного вікна
W1, W2 - Координати Y поточного вікна i1 - номер рядка іншого
вікна
NN - кількість елементів (N_Nodes, N_Elem)
NNN - кількість стовпців (N - 11, E - 10) } Label 0, 1, 2, 3,4, 51;
Var i, i1, k : Byte; j : Integer; Str1 : String80;
W1, W2 : Byte; N, M, NC, MC, NCC, MCC : Byte;
NN, NNN : Byte; SC : Word;
Dr : Char Absolute SC; Dr_b : Byte;
{-----}
Begin 4: If Nodes_On then begin W1 := W_N1; W2 := W_N2; NNN
:= 11;
NN := N_Nodes; end Else begin

```

```

W1 := W_E1;   W2 := W_E2;   NNN := 10;   NN := N_Elem; end;
N := 1;   M := 1;   NC := 1;   MC := 1;   NCC := 1;   MCC := 1;
i := 1 + NC - N;   i1 := 1 + MC - M;   Insert := True;
NormalCursor;   TextBackGround(Blue);
Window(1,W1+1,80,W2);   Cur_Nodes(i, YellowOnRed);   GotoXY(2, i+2);
Repeat
SC := ReadKeyWord; 0: Case Dr of   #0: Begin   Dr_b := Hi(SC);
1:   Case Dr_b of   $50:   { Down }
Begin EndStr := False;   If NC < NN then   If i < W2-W1-2 then
Begin
Cur_Nodes(i, YellowOnBlue);
Inc(i); Inc(NC);   Cur_Nodes(i, YellowOnRed);   Goto 3; End
Else begin   Cur_Nodes(i, YellowOnBlue);
ScrollWindowUp(1, W1+3, 80, W2, 1);
Inc(N); Inc(NC);   If Nodes_On then Str1 := Str_Nodes(NC)
Else Str1 := Str_Elem(NC);   FastWriteWindow(Str1, i+2, 1,
YellowOnBlue);
Cur_Nodes(i, YellowOnRed);   Goto 3;   End
Else begin   Goto 3;   end;   End;   $48:   { Up } Begin
EndStr := False;   If (NC > N) then   Begin
Cur_Nodes(i, YellowOnBlue); Dec(i); Dec(NC); Cur_Nodes(i, YellowOnRed);
Goto 3;   End Else If (NC = N) and (N > 1) then   begin
Cur_Nodes(i, YellowOnBlue);   ScrollWindowDown(1, W1+3, 80, W2, 1);
Dec(N); Dec(NC);   If Nodes_On then Str1 := Str_Nodes(NC)
Else Str1 := Str_Elem(NC);   FastWriteWindow(Str1, i+2, 1, YellowOnBlue);
Cur_Nodes(i, YellowOnRed);   Goto 3;   End
Else begin   Goto 3;   end; End; $4B,   { Left }   $47:   { Home }
Begin
EndStr := True;   If (NCC > 1) then Dec(NCC)   Else NCC := NNN;
Goto 3;   { As End & Right }   end;
$4D,   { Right }   $4F:   { End }   Goto 2;   $52:   { Ins } Begin
Insert := Not Insert;   If Insert then NormalCursor   Else BlockCursor;
End; $40:   { F6 - Switch Windows } Begin
Cur_Nodes(i, LtCyanOnBlue);   ChangeAttribute(80, W1, 1, GrayOnCyan);
W1 := i; i := i1; i1 := W1;   W1 := N; N := M; M := W1;

```

```

W1 := NC; NC := MC; MC := W1; W1 := NCC; NCC := MCC; MCC := W1;
Nodes_On := Not Nodes_On;      If Nodes_On then      begin
N_Elem := NN;  NN := N_Nodes;    W1 := W_N1;      W2 := W_N2;
NNN := 11;    end      Else      begin      N_Nodes := NN;
NN := N_Elem;  W1 := W_E1;    W2 := W_E2;    NNN := 10; end;
Window(1,W1+1,80,W2);      ChangeAttribute(80, W1, 1, YellowOnCyan);
Cur_Nodes(i, YellowOnRed);      Goto 3; end;
$41:  { F7 - Insert String }      If (NN < N_Max) then      begin      Inc(NN);
If Nodes_On then      begin For j := NN downto NC+1 do
begin      For k := 1 to 9 do
Nodes[j,k] := Nodes[j-1,k];      GU[j] := GU[j-1]; end;
For k := 1 to 9 do      Nodes[NC+1,k] := 0;      GU[NC+1] := 0;      end
Else begin      For j := NN downto NC+1 do      begin
Q_E[j,1] := Q_E[j-1,1];      Q_E[j,2] := Q_E[j-1,2];
For k := 1 to 4 do      F_E[j,k] := F_E[j-1,k];
For k := 1 to 3 do      Elem[j,k] := Elem[j-1,k];      end;
Q_E[NC+1,1] := 0;      Q_E[NC+1,2] := 0;
For k := 1 to 3 do      Elem[NC+1,k] := 1;      end;
If (i+2+W1 < W2) then      begin ScrollWindowDown(6, i+3+W1, 80, W2,
1);
If Nodes_On then Str1 := Str_Nodes(NC+1) Else Str1 := Str_Elem(NC+1);
FastWriteWindow(Str1, i+3, 1, YellowOnBlue);
If (NN-NC+i+1 < W2-W1) then      begin      Str(NN:3, Str1);
FastWriteWindow(Str1, NN-NC+i+2, 1, YellowOnBlue);      end;      end;
Dr_b := $50;      Goto 1;      end; $42:  { F8 - Delete String }
If (Nodes_On and (NN > 2)) or ((Not Nodes_On) and (NN > 1)) then
begin      Dec(NN);      If Nodes_On then      begin
For j := NC to NN do      begin      For k := 1 to 9 do
Nodes[j,k] := Nodes[j+1,k];      GU[j] := GU[j+1];      end;      end
Else begin      For j := NC to NN do      begin
Q_E[j,1] := Q_E[j+1,1];      Q_E[j,2] := Q_E[j+1,2];
For k := 1 to 4 do      F_E[j,k] := F_E[j+1,k];
For k := 1 to 3 do      Elem[j,k] := Elem[j+1,k];      end;      end;
If Nodes_On then Str1 := Str_Nodes(NC)      Else Str1 := Str_Elem(NC);
Str1 := Copy(Str1, 4, 80-4);      If (W2 >= i+3+W1) then

```

```

ScrollWindowUp(6, i+3+W1, 80, W2, 1);
FastWriteWindow(Str1, i+2, 4, YellowOnBlue); Cur_Nodes(i, YellowOnRed);
If (NC+W2-W1-i-3 < NN) then      If (i < W2-W1-2) then      begin
  If Nodes_On then Str1 := Str_Nodes(NC+W2-W1-i-1)
Else Str1 := Str_Elem(NC+W2-W1-i-1); Str1 := Copy(Str1, 4, 80-4);
FastWriteWindow(Str1, W2-W1, 4, YellowOnBlue);      end Else
Else
begin      If Nodes_On then FastWriteWindow(Str_N, W2-W1, 1,
YellowOnBlue)
Else FastWriteWindow(Str_E, W2-W1, 1, YellowOnBlue);
FastTextWindow(' ', NN-NC+i+3, 1);      end; If (NC > NN) then
begin
If Nodes_On then FastWriteWindow(Str_N, i+2, 1, YellowOnBlue)
Else FastWriteWindow(Str_E, i+2, 1, YellowOnBlue); Dr_b:=$48; Goto 1; end
Else      begin      Goto 3;      end; end      Else Goto 3;      $44 : { F10 }
Begin  If Nodes_On then N_Nodes:=NN else N_Elem:=NN;
NormalCursor; Window(1,W1,80,W2);      Exit;      end;      End;      end;
#$09 : { Tab }
Begin 2: EndStr := False; If (NCC < NNN) then Inc(NCC) Else NCC := 1;
3: If (NCC > 1) then      begin      If Nodes_On then      If (NCC = 5) then
SC := Edit_ByteB(GU[NC], Nodes_Pos[4]+2, i+2, 6, YellowOnRed)
Else      If NCC<5 then      SC := Edit_Real(Nodes[NC,NCC-1],3,
-9.999, 99.999,Nodes_Pos[NCC-1]+2, i+2, 6,      YellowOnRed)
else      SC := Edit_Real(Nodes[NC,NCC-2],0,
-999, 9999,Nodes_Pos[NCC-1]+2, i+2, 5,      YellowOnRed)
Else      Case NCC of      2,3 : SC := Edit_Byte(Elem[NC,NCC-1], 1, N_Nodes,
Elem_Pos[NCC-1]+2, i+2, 4, YellowOnRed);      6 : Begin
SC := Edit_Byte(Elem[NC,3], 1, N_Elem,
Elem_Pos[NCC-1]+2, i+2, 4, YellowOnRed); Str1 := Str_Elem(NC); End;
4,5 : SC := Edit_Real(Q_E[NC,NCC-3], 3,      -1.000, 1.000,
Elem_Pos[NCC-1]+2, i+2, 6, YellowOnRed);      7..10: Begin
SC := Edit_Real(F_E[Elem[NC,3],NCC-6], 1,      0.0, 9999.9,
Elem_Pos[NCC-1]+2, i+2, 6, YellowOnRed);
Str(F_E[Elem[NC,3],NCC-6]:6:1,Str1);      For j := i+2 to W2-W1 do
If (Elem[N+j-3,3] = Elem[NC,3]) and (N+j-3<= NN) then

```

```

FastTextWindow(Str1, j, Elem_Pos[NCC-1]+2); End; End; Goto 0; end
Else GotoXY(2, i+2); end; End; Until False;
If Nodes_On then N_Nodes := NN Else N_Elem := NN;
Window(1,W1,80,W2); end;
{-----}
Begin {-----}
FillChar(Str_N, SizeOf(Str_N), ' '); Str_N[0] := #80;
For i := 1 to 10 do Str_N[Nodes_Pos[i]]:= '';
{-----}
FillChar(Str_E, SizeOf(Str_E), ' '); Str_E[0] := #80;
For i := 1 to 9 do Str_E[Elem_Pos[i]]:= ''; Zooming := False; Nodes_On :=
True;
{-----} i := 1; 0:
Window(1,2,80,24); TextBackGround(Blue); ClrScr; Window(1,1,80,1);
TextBackGround(LightGray); TextColor(Black); ClrScr;
FastCenter( 'ПРОГРАМА РОЗРАХУНКУ НА МІЦНІСТЬ І ЖОРСТКІСТЬ
ПРОСТОРОВИХ СТРЖИНЕВИХ СИСТЕМ', 1, $70);
Window(1,25,80,25); TextBackGround(LightGray); ClrScr; FastCenter(
'F7-додаток, F8- видалення, Табл - введення даних, F6-зміна вікон, F10-
вихід',1, $70); W_N1 := 2; W_N2 := 13; W_E1 := 14; W_E2 := 24;
TextBackGround(Cyan); Window(1,W N1,80,W N2); ClrEol;
FastCenter('Координати вузлів, граничні умови і навантаження у вузлах', 1
$3E); Window(1,W E1,80,W E2); ClrEol; FastCenter(
зв'язки ел-тів, косинуси кутів, тип перерізу, геом. характеристики ',
1, GrayOnCyan); 3: For i:=1 to N_Nodes do begin GU[i]:=0;
GDU[i]:=Round(GDU[i]); For J:=1 to 6 do begin
If Round(GDU[i])=0 then goto 5; If Round(GDU[i]/2)*2=GDU[i] then goto 1;
GU[i]:=GU[i]+Round(exp((6-J)*ln(2))); GDU[i]:=GDU[i]-1;
1:GDU[i]:=GDU[i]/10; end; 5: end; 4: Init_Windows(1, 1);
Window(1,W_N1,80,W_N2); Edit_Dat; i1:=Menu_D(i1);
If i1 = 4 then Exit_Prog; If i1 = 3 then Goto 4;
For i:=1 to N_Nodes do begin GDU[i]:=0; For J:=1 to 6 do begin
If GU[i]=0 then goto 6; If GU[i]/(0.9999*exp((6-J)*ln(2)))<1 then goto 2;
GDU[i]:=GDU[i]+Round(exp((J-1)*ln(10)));
GU[i]:=GU[i]-Round(exp((6-J)*ln(2))); 2: end;

```



```

6:{ GU[i]:=Round(GDU[i]);} end; If i1=2 then begin HiddenCursor;
If MakeWindow(W,30,10,62,12,True,True,False,$1E,$1E,$1E, ") then;
If ActivateWindow(W) then; Str1 := "; Keys_Move := $80;
FastTextWindow('Введіть назву файла:', 1, 3);
SC := Edit_String(Str1, 22, 1, 8, $4E); NormalCursor;
W := EraseTopWindow; DisposeWindow(W);
If (Str1 = ") then Str1 := 'NONAME'; Str1:=Str1+'.DAT';
Assign(F,Str1); Rewrite(F); Writeln(F,N_Nodes:6,N_Elem:6);
For i:=1 to N_Nodes do
Writeln(F,Nodes[i,1]:7:3,Nodes[i,2]:7:3,Nodes[i,3]:7:3,GDU[i]:8:0,
Nodes[i,4]:8:0,Nodes[i,5]:8:0,Nodes[i,6]:8:0
,Nodes[i,7]:8:0,Nodes[i,8]:8:0,Nodes[i,9]:8:0);
For i:=1 to N_Elem do
Writeln(F,Elem[i,1]:6,Elem[i,2]:6,Q_E[i,1]:9:3,Q_E[i,2]:9:3,
Elem[i,3]:6,F_E[i,1]:9:2,F_E[i,2]:9:0,F_E[i,3]:9:0 ,F_E[i,4]:9:0);
Close(F); Goto 3; end; end;
{-----}
Function Menu_V (i :integer) : integer;
Var Dr : Char; begin Window(24,8,57,11); TextBackGround(Blue);
TextColor(Yellow); ClrScr; HiddenCursor; SetFrameChars('|','=','-','_','L','r');
FrameWindow(23, 7, 58, 12, YellowOnBlue, YellowOnBlue,");
WriteLn(' Епюра подовжніх сил'); WriteLn(' Епюра поперечних сил');
WriteLn(' Епюра згинальних моментів'); Write(' Рішення нової задачі');
ChangeAttributeWindow(32, i, 2, YellowOnRed);
Repeat Dr := ReadKey; Case Dr of #50: {Down}
If i < 5 then Begin
ChangeAttributeWindow(32, i, 2, YellowOnBlue); Inc(i);
If i = 5 then i := 1; ChangeAttributeWindow(32, i, 2, YellowOnRed); End;
#548: {Up} If i > 0 then Begin
ChangeAttributeWindow(32, i, 2, YellowOnBlue); Dec(i);
If i = 0 then i := 4; ChangeAttributeWindow(32, i, 2, YellowOnRed); End;
end;
Until dr=#13; Menu_V:=i; end;
{-----}
Procedure DATA;

```

```

Var R2 :array [1..3] of Real; IC,NQ :Integer; Label
1,2;
begin E:=200000000; GE:=800000000; Assign(F,S1); Reset (F);
Read(F,N_Nodes); Read(F,N_Elem);
For i:=1 to N_Nodes do Readln(F,Nodes[i,1],Nodes[i,2],Nodes[i,3],GDU[i],
Nodes[i,4],Nodes[i,5],Nodes[i,6],Nodes[i,7], Nodes[i,8],Nodes[i,9]);
For i:=1 to N_Elem do Readln(F,Elem[i,1],Elem[i,2],Q_E[i,1],Q_E[i,2],
Elem[i,3],F_E[i,1],F_E[i,2],F_E[i,3],F_E[i,4]); Close(F); end;
{-----}

Procedure Serv1(var FileName:String);
Label 0; Var i : Byte; Dr : Char; CONST Ext = 'DAT';
begin i := 1; 0: HiddenCursor;
Window(1,2,80,24); TextBackGround(Blue); ClrScr;
Window(1,1,80,1); TextBackGround(LightGray); TextColor(Black); ClrScr;
FastCenter( 'ПРОГРАМА РОЗРАХУНКУ НА МІЦНІСТЬ І ЖОРСТКІСТЬ
ПЛОСКИХ СТРИЖНЕВИХ СИСТЕМ', 1, $70);
Window(1,25,80,25); TextBackGround(LightGray); ClrScr;
FastCenter(' ', 1, $70); i := Menu(i); Case i of 1 : Begin
FilesTab(WhiteOnBlue, MagentaOnBlue, YellowOnRed, Ext, Off, FileName);
If FileName = #$FF then Goto 0; TextBackGround(Blue);
Window(1,2,80,24); ClrScr; HiddenCursor; NormalCursor;
DATA; End; 2 :Begin N_Nodes:=2; N_Elem:=1;
For j:=1 to 2 do begin GU[j]:=0;
For K:=1 to 9 do Nodes[j,k]:=0; end;
For k:=1 to 3 do Elem[1,k]:=1; F_E[1,1]:=60.0;
F_E[1,2]:=500.0; F_E[1,3]:=20.0; F_E[1,4]:=5.0;
Q_E[1,1]:=0; Q_E[1,2]:=0; End;
3 : Begin Instruk; Goto 0; End; 4 : quitter:=true; End; end;
{-----}

Procedure INFORM;
Label 1,2,3;
begin ClrScr; Z1:=1000000; Z2:=-1000000; T1:=1000000; T2:=-1000000;
NSZ:=6*N_Nodes;
For i:=1 to N_Nodes do begin For j:=1 to 6 do
R1[6*(i-1)+j]:=Nodes[i,j+3]; end; end;

```

{-----}

Procedure ELEMENT;

```
Var   N1       :Integer;  A1,A2,A3,DE           :Real;
BG1   :array [1..12,1..12] of Real;  Label      1,2;
begin For i:=1 to 12 do          begin      For k:=1 to 12 do          begin
T[i,k]:=0;  BE[i,k]:=0;          end;          end; k:=Elem[n,1];  i:=Elem[n,2];
A1:=Nodes[i,1]-Nodes[k,1];  A2:=Nodes[i,2]-Nodes[k,2];
A3:=Nodes[i,3]-Nodes[k,3];  DE:=sqrt(A1*A1+A2*A2+A3*A3);
If DE=0 then goto 1;  T[1,1]:=A1/DE;  T[1,2]:=A2/DE;  T[1,3]:=A3/DE;
T[2,1]:=Q_E[n,1];  T[2,2]:=Q_E[n,2];
T[2,3]:=sqrt(abs(1-T[2,1]*T[2,1]-T[2,2]*T[2,2]));
T[3,1]:=T[1,2]*T[2,3]-T[1,3]*T[2,2];  T[3,2]:=T[1,3]*T[2,1]-T[1,1]*T[2,3];
T[3,3]:=T[1,1]*T[2,2]-T[1,2]*T[2,1]; For N1:=1 to 3 do begin      M:=3*N1;
For j:=1 to 3 do  begin      L:=J+M;
For i:=1 to 3 do  begin  K:=I+M;  T[K,L]:=T[I,J];  end;  end;          end;
BE[1,1]:=E*F_E[n,1]/DE/10000;          BE[1,7]:=-BE[1,1];
BE[5,5]:=4*E*F_E[n,2]/DE/100000000;          BE[5,11]:=BE[5,5]/2;
BE[5,9]:=3*BE[5,11]/DE;          BE[6,6]:=4*E*F_E[n,3]/DE/100000000;
BE[6,12]:=BE[6,6]/2;  BE[6,8]:=-BE[6,12]*3/DE;  BE[2,6]:=-BE[6,8];
BE[2,2]:=BE[2,6]*2/DE;  BE[2,8]:=-BE[2,2];          BE[2,12]:=BE[2,6];
BE[3,5]:=-BE[5,9];  BE[3,3]:=BE[5,9]*2/DE;          BE[3,9]:=-BE[3,3];
BE[3,11]:=BE[3,5];          BE[4,4]:=GE*F_E[n,4]/DE/100000000;
BE[4,10]:=-BE[4,4];  BE[7,7]:=BE[1,1];  BE[8,8]:=BE[2,2];
BE[8,12]:=BE[6,8];  BE[9,9]:=BE[3,3];          BE[9,11]:=BE[5,9];
BE[10,10]:=BE[4,4];  BE[11,11]:=BE[5,5];          BE[12,12]:=BE[6,6];
For i:=5 to 12 do          begin      I1:=I-1;
For k:=1 to I1 do          begin      BE[I,K]:=BE[K,I];  end;          end;
For i:=1 to 12 do          begin
For k:=1 to 12 do          begin      BG1[i,k]:=0;
For j:=1 to 12 do          begin      BG1[i,k]:=BG1[i,k]+T[j,i]*BE[j,k];  end;          end;
end;
For i:=1 to 12 do          begin      For k:=1 to 12 do          begin
BG[i,k]:=0;
For j:=1 to 12 do          begin      BG[i,k]:=BG[i,k]+T[j,k]*BG1[i,j];
end;          end;          end;          goto 2;
```

```

1: Writeln( ' Невірно введені координати або зв'язки елемента
N:4,' Його зв'язки ', Elem[n,1]:4,Elem[n,2]:4); Halt; 2: end;
{-----}
Procedure MATRICA;
Var NK,JJ,NR,KK,NC1,ICON,NRO,NC : Integer; NX : Real;
Label 1,2,3,4; begin NBN:=0;
For N:=1 to N_Elem do begin NK:=6*(abs(Elem[N,2]-Elem[N,1])+1);
If NK>NBN then NBN:=NK; end;
If NBN<=48 then goto 3; Writeln( ' Ширина стрічки рівна ',NBN);
Writeln(' Різниця між вузлами одного елемента'
' більше п'ятнадцяти, необхідне перенумерувати ' елементи'); Halt;
3: For N:=1 to NSZ do begin
For M:=1 to NBN do begin SK[n,m]:=0; end; end;
For N:=1 to N_Elem do begin
ELEMENT;
For JJ:=1 to 2 do begin NR:=6*(Elem[N,JJ]-1);
For J:=1 to 6 do begin NR:=NR+1; I:=(JJ-1)*6+J;
For KK:=1 to 2 do begin NC1:=6*(Elem[N,KK]-1);
For K:=1 to 6 do begin L:=6*(KK-1)+K; NC:=NC1+K+1-NR;
If NC>0 then SK[NR,NC]:=SK[NR,NC]+ BG[L,L];
end; end; end; end; end;
{ Врахування граничних умов }
For N:=1 to N_Nodes do begin IF Round(GDU[n])=0 then goto 4;
NX:=100000; NR:=6*(N-1);
For M:=1 to 6 do begin NR:=NR+1; ICON:=Round(GDU[n]/NX);
If ICON<=0 then goto 1; SK[NR,1]:=1; R1[NR]:=0;
For J:=2 to NBN do begin SK[NR,J]:=0; NRO:=NR+1-J;
If NRO>0 then SK[NRO,J]:=0; end;
GDU[n]:=GDU[n]-Round(NX)*ICON; 1: NX:=Round(NX/10);
end; 4:; end; end;
{-----}
Procedure SISTEM;
Var C :Real;
Label 1,2,3;
begin For N:=1 to NSZ do begin I:=N;

```

```

For L:=2 to NBN do   begin   I:=I+1;   If SK[n,L]=0 then goto 1;
C:=SK[n,L]/SK[n,1];   J:=0; For K:=L to NBN do   begin
J:=J+1;
If SK[n,k]<>0 then SK[i,j]:=SK[i,j]-C*SK[n,k];   end;
SK[n,L]:=C;   R1[i]:=R1[i]-C*R1[n];   1: end;
R1[n]:=R1[n]/SK[n,1];   end;
N:=NSZ;   3: N:=N-1;   If N<=0 then goto 2;   L:=N;
For K:=2 to NBN do   begin   L:=L+1;
f SK[n,k]<>0 then R1[n]:=R1[n]-SK[n,k]*R1[L]; end; goto 3;   2: ;
end;
{-----}
Procedure SIGMA;
Var   I1,K1,t0   :Integer;   RG,FL   :array [1..12] of Real;
A1,A2,M1,P1,DL,Mmax   :Real; Label   1,2,8;
begin   For N:=1 to N_Elem do   begin
For I:=1 to 6 do   begin   I1:=6*(Elem[N,1]-1)+I; RG[i]:=R1[I1];
end;
For K:=1 to 6 do   begin   K1:=6*(Elem[N,2]-1)+k;   I:=K+6;
RG[i]:=R1[K1];   end;
ELEMENT;
For K:=1 to 12 do   begin   SK[n,k]:=0;
For I:=1 to 12 do   begin   SK[n,k]:=SK[n,k]+T[k,i]*RG[i]; end; end;
For K:=1 to 12 do   begin   FL[k]:=0;
For I:=1 to 12 do   begin   FL[k]:=FL[k]+BE[k,i]*SK[n,i]; end; end;
For m:=1 to 2 do   begin
For t0:=1 to 6 do   begin   end;   end;
For i:=13 to 24 do   begin   SK[n,i]:=FL[i-12]; end; end; end;
{---MAIN PROGRAM-----}
begin   InitGraph2;   RestoreCRTMode;   i1:=1;   quitter:=false;
3 : ClrScr; SERV1(s1); if quitter then Exit_Prog;   SERVIS;   1:INFORM;
Assign(F,'RES.DAT');   ReWrite(F);
WriteLn(F,'   РОЗРАХУНОК СТРИЖНЕВОЇ ПРОСТОРОВОЇ',
' КОНСТРУКЦІЇ   ');
WriteLn(F, '   МЕТОДОМ КІНЦЕВИХ ЕЛЕМЕНТІВ');

```

```

Writeln(F,' ');      Writeln(F,' '); Writeln(F,'          Початкові дані');
Writeln(F,'_____');
'_____');
Write      (F,' Модулі пружності :першого і другого роду в МПа ');
Writeln(F,E/1000:10:0,GE/1000:10:0);
Writeln(F,'_____');
'_____');      Writeln(F,' ');
Writeln(F,'          Координати вузлів і вид їх закріплення ');
Writeln(F,'_____');
'_____');
Writeln(F,' Вузол  X   Y   Z  Гр. умов.', ' Узел  X   Y   Z  Гр. умов. ');
Writeln(F,'_____');
'_____'); For i:=1 to N_Nodes do          begin
If i=Round(i/2)*2 then          Writeln(F,i:5,Nodes[i,1]:8:3,Nodes[i,2]:8:3,
Nodes[i,3]:8:3,GDU[i]:7:0) else Write(F,i:5,
Nodes[i,1]:8:3,Nodes[i,2]:8:3,Nodes[i,3]:8:3,GDU[i]:7:0);          end;
Writeln(F,' ');
Writeln(F,'_____');
'_____'); Writeln(F,' ');
Writeln(F,'          Навантаження в вузлах: сили в кН, моменти в кН*м');
Writeln(F,'_____');
'_____');
Writeln(F,' Вузол  Px   Py   Pz   Mx','   My   Mz');
Writeln(F,'_____');
'_____');
For i:=1 to N_Nodes do          begin          K1:=0;
For j:=1 to 6 do          begin          If R1[6*(i-1)+j]<>0 then K1:=1;          end;
If K1=1 then          Writeln(F,i:6,R1[6*(i-1)+1]:11:2,R1[6*(i-1)+2]:11:2,
R1[6*(i-1)+4]:11:2,R1[6*(i-1)+4]:11:2,R1[6*(i-1)+5]:11:2,
R1[6*(i-1)+6]:11:2);          end;
Writeln(F,'_____');
Writeln(F,' ');      Writeln(F,' Зв'язки елементів, площа, моменти інерції ',
' і направ. косинуси ел-тів');
Writeln(F,'_____');
Writeln(F,' Ел-т  Зв'язки  F  Ix  Iy  Iкр'  '  Косинуси е-тів');
Writeln(F,'_____');

```

```

For i:=1 to N_Elem do      begin
Writeln(F,i:4,Elem[i,1]:5,Elem[i,2]:5,F_E[i,1]:10:2,
F_E[i,2]:10:2,F_E[i,3]:10:2,F_E[i,4]:10:2,Q_E[i,1]:9:3,Q_E[i,2]:9:3);      end;
Writeln(' Чекайте, формується і вирішується система ', NSZ,' рівнянь');
      MATRICA;
      СИСТЕМ;
Writeln(' Система рівнянь розв'язана');
SIGMA;
Writeln(' Результаті розрахунків дивіться у файлі RES.DAT');
Writeln(F,'_____');
'_____');
Writeln(F,' Узел  X,cm  Y,cm  Z,cm  0X',
'  0Y  0Z');
Writeln(F,'_____');
'_____');
For N:=1 to N_Nodes do      begin
Writeln(F,N:5,100*R1[6*(n-1)+1]:11:4,      100*R1[6*(n-1)+2]:11:4,
100*R1[6*(n-1)+3]:11:4,R1[6*(n-1)+4]:11:4,
R1[6*(n-1)+5]:11:4,R1[6*(n-1)+6]:11:4);      end;
Writeln(F,'_____');
'_____');  Writeln(F,' ');
Writeln(F,'      Результати розрахунку -',      'внутрішні зусилля в
стрижнях');
Writeln(F,'_____');
'_____');  Writeln(F,' Эл-т  N, кН  Qy,кН  Qz,кН ',
'Mx,кН*см  My,кН*см  Mz,кН*см');
Writeln(F,'_____');
'_____');
For N:=1 to N_Elem do      begin
Writeln(F,N:5,SK[n,13]:11:2,SK[n,14]:11:2,
SK[n,15]:11:2,100*SK[n,16]:11:2,100*SK[n,17]:11:2, 100*SK[n,18]:11:2);
Writeln(F,' ',SK[n,19]:11:2,SK[n,20]:11:2,
SK[n,21]:11:2,100*SK[n,22]:11:2,100*SK[n,23]:11:2, 100*SK[n,24]:11:2);
end;
Writeln(F,'_____');
'_____'); Close(F);      5: end.

```

ДОДАТОК Б

Текст програми плоскої задачі теорії пружності

```
Program PLASTINA;
Uses Tpcrt,Graph,User;
Var Xmg,Ymg,gd,gm,t0,i1,KD,XC,YC :Integer;
X1,X2,Y1,Y2,G1,Xa,Ya,Xk,Yk,Vn,Vq,Vm :Real;
NP,NE,NB,NSZ,NSN,K1,K2,NGH :Integer;
XG,YG,XG1,YG1 :array [1..2] of Integer;
i,j,k,n,L,M,NBN,KF :Integer; X :array [1..100,1..2] of Real;
NOP :array [1..200,1..3] of Integer; BE,BG,BG1 :array [1..6,1..6] of Real;
NBC,NFIX,NBC1 :array [1..30] of Integer; R1 :array [1..200] of Real;
SK :array [1..200,1..48] of Real; F,Z,W :Text;
S1,S2 :String[79]; Z1,Z2,T1,T2,Fmax :Real;
P :array [1..3] of PointType; Label 1,3,5,6,8;
{-----}

Procedure INSTRUK;
begin ClrScr; Writeln(' ПРОГРАМА РОЗРАХУНКУ НА МІЦНІСТЬ І ',
'ЖОРСТКІСТЬ ПЛОСКОЇ ЗАДАЧІ ТЕОРІЇ ПРУЖНОСТІ');
Writeln(' Автор -'); Writeln(' '); Write(
' Програма дозволяє розраховувати на міцність і жорсткість плоскої ',
'задачу теорії пружності методом кінцевих елементів . Максимальна
кількість вузлів – 40' максимальна кількість елементів – 50, різниця між
вузлами'
' одного елемента'); Write (' не повинна перевищувати- 19. ');
Writeln(
' Номерація вузлів і елементів довільна . ');
Writeln(' Координати вузлів задаються в довільній глобальній системі
коорди','нат. '); Writeln(' Закріпленню в якому-небудь напрямку відповідає
1, відсутність'
' закріплення -0. Послідовність накладання зв'язків: уздовж осей X і Y. ',
' Наприклад: 2 10 - у вузлі 2 відсутнє переміщення уздовж осі'
' X, але можливе переміщення уздовж осі Y. ');
```



```

Write(' Додатні знаки зовнішніх сил – вдовж додатних напрямків осей
координат');
Writeln(' Введення навантажень повинне закінчуватися останнім вузлом ',
' незалежно від того є в ньому навантаження чи ні. ');
Writeln(' У результаті рішення одержуємо переміщення вузлів у гло'
'бальній системі координат, напруження в центрах ваги елементів. ');
Writeln(' '); end;
{-----}
Procedure SETKA;
const IC : array [1..16] of Integer =(-1,1,1,-1,1,-1,-1,1,1,-1,      -1,1,-1,1,1,-1);
Var      XE,YE          :array [1..100] of Real;
NE1      :array [1..100] of Integer;      DIAG1,DIAG2      :Real;
NR,LB    :array [1..4] of Integer;
NP1,I,J1,J2,J3,NEL,NBW,NELBW,IK      :Integer;
II,KK,TR,IR,KN1,JK,NB1,N1,IR1      :Integer;
KN2,KS1,KS2,NRT,NRTS,JL,KJ,KD,KD1  :Integer;
INRG,INGP,IPCH,NRG,NROWS,NCOL      :Integer;
XP,YP    :array [1..50] of Real;R      :array [1..200] of Real;
XRG,YRG,N2 :array [1..4] of Real;      JT :array[1..20,1..4] of Integer;
NDN      :array[1..20,1..4] of Integer;  XC,YC :array [1..21,1..21] of Real;
NN      :array [1..21,1..21] of Integer;
NNRB    :array [1..20,1..4,1..21] of Integer;
DETA,ETA,DSI,SI      :Real;
Label    1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17;
begin    ClrScr;      Writeln('0- Вихід з програми');
Writeln('2- Введення початкових даних з дисплея');
Writeln('3- Введення початкових даних з файла');      NEL:=0;
NELBW:=0;      NBW:=0;      NB1:=0;      N1:=0;      NP:=0;
Readln(KD1);      If KD1=0 then Halt;      If KD1=2 then begin
Writeln(' Введіть ім'я файла, куди будуть записані початкові дані ');
Readln(S2);      S2:=S2+'.DAT';      Assign(F,S2);      Rewrite(F);
Writeln(' Число зон');      Readln(INRG); Writeln(' Загальна кількість точок');
Readln(INGP);      Writeln(F,INRG:5);      Writeln(F,INGP:5);
Writeln(' Введіть координати вузлів');
For i:=1 to INGP do begin Writeln(' Координати X і Y', i:5, '-го вузла');

```

```

Readln(XP[i],YP[i]);      Writeln(F,XP[i]:10:3,YP[i]:10:3);      end;
For i:=1 to INRG do      begin
Writeln(' Введіть номер ',i:5,'-ой зони і номери граничних зон');
Readln( NRG,JT[nrg,1],JT[nrg,2],JT[nrg,3],JT[nrg,4]);
Writeln(F, NRG:5,JT[nrg,1]:5,JT[nrg,2]:5,JT[nrg,3]:5,JT[nrg,4]:5);      end;
For KK:=1 to INRG do      begin
Writeln(' Номер ',KK:4,'-ой зони, Кількість рядків вузлів, кількість
стовпців'
' вузлів, Глобальні номери вузлів зони');
Readln(NRG,NROWS,NCOL,NDN[kk,1],NDN[kk,2],NDN[kk,3],NDN[kk,4]);
Writeln(F,NRG,NROWS:5,NCOL:5,NDN[kk,1]:5,NDN[kk,2]:5,NDN[kk,3]:5,
NDN[kk,4]:5);      end;      end      else      begin
Writeln(' Введіть ім'я файла');      Readln(S1);      S1:=S1+'.DAT';
Assign(Z,S1);      Reset(Z);      Readln(Z,INRG);      Readln(Z,INGP);
Writeln('INRG=',INRG:5,' INGP=',INGP:5);      For i:=1 to INGP do
Readln(Z,XP[i],YP[i]);      end;
Writeln(' Початкові дані введені');      If KD1=2 then goto 13;
For KK:=1 to INRG do      begin
Readln( Z,NRG,JT[kk,1],JT[kk,2],JT[kk,3],JT[kk,4]);
Writeln( NRG:5,JT[kk,1]:5,JT[kk,2]:5,JT[kk,3]:5,JT[kk,4]:5);      end;      13::
For KK:=1 to INRG do      begin      If KD1=2 then goto 14;
Readln(Z,NRG,NROWS,NCOL,NDN[kk,1],NDN[kk,2],NDN[kk,3],NDN[kk,4]
);
Writeln(NRG:5,NROWS:5,NCOL:5,NDN[kk,1]:5,NDN[kk,2]:5,NDN[kk,3]:5,
NDN[kk,4]:5);      14::      For i:=1 to 4 do      begin      II:=NDN[kk,I];
XRG[I]:=XP[II]; YRG[I]:=YP[II];      end;      TR:=NROWS-1;
DETA:=1/TR;
TR:=NCOL-1;      DSI:=1/TR; For I:=1 TO NROWS do      begin      TR:=I-1;
ETA:=TR*DETA; For J:=1 to NCOL do      begin      TR:=J-1;      SI:=TR*DSI;
N2[1]:=ETA-SI*ETA;      N2[2]:=SI*ETA;      N2[3]:=SI-SI*ETA;
N2[4]:=1-SI-ETA+SI*ETA;      XC[I,J]:=0;      YC[I,J]:=0;
For k:=1 to 4 do      begin      XC[I,J]:=XC[I,J]+XRG[K]*N2[K];
YC[I,J]:=YC[I,J]+YRG[K]*N2[K];      end;      end;      end;
KN1:=1;      KS1:=1;      KN2:=NROWS;      KS2:=NCOL; For i:=1 to 4 do
begin      NRT:=JT[NRG,I];      If (NRT=0) or (NRT>NRG) then goto 5;

```

```

For j:=1 to 4 do      If JT[NRT,j]=NRG then NRTS:=j;      K:=NCOL;
If (I=2) or (I=4) then K:=NROWS;      JL:=1;      JK:=IC[I+4*(NRTS-1)];
If JK=-1 then JL:=K;      For J:=1 to K do      begin      Case I of      1: Begin
NN[NROWS,J]:=NNRB[NRT,NRTS,JL];      KN2:=NROWS-1;      end;
2: Begin      NN[J,NCOL]:=NNRB[NRT,NRTS,JL]; KS2:=NCOL-1;      end;
3: Begin      NN[1,J]:=NNRB[NRT,NRTS,JL];      KN1:=2;      end;
4: Begin      NN[J,1]:=NNRB[NRT,NRTS,JL];      KS1:=2;      end;      end;
JL:=JL+JK;      end;      5: end;
If KN1>KN2 then goto 6;  If KS1>KS2 then goto 6;  For I:=KN1 to KN2 do
begin  For J:=KS1 to KS2 do begin NB1:=NB1+1; NN[I,J]:=NB1; end; end;
For I:=1 to NCOL do      begin      NNRB[NRG,1,I]:=NN[NROWS,I];
NNRB[NRG,3,I]:=NN[1,I];      end;
For I:=1 to NROWS do      begin      NNRB[NRG,2,I]:=NN[I,NCOL];
NNRB[NRG,4,I]:=NN[I,1];      end;      6: ;      K:=1;
For I:=1 to NROWS do      begin      For J:=1 to NCOL do
begin      XE[k]:=XC[i,j];      YE[k]:=YC[i,j]; E1[k]:=NN[i,j];
K:=K+1;      end;      end; L:=NROWS-1;
For I:=1 to L do      begin      For J:=2 to NCOL do      begin
DIAG1:=sqrt(sqr(XC[i,j]-XC[i+1,j-1])+      sqr(YC[i,j]-YC[i+1,j-1]));
DIAG2:=sqrt(sqr(XC[i+1,j]-XC[i,j-1])+      sqr(YC[i+1,j]-YC[i,j-1]));
NR[1]:=NCOL*I+J-1;      NR[2]:=NCOL*I+J;      NR[3]:=NCOL*(I-1)+J;
NR[4]:=NCOL*(I-1)+J-1;      For ij:=1 to 2 do      begin
NEL:=NEL+1;      If DIAG1/DIAG2>1.02 then goto 9;      J1:=NR[1];
J2:=NR[IJ+1];      J3:=NR[IJ+2];      goto 10;      9:J1:=NR[IJ];
J2:=NR[IJ+1]; J3:=NR[4];      10:LB[1]:=Round(abs(NE1[J1]-NE1[J2]))+1;
LB[2]:=Round(abs(NE1[J2]-NE1[J3]))+1; LB[3]:=Round(abs(NE1[J1]-
NE1[J3]))+1;
For IK:= 1 to 3 do      begin
If LB[iK]<=NBW then goto 11; NBW:=LB[ik];      NELBW:=NEL; 11:; end;
N1:=N1+1;      NOP[n1,1]:=NE1[j1];      NOP[n1,2]:=NE1[j2];
NOP[n1,3]:=NE1[j3];      X[NE1[j1],1]:=XE[j1];      X[NE1[j2],1]:=XE[j2];
X[NE1[j3],1]:=XE[j3];      X[NE1[j1],2]:=YE[j1];
X[NE1[j2],2]:=YE[j2];      X[NE1[j3],2]:=YE[j3];
For ii:=1 to 3 do      If NP<NOP[n1,ii] then NP:=NOP[n1,ii];
12: end;      end;      end;      end;      If KD=2 then

```

```

begin  Writeln(' Введіть число закріплених вузлів');  Readln(NB);
Writeln(F,NB:5);
For i:=1 to NB do      begin  Writeln(' Введіть номер', i:4,'-ого вузла і
тип граничної умови');      Readln(NBC1[i],NFIX[i]);
Writeln(F,NBC1[i]:5,NFIX[i]:5);      end;      end  else      begin
Readln(Z,NB);      For i:=1 to NB do      Readln(Z,NBC1[i],NFIX[i]);  end;
For i:=1 to NB do      begin      For j:=1 to NP do
If (X[j,1]=XP[NBC1[i]]) and (X[j,2]=YP[NBC1[i]]) then      NBC[i]:=J;  end;
For j:=1 to NB do      Writeln(' NBC=',NBC[j]:5,' NFIX=',NFIX[j]:5);
IR1:=2*INGP;      For J:=1 to IR1 do      R[j]:=0;      NP1:=2*NP;
For j:=1 to NP1 do      R1[j]:=0;      If KD1=2 then      begin
Writeln(' Введіть навантаження у вузлах');
15:  Writeln(' Введіть номер вузла і навантаження Px і Py');
Readln(i,R[2*i-1],R[2*i]);      Writeln(F,i:5,R[2*i-1]:10:3,R[2*i]:10:3);
If I<INGP then goto 15;      end      else      begin
16:Readln(Z,i,R[2*i-1],R[2*i]);      If I<INGP then goto 16;      end;
For i:=1 to INGP do      begin      For j:=1 to NP do      begin
If (X[j,1]<>XP[i]) or (X[j,2]<>YP[i]) then goto 17;      R1[2*j-1]:=R[2*i-1];
R1[2*j]:=R[2*i]; 17:end; end;  If KD1=2 then Close(F) else Close(Z);  end;
{-----}
      Procedure DATA;
Var      R2      :array [1..3] of Real;      IC,NQ,KR      :Integer;
Label  1,2,3,4,5,6,11,12,14,15,16;
begin  ClrScr;      If KD=3 then      begin
Writeln(' Введіть ім'я файла');      Readln(S1);  S1:=S1+'.DAT';
Assign(F,S1);      Reset (F);      Read(F,NP,NE,NB);
For i:=1 to NE do      begin      If (i=Round(i/4)*4) or (i=NE) then
Readln(F,N,NOP[i,1],NOP[i,2],NOP[i,3]) else
Read(F,N,NOP[i,1],NOP[i,2],NOP[i,3]);      end;
For i:=1 to NP do      begin      If (i=Round(i/4)*4) or (i=NP) then
Readln(F,N,X[i,1],X[i,2]) else  Read(F,N,X[i,1],X[i,2]);      end;
For i:=1 to NB do      begin  Readln(F,NBC[i],NFIX[i]);      end;
1:  Readln(F,NQ,R2[1],R2[2]);
For i:=1 to 2 do      begin  IC:=(NQ-1)*2+i;      R1[IC]:=R2[i];      end;
If NQ<NP then goto 1;      4: ;      Close(F);      end  Else

```

```

begin      Writeln(' Якщо проводиться автоматичне розбиття на елементи'
' -ввести 1,ні -0');      Readln(KR);      If KR=1 then goto 15;
Writeln('  Характеристика завдання');  Writeln(' ВУЗЛІВ      ');
Readln(NP);
Writeln(' Елементів      ');      Readln(NE);
Writeln(' Закріплених вузлів ');      Readln(NB);
Writeln(' Введіть координати вузлів');      For i:=1 to NP do
begin      Writeln(' Координати X і Y ',i,'-ого вузла');
Readln(X[i,1],X[i,2]);      R1[2*(i-1)+1]:=0;  R1[2*(i-1)+2]:=0;      end;
Writeln(' Введіть зв'язки елементів ');
For i:=1 to NE do      begin      Writeln(' Зв'язки ',i,'-ого елемента');
Readln(NOP[i,1],NOP[i,2],NOP[i,3]);      end;
Writeln(' Введіть граничні умови ');      For i:=1 to NB do
begin      Writeln(' номер вузла і тип його закріплення');
Readln(NBC[i],NFIK[i]);      end;
11:  Writeln(' Навантаження у вузлах Px, Py');  Readln(NQ,R2[1],R2[2]);
For i:=1 to 2 do      begin      IC:=(NQ-1)*2+i;      R1[IC]:=R2[i];      end;
If NQ<NP then goto 11;      end;      goto 16;
15 : SETKA;      16:;      NSZ:=2*NP;      end;
{-----}

```

Procedure INFORM;

```

Label  1,2,3;
begin      If KD=3 then goto 2;
Writeln(' Якщо дані необхідно записати у файл -введіть 1'      ' немає -0');
Readln(KF);  Writeln(' Введіть ім'я файла, куди будуть записані Ваші
початкові
' дані');  If KF=0 then S2:='D1.DAT' else Readln(S2); S2:=S2+'.DAT';
goto 3;      2:S2:=S1;      3:;  Assign(F,S2);  ReWrite(F);  ClrScr;
Z1:=1000000;      Z2:=-1000000;      T1:=1000000;      T2:=-1000000;
Writeln(F,NP:6,NE:6,NB:6);
For i:=1 to NE do      begin      If (i=Round(i/4)*4) or (i=NE) then
Writeln(F,i:5,NOP[i,1]:5,NOP[i,2]:5,NOP[i,3]:5) else
Write(F,i:5,NOP[i,1]:5,NOP[i,2]:5,NOP[i,3]:5);      end;
For i:=1 to NP do      begin      If (i=Round(i/4)*4) or (i=NP) then
Writeln(F,i:4,X[i,1]:8:3,X[i,2]:8:3)      else Write(F,i:4,X[i,1]:8:3,X[i,2]:8:3);

```

```

If X[i,1]<Z1 then Z1:=X[i,1];           If X[i,1]>Z2 then Z2:=X[i,1];
If X[i,2]<T1 then T1:=X[i,2];         If X[i,2]>T2 then T2:=X[i,2];  end;
For i:=1 to NB do      begin           Writeln(F,NBC[i]:6,NFIX[i]:6);  end;
For I:=1 to NP do    begin              Writeln(F,I:6,R1[2*(i-
1)+1]:10:3,
R1[2*(i-1)+2]:10:3);      end;      Close(F);
Write(' Початкові дані закінчилися, для продовження'
' наберіть 1 і натисніть введення ');      Readln(KF);  end;
{-----}
      Procedure SILA;
Var  M1,P1,A1,A2,DL,PX,PY,Ld  :Real;  S      :String;
Label  1,2,3,4,5,6,7,8,9,10,11,12;
begin  gd:=detect;  initgraph(gd,gm,"");  Xmg:=GetMaxX;  mg:=GetMaxY;
SetColor(Yellow);  If (Z2-Z1)>(T2-T1) then G1:=Z2-Z1 else G1:=T2-T1;
X2:=0.8*Xmg/G1;  Y2:=0.7*Ymg/G1;  X1:=0.1*Xmg+X2*abs(Z1);
Y1:=0.8*Ymg-Y2*abs(T1);  Rectangle(0,0,Xmg,Ymg);
For i:=1 to 35 do  begin
Line(Round(X1+(i-1)*X2*G1/30),Round(Y1),Round(X1+i*X2*G1/30-7),
Round(Y1));  Line(Round(X1),Round(Y1-(i-1)*Y2*G1/30),Round(X1),
Round(Y1-i*Y2*G1/30+2));  end;
OutTextXY(Round(X1+0.87*Xmg*(Z2-Z1)/G1),Round(0.75*Ymg),'X');
OutTextXY(Round(X1-Xmg/100),Round(Y1-0.75*Ymg*(T2-T1)/G1),'Y');
SetFillStyle(EmptyFill,White);
For i:=1 to NE do  begin  P[1].X:=Round(X1+X2*X[NOP[i,1],1]);
P[2].X:=Round(X1+X2*X[NOP[i,2],1]);
P[3].X:=Round(X1+X2*X[NOP[i,3],1]);
P[1].Y:=Round(Y1-Y2*X[NOP[i,1],2]);  P[2].Y:=Round(Y1-
Y2*X[NOP[i,2],2]);
P[3].Y:=Round(Y1-Y2*X[NOP[i,3],2]);  FillPoly(3,P);
Xc:=Round((P[1].X+P[2].X+P[3].X)/3-Xmg/100);
Yc:=Round((P[1].Y+P[2].Y+P[3].Y)/3-Ymg/50);
Circle(Round(Xc+Xmg/140),Round(Yc+Ymg/70),12);
RealToString(S,I,10,10,0);
OutTextXY(Xc,Yc,S);  end;

```

```

Line(Round(X1-Xmg/12),Round(Y1-Y2*(T2-T1)),Round(X1-
Xmg/12),Round(Y1));
Line(Round(X1),Round(0.95*Ymg),Round(X1+X2*(Z2-
Z1)),Round(0.95*Ymg));
For i:=1 to 6 do      begin
Line(Round(X1-Xmg/12),Round(Y1-Y2*(T2-T1)*(i-1)/5),Round(X1-Xmg/14),
Round(Y1-Y2*(T2-T1)*(i-1)/5));
Line(Round(X1+X2*(Z2-Z1)*(i-1)/5),Round(0.95*Ymg),
Round(X1+X2*(Z2-Z1)*(i-1)/5),Round(0.93*Ymg));
RealToString(S,Z1+(i-1)*(Z2-Z1)/5,10,10,1);
OutTextXY(Round(X1+X2*(Z2-Z1)*(i-1)/5),Round(0.9*Ymg),S);
RealToString(S,T1+(i-1)*(T2-T1)/5,10,10,1);
OutTextXY(Round(X1-Xmg/15),Round(Y1-Y2*(T2-T1)*(i-1)/5),S);      end;
SetColor(Red);      For n:=1 to NP do      begin
XG[1]:=Round(X1+X2*X[n,1]);      YG[1]:=Round(Y1-Y2*X[n,2]);
If R1[2*(n-1)+1]=0 then goto 1;      YG[2]:=YG[1];
XG[2]:=XG[1]-Round(abs(R1[2*(n-1)+1])*Xmg/(10*R1[2*(n-1)+1]));
XG1[1]:=XG[1]-Round(abs(R1[2*(n-1)+1])*Xmg/(40*R1[2*(n-1)+1]));
YG1[1]:=YG[1]-Round(Xmg/200);      YG1[2]:=YG[1]+Round(Xmg/200);
Line(XG[1],YG[1],XG[2],YG[2]);      Line(XG[1],YG[1],XG1[1],YG1[1]);
Line(XG[1],YG[1],XG1[1],YG1[2]);
Line(XG1[1],YG1[1],XG1[1],YG1[2]);
RealToString(S,abs(R1[2*n-1]),10,10,1);
OuTTextXY(XG[1]-Round(Xmg/25*abs(R1[2*n-1])/R1[2*n-1]),YG[1]-
Round(Ymg/15),'P='+S);      1:If R1[2*(n-1)+2]=0 then goto 2;
XG[2]:=XG[1];
YG[2]:=YG[1]+Round(abs(R1[2*(n-1)+2])*Xmg/(30*R1[2*(n-1)+2]));
YG1[1]:=YG[1]+Round(abs(R1[2*(n-1)+2])*Xmg/(150*R1[2*(n-1)+2]));
XG1[1]:=XG[1]-Round(Xmg/200);      XG1[2]:=XG[1]+Round(Xmg/200);
Line(XG[1],YG[1],XG[2],YG[2]);      Line(XG[1],YG[1],XG1[1],YG1[1]);
Line(XG[1],YG[1],XG1[2],YG1[1]);
Line(XG1[1],YG1[1],XG1[2],YG1[1]);
RealToString(S,abs(R1[2*n]),10,10,1); OuTTextXY(XG[1]-
Round(Xmg/25),YG[1]+ Round(Ymg/15*abs(R1[2*n])/R1[2*n]),'P='+S);
2: RealToString(S,N,10,10,0);

```

```

OutTextXY(XG[1]-Round(Xmg/35),YG[1]-Round(Ymg/25),S);    end;
For i:=1 to NB do          begin          K:=0;
For N:=1 to NE do          begin
If NBC[i]=NOP[n,1] then K:=NOP[n,1];    If NBC[i]=NOP[n,2] then
K:=NOP[n,2];
If NBC[i]=NOP[n,3] then K:=NOP[n,3];          If K=0 then goto 12;
11: A1:=(X[NOP[n,2],1]+X[NOP[n,1],1]+X[NOP[n,3],1])/3;
A2:=(X[NOP[n,2],2]+X[NOP[n,1],2]+X[NOP[n,3],2])/3;
If A1-X[NBC[i],1]>0 then K1:=1 else K1:=-1;
If A2-X[NBC[i],2]>0 then K2:=1 else K2:=-1;
XG[1]:=Round(X1+X2*X[NBC[i],1]);
YG[1]:=Round(Y1-Y2*X[NBC[i],2]);
8:If(NFIX[i]<>10) and (NFIX[i]<>11) then goto 9;
Circle(XG[1]-K1*Round(Xmg/200),YG[1],
Round(Xmg/200));          Circle(XG[1]-K1*Round(Xmg/50),YG[1],
Round(Xmg/200));          Line(XG[1]-K1*Round(Xmg/200),YG[1],
XG[1]-K1*Round((Xmg/50-Xmg/200)),YG[1]);
Line(XG[1]-K1*Round(Xmg/40),YG[1]-Round(Xmg/100),
XG[1]-K1*Round(Xmg/40),YG[1]+Round(Xmg/100));
9: If (NFIX[i]<>11) and (NFIX[i]<>1) then goto 10;          If NFIX[i]=1
then
Circle(XG[1],YG[1]+Round(K2*Xmg/200),          Round(Xmg/200));
Circle(XG[1],YG[1]+Round(K2*Xmg/80),          Round(Xmg/200));
Line(XG[1],YG[1]+Round(K2*Xmg/200),
XG[1],YG[1]+Round(K2*(Xmg/80-Xmg/200)));
Line(XG[1]-Round(Xmg/75),YG[1]+Round(K2*Xmg/70),
XG[1]+Round(Xmg/75),YG[1]+Round(K2*Xmg/70));    goto 10;
12: end;          10:;          end; end;
{-----}
Procedure ELEMENT;
Var    AJ,AK,BJ,BK,DE,COMM,E,G          :Real;
Label    1,2;
begin    For i:=1 to 6 do          begin    For j:=1 to 6 do
begin    BE[i,j]:=0;    BG[i,j]:=0;    end;    end;
i:=NOP[n,1];    j:=NOP[n,2];    k:=NOP[n,3];    AJ:=(X[j,1]-X[i,1])/1000;

```



```

AK:=(X[k,1]-X[i,1])/1000;   BJ:=(X[j,2]-X[i,2])/1000;
BK:=(X[k,2]-X[i,2])/1000;   DE:=(AJ*BK-AK*BJ)/2;   If DE=0 then goto
1;
BE[1,1]:=BJ-BK;   BE[1,3]:=BK;   BE[1,5]:=-BJ; BE[2,2]:=AK-AJ;
BE[2,4]:=-AK;   BE[2,6]:=AJ; BE[3,1]:=AK-AJ;   BE[3,2]:=BJ-BK;
BE[3,3]:=-AK;   BE[3,4]:=BK;   BE[3,5]:=AJ; BE[3,6]:=-BJ;
E:=200000;   G:=0.3;   COMM:=E/((1+G)*(1-G*2)*DE);
BG[1,1]:=COMM*(1-G);   BG[1,2]:=COMM*G;   BG[2,1]:=BG[1,2];
BG[2,2]:=BG[1,1];   BG[3,3]:=E/(2*(1+G)*DE);
For i:=1 to 3 do   begin   For j:=1 to 6 do
begin   BG1[i,j]:=0;   For k:=1 to 3 do
begin   BG1[i,j]:=BG1[i,j]+BG[L,K]/
2*BE[k,j];   end;   end;   end;
For i:=1 to 6 do   begin   For j:=1 to 6 do   begin   BG[i,j]:=0;
For k:=1 to 3 do   begin   BG[i,j]:=BG[i,j]+BG1[k,i]/2*BE[k,j];
end;   end;   end;   goto 2;
1: Writeln( '   Невірно введені координати або зв'язки елемента',
N:4, '   Его зв'язи ', NOP[n,1]:4,NOP[n,2]:4,NOP[n,3]:4);   Halt; 2;;   end;
{-----}
Procedure MATRICA;
Var   NK, JJ, NR, KK, NC, NC1, NX, ICON, NRO   : Integer;
Label   1,2,3;
begin   NBN:=0;   For N:=1 to NE do   begin
NK:=2*(abs(NOP[N,2]-NOP[N,1])+1);   If NK>NBN then NBN:=NK;
NK:=2*(abs(NOP[N,2]-NOP[N,3])+1);   If NK>NBN then NBN:=NK;
NK:=2*(abs(NOP[N,3]-NOP[N,1])+1);   If NK>NBN then NBN:=NK;   end;
If NBN<=60 then goto 3;   Writeln ( '   Ширина стрічки рівна
',NBN);
Writeln( '   Різниця між вузлами одного елемента'
'   більше дев'ятнадцяти, необхідно перенумерувати'
'   елементи');   Halt;
3:   For N:=1 to NSZ do   begin   For M:=1 to NBN do   begin
SK[n,m]:=0;   end;   end; For N:=1 to NE do   begin
ELEMENT;
{   Writeln(F, '   Матриця [B]',N:5, '-ого елемента');

```

```

For i:=1 to 3 do      For j:=1 to 6 do
If j=6 then writeln(F,BE[i,j]:12:4) else write(F,BE[i,j]:12:4);
Writeln(F,' Матриця жорсткості',N:5,'-ого елемента');      For i:=1 to 6 do
For j:=1 to 6 do  If j=6 then writeln(F,BG[i,j]:12:1) else write(F,BG[i,j]:12:1);
Writeln(F,' Матриця [D]*[B]',N:5,'-ого елемента');      For i:=1 to 3 do
For j:=1 to 6 do      If j=6 then writeln(F,BG1[i,j]:12:1) else
write(F,BG1[i,j]:12:1);  Writeln(F,' ');}
For JJ:=1 to 3 do      begin      NR:=2*(NOP[N,JJ]-1);
For J:=1 to 2 do      begin      NR:=NR+1;      I:=(JJ-1)*2+J;
For KK:=1 to 3 do      begin      NC1:=2*(NOP[N,KK]-1);
For K:=1 to 2 do      begin      L:=2*(KK-1)+K;      NC:=NC1+K+1-NR;
If NC>0 then      SK[NR,NC]:=SK[NR,NC]+ BG[I,L]; end;      end;
end;      end;      end;
{  Writeln(F,' Матриця жорсткості всієї конструкції без урахування гр.
умов');
For n:=1 to NSZ do      For i:=1 to NBN do
If i=NBN then Writeln(F,SK[n,i]:8:0) else Write(F,SK[n,i]:8:0); }
{  Облік граничних умов  }      For N:=1 to NB do      begin
NX:=10;      I:=NBC[n];      NR:=2*(I-1);      For M:=1 to 2 do begin
NR:=NR+1;      ICON:=Round(NFIX[n]/NX);      If ICON<=0 then
goto 1;
SK[NR,1]:=1;      R1[NR]:=0;      For J:=2 to NBN do      begin
SK[NR,J]:=0;      NRO:=NR+1-J;      If NRO>0 then SK[NRO,J]:=0; end;
NFIX[n]:=NFIX[n]-NX*ICON;      1: NX:=Round(NX/10); end; end;
{  Writeln(F,' Матриця жорсткості всієї конструкції після обліку гр. умов');
For n:=1 to NSZ do      For i:=1 to NBN do
If i=NBN then Writeln(F,SK[n,i]:8:0) else Write(F,SK[n,i]:8:0);}      end;
{-----}
      Procedure SISTEM;
Var      C      :Real; Label      1,2,3,25;
begin      For N:=1 to NSZ do      begin      I:=N;
For L:=2 to NBN do      begin      I:=I+1;      If SK[n,L]=0 then goto 1;
C:=SK[n,L]/SK[n,1];      J:=0;      For K:=L to NBN do      begin
J:=J+1;      If SK[n,k]<>0 then SK[i,j]:=SK[i,j]-C*SK[n,k];      end;
SK[n,L]:=C;      R1[i]:=R1[i]-C*R1[n];      1: end;

```

```

R1[n]:=R1[n]/SK[n,1]; end; N:=NSZ; 3: N:=N-1; If N<=0 then
goto 2;
L:=N; For K:=2 to NBN do begin L:=L+1;
If SK[n,k]<>0 then R1[n]:=R1[n]-SK[n,k]*R1[L]; end; goto 3; 2: ; end;
{-----}
Procedure SIGMA;
Var I1,K1,t0 ,IA :Integer;
RG,FL :array [1..6] of Real; A,C :Real; Label
1,2,7,8;
begin Fmax:=-100000; For N:=1 to NE do begin
ELEMENT;
For I:=1 to 3 do begin M:=NOP[n,i]; If M=0 then goto 2;
K:=2*(i-1); For J:=1 to 2 do begin I1:=J+K;
RG[i1]:=R1[2*(m-1)+j]/1000; end; 2:end; IA:=K+2;
For i:=1 to 3 do begin SK[n,i]:=0; For J:=1 to IA do begin
SK[n,i]:=SK[n,i]+BG1[i,j]*RG[j]; end; end; end;
For N:=1 to NE do begin C:=(SK[n,1]+SK[n,2])/2;
A:=sqrt(sqr((SK[n,2]-SK[n,1])/2)+sqr(SK[n,3])); SK[n,4]:=C+A;
SK[n,5]:=C-A; If(SK[n,2]=SK[n,5]) then goto 7;
SK[n,6]:=57.29578*arctan(SK[n,3]/(SK[n,2]-SK[n,5])); goto 8;
7: SK[n,6]:=90; 8:;
If Fmax<abs(SK[n,4]) then Fmax:=abs(sK[n,4]);
If Fmax<abs(SK[n,5]) then Fmax:=abs(sK[n,5]); end;
Repeat Until KeyPressed; end;
{-----}
Procedure GR;
Var RG,FL :array [1..6] of Real; A1,A2 :Real;
Label 1,2;
begin SetColor(Yellow); Rectangle(0,0,Xmg,Ymg);
If Fmax=0 then Vn:=Xmg/40 else Vn:=Xmg/(40*Fmax); SetColor(Red);
For N:=1 to NE do begin
P[1].X:=Round(X1+X2*X[NOP[n,1],1]);
P[2].X:=Round(X1+X2*X[NOP[n,2],1]);
P[3].X:=Round(X1+X2*X[NOP[n,3],1]); P[1].Y:=Round(Y1-
Y2*X[NOP[n,1],2]);

```

```

P[2].Y:=Round(Y1-Y2*X[NOP[n,2],2]);   P[3].Y:=Round(Y1-
Y2*X[NOP[n,3],2]);
FillPoly(3,P);      I:=NOP[n,1];      K:=NOP[n,2];      J:=NOP[n,3];
A1:=(X[k,1]+X[i,1]+X[j,1])/3;          A2:=(X[k,2]+X[i,2]+X[j,2])/3;
XG1[1]:=Round(X1+X2*A1);          YG1[1]:=Round(Y1-Y2*A2);
XG[1]:=XG1[1]+Round(0.5*Vn*SK[n,4]*cos(SK[n,6]/57.3));
YG[1]:=YG1[1]-Round(0.5*Vn*SK[n,4]*sin(SK[n,6]/57.3));
XG[2]:=XG1[1]-Round(0.5*Vn*SK[n,4]*cos(SK[n,6]/57.3));
YG[2]:=YG1[1]+Round(0.5*Vn*SK[n,4]*sin(SK[n,6]/57.3));
Line(XG[1],YG[1],XG[2],YG[2]);
XG[1]:=XG1[1]+Round(0.5*Vn*SK[n,5]*cos((90+SK[n,6])/57.3));
YG[1]:=YG1[1]-Round(0.5*Vn*SK[n,5]*sin((90+SK[n,6])/57.3));
XG[2]:=XG1[1]-Round(0.5*Vn*SK[n,5]*cos((90+SK[n,6])/57.3));
YG[2]:=YG1[1]+Round(0.5*Vn*SK[n,5]*sin((90+SK[n,6])/57.3));
Line(XG[1],YG[1],XG[2],YG[2]);          end;  end;
{-----}
begin          3: ClrScr;          Writeln(' 0. Вихід з програми');
Writeln(' 1. Інструкція');          Writeln(' 2. Введення початкових даних з
дисплею');
Writeln(' 3. Введення початкових даних з файла');
Readln(KD);
If KD=0 then goto 5;          If (KD=2) or (KD=3) then goto 1;
If KD=1 then INSTRUK;          Repeat          Until KeyPressed;          goto 3;          1:
          DATA;
          INFORM;
Assign(F,'RES.DAT');          ReWrite(F);
Writeln(F,'          РОЗРАХУНОК ПЛОСКОЇ ЗАДАЧІ ТЕОРІЇ ',
'ПРУЖНОСТІ          ');
Writeln(F '          МЕТОДОМ КІНЦЕВИХ ЕЛЕМЕНТІВ');
Writeln(F,' Товщина пластини прийнята 1 см. ');          Writeln(F, ' ');
Writeln(F '          Початкові дані');
Writeln(F,'_____';
'_____');
Writeln(F,' Вузлів-',NP,' Елементів-',NE,' Закріплених' ' вузлів-',NB);
Writeln(F,'_____');

```

```

'_____'); Writeln(F, ' ');
Writeln(F, '_____ Координати вузлів, в мм. ');
Writeln(F, '_____');
'_____');
Writeln(F, ' Вузол X Y ');
' Вузол X Y Вузол X Y ');
Writeln(F, '_____');
'_____');
For i:=1 to NP do begin If (i=Round(i/3)*3) or (i=NP)
then
Writeln(F,i:6,X[i,1]:9:2,X[i,2]:9:2) else Write(F,i:6,X[i,1]:9:2, X[i,2]:9:2);
end;
Writeln(F, '_____');
'_____'); Writeln(F, ' ');
Writeln(F, '_____ Навантаження у вузлах, в кН');
Writeln(F, '_____');
'_____');
Writeln(F, ' Вузол Px Py Вузол Px ',
' Py ');
Writeln(F, '_____');
'_____'); M:=0;
For i:=1 to NP do begin K1:=0;
For j:=1 to 2 do begin If R1[2*(i-1)+j]<>0 then K1:=1; end;
If K1=0 then goto 8; M:=M+1;
If (M=Round(M/2)*2) or (i=NP) then
Writeln(F,i:5,R1[2*(i-1)+1]:10:5,R1[2*(i-1)+2]:10:5)
else Write(F,i:5,R1[2*(i-1)+1]:10:5,R1[2*(i-1)+2]:10:5); 8: end;
Writeln(F, '_____');
'_____'); Writeln(F, ' '); 6:;
Writeln(F, '_____ Зв'язки елементів ');
Writeln(F, '_____');
'_____'); Writeln(F, ' Ел-т Зв'язки ', ' Ел-т Зв'язки
';
' Ел-т Зв'язки ');
Writeln(F, '_____');

```

```

'_____');
For i:=1 to NE do      begin      If (i=Round(i/3)*3) or (i=NE) then
Writeln(F,i:5,NOP[i,1]:6,NOP[i,2]:5,NOP[i,3]:5)
else Write(F,i:5,NOP[i,1]:6,NOP[i,2]:5,NOP[i,3]:5);      end;
Writeln(F,'_____');
'_____');      Writeln(F,' ');
Writeln(F,'      Граничні умови: вузол і вид його закріплення');
Writeln(F,'_____');
'_____');
For i:=1 to NB do      begin      If (i=Round(i/6)*6) or (i=NB)
then
Writeln(F,NBC[i]:6,NFIX[i]:6) else      Write(F,NBC[i]:6,NFIX[i]:6);      end;
Writeln(' Чекайте, формується і вирішується система ', NSZ,' рівнянь');
      SILA;
      MATRICA;
      SISTEM;
      SIGMA;
ClrScr;  GR;      Readln;  Repeat      Until KeyPressed;  CloseGraph;
Writeln(' Результати розрахунків дивіться у файлі RES.DAT');
Writeln(F,'_____');
'_____');      Writeln(F,' ');      Writeln(F,' ');
Writeln(F,'      Результати розрахунків – переміщення вузлів, мм');
Writeln(F,'_____');
'_____');      Writeln(F,' Вузол  X      Y ');
' Вузол  X      Y ',' Вузол  X      Y ');
Writeln(F,'_____');
'_____');
For N:=1 to NP do      begin      If (n=Round(n/3)*3) or (N=NP)
then
Writeln(F,N:5,R1[2*(n-1)+1]:10:5,R1[2*(n-1)+2]:10:5)      else
Write(F,N:5
R1[2*(n-1)+1]:10:5,R1[2*(n-1)+2]:10:5) ;      end;
Writeln(F,'_____');
'_____');      Writeln(F,' ');      Writeln(F,'      Результати
розрахунків -',

```

```

'Напруження в центрах ваги елементів, в МПа');
Writeln(F,'_____');
'_____');      Writeln(F,'Эл-т N   SIGX   SIGY ',
' TAUXY   SIGMAX   SIGMIN   ALFA ');
Writeln(F,'_____');
'_____');
For N:=1 to NE do   begin
Writeln(F,N:5,SK[n,1]:11:4,SK[n,2]:11:4,SK[n,3]:11:4
,SK[n,4]:11:4,SK[n,5]:11:4,SK[n,6]:11:4);      end;
Writeln(F,'_____');
'_____'); Close(F);      5: end.

```

ДОДАТОК В

Текст програми розрахунку вісесиметричних товстостінних циліндрів

```
unit cillserv; interface
Procedure SERVIS;
implementation
Uses TpCrt, TpWindow, User, Printer, Dos, cilldata, edt;
    Procedure SERVIS;
Label 0,1,2,3,4,5,6; Var W : WindowPtr; Str_N : String80; Str_E :
String80;
Str1 : String; SC : Word; Zooming, Nodes_On : Boolean;
    Procedure Cur_Nodes(N, Attr : Byte);
{ N - номер рядка Attr - атрибути курсора }
Var i, j, до, s : Byte; P : ^Byte; OffSet_P : Word;
begin N := N + 2; If Nodes_On then
begin s := 6; OffSet_P := Ofs(Nodes_Pos); end
Else begin s := 7; OffSet_P := Ofs(Elem_Pos); end;
For i := 1 to s do begin If i = 1 then j := 2 Else
begin P := Ptr(DSEG, OffSet_P+i-2); j := P^ + 2; end;
If i = s then до := 79 Else begin P := Ptr(DSEG, OffSet_P+i-1);
до := P^ - 2; end; до := до - j + 1;
ChangeAttributeWindow(до, N, j, Attr); end; end;
{-----}
    Function Menu_D (i :integer) : integer;
Var Dr : Char;
begin Window(24,8,57,11); TextBackGround(Blue); TextColor(Yellow);
ClrScr; HiddenCursor; SetFrameChars('|','=','-','-', 'L','Г');
FrameWindow(23, 7, 58, 12, YellowOnBlue, YellowOnBlue, '');
WriteLn(' Продовжити роботу'); WriteLn(' Дані записати у файл');
WriteLn(' Редагування даних'); Write(' Закінчити роботу');
ChangeAttributeWindow(32, i, 2, YellowOnRed); Repeat
Dr := ReadKey; Case Dr of #50: {Down} If i < 5 then
Begin ChangeAttributeWindow(32, i, 2, YellowOnBlue); Inc(i);
If i = 5 then i := 1; ChangeAttributeWindow(32, i, 2, YellowOnRed); End;
#$48: {Up} If i > 0 then Begin
```



```

ChangeAttributeWindow(32, i, 2, YellowOnBlue); Dec(i); If i = 0 then i := 4;
ChangeAttributeWindow(32, i, 2, YellowOnRed); End; end;
Until dr=#13; Menu_D:=i;end;
{-----}
Function Str_Nodes(N : Byte) : String80;
{ N - номер строки }
Var Str1 : String[80]; Str2 : String[20]; Var i, j : Byte;
begin Str1 := Str_N; Str(N:2, Str2); Move(Str2[1], Str1[2], 2);
For j := 1 to 4 do begin Str2 := Real_Right(Nodes[N,j],3,11,i);
If (j < 3) then Move(Str2[1], Str1[Nodes_Pos[j]+2], 11)
Else Move(Str2[1], Str1[Nodes_Pos[j+1]+2], 10); end;
Str2 := ByteB_StrR(GU[N],2); Move(Str2[1], Str1[Nodes_Pos[3]+2], 2);
Str_Nodes := Str1; end;
{-----}
Function Str_Elem(N : Byte) : String80;
{ N - номер строки }
Var Str1 : String[80]; Str2 : String[20]; Var i, j, k : Byte;
begin k := 5; Str1 := Str_E; Str(N:2, Str2); Move(Str2[1], Str1[2], 2);
For j := 1 to 4 do begin If j = 4 then begin i := 4; k := 4; end
Else i := j; Str(Elem[N,j]:k, Str2); Move(Str2[1], Str1[Elem_Pos[i]+1], k); end;
Str2 := Real_Right(F_E[Elem[N,4],1], 0, 11, i); Move(Str2[1],
Str1[Elem_Pos[5]+2], 11); Str2 := Real_Right(F_E[Elem[N,4],2], 3, 9, i);
Move(Str2[1], Str1[Elem_Pos[6]+2], 9); Str_Elem := Str1; end;
{-----}
Var W_N1, W_N2, W_E1, W_E2 : Byte;
{-----}
Procedure Init_Windows(N, M : Byte);
{ N - перший рядок у вікні N M - перший рядок у вікні E }
Var Str1, Str2, Str3 : String[80]; { Str_N - вертикальні лінії
Str1 - з текстом Str2 - перетин ліній } i, j, до : Byte;
begin Window(1,W_N1+1,80,W_N2); TextColor(Yellow);
FillChar(Str2, SizeOf(Str2), '='); Str2[0] := #80;
For i := 1 to 6 do begin Str2[Nodes_Pos[i]]:= '⌘'; end;
Str1 := Str_N; Str1[2] := 'N'; Str1[3] := '%';
Str1[11] := 'R'; Str1[12] := ','; Str1[13] := 'M'; Str1[14] := 'M';

```

```

Str1[28] := 'Y'; Str1[29] := ','; Str1[30] := 'm'; Str1[31] := 'm';
Str1[42] := 'T'; Str1[43] := 'Y';
Str1[53] := 'P'; Str1[54] := 'r'; Str1[55] := ','; Str1[56] := 'k'; Str1[57] := 'H';
Str1[67] := 'P'; Str1[68] := 'y'; Str1[69] := ','; Str1[70] := 'k'; Str1[71] := 'H';
FastWriteWindow(Str1, 1, 1, YellowOnBlue);
FastWriteWindow(Str2, 2, 1, YellowOnBlue);
For i := 3 to W_N2-W_N1 do
begin If (N <= N_Nodes) then Str3 := Str_Nodes(N) Else Str3 := Str_N;
Inc(N); FastWriteWindow(Str3, i, 1, YellowOnBlue); end;
{----- Init_Elem -----}
Window(1,W_E1+1,80,W_E2); TextColor(Yellow);
FillChar(Str2, SizeOf(Str2), '='); Str2[0] := #80;
For i := 1 to 6 do Str2[Elem_Pos[i]]:= '⚡'; Str2[Elem_Pos[2]]:= '⚡';
Str2[Elem_Pos[3]]:= '⚡'; Str1 := Str_E; Str1[2] := 'N'; Str1[3] := '%';
Str1[10] := 'C'; Str1[12] := 'B'; Str1[14] := 'Я'; Str1[16] := 'З'; Str1[18] := 'И';
Str1[22] := 'Э'; Str1[24] := 'Л'; Str1[26] := 'Е'; Str1[28] := 'М';
Str1[30] := 'е'; Str1[31] := ' '; Str1[32] := 'И'; Str1[34] := 'Т'; Str1[36] := 'а';
Str1[45] := 'Т'; Str1[47] := 'И'; Str1[49] := 'П';
Str1[60] := 'Е'; Str1[61] := ','; Str1[62] := 'М'; Str1[63] := 'П';
Str1[64] := 'а'; Str1[75] := 'v'; FastWriteWindow(Str1, 1, 1, YellowOnBlue);
FastWriteWindow(Str2, 2, 1, YellowOnBlue);
For i := 3 to W_E2-W_E1 do begin If (M <= N_Elem) then Str3 :=
Str_Elem(M) Else Str3 := Str_E; Inc(M); FastWriteWindow(Str3, i, 1,
YellowOnBlue); end;
Window(1,W_E1,80,W_E2); end;
{-----}

```

Procedure Edit_Dat;

{ N, M - перший рядок у вікні (1,2) NC, MC - положення строкового курсора

NCC, MCC - положення курсора (N% стовпця) і - номер рядка поточного вікна

W1, W2 - Координати Y поточного вікна і1 - номер рядка іншого вікна

NN - кількість елементів (N_Nodes, N_Elem)

NNN - кількість стовпців (N - 7, E - 8) } Label 0, 1, 2, 3,4, 51;

```

Var i, i1, k      : Byte;   j      : Integer;  Str1      : String80;
W1, W2          : Byte;   N, M, NC, MC, NCC, MCC  : Byte;
NN, NNN        : Byte;   SC      : Word;
Dr              : Char Absolute SC;   Dr_b      : Byte; {-----}
Begin 4: If Nodes_On then begin W1 := W_N1; W2 := W_N2;
NNN := 6; NN := N_Nodes; end Else begin W1 := W_E1;
W2 := W_E2; NNN := 7; NN := N_Elem; end;
N := 1; M := 1; NC := 1; MC := 1; NCC := 1; MCC := 1; i := 1 + NC - N;
i1 := 1 + MC - M; Insert := True; NormalCursor; TextBackGround(Blue);
Window(1,W1+1,80,W2); Cur_Nodes(i, YellowOnRed); GotoXY(2, i+2);
Repeat SC := ReadKeyWord;0: Case Dr of #0: Begin Dr_b := Hi(SC);
1: Case Dr_b of $50: { Down } Begin EndStr := False;
If NC < NN then If i < W2-W1-2 then Begin
Cur_Nodes(i, YellowOnBlue); Inc(i); Inc(NC); Cur_Nodes(i, YellowOnRed);
Goto 3; End Else begin Cur_Nodes(i, YellowOnBlue);
ScrollWindowUp(1, W1+3, 80, W2, 1); Inc(N); Inc(NC);
If Nodes_On then Str1 := Str_Nodes(NC) Else Str1 := Str_Elem(NC);
FastWriteWindow(Str1, i+2, 1, YellowOnBlue); Cur_Nodes(i, YellowOnRed);
Goto 3; End Else begin Goto 3; end; End;
$48: { Up } Begin EndStr := False; If (NC > N) then
Begin Cur_Nodes(i, YellowOnBlue); Dec(i); Dec(NC);
Cur_Nodes(i, YellowOnRed); Goto 3; End
Else If (NC = N) and (N > 1) then begin Cur_Nodes(i, YellowOnBlue);
ScrollWindowDown(1, W1+3, 80, W2, 1); Dec(N); Dec(NC);
If Nodes_On then Str1 := Str_Nodes(NC) Else Str1 := Str_Elem(NC);
FastWriteWindow(Str1, i+2, 1, YellowOnBlue); Cur_Nodes(i, YellowOnRed);
Goto 3; End Else begin Goto 3; end; End;
$4B, { Left } $47: { Home } Begin EndStr := True;
If (NCC > 1) then Dec(NCC) Else NCC := NNN;
Goto 3; {As End & Right} end;
$4D, { Right } $4F: { End } Goto 2; $52: { Ins }
Begin Insert := Not Insert; If Insert then NormalCursor
Else BlockCursor; End; $40: { F6 - Switch Windows }
Begin Cur_Nodes(i, LtCyanOnBlue);
ChangeAttribute(80, W1, 1, GrayOnCyan); W1 := i; i := i1; i1 := W1;

```

```

W1 := N; N := M; M := W1;      W1 := NC; NC := MC; MC := W1;
W1 := NCC; NCC := MCC; MCC := W1;      Nodes_On := Not Nodes_On;
If Nodes_On then      begin      N_Elem := NN;      NN := N_Nodes;
W1 := W_N1;      W2 := W_N2;      NNN := 6;      end      Else
begin      N_Nodes := NN;      NN := N_Elem;      W1 := W_E1;
W2 := W_E2;      NNN := 7;      end;
Window(1,W1+1,80,W2);      ChangeAttribute(80, W1, 1, YellowOnCyan);
Cur_Nodes(i, YellowOnRed);      Goto 3;      end;
$41:  { F7 - Insert String }      If (NN < N_Max) then      begin
Inc(NN);      If Nodes_On then      begin
For j := NN downto NC+1 do      begin      For k := 1 to 4 do
Nodes[j,k] := Nodes[j-1,k];      GU[j] := GU[j-1];      end;
For k := 1 to 4 do      Nodes[NC+1,k] := 0;      GU[NC+1] := 0;      end
Else begin      For j := NN downto NC+1 do      begin
F_E[j] := F_E[j-1];      For k := 1 to 4 do      Elem[j,k] := Elem[j-1,k];      end;
For k := 1 to 4 do      Elem[NC+1,k] := 1;      end;
If (i+2+W1 < W2) then begin      ScrollWindowDown(6, i+3+W1, 80, W2, 1);
If Nodes_On then Str1 := Str_Nodes(NC+1)
Else Str1 := Str_Elem(NC+1); FastWriteWindow(Str1, i+3, 1, YellowOnBlue);
If (NN-NC+i+1 < W2-W1) then      begin      Str(NN:3, Str1);
FastWriteWindow(Str1, NN-NC+i+2, 1, YellowOnBlue);      end;      end;
Dr_b := $50;      Goto 1;      end; $42:  { F8 - Delete String }
If (Nodes_On and (NN > 2)) or ((Not Nodes_On) and (NN > 1)) then
begin      Dec(NN);      If Nodes_On then      begin
For j := NC to NN do      begin
For k := 1 to 4 do      Nodes[j,k] := Nodes[j+1,k]; GU[j] := GU[j+1]; end;      end
Else begin      For j := NC to NN do      begin
For k:=1 to 2 do      F_E[j,k] := F_E[j+1,k];
For k := 1 to 4 do      Elem[j,k] := Elem[j+1,k];      end;      end;
If Nodes_On then Str1 := Str_Nodes(NC)      Else Str1 := Str_Elem(NC);
Str1 := Copy(Str1, 4, 80-4);      If (W2 >= i+3+W1) then
ScrollWindowUp(6, i+3+W1, 80, W2, 1);
FastWriteWindow(Str1, i+2, 4, YellowOnBlue);
Cur_Nodes(i, YellowOnRed);
If (NC+W2-W1-i-3 < NN) then      If (i < W2-W1-2) then

```

```

begin          If Nodes_On then Str1 := Str_Nodes(NC+W2-W1-i-1)
Else Str1 := Str_Elem(NC+W2-W1-i-1);          Str1 := Copy(Str1, 4, 80-4);
FastWriteWindow(Str1, W2-W1, 4, YellowOnBlue); end Else Else
begin  If Nodes_On then FastWriteWindow(Str_N, W2-W1, 1, YellowOnBlue)
Else FastWriteWindow(Str_E, W2-W1, 1, YellowOnBlue);
FastTextWindow(' ', NN-NC+i+3, 1);          end;
If (NC > NN) then          begin
If Nodes_On then FastWriteWindow(Str_N, i+2, 1, YellowOnBlue)
Else FastWriteWindow(Str_E, i+2, 1, YellowOnBlue);
Dr_b := $48;          Goto 1;          end Else          begin
Goto 3;          end;          end Else Goto 3;  $44 :  { F10 }
Begin          If Nodes_On then N_Nodes:=NN else N_Elem:=NN;
NormalCursor;  Window(1,W1,80,W2); Exit;          end; End; end;
#$09 :  { Tab }  Begin2:  EndStr := False;
If (NCC < NNN) then Inc(NCC) Else NCC := 1;3:  If (NCC > 1) then
begin  If Nodes_On then  If (NCC = 4) then
SC := Edit_ByteB(GU[NC], Nodes_Pos[3]+2, i+2, 2, YellowOnRed) Else
SC := Edit_Real(Nodes[NC,NCC-1-(NCC div 4)],3,
-99999.999, 999999.999,Nodes_Pos[NCC-1]+2, i+2, 10,          YellowOnRed)
Else  Case NCC of  2,3,4 : SC := Edit_Byte(Elem[NC,NCC-1], 1, N_Nodes,
Elem_Pos[NCC-1]+2, i+2, 4, YellowOnRed);  5 : Begin
SC := Edit_Byte(Elem[NC,4], 1, N_Elem,
Elem_Pos[NCC-1]+2, i+2, 3, YellowOnRed);
For j := 1 to N_Elem do If (Elem[NC,4] = Elem[j,4]) and (j <> NC) then Goto
51;
F_E[Elem[NC,4],1] := 200000;  F_E[Elem[NC,4],2] := 0.3;
51:          Str1 := Str_Elem(NC);  FastTextWindow(Str1, i+2, 1);          End;
6,7: Begin          If NCC=6 then
SC := Edit_Real(F_E[Elem[NC,4],NCC-5], 0, 0, 9999999,
Elem_Pos[NCC-1]+2, i+2, 11, YellowOnRed)
else SC := Edit_Real(F_E[Elem[NC,4],NCC-5], 3,0.000, 0.500,
Elem_Pos[NCC-1]+2, i+2, 9, YellowOnRed);          If NCC=6 then
Str(F_E[Elem[NC,4],NCC-5]:11:0,Str1) else Str(F_E[Elem[NC,4],NCC-
5]:9:3,Str1);

```

```

For j := i+2 to W2-W1 do If (Elem[N+j-3,4] = Elem[NC,4]) and (N+j-3<= NN)
then FastTextWindow(Str1, j, Elem_Pos[NCC-1]+2); End; End;
Goto 0; end Else GotoXY(2, i+2); end; End;
Until False; If Nodes_On then N_Nodes := NN Else N_Elem := NN;
Window(1,W1,80,W2); end;
{-----}
begin
{-----}
FillChar(Str_N, SizeOf(Str_N), ' '); Str_N[0] := #80;
For i := 1 to 6 do Str_N[Nodes_Pos[i]]:= ' | ';
{-----}
FillChar(Str_E, SizeOf(Str_E), ' '); Str_E[0] := #80;
For i := 1 to 6 do Str_E[Elem_Pos[i]]:= ' | '; Zooming := False; Nodes_On :=
True;
{-----}
i := 1;0: Window(1,2,80,24); TextBackGround(Blue); ClrScr;
Window(1,1,80,1); TextBackGround(LightGray); TextColor(Black); ClrScr;
FastCenter( 'РОЗРАХУНОК ТОВСТОСТІННИХ ЦИЛІНДРІВ МЕТОДОМ
КІНЦЕВИХ ЕЛЕМЕНТІВ', 1, $70);
Window(1,25,80,25); TextBackGround(LightGray); ClrScr; FastCenter(
'F7-добавлення, F8- видалення, Табл-введення даних, F6-зміна вікон, F10-
вихід',1, $70);
W_N1 := 2; W_N2 := 13; W_E1 := 14; W_E2 := 24;
TextBackGround(Cyan); Window(1,W N1,80,W N2); ClrEol;
FastCenter('Координати вузлів, граничні умови і навантаження у вузлах', 1
$3E); Window(1,W E1,80,W E2); ClrEol;
FastCenter('Зв'язки ел-тів, тип перерізу, модуль пружності і коефіцієнт
Пуасона', 1, GrayOnCyan);
3: For i:=1 to N_Nodes do begin GU[i]:=0; GDU[i]:=Round(GDU[i]);
For J:=1 to 2 do begin If Round(GDU[i])=0 then goto 5;
If Round(GDU[i]/2)*2=GDU[i] then goto 1;
GU[i]:=GU[i]+Round(exp((2-J)*ln(2))); GDU[i]:=GDU[i]-1;
1:GDU[i]:=GDU[i]/10; end; 5: end;
4: Init_Windows(1, 1); Window(1,W_N1,80,W_N2); Edit_Dat;
i1:=Menu_D(i1); If i1 = 4 then Exit_Prog; If i1 = 3 then Goto 4;

```

```

For i:=1 to N_Nodes do begin GDU[i]:=0; For J:=1 to 2 do
begin If GU[i]=0 then goto 6; f GU[i]/(0.9999*exp((2-J)*ln(2)))<1 then goto
2;
GDU[i]:=GDU[i]+Round(exp((J-1)*ln(10)));
GU[i]:=GU[i]-Round(exp((2-J)*ln(2))); 2: end; 6: end;
If i1=2 then begin HiddenCursor;
If MakeWindow(W,30,10,62,12,True,True,False,$1E,$1E,$1E, ") then;
If ActivateWindow(W) then; Str1 := "; Keys_Move := $80;
FastTextWindow('Введіть назву файла:', 1, 3);
SC := Edit_String(Str1, 22, 1, 8, $4E); NormalCursor;
W := EraseTopWindow; DisposeWindow(W);
If (Str1 = ") then Str1 := 'NONAME'; Str1:=Str1+'.DAT'; Assign(F,Str1);
Rewrite(F); Writeln(F,N_Nodes:6,N_Elem:6); For i:=1 to N_Nodes do
Writeln(F,Nodes[i,1]:10:3,Nodes[i,2]:10:3,GDU[i]:4,
Nodes[i,3]:10:3,Nodes[i,4]:10:3);
For i:=1 to N_Elem do Writeln(F,Elem[i,1]:6,Elem[i,2]:6,
Elem[i,3]:6,Elem[i,4]:6,F_E[i,1]:15:3,F_E[i,2]:15:3);
Close(F); Goto 3; end; end;end.

```

ДОДАТОК Г

Текст програми автоматичного розбиття конструкції на елементи

```
{-----}
Program RAZBIV;
Uses TpCrt, TpWindow, Graph, User, BGI, Fonts, Printer, Dos;
CONST
{ Blue = 0;} YellowOnBlue = $1E { $0E};
YellowOnRed = $4E { $40}; YellowOnCyan = $3E { $30};
GrayOnCyan = $38 { $30}; LtCyanOnBlue = $1B { $30};
WhiteOnBlue = $1F { $0F}; MagentaOnBlue = $1C { $0C};
BlackOnGray = $70 { $70}; WhiteOnGray = $7F { $70};
CONST Ext = 'DAT'; N_Max = 100; E_Max = 20;
Nodes_Pos : Array [1..5] of Byte = (5,20,37,49,63);
Elem_Pos : Array [1..12] of Byte = (5,11,17,23,29,35,41,47,53,59,65,74);
CONST GDU :array [1..3] of Byte =(10,1,11);
IC : array [1..16] of Integer =(-1,1,1,-1,1,-1,-1,1,1,-1,
-1,1,-1,1,1,-1);
TYPE String3 = String[3];
{-----}
Var Nodes,Nodes1 : Array [1..100, 1..4] of Real; { Координати і сили }
GU : Array [1..100] of Byte; { Граничні умови }
N_Nodes : Byte; { Кількість вузлів }
Elem1 : Array [1..300, 1..4] of Byte; { Зв'язки і тип перерізу }
F_E : Array [1..30,1..2] of Real;
F_E1 : Array [1..300,1..2] of Real; { Модулі пружності }
N_Elem : Byte; { Кількість елементів } i, j : Byte;
FileName : String; Dr : Char; Str1 : String; SC : Word;
W : WindowPtr; KD1,Xmg,Ymg,gd,gm,t0,i1,KD,XC,YC
:Integer;
X1,X2,Y1,Y2,G1,Xa,Ya,Xk,Yk,Vn,Diam :Real;
NSZ,NSN,K1,K2 :Integer; XG,YG,XG1,YG1 :array [1..2] of Integer;
k,n,L,M,NBN,KF :Integer; R1 :array [1..200] of Real;
GU1 : Array [1..100] of Byte; N_Nodes1 : Byte;
Elem : Array [1..30, 1..10] of Byte; N_Elem1 : Byte;
XE,YE :array [1..100] of Real; NE1 :array [1..100] of Integer;
```



```

DIAG1,DIAG2      :Real;    NR,LB          :array [1..4] of Integer;
NP1,IJ,J1,J2,J3,NEL,NBW,NELBW,IK      :Integer;
NB, II,KK,TR,IR,KN1,JK,NB1,N1,IR1     :Integer;
KN2,KS1,KS2,NRT,NRTS,JL,KJ   :Integer;
INRG,INGP,IPCH,NRG,NROWS,NCOL   :Integer;
XP,YP           :array [1..100] of Real;  R           :array [1..200] of Real;
XRG,YRG,N2      :array [1..4] of Real; JT      :array[1..20,1..4] of
Integer;
NDN             :array[1..20,1..4] of Integer;
XC1,YC1         :array [1..21,1..21] of Real;
NN             :array [1..21,1..21] of Integer;
NNRB           :array [1..20,1..4,1..21] of Integer;
DETA,ETA,DSI,SI :Real;    F,Z           :Text;
S1,S2          :String[79];  AZ,AR,Del,Z1,Z2,T1,T2,Fmax      :Real;
P              :array [1..3] of PointType;  quitter          :Boolean;
Label          1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,20,21;
{$I INITGRAF.PAS} { InitGraf - підключення графіки
PrintScr - друк екрану. } {$I PRINT.INC}
{-----}
Function Menu_D (i :integer) : integer;
Var   Dr : Char;
begin Window(24,8,57,11); TextBackGround(Blue); TextColor(Yellow);
ClrScr; HiddenCursor; SetFrameChars('||','=','⌂','⌂','⌂','⌂');
FrameWindow(23, 7, 58, 12, YellowOnBlue, YellowOnBlue, "");
WriteLn(' Продовжити роботу'); WriteLn(' Дані записати у файл');
WriteLn(' Редагування даних'); Write(' Закінчити роботу');
ChangeAttributeWindow(32, i, 2, YellowOnRed);
Repeat  Dr := ReadKey; Case Dr of  #50:  {Down}
If i < 5 then Begin ChangeAttributeWindow(32, i, 2, YellowOnBlue);
Inc(i); If i = 5 then i := 1; ChangeAttributeWindow(32, i, 2, YellowOnRed);
End;  #548:  {Up}
If i > 0 then Begin ChangeAttributeWindow(32, i, 2, YellowOnBlue);
Dec(i); If i = 0 then i := 4; ChangeAttributeWindow(32, i, 2, YellowOnRed);
End; end; Until dr=#13; Menu_D:=i;end;
{-----}

```

```

Procedure Exit_Prog;
begin (* RestoreCRTMode; *) Window(1,1,80,25); TextColor(LightGray);
TextBackGround(Black); ClrScr; TextMode(CO80); WriteLn('Good Luck!');
NormalCursor; Halt(1); end;
{-----}
Procedure INSTRUK;
Var i, j : Byte; W1, W2 : Integer; Dr : Char; CONST
N_Help = 26; S_Help : Array [1..N_Help] of String[57] = (
' Програма дозволяє виконати автоматичну розбивку',
' довільної області на елементи і є підгото-',
' вкою при розрахунку плоского задачі теорії пружності і ',
' товстостінних циліндрів методом кінцевих елементів. ',
' Виконується наступна підготовча робота. Об-',
' ласть розбивається на довільні чотирикутні зони ',
' із загальними вершинами, номеруються зони і їх вершини за ',
' суцільною нумерацією. У довільній системі координат ',
' (для товстостінних циліндрів вісь Y співпадає з віссю ци-',
' ліндра) визначаються координати вершин, фіксуються су-',
' сідні зони кожного боку чотирикутника, записується ',
' кількість стовпців і рядків після розбиття на елементи ',
' а також номери вузлів, що входять до даного чоти-',
' рикутника, причому, в тій же послідовності, в ',
' якій фіксувалися сусідні зони. Максимальноа кількість зон – ',
' 20. ',
' Послідовність накладання зв'язків: уздовж осей R і Y ',
' Наприклад: 2 10 - у вузлі 2 відсутнє переміщення упродовж ',
' осі R, але можливо переміщення уздовж осі Y. Додат ',
' ні знаки зовнішніх сил - подовж додатних напрямків ',
' осей координат. ',
' У результаті рішення одержуємо масив даних (координа-',
' ти вузлів, зв'язки елементів, граничні умови і навантаження ',
' у вузлах), готовий до використання програмами PLASTINA ',
' і CILINDR. ',
' Результати розрахунків дивися у файлі CIL.DAT ');
Begin HiddenCursor;

```

```

If MakeWindow(W,10,4,70,21,True,True,False,$70,$7F,$7F, ' Help ') then;
If ActivateWindow(W) then;
For i := 1 to 15 do FastTextWindow(S_Help[i], 1+i, 2); W1 := 1; W2 := 16;
Repeat Dr := ReadKey; If (Dr = #0) and KeyPressed then
Begin Dr := ReadKey; Case Dr of #50: {Down}
If W2 < N_Help+1 then Begin ScrollWindow(True, 1);
FastTextWindow(S_Help[W2], 16, 2); Inc(W2); Inc(W1); End;
#548: {Up} If W1 > 1 then Begin ScrollWindow(False, 1);
Dec(W1);Dec(W2); If W1 > 1 then FastTextWindow(S_Help[W1-1], 1, 2);
End; #551: {Page Down} If W2 < N_Help+1 then
Begin ScrollWindow(True, 15); Inc(W1,15);Inc(W2,15);
For i := 1 to 15 do If W1-1+i < N_Help+1 then FastTextWindow(S_Help[W1-
1+i], 1+i, 2); End;
#549: {Page Up} If W1 > 1 then Begin If W1 > 15 then j := 15
Else j := W1-1; ScrollWindow(False, j); Dec(W1,j);Dec(W2,j);
For i := 1 to j do If W1+i > 1 then FastTextWindow(S_Help[W1-
2+i], i, 2); End; End; end Else If (Dr = #27) then begin
NormalCursor; DisposeWindow(W); Exit; end; Until False; end;
{-----}
Function Menu (i : Byte) : Byte;
Var Dr : Char;
begin Window(24,8,57,11); TextBackGround(Blue); TextColor(Yellow);
ClrScr; HiddenCursor; SetFrameChars('||','=','┘','┙','┚','┛');
FrameWindow(23, 7, 58, 12, $1E, $1E,");
WriteLn(' Введення початкових даних з файла');
WriteLn(' Введення початкових даних з дисплею');
WriteLn(' Інструкція'); Write(' Вихід з програми');
ChangeAttributeWindow(32, i, 2, $4E);
Repeat Dr := ReadKey; KD:=i; If (Dr = #0) and KeyPressed then
Begin Dr := ReadKey; Case Dr of #50: {Down} If i < 5 then
Begin ChangeAttributeWindow(32, i, 2, $1E); Inc(i);
If i = 5 then i := 1; ChangeAttributeWindow(32, i, 2, $4E); End;
#548: {Up} If i > 0 then Begin ChangeAttributeWindow(32, i, 2, $1E);
Dec(i); If i = 0 then i := 4; ChangeAttributeWindow(32, i, 2, $4E);
End; End; end Else If (Dr = #27) or (Dr = #13) then begin

```

```

If Dr = #27 then Menu := 4      Else Menu := i;      NormalCursor;
Exit;      end;  Until False;end;
{-----}
      Procedure SERVIS;
Label 0,1,2,3,4;
Var W      : WindowPtr;  {$I EDT.PAS} (* Edit Real, Byte & Bits *)
Var  Str_N : String80;   Str_E : String80; Zooming, Nodes_On : Boolean;
      Procedure Cur_Nodes(N, Attr : Byte);
      { N - номер строки  Attr - атрибуты курсора }
Var i, j, k, s : Byte;   P      : ^Byte;   OffSet_P : Word;
begin  N := N + 2; If Nodes_On then begin  s := 6; OffSet_P :=
Ofs(Nodes_Pos); end  Else  begin s := 13; OffSet_P := Ofs(Elem_Pos); end;
For i := 1 to s do  begin  If i = 1 then j := 2  Else  begin
P := Ptr(DSEG, OffSet_P+i-2);   j := P^ + 1;   end;  If i = s then k := 79
Else  begin  P := Ptr(DSEG, OffSet_P+i-1); k := P^ - 1; end; k := k - j + 1;
ChangeAttributeWindow(k, N, j, Attr);  end; end;
{-----}
      Function Str_Nodes(N : Byte) : String80;
      { N - номер рядка }
Var Str1 : String[80];   Str2 : String[20]; Var i, j : Byte;
begin  Str1 := Str_N;   Str(N:2, Str2);  Move(Str2[1], Str1[2], 2);
For j := 1 to 4 do  begin  Str2 := Real_Right(Nodes[N,j],3,11,i);
If (j < 3) then Move(Str2[1], Str1[Nodes_Pos[j]+2], 11)
Else Move(Str2[1], Str1[Nodes_Pos[j+1]+2], 11);  end;
Str2 := ByteB_StrR(GU[N],2);  Move(Str2[1], Str1[Nodes_Pos[3]+2], 2);
Str_Nodes := Str1; end;
{-----}
Function Str_Elem(N : Byte) : String80;
      { N - номер рядка }
Var Str1 : String[80];   Str2 : String[20]; Var i, j, k : Byte;
begin  k := 5;  Str1 := Str_E;  Str(N:2, Str2);  Move(Str2[1], Str1[2], 2);
For j := 1 to 12 do  begin  i := j;  Str(Elem[N,j]:k, Str2);
Move(Str2[1], Str1[Elem_Pos[i]+1], k); end; Str2 := Real_Right(F_E[N,1], 0,
7, k);  Move(Str2[1], Str1[Elem_Pos[11]+2], 7);
Str2 := Real_Right(F_E[N,2], 2, 4, k); Move(Str2[1], Str1[Elem_Pos[12]+2], 4);

```

```

Str_Elem := Str1; end;
{-----}
Var W_N1, W_N2, W_E1, W_E2 : Byte;
{-----}
    Procedure Init_Windows(N, M : Byte);
{ N - перший рядок у вікні N M - перший рядок у вікні E }
Var Str1, Str2, Str3 : String[80]; { Str_N - вертикальні лінії
Str1 - з текстом Str2 - перетин ліній } i, j, до : Byte;
begin Window(1,W_N1+1,80,W_N2); TextColor(Yellow);
FillChar(Str2, SizeOf(Str2), '='); Str2[0] := #80; For i := 1 to 6 do
begin Str2[Nodes_Pos[i]]:= '⚡'; end; Str1 := Str_N; Str1[2] := 'N'; Str1[3] :=
'%'; Str1[11] := 'R'; Str1[12] := ','; Str1[13] := 'M'; Str1[14] := 'M';
Str1[28] := 'Y'; Str1[29] := ','; Str1[30] := 'M'; Str1[31] := 'M';
Str1[42] := 'T'; Str1[43] := 'Y'; Str1[59] := '*'; Str1[73] := '*';
Str1[52] := 'P'; Str1[53] := 'r'; Str1[54] := ','; Str1[56] := 'M';
Str1[57] := 'П'; Str1[58] := 'a'; Str1[60] := 'M'; Str1[61] := 'M';
Str1[66] := 'P'; Str1[67] := 'y'; Str1[68] := ','; Str1[70] := 'M';
Str1[71] := 'П'; Str1[72] := 'a'; Str1[74] := 'M'; Str1[75] := 'M';
FastWriteWindow(Str1, 1, 1, YellowOnBlue);
FastWriteWindow(Str2, 2, 1, YellowOnBlue);
For i := 3 to W_N2-W_N1 do begin If (N <= N_Nodes) then Str3 :=
Str_Nodes(N) Else Str3 := Str_N; Inc(N); FastWriteWindow(Str3, i, 1,
YellowOnBlue); end;
{----- Init_Elem -----} Window(1,W_E1+1,80,W_E2);
TextColor(Yellow);
FillChar(Str2, SizeOf(Str2), '='); Str2[0] := #80; For i := 1 to 12 do
Str2[Elem_Pos[i]]:= '⚡'; Str2[Elem_Pos[2]]:= '⚡'; Str2[Elem_Pos[3]]:= '⚡';
Str2[Elem_Pos[4]]:= '⚡'; Str2[Elem_Pos[8]]:= '⚡'; Str2[Elem_Pos[9]]:= '⚡';
Str2[Elem_Pos[10]]:= '⚡'; Str1 := Str_E; Str1[2] := 'N'; Str1[3] := '%';
Str1[12] := ','; Str1[17] := ','; Str1[11] := '3'; Str1[15] := 'o'; Str1[19] := 'H';
Str1[23] := 'H'; Str1[30] := 'C'; Str1[31] := 'T'; Str1[32] := 'p'; Str1[33] := 'o';
Str1[34] := 'κ'; Str1[36] := 'C'; Str1[37] := '-'; Str1[38] := 'H'; Str1[39] := 'o';
Str1[40] := 'B'; Str1[47] := 'Y'; Str1[51] := '3'; Str1[55] := 'л'; Str1[59] := 'H';
Str1[53] := ','; Str1[67] := 'E'; Str1[68] := ','; Str1[70] := 'M'; Str1[71] := 'П';
Str1[72] := 'a'; Str1[78] := 'v'; FastWriteWindow(Str1, 1, 1, YellowOnBlue);

```

```

FastWriteWindow(Str2, 2, 1, YellowOnBlue);
For i := 3 to W_E2-W_E1 do begin If (M <= N_Elem) then Str3 :=
Str_Elem(M) Else Str3 := Str_E; Inc(M);
FastWriteWindow(Str3, i, 1, YellowOnBlue); end;
Window(1,W_E1,80,W_E2); end;
{-----}
      Procedure Edit_Dat;
{ N, M - перший рядок у вікні (1,2) NC, MC - положення строкового
курсора
NCC, MCC - положення курсора (N стовпця) i - номер рядка поточного
вікна W1, W2 - Координати Y поточного вікна i1 - номер рядка
іншого вікна
NN - кількість елементів (N_Nodes, N_Elem)
NNN - кількість стовпців (N - 7, E - 8) } Label 0, 1, 2, 3,4, 51;
Var i, i1, k : Byte; j : Integer; Str1 : String80;
W1, W2 : Byte; N, M, NC, MC, NCC, MCC : Byte;
NN, NNN : Byte; SC : Word;
Dr : Char Absolute SC; Dr_b : Byte;
{-----}
begin 4: If Nodes_On then begin W1 := W_N1; W2 := W_N2;
NNN := 6; NN := N_Nodes; end Else begin W1 := W_E1;
W2 := W_E2; NNN := 13; NN := N_Elem; end;
N := 1; M := 1; NC := 1; MC := 1; NCC := 1; MCC := 1; i := 1 + NC - N;
i1 := 1 + MC - M; Insert := True; NormalCursor; TextBackGround(Blue);
Window(1,W1+1,80,W2); Cur_Nodes(i, YellowOnRed); GotoXY(2, i+2);
Repeat SC := ReadKeyWord;0: Case Dr of #0: Begin Dr_b := Hi(SC);
1: Case Dr_b of $50: { Down } Begin EndStr := False;
If NC < NN then If i < W2-W1-2 then Begin Cur_Nodes(i, YellowOnBlue);
Inc(i); Inc(NC); Cur_Nodes(i, YellowOnRed); Goto 3; End
Else begin Cur_Nodes(i, YellowOnBlue);
ScrollWindowUp(1, W1+3, 80, W2, 1); Inc(N); Inc(NC);
If Nodes_On then Str1 := Str_Nodes(NC) Else Str1 := Str_Elem(NC);
FastWriteWindow(Str1, i+2, 1, YellowOnBlue); Cur_Nodes(i, YellowOnRed);
Goto 3; End Else begin Goto 3; end; End;
$48: { Up } Begin EndStr := False; If (NC > N) then

```

```

Begin      Cur_Nodes(i, YellowOnBlue);      Dec(i); Dec(NC);
Cur_Nodes(i, YellowOnRed);      Goto 3;      End
Else If (NC = N) and (N > 1) then      begin
Cur_Nodes(i, YellowOnBlue);      ScrollWindowDown(1, W1+3, 80, W2, 1);
Dec(N); Dec(NC);      If Nodes_On then Str1 := Str_Nodes(NC)
Else Str1 := Str_Elem(NC);      FastWriteWindow(Str1, i+2, 1, YellowOnBlue);
Cur_Nodes(i, YellowOnRed);      Goto 3;      End      Else begin
Goto 3;      end;      End; $4B,      { Left }      $47:      { Home }      Begin
EndStr := True;      If (NCC > 1) then Dec(NCC)      Else NCC := NNN;
Goto 3;      { As End & Right }      end;
$4D,      { Right }      $4F:      { End }      Goto 2;
$52:      { Ins }      Begin      Insert := Not Insert;      If Insert then NormalCursor
Else BlockCursor;      End;
$40:      { F6 - Switch Windows }      Begin      Cur_Nodes(i, LtCyanOnBlue);
ChangeAttribute(80, W1, 1, GrayOnCyan);      W1 := i; i := i1; i1 := W1;
W1 := N; N := M; M := W1;      W1 := NC; NC := MC; MC := W1;
W1 := NCC; NCC := MCC; MCC := W1;      Nodes_On := Not Nodes_On;
If Nodes_On then      begin      N_Elem := NN;      NN := N_Nodes;
W1 := W_N1;      W2 := W_N2;      NNN := 6;      end
Else      begin      N_Nodes := NN;      NN := N_Elem;
W1 := W_E1;      W2 := W_E2;      NNN := 13;      end;
Window(1,W1+1,80,W2);      ChangeAttribute(80, W1, 1, YellowOnCyan);
Cur_Nodes(i, YellowOnRed);      Goto 3;      end;
$41:      { F7 - Insert String }      If (NN < N_Max) then      begin      Inc(NN);
If Nodes_On then      begin      For j := NN downto NC+1 do      begin
For k := 1 to 4 do      Nodes[j,k] := Nodes[j-1,k];      GU[j] := GU[j-1];      end;
For k := 1 to 4 do      Nodes[NC+1,k] := 0;      GU[NC+1] := 0;      end
Else begin      For j := NN downto NC+1 do      begin
For k := 1 to 10 do      Elem[j,k] := Elem[j-1,k];      For k := 1 to 2 do
F_E[j,k] := F_E[j-1,k];      end;      For k := 1 to 10 do      Elem[NC+1,k] := 1;
F_E[NC+1,1]:=200000;      F_E[NC+1,2]:=0.30;      end;
If (i+2+W1 < W2) then      begin
ScrollWindowDown(6, i+3+W1, 80, W2, 1);
If Nodes_On then Str1 := Str_Nodes(NC+1)      Else Str1 := Str_Elem(NC+1);
FastWriteWindow(Str1, i+3, 1, YellowOnBlue);

```

```

If (NN-NC+i+1 < W2-W1) then      begin      Str(NN:3, Str1);
FastWriteWindow(Str1, NN-NC+i+2, 1, YellowOnBlue);      end;      end;
Dr_b := $50;      Goto 1;      end; $42:  { F8 - Delete String }
If (Nodes_On and (NN > 2)) or ((Not Nodes_On) and (NN > 1)) then
begin      Dec(NN);      If Nodes_On then      begin
For j := NC to NN do      begin      For k := 1 to 4 do
Nodes[j,k] := Nodes[j+1,k];      GU[j] := GU[j+1];      end;      end
Else begin      For j := NC to NN do      begin      For k := 1 to 10 do
Elem[j,k] := Elem[j+1,k]; For k := 1 to 2 do  _E[j,k] := F_E[j+1,k]; end; end;
If Nodes_On then Str1 := Str_Nodes(NC)      Else Str1 := Str_Elem(NC);
Str1 := Copy(Str1, 4, 80-4);      If (W2 >= i+3+W1) then
ScrollWindowUp(6, i+3+W1, 80, W2, 1);
FastWriteWindow(Str1, i+2, 4, YellowOnBlue);
Cur_Nodes(i, YellowOnRed);
If (NC+W2-W1-i-3 < NN) then  If (i < W2-W1-2) then      begin
If Nodes_On then Str1 := Str_Nodes(NC+W2-W1-i-1)
Else Str1 := Str_Elem(NC+W2-W1-i-1);      Str1 := Copy(Str1, 4, 80-4);
FastWriteWindow(Str1, W2-W1, 4, YellowOnBlue); end Else      Else
begin  If Nodes_On then FastWriteWindow(Str_N, W2-W1, 1, YellowOnBlue)
Else FastWriteWindow(Str_E, W2-W1, 1, YellowOnBlue);
FastTextWindow(' ', NN-NC+i+3, 1); end; If (NC > NN) then
begin
If Nodes_On then FastWriteWindow(Str_N, i+2, 1, YellowOnBlue)
Else FastWriteWindow(Str_E, i+2, 1, YellowOnBlue);      Dr_b := $48;
Goto 1;      end      Else      begin      Goto 3;      end;      end
Else Goto 3;  $44 :  { F10 }      Begin
If Nodes_On then N_Nodes:=NN else N_Elem:=NN;
NormalCursor;  Window(1,W1,80,W2);  Exit;  end;  End;  end;
#$09 :  { Tab }  Begin2:  EndStr := False;  If (NCC < NNN) then Inc(NCC)
Else NCC := 1;3: If (NCC > 1) then begin If Nodes_On then If (NCC = 4) then
SC := Edit_ByteB(GU[NC], Nodes_Pos[3]+2, i+2, 2, YellowOnRed)
Else      SC := Edit_Real(Nodes[NC,NCC-1-(NCC div 4)],3,
-999999.999, 9999999.999,Nodes_Pos[NCC-1]+2, i+2, 11,      YellowOnRed)
Else  begin  If (NCC>1) and (NCC<=11) then
SC := Edit_Byte(Elem[NC,NCC-1], 0, 30,

```



```

Elem_Pos[NCC-1]+2, i+2, 4, YellowOnRed); If NCC=12 then
SC := Edit_Real(F_E[NC,1], 0,0, 9999999,
Elem_Pos[NCC-1]+2, i+2, 7, YellowOnRed); If NCC=13 then
SC := Edit_Real(F_E[NC,2], 2,0, 1.00, Elem_Pos[NCC-1]+2, i+2, 4,
YellowOnRed); end; Goto 0; end Else GotoXY(2, i+2); end; End;
Until False; If Nodes_On then N_Nodes := NN Else N_Elem := NN;
Window(1,W1,80,W2); end;
{-----}
begin
{-----}
FillChar(Str_N, SizeOf(Str_N), ' '); Str_N[0] := #80;
For i := 1 to 6 do Str_N[Nodes_Pos[i]]:= ' | ';
{-----}
FillChar(Str_E, SizeOf(Str_E), ' '); Str_E[0] := #80;
For i := 1 to 12 do Str_E[Elem_Pos[i]]:= ' | '; Zooming := False;
Nodes_On := True;
{-----}
i := 1;0: Window(1,2,80,24); TextBackGround(Blue); ClrScr;
Window(1,1,80,1); TextBackGround(LightGray); TextColor(Black); ClrScr;
FastCenter( 'ПРОГРАМА АВТОМАТИЧНОГО РОЗБИВАННЯ НА
КІНЦЕВІ ЕЛЕМЕНТИ',1, $70); Window(1,25,80,25);
TextBackGround(LightGray); ClrScr; FastCenter(
'F7-додавання, F8- видалення, Табл-введення даних, F6-зміна вікон, F10-
вихід',1, $70);
W_N1 := 2; W_N2 := 13; W_E1 := 14; W_E2 := 24;
TextBackGround(Cyan); Window(1,W N1,80,W N2); ClrEol;
FastCenter('Координати вузлів, граничні умови і навантаження у вузлах', 1
$3E);
Window(1,W E1,80,W E2); ClrEol; FastCenter(
'Сусідні зони, кількість стовпців і рядків, а також вузли, що входять до
зони', 1, GrayOnCyan);
3: For i:=1 to N_Nodes do begin For J:=1 to 3 do begin
If GU[i]=0 then goto 1; If GU[i] <> GDU[j] then goto 2; GU[i]:=j;
2: end; 1: end; 4: Init_Windows(1, 1); Window(1,W_N1,80,W_N2);
Edit_Dat; i1:=Menu_D(i1); If i1 = 4 then Exit_Prog; If i1 = 3 then Goto 4;

```

```

For n:=1 to N_Nodes do   If GU[n]>0 then GU[n]:=GDU[GU[n]];
If i1=2 then   begin   HiddenCursor;
If MakeWindow(W,30,10,62,12,True,True,False,$1E,$1E,$1E, ") then;
If ActivateWindow(W) then;   Str1 := "; Keys_Move := $80;
FastTextWindow('Введіть ім'я файла:', 1, 3);
SC := Edit_String(Str1, 22, 1, 8, $4E);
NormalCursor;   W := EraseTopWindow;   DisposeWindow(W);
If (Str1 = ") then Str1 := 'NONAME';   Str1:=Str1+'.DAT';   Assign(F,Str1);
Rewrite(F);   Writeln(F,N_Nodes:6,N_Elem:6);   For i:=1 to N_Nodes do
Writeln(F,Nodes[i,1]:15:3,Nodes[i,2]:15:3,GU[i]:6,
Nodes[i,3]:15:3,Nodes[i,4]:15:3);
For i:=1 to N_Elem do   begin   For j:=1 to 10 do   Write(F,Elem[i,j]:6);
Writeln(F,F_E[i,1]:12:1,F_E[i,2]:6:3);   end;   Close(F);   Goto 3;   end; end;
{-----}
      Procedure DATA;
Var      R2          :array [1..3] of Real;   IC,NQ,KR      :Integer;
begin      Assign(F,S1);      Reset (F);      Read(F,N_Nodes);
Read(F,N_Elem);   For i:=1 to N_Nodes do
Readln(F,Nodes[i,1],Nodes[i,2],GU[i],Nodes[i,3],      Nodes[i,4]);
For i:=1 to N_Elem do      begin      For j:=1 to 10 do
Read(F,Elem[i,j]);   Readln(F,F_E[i,1],F_E[i,2]);   end;   Close(F);   end;
{-----}
Procedure Serv1(var FileName:String);
Label 0;
Var i : Byte;   Dr : Char; CONST Ext = 'DAT';begin   i := 1; 0:
HiddenCursor;   Window(1,2,80,24);   TextBackGround(Blue);   ClrScr;
Window(1,1,80,1);   TextBackGround(LightGray);   TextColor(Black);
ClrScr;   FastCenter(
'ПРОГРАМА АВТОМАТИЧНОГО РОЗБИВАННЯ НА КІНЦЕВІ
ЕЛЕМЕНТИ',
1, $70);   Window(1,25,80,25);   TextBackGround(LightGray);   ClrScr;
i := Menu(i); Case i of   1 : Begin   FilesTab(WhiteOnBlue,
MagentaOnBlue, YellowOnRed, Ext, Off, FileName);   If FileName = #$FF
then Goto 0;
TextBackGround(Blue);   Window(1,2,80,24);   ClrScr; HiddenCursor;

```

```

NormalCursor; DATA;      End;
2 :Begin    N_Nodes:=2;    N_Elem:=1;    For j:=1 to 2 do
begin      GU[j]:=0;      For K:=1 to 5 do    Nodes[j,k]:=0;    end;
For k:=1 to 10 do Elem[1,k]:=1; F_E[1,1]:=200000; F_E[1,2]:=0.30; End;
3 : Begin    Instruk;      Goto 0;      End;  4 : quitter:=true; End; end;
{-----}

      Procedure SETKA;
Label 5,6,9,10,11,15,17;
begin      NEL:=0;      NELBW:=0;      NBW:=0;      NB1:=0; N1:=0;
N_Nodes1:=0;    For kk:=1 to N_Elem do    begin    For i:=1 to 4 do
begin II:=Elem[kk,I+6]; XRG[I]:=Nodes[II,1]; YRG[I]:=Nodes[II,2]; end;
TR:=Elem[kk,5]-1; DETA:=1/TR; TR:=Elem[kk,6]-1; DSI:=1/TR;
For I:=1 to Elem[kk,5] do    begin    TR:=I-1;    ETA:=TR*DETA;
For J:=1 to Elem[kk,6] do    begin    TR:=J-1;    SI:=TR*DSI;
N2[1]:=ETA-SI*ETA;    N2[2]:=SI*ETA;    N2[3]:=SI-SI*ETA;
N2[4]:=1-SI-ETA+SI*ETA;    XC1[I,J]:=0;    YC1[I,J]:=0; For k:=1 to 4 do
begin    XC1[I,J]:=XC1[I,J]+XRG[K]*N2[K];
YC1[I,J]:=YC1[I,J]+YRG[K]*N2[K];    end;    end;    end;
KN1:=1;    KS1:=1;    KN2:=Elem[kk,5];    KS2:=Elem[kk,6];
For i:=1 to 4 do    begin    NRT:=Elem[kk,I];    If (NRT=0) or (NRT>kk) then
goto 5;
For j:=1 to 4 do    If Elem[NRT,j]=kk then NRTS:=j;    K:=Elem[kk,6];
If (I=2) or (I=4) then K:=Elem[kk,5];    JL:=1;    JK:=IC[I+4*(NRTS-1)];
If JK=-1 then JL:=K;    For J:=1 to K do    begin    Case I of
1: Begin    NN[Elem[kk,5],J]:=NNRB[NRT,NRTS,JL];
KN2:=Elem[kk,5]-1;    end;    2: Begin
NN[J,Elem[kk,6]]:=NNRB[NRT,NRTS,JL];    KS2:=Elem[kk,6]-1;    end;
3: Begin    NN[1,J]:=NNRB[NRT,NRTS,JL];    KN1:=2;    end;
4: Begin    NN[J,1]:=NNRB[NRT,NRTS,JL];    KS1:=2;    end;    end;
JL:=JL+JK;    end;    5: end;
If (KN1>KN2) or (KS1>KS2) then goto 6;
For I:=KN1 to KN2 do    begin    For J:=KS1 to KS2 do    begin
NB1:=NB1+1;    NN[I,J]:=NB1;    end;    end;
For I:=1 to Elem[kk,6] do    begin    NNRB[kk,1,I]:=NN[Elem[kk,5],I];
NNRB[kk,3,I]:=NN[1,I];    end;    For I:=1 to Elem[kk,5] do    begin

```

```

NNRB[kk,2,I]:=NN[I,Elem[kk,6]];      NNRB[kk,4,I]:=NN[I,1];   end; 6;
K:=1;      For I:=1 to Elem[kk,5] do      begin      For J:=1 to Elem[kk,6] do
begin      XE[k]:=XC1[i,j];      YE[k]:=YC1[i,j];  NE1[k]:=NN[i,j];
K:=K+1;      end;      end;      L:=Elem[kk,5]-1;
For I:=1 to L do      begin      For J:=2 to Elem[kk,6] do      begin
DIAG1:=sqrt(sqr(XC1[i,j]-XC1[i+1,j-1])+      sqr(YC1[i,j]-YC1[i+1,j-1]));
DIAG2:=sqrt(sqr(XC1[i+1,j]-XC1[i,j-1])+      sqr(YC1[i+1,j]-YC1[i,j-1]));
NR[1]:=Elem[kk,6]*I+J-1;      NR[2]:=Elem[kk,6]*I+J;
NR[3]:=Elem[kk,6]*(I-1)+J;      NR[4]:=Elem[kk,6]*(I-1)+J-1;
For ij:=1 to 2 do      begin      NEL:=NEL+1;
If DIAG1/DIAG2>1.02 then goto 9;      J1:=NR[1];      J2:=NR[IJ+1];
J3:=NR[IJ+2];      goto 10;      9:J1:=NR[IJ];      J2:=NR[IJ+1];      J3:=NR[4];
10:LB[1]:=Round(abs(NE1[J1]-NE1[J2]));
LB[2]:=Round(abs(NE1[J2]-NE1[J3])); LB[3]:=Round(abs(NE1[J1]-NE1[J3]));
For IK:= 1 to 3 do      begin      If LB[iK]<=NBW then goto 11;
NBW:=LB[ik];      NELBW:=NEL;      11:;      end;
N1:=N1+1;      F_E1[n1,1]:=F_E[kk,1];      F_E1[n1,2]:=F_E[kk,2];
Elem1[n1,1]:=NE1[j1];      Elem1[n1,2]:=NE1[j2]; Elem1[n1,3]:=NE1[j3];
Elem1[n1,4]:=kk; Nodes1[NE1[j1],1]:=XE[j1]; Nodes1[NE1[j2],1]:=XE[j2];
Nodes1[NE1[j3],1]:=XE[j3]; Nodes1[NE1[j1],2]:=YE[j1];
Nodes1[NE1[j2],2]:=YE[j2]; Nodes1[NE1[j3],2]:=YE[j3];
For ii:=1 to 3 do      If N_Nodes1<Elem1[n1,ii] then N_Nodes1:=Elem1[n1,ii];
end; end; end;      end;
N_Elem1:=NEL;      For i:=1 to N_Nodes1 do      GU1[i]:=0;
For i:=1 to N_Nodes do      begin      For j:=1 to N_Nodes1 do
If (Nodes1[j,1]=Nodes[i,1]) and (Nodes1[j,2]=Nodes[i,2]) then GU1[j]:=GU[i];
end;
For J:=1 to N_Nodes1 do begin Nodes1[j,3]:=0; Nodes1[j,4]:=0; end; 15;;
For i:=1 to N_Nodes do      begin      For j:=1 to N_Nodes1 do      begin
If (Nodes1[j,1]<>Nodes[i,1]) or (Nodes1[j,2]<>Nodes[i,2]) then goto 17;
Nodes1[j,3]:=Nodes[i,3]; Nodes1[j,4]:=Nodes[i,4]; 17:end; end; end;
{-----}
Procedure INFORM;
Label 1,2,3;
begin      ClrScr;      NSZ:=2*N_Nodes;      Z1:=1000000;      Z2:=-1000000;

```

```

T1:=1000000;    T2:=-1000000;          NSZ:=2*N_Nodes1;
For i:=1 to N_Nodes1 do          begin    R1[2*(i-1)+1]:=Nodes1[i,3];
R1[2*(i-1)+2]:=Nodes1[i,4];  end;    For i:=1 to N_Nodes1 do          begin
If Nodes1[i,1]<Z1 then Z1:=Nodes1[i,1];
If Nodes1[i,1]>Z2 then Z2:=Nodes1[i,1];
If Nodes1[i,2]<T1 then T1:=Nodes1[i,2];
If Nodes1[i,2]>T2 then T2:=Nodes1[i,2];          end;    end;
{-----}
    Procedure UGOL;
begin
P[1].X:=Round(X1+X2*(Nodes1[Elem1[n,1],1]-Z1));
P[2].X:=Round(X1+X2*(Nodes1[Elem1[n,2],1]-Z1));
P[3].X:=Round(X1+X2*(Nodes1[Elem1[n,3],1]-Z1));
P[1].Y:=Round(Y1-Y2*(Nodes1[Elem1[n,1],2]-T1));
P[2].Y:=Round(Y1-Y2*(Nodes1[Elem1[n,2],2]-T1));
P[3].Y:=Round(Y1-Y2*(Nodes1[Elem1[n,3],2]-T1));  FillPoly(3,P);  end;
{-----}
    Procedure SILA;
Var ij  :Char;  M1,P1,A1,A2,DL,PX,PY,Ld  :Real;  S          :String;
Label   1,2,3,4,5,6,7,8,9,10,11,12;
begin  gd:=detect;  initgraph(gd,gm,"");  Xmg:=GetMaxX;
Ymg:=GetMaxY;  SetColor(Yellow);
If (Z2-Z1)>(T2-T1) then G1:=Z2-Z1 else G1:=T2-T1;
X2:=0.8*Xmg/G1;  Y2:=0.7*Ymg/G1;  X1:=0.1*Xmg;  Y1:=0.8*Ymg;
Rectangle(0,0,Xmg,Ymg);  For i:=1 to 35 do
begin  If (i-3)*G1/30<=Z2-Z1 then
Line(Round(X1+(i-1)*X2*G1/30),Round(Y1),Round(X1+i*X2*G1/30-7),
Round(Y1));  If (i-3)*G1/30<=T2-T1 then
Line(Round(X1),Round(Y1-(i-1)*Y2*G1/30),Round(X1),
Round(Y1-i*Y2*G1/30+2));  end;
OutTextXY(Round(X1+0.85*Xmg*(Z2-Z1)/G1),Round(0.765*Ymg),'R');
OutTextXY(Round(X1-Xmg/50),Round(Y1-0.75*Ymg*(T2-T1)/G1),'Y');
SetFillStyle(EmptyFill,White);
For n:=1 to N_Elem1 do  begin  UGOL;
Xc:=Round((P[1].X+P[2].X+P[3].X)/3-Xmg/100);

```

```

Yc:=Round((P[1].Y+P[2].Y+P[3].Y)/3-Ymg/50);
Circle(Round(Xc+Xmg/140),Round(Yc+Ymg/70),12);
RealToString(S,N,10,10,0);    OutTextXY(Xc,Yc,S);    end;
Line(Round(X1-Xmg/12),Round(Y1-Y2*(T2-T1)),Round(X1-
Xmg/12),Round(Y1));
Line(Round(X1),Round(0.9*Ymg),Round(X1+X2*(Z2-Z1)),Round(0.9*Ymg));
For i:=1 to 6 do    begin    Line(Round(X1-Xmg/12),Round(Y1-Y2*(T2-
T1)*(i-1)/5),Round(X1-Xmg/14),
Round(Y1-Y2*(T2-T1)*(i-1)/5));
Line(Round(X1+X2*(Z2-Z1)*(i-1)/5),Round(0.9*Ymg),
Round(X1+X2*(Z2-Z1)*(i-1)/5),Round(0.88*Ymg));
RealToString(S,Z1+(i-1)*(Z2-Z1)/5,10,10,1);
OutTextXY(Round(X1+X2*(Z2-Z1)*(i-1)/5),Round(0.85*Ymg),S);
RealToString(S,T1+(i-1)*(T2-T1)/5,10,10,1);
OutTextXY(Round(X1-Xmg/15),Round(Y1-Y2*(T2-T1)*(i-1)/5),S);    end;
OutTextXY(Round(0.25*Xmg),Round(0.94*Ymg),
'Reзультати розрахунків дивися в файлі CIL.DAT'); SetColor(Red);
For n:=1 to N_Nodes1 do    begin
XG[1]:=Round(X1+X2*(Nodes1[n,1]-Z1));
YG[1]:=Round(Y1-Y2*(Nodes1[n,2]-T1));    If R1[2*(n-1)+1]=0 then goto 1;
YG[2]:=YG[1];
XG[2]:=XG[1]-Round(abs(R1[2*(n-1)+1])*Xmg/(10*R1[2*(n-1)+1]));
XG1[1]:=XG[1]-Round(abs(R1[2*(n-1)+1])*Xmg/(40*R1[2*(n-1)+1]));
YG1[1]:=YG[1]-Round(Xmg/200);    YG1[2]:=YG[1]+Round(Xmg/200);
Line(XG[1],YG[1],XG[2],YG[2]);    Line(XG[1],YG[1],XG1[1],YG1[1]);
Line(XG[1],YG[1],XG1[1],YG1[2]);    Line(XG1[1],YG1[1],XG1[1],YG1[2]);
RealToString(S,abs(R1[2*n-1]),10,10,1);
OuTTextXY(XG[1]-Round(Xmg/25*abs(R1[2*n-1])/R1[2*n-1]),YG[1]-
Round(Ymg/15),'P='+S);    1:If R1[2*(n-1)+2]=0 then goto 2; XG[2]:=XG[1];
YG[2]:=YG[1]+Round(abs(R1[2*(n-1)+2])*Xmg/(30*R1[2*(n-1)+2]));
YG1[1]:=YG[1]+Round(abs(R1[2*(n-1)+2])*Xmg/(150*R1[2*(n-1)+2]));
XG1[1]:=XG[1]-Round(Xmg/200);    XG1[2]:=XG[1]+Round(Xmg/200);
Line(XG[1],YG[1],XG[2],YG[2]);    Line(XG[1],YG[1],XG1[1],YG1[1]);
Line(XG[1],YG[1],XG1[2],YG1[1]);    Line(XG1[1],YG1[1],XG1[2],YG1[1]);
RealToString(S,abs(R1[2*n]),10,10,1);

```

```

OuTTextXY(XG[1]-Round(Xmg/25),YG[1]+
Round(Ymg/15*abs(R1[2*n])/R1[2*n]),'P='+S);
2: RealToString(S,N,10,10,0); OutTextXY(XG[1]-Round(Xmg/35),YG[1]-
Round(Ymg/25),S);      end; For i:=1 to N_Nodes1 do          begin
For j:=1 to 2 do      R1[2*(i-1)+j]:=6.2832*R1[2*(i-1)+j]*abs(Nodes1[i,1]);
If GU1[i]=0 then goto 10;          K:=0;
For N:=1 to N_Elem1 do          begin
If i=Elem1[n,1] then K:=Elem1[n,1];      If i=Elem1[n,2] then K:=Elem1[n,2];
If i=Elem1[n,3] then K:=Elem1[n,3];          If K=0 then goto 12;
11: A1:=(Nodes1[Elem1[n,2],1]+Nodes1[Elem1[n,1],1]+
Nodes1[Elem1[n,3],1])/3;  A2:=(Nodes1[Elem1[n,2],2]+Nodes1[Elem1[n,1],2]+
Nodes1[Elem1[n,3],2])/3;      If A1-Nodes1[i,1]>0 then K1:=1 else K1:=-1;
If A2-Nodes1[i,2]>0 then K2:=1 else K2:=-1;
XG[1]:=Round(X1+X2*(Nodes1[i,1]-Z1));
YG[1]:=Round(Y1-Y2*(Nodes1[i,2]-T1));
8:If(GU1[i]<>10) and (GU1[i]<>11) then goto 9;
Circle(XG[1]-K1*Round(Xmg/200),YG[1],          Round(Xmg/200));
Circle(XG[1]-K1*Round(Xmg/50),YG[1],          Round(Xmg/200));
Line(XG[1]-K1*Round(Xmg/200),YG[1],
XG[1]-K1*Round((Xmg/50-Xmg/200)),YG[1]);
Line(XG[1]-K1*Round(Xmg/40),YG[1]-Round(Xmg/100),
XG[1]-K1*Round(Xmg/40),YG[1]+Round(Xmg/100));
9: If (GU1[i]<>11) and (GU1[i]<>1) then goto 10;
If GU1[i]=1 then          Circle(XG[1],YG[1]+Round(K2*Xmg/200),
Round(Xmg/200));          Circle(XG[1],YG[1]+Round(K2*Xmg/80),
Round(Xmg/200));          Line(XG[1],YG[1]+Round(K2*Xmg/200),
XG[1],YG[1]+Round(K2*(Xmg/80-Xmg/200)));
Line(XG[1]-Round(Xmg/75),YG[1]+Round(K2*Xmg/70),
XG[1]+Round(Xmg/75),YG[1]+Round(K2*Xmg/70));  goto 10;  12: end;
10:;          end;{          Repeat          Until KeyPressed;}
ij:=Readkey;  end; {-----}
Begin InitGraf; RestoreCRTMode; i1:=1; quitter:=false;3 : ClrScr;
SERV1(s1); if quitter then Exit_Prog;
SERVIS; SETKA; INFORM; SILA;
CloseGraph;  Str1:='Cil.dat';  Assign(F,Str1);  Rewrite(F);

```

```
Writeln(F,N_Nodes1:5,NEL:5);      For i:=1 to N_Nodes1 do
Writeln(F,Nodes1[i,1]:15:3,Nodes1[i,2]:15:3,GU1[i]:6,
Nodes1[i,3]:15:3,Nodes1[i,4]:15:3);
For i:=1 to NEL do      Writeln(F,Elem1[i,1]:5,Elem1[i,2]:5,Elem1[i,3]:5,
Elem1[i,4]:5,F_E1[i,1]:12:0,F_E1[i,2]:9:4);
Writeln(F,' Максимальна різниця між номерами елемента =',NBW:5);
Close(F);  I:=0;  Writeln(' Результати розрахунків див. у файлі
CIL.DAT'); end.
```


ДОДАТОК Д

Текст програми розрахунку тонкостінних просторових систем

```
{ $N+ } Program WC; { Wall`s Compiler } Uses Crt,Dos
TpVarray { Реалізація віртуальних масивів }
Variable { Основні визначення змінних і масивів } NewData { Читання
даних з файлів } Matrix { Формування матриці жорсткості конструкції }
Deform; { Визначення напружень }
Var i,j : Integer; ch : Char; NumPage : Word; Dir : DirStr; Name :
NameStr; Ext : ExtStr; Year, Month, Day, MayDay, Hour, Minute, Sec,
Sec100 : Word;
Procedure ViewHelp;
begin writeln; writeln(' Use: WC[.EXE]<filename>');
writeln(' <filename> - ім'я файла даних завдання'); writeln;end; Procedure
Title;
begin NormVideo; TextColor(11); TextColor(15);
writeln(' << Wall`s Compiler >> alfa Version 1.01.'); TextColor(14);
TextColor(15);
writeln(' Copyright (c) 1995, CAD, Rudnikov Roman & Spodinec Denis. ');
if ParamCount = 0 then begin ViewHelp; Halt(1); end;end;
Begin Title; Km1:= 0; Km2:= 0; { Читання даних з файлів }
writeln; write(' Читання даних з файлів...'); DATFileName:= ParamStr(1);
{ $I- }
Assign(DATFile, DATFileName); Reset(DATFile); Close(DATFile); { $I+ }
if IOResult <> 0 then begin writeln('Error!'); HighVideo; writeln;
writeln(' Помилка: Файл даних не знайдений !'); NormVideo; Halt(1);
end;
GDATA;
writeln('Ok. '); { Кількість рівнянь у системі } NSZF:= NP * NDF;
{ Формування матриці жорсткості конструкції }
writeln; writeln(' Вузлів: ', NP ' Елементів: ', NE);
write(' Формування матриці жорсткості конструкції...');
Matrica;
{ Рішення системи рівнянь }
```

```

writeln; writeln(' Кількість рівнянь у системі: ', NSZF);
write(' Рішення системи рівнянь...'); Sistem;      writeln;
write(' Формування файлу переміщень...');
FSplit(DATFileName, Dir, Name, Ext); OUTFileName:= Dir + Name +
'.WRF'; Assign(OUTFile, OUTFileName); Rewrite(OUTFile);
etDate(Year, Month, Day, MayDay); GetTime(Hour, Minute, Sec, Sec100);
writeln(OUTFile ' << Wall`s Compiler >>  Date: ',Day, '/',Month, '/',Year
      ' Time: ',Hour, ':',Minute, ':',Sec, ':',Sec100); writeln(OUTFile);
writeln(OUTFile ' ЗАВДАННЯ: '+ Name); writeln(OUTFile);
writeln(OUTFile ' 1. Переміщення вузлів конструкції...'); writeln(OUTFile);
writeln(OUTFile, 'Вузол   X       Y       Z       x       y       z');
for i:= 1 to NP do begin  write(OutFile,[' ',i:3, ']);
for j:= 1 to 6 do write(OutFile, P[6*(i-1)+j]:11:5);  writeln(OutFile); end;
writeln('Ok. '); { Визначення напружень }
writeln; write(' Визначення напружень...'); writeln(OUTFile);
writeln(OUTFile ' 2. Напруження в елементах конструкції...');
writeln(OUTFile);
writeln(OUTFile, 'Еле-нт'); Deformation; Close(OUTFile);End.
{$S-,R-,V-,I-,B-,F-} {$IFDEF Ver40} {$F-} {$ELSE} {$F+} {$I
OPLUS.INC} {$ENDIF}
{*****}
{*          TPVARRAY.PAS 5.05          *}
{*    Copyright (c) TurboPower Software 1987.    *}
{* Portions copyright (c) Sunny Hill Software 1985, 1986 *}
{* and used under license to TurboPower Software  *}
{*          All rights reserved.          *}
{*****}
unit TpVarray;
{-Dynamically allocated arrays with virtual memory support}
Interface uses Dos, TpMemChk; type TpArray = Pointer;      {Generic type
of TpArray} nitType =      {Initialization options for new arrays}
(ExactInit, FastInit); const {Settings for error handling}
WriteError : Boolean = True; {Display an error message}
HaltOnError : Boolean = True; {Halt when any error encountered}

```

```

RangeCheck : Boolean = False; {Check array indices - up to 40% performance
penalty}
KeepDiskCurrent : Boolean = False; {If true, force disk update after every
operation}
function ERRORA : Word; {-Return the last error number and reset the error
variable}
procedure RETA(A : TpArray; R, C : Word; var Value); {-Return value of
A[r,c]}
procedure SETA(A : TpArray; R, C : Word; var Value);
{-Set array element A[r,c] to value}
procedure CLEARA(A : TpArray; var Value; Initialize : InitType);
{-Initialize all array values to Value} procedure MakeAPrimitive(var A :
TpArray;
RowsInPage : Word;      ColsInPage : Word;          PagesDown : Word;
PagesAcross : Word;      ElementSize : Word;        FileName : string;
RamBytesForData : LongInt);
{-Allocate array memory, with full control over page structure}
procedure MAKEA(var A : TpArray;      Rows : Word;    Cols : Word;
ElementSize : Word;  FileName : string;      RamBytesForData : LongInt);
{-Allocate array the easy way} procedure FLUSHA(A : TpArray);
{-Update the disk image of the array} procedure STOREA(A : TpArray);
{-Write array to disk and close file}
procedure DISPOSEA(var A : TpArray; DeleteFile : Boolean);
{-Deallocate array in memory, and delete array file if requested}
procedure LOADA(var A : TpArray; FileName : string; RamBytesForData :
LongInt);
{-Allocate and read array from disk}
{=====}
Implementation const
MaxHeapBlock = 65519;    {Largest single block on heap, less one}
NotInRAM = $FFFF;      {Flag that a page is not currently in RAM}
RPPOfs = 0;           {Offsets within header record of each field}
CPPOfs = 2;          { used within inline macro below}

```

```

PDOfs = 4; PAOfs = 6; ESOfs = 8; IdString : string[6]= 'VARRAY'; {Written
at the start of array files} type {Data structures for management of large
arrays}
{Header for description of an array}
{DO NOT CHANGE WITHOUT UPDATING RECORD OFFSETS ABOVE}
ArrayHeader = record   RowsInPage : Word;   ColsInPage : Word;
PagesDown : Word;   PagesAcross : Word;
ElSize : Word;      {Bytes in one array element}
PageSize : Word;    {Bytes of data in one page}
PageCount : Word;   {Number of pages in entire array}
NextAvail : LongInt; {Next file position for new page}
ID : string[6];     {Identifies array files} end;
{Data storage}   DataPage = array[0..MaxHeapBlock] of Byte;
{Descriptor of a page kept in RAM}
PageRec = record   PageNum : Word;      {Index into disk pointer table}
LRU : Word;        {Quasi-time when page last used}
Used : Boolean;    {True when page is loaded with data}
Dirty : Boolean;   {True when RAM page differs from disk page}
Data : DataPage;  {Actual data for page} end; PageRecPtr = ^PageRec;
RamPageType = 0..16000; DiskPageType = 0..16000; {Pointers to pages in
RAM}
RamPageIndex = array[RamPageType] of PageRecPtr;
{Map between disk page and RAM page, holds NotInRAM if not in RAM}
RamPageMap = array[DiskPageType] of Word;
{Offset within disk file of each page}
DiskPageMap = array[DiskPageType] of LongInt; TpArrayRec = record
RamPages : ^RamPageIndex; {Points to pages of data in RAM}
RamIndex : ^RamPageMap; {Index of pages currently in RAM}
DiskIndex : ^DiskPageMap; {Index of pages on disk}
PagesInRam : Word;   {Number of pages in RAM}
Header : ArrayHeader; {Descriptor of array configuration}
F : file;            {File variable}   Opened : Boolean;   {True when file open}
LRUcount : Word; {Current quasi-time} end; TpArrayInternal =
^TpArrayRec;
var ArrayErrorLast : Word;   {Holds the last error number encountered}

```

```

Pnum : Word;          {Global results of address calculations}
Pofs : Word; procedure Flush(var F : file);  {-Flush DOS file buffers}
var  handle : Word absolute F;  regs : registers;
begin  with regs do begin  ah := $45;  bx := handle;  msdos(regs);
if Odd(flags) then  Exit; bx := ax;  ah := $3E;  msdos(regs);  end; end;
procedure Error(ErrNum : Word);
{-Store, report and/or halt on errors}
begin  ArrayErrorLast := ErrNum;  if WriteError then  case ErrNum of
1 : WriteLn('Row index out of range'); 2 : WriteLn('Column index out of range');
3 : WriteLn('Invalid element size'); 4 : WriteLn('Invalid row or column limit');
5 : WriteLn('Invalid page size'); 6 : WriteLn('Invalid page index size');
7 : WriteLn('Insufficient memory to allocate array'); 8 : WriteLn('File creation
error');
9 : WriteLn('File write error'); 10 : WriteLn('Error while closing file');
11 : WriteLn('File not found'); 12 : WriteLn('File read error');
13 : WriteLn('Number of in-RAM pages less than 1');
17 : WriteLn('Invalid file format');  end; if HaltOnError then  Halt(1);  end;
procedure DisposePrimitive(var A : TpArray; Level, Pages : Word);
{-Deallocate RAM space for selected components of array}
var  AP : TpArrayInternal absolute A;  P : Word;
begin  with AP^, Header do begin  if Level >= 5 then  for P := 0 to
Pred(Pages) do
FreeMemCheck(RamPages^[P], PageSize+(SizeOf(PageRec) -
SizeOf(DataPage))); if Level >= 4 then  FreeMemCheck(RamPages,
SizeOf(Pointer)*PagesInRam);
if Level >= 3 then  FreeMemCheck(DiskIndex,
SizeOf(LongInt)*PageCount);
if Level >= 2 then  FreeMemCheck(RamIndex, SizeOf(Word)*PageCount);
if Level >= 1 then  FreeMemCheck(A, SizeOf(TpArrayRec));  end; end;
function DivUp(X, Y : LongInt): Word;  {-Compute X div Y and round up}
var  result : Word;
begin result := X div Y; if X mod Y <> 0 then Inc(result); DivUp := result; end;
function ERRORA : Word;
{-Return the last error number and reset the error variable}
begin  ERRORA := ArrayErrorLast;  ArrayErrorLast := 0;  end;

```

```

function RangeError(var H : ArrayHeader; R, C : Word): Boolean;
{-Check for addressing range errors}
begin  RangeError := True; with H do begin  if R >= RowsInPage*PagesDown
then begin  Error(1);  Exit;  end;
if C >= ColsInPage*PagesAcross then begin  Error(2);  Exit;  end;  end;
RangeError := False; end;
function WriteIndex(A : TpArray): Boolean;
{-Write the current header and page index to disk}
var  AP : TpArrayInternal absolute A;  Bytes : Word;  DiskIndexSize :
Word; begin  WriteIndex := False; with AP^ do begin  if not Opened then  Exit;
Reset(F, 1);  BlockWrite(F, Header, SizeOf(ArrayHeader), Bytes);
if (IoResult <> 0) or (Bytes <> SizeOf(ArrayHeader)) then begin  Error(9);
Exit;  end; DiskIndexSize := SizeOf(LongInt)*Header.PageCount;
BlockWrite(F, DiskIndex^, DiskIndexSize, Bytes);
if (IoResult <> 0) or (Bytes<>DiskIndexSize) then begin  Error(9);Exit; end;end;
WriteIndex := True; end;
procedure InitRamIndex(A : TpArray);
{-Initialize the RAM page index} var  AP : TpArrayInternal absolute A;  P :
Word;
begin  with AP^, Header do begin  {Mark all pages as not in RAM}
FillChar(RamIndex^, SizeOf(Word)*PageCount, $FF);
{Initialize each in-RAM page descriptor}
for P := 0 to Pred(PagesInRam) do with RamPages^[P]^ do begin  PageNum:= 0;
LRU := 0;  Used := False;  Dirty := False;  end;  end; end;
function WritePage(A : TpArray; PIR, Pnum : Word): Boolean;
{-Write from RAM page PIR to disk page Pnum}
var  AP : TpArrayInternal absolute A;  Bytes : Word;
begin  WritePage := False;  with AP^, Header do begin
{Allocate disk space for page if needed}
if DiskIndex^[Pnum]= 00 then begin  DiskIndex^[Pnum]:= NextAvail;
Inc(NextAvail, PageSize); {Update file image of index immediately if desired}
if KeepDiskCurrent then  if not WriteIndex(A) then  Exit;  end; {Seek and
write}
Seek(F, DiskIndex^[Pnum]); BlockWrite(F, RamPages^[PIR]^Data, PageSize,
Bytes); if (IoResult <> 0) or (Bytes <> PageSize) then begin  Error(9); Exit; end;

```

```

{Page is no longer dirty}
RamPages^[PIR]^Dirty := False; {Flush DOS file buffers if requested}
if KeepDiskCurrent then Flush(F); end; WritePage := True; end;
function GetPage(A : TpArray; Pnum : Word): Word;
{-Read disk page into RAM and return page number there}
Var AP : TpArrayInternal absolute A; P : Word; PIR : Word; MINLRU : Word;
Bytes : Word; PageUsed : Boolean; begin with AP^, Header do begin
{Find the next unused or oldest page in RAM}
MINLRU := $FFFF; P := 0; PIR := 0; PageUsed := True;
while P < PagesInRam do begin with RamPages^[P]^ do if not(Used) then
begin
PIR := P; PageUsed := False; {Force exit from loop} P := PagesInRam;
end else if LRU < MINLRU then begin PIR := P;
MINLRU := LRU; end; Inc(P); end;
with RamPages^[PIR]^ do begin if PageUsed then begin
{A previous page was in this location}
if Dirty then begin if not WritePage(A, PIR, PageNum) then Exit; end;
{Indicate that the page is no longer in RAM}
RamIndex^[PageNum]:= NotInRAM; end; {Read in the new page}
if DiskIndex^[Pnum]<> 00 then begin {There is already data stored on disk}
Seek(F, DiskIndex^[Pnum]); BlockRead(F, Data, PageSize, Bytes);
if (IoResult <> 0) or (Bytes <> PageSize) then begin Error(12); Exit; end; end;
{Indicate that the new page is in RAM}
RamIndex^[Pnum]:= PIR; {Update page flags} Used := True;
Dirty := False; PageNum := Pnum; end; {With RamPages^[PIR]^}
end; {With AP^, Header} {Return the page number where the data was
loaded}
GetPage := PIR; end;
procedure ArrayAddress(H : ArrayHeader; R, C : Word);
{-Return the page parameters for a given row and column}
nline ($5B/ {pop bx ;save column in bx}
$58/ {pop ax ;save row in ax}
$5F/ {pop di ;offset of array header}
$07/ {pop es ;segment of array header}
$31/$D2/ {xor dx,dx ;prepare to divide} $26/ {es:}

```

```

$F7/$75/<RPPOfs/      { div word ptr [di+<RPPOfs] ;divide by RowsInPage
(POR in DX, PR in AX)}  $89/$D1/      { mov cx,dx      ;save POR in cx}
$89/$C6/              { mov si,ax      ;save PR in si}
$89/$D8/              { mov ax,bx      ;col into ax}
$31/$D2/              { xor dx,dx      ;prepare to divide}  $26/
{es;}
$F7/$75/<CPPOfs/      { div word ptr [di+<CPPOfs] ;divide by ColsInPage
(POC in DX, PC in AX)}  $89/$C3/      { mov bx,ax      ;save PC in BX}
$26/  {es;}  $8B/$45/<PDOfs/ { mov ax[di+<PDOfs];get PagesDown in AX}
$52/                  { push dx          ;save POC a moment}
$F7/$E3/              { mul bx          ;PagesDown*PC}
$01/$F0/              { add ax,si       ;Pnum = PR+PagesDown*PC}
$A3/>Pnum/            { mov [>Pnum],ax   ;Store Pnum}
$58/                  { pop ax           ;get POC back}
$26/  {es;}  $F7/$65/<RPPOfs/ { mul word ptr [di+<RPPOfs]
;RowsPerPage*POC}
$01/$C8/              { add ax,cx      ;POR+RowsPerPage*POC}  $26/  {es;}
$F7/$65/<ESOfs/      { mul word ptr [di+<ESOfs]
;ElSize*(POR+RowsPerPage*POC)}
$A3/>Pofs);           { mov [>Pofs],ax   ;Store Pofs}
function ChangesWereWritten(A : TpArray): Boolean;
{-Write any changed pages and return true if any were written}
var  AP : TpArrayInternal absolute A;  PIR : Word;
begin  ChangesWereWritten := False;  with AP^ do
for PIR := 0 to Pred(PagesInRam) do  with RamPages^[PIR]^ do
if Dirty then begin  if not WritePage(A, PIR, PageNum) then  Exit;
ChangesWereWritten := True;  end; end;
procedure FLUSHA(A : TpArray); {-Update the disk image of the array}
var  AP : TpArrayInternal absolute A;  Changed : Boolean;
begin  with AP^ do begin  if not Opened then  Exit; {Write any changed
pages}
Changed := ChangesWereWritten(A);  if not(KeepDiskCurrent) then
{Update the index if WritePage didn't already do so}
if not WriteIndex(A) then  Exit;  if Changed or not(KeepDiskCurrent) then
{Assure DOS buffers are flushed}  Flush(F);  end; end;

```



```

procedure RESETLRU(A : TpArray); {-Reset the LRU values for all pages in
RAM}
var AP : TpArrayInternal absolute A; P : Word;
begin with AP^ do for P := 0 to Pred(PagesInRam) do RamPages^[P]^LRU :=
0; end;
procedure RETA(A : TpArray; R, C : Word; var Value);
{-Return the value of A[R,C]}
var AP : TpArrayInternal absolute A; PIR : Word; P : Word;
begin with AP^, Header do begin if RangeCheck then
if RangeError(Header, R, C) then Exit;
{Compute page number and offset into page - returned in globals}
ArrayAddress(Header, R, C); {Is page in RAM? If not, put it there now}
PIR := RamIndex^[Pnum]; if PIR = NotInRAM then PIR := GetPage(A, Pnum);
{Update quasi-time} Inc(LRUcount); if LRUcount = 0 then
RESETLRU(A);
{Update LRU for this page, and pass back data value}
with RamPages^[PIR]^ do begin LRU := LRUcount; Move(Data[Pofs], Value,
ElSize);
inline($FC); end; end; end;
procedure SETA(A : TpArray; R, C : Word; var Value);
{-Set array element A[R,C] to value}
var AP : TpArrayInternal absolute A; PIR : Word; P : Word;
begin with AP^, Header do begin if RangeCheck then
if RangeError(Header, R, C) then Exit;
{Compute page number and offset into page - returned in globals}
ArrayAddress(Header, R, C); {Is page in RAM? If not, put it there now}
PIR := RamIndex^[Pnum]; if PIR = NotInRAM then PIR := GetPage(A, Pnum);
{Update quasi-time} Inc(LRUcount); if LRUcount = 0 then
RESETLRU(A);
{Update LRU for this page, and pass back data value}
with RamPages^[PIR]^ do begin Dirty := True; LRU := LRUcount;
Move(Value, Data[Pofs], ElSize); inline($FC); end; end; end;
procedure CLEARA(A : TpArray;
var Value; Initialize : InitType); {-Reset all array values to Value}
var AP : TpArrayInternal absolute A; B : Byte absolute Value; R : Word;

```

```

C : Word;  P : Word;  Save : Boolean;
begin  {Don't waste time on updating disk index until done}
Save := KeepDiskCurrent;  KeepDiskCurrent := False;  with AP^, Header do
if Initialize = FastInit then begin    {Initialize in blocks by page}
FillChar(RamPages^[0]^Data, PageSize, B);  {Write all the pages to disk}
for P := 0 to Header.PageCount-1 do    if not WritePage(A, 0, P) then  Exit;
{Mark in-RAM pages as available}    InitRamIndex(A);  end else
{Initialize one element at a time}  for R := 0 to Pred(RowsInPage*PagesDown)
do
for C := 0 to Pred(ColsInPage*PagesAcross) do    SETA(A, R, C, Value);
{Update index and buffers if requested}  KeepDiskCurrent := Save;
FLUSHA(A); end;
function ALLOCATEA(var A : TpArray;
RowsInPage : Word;    ColsInPage : Word;    PagesDown : Word;
PagesAcross : Word; ElementSize : Word; RamBytesForData : LongInt) :
Boolean;
{-Validate and allocate heap space for array}
var  AP : TpArrayInternal absolute A;  RamIndexSize : LongInt;
DiskIndexSize : LongInt;  PageCount : LongInt;  PageSize : LongInt;
PagesInRamL : LongInt; PageOverHead : LongInt; P : Word; {$IFDEF debug}
HeapUsage : LongInt;  {$ENDIF} begin  ALLOCATEA := False;
{Assure element size is valid} if ElementSize = 0 then begin  Error(3);
Exit;  end;
{Assure row and column dimensions are valid}
if (PagesDown or RowsInPage = 0) or (PagesAcross or ColsInPage = 0) then
begin Error(4);  Exit;  end;
if LongInt(PagesDown)*RowsInPage > 65535 then begin  Error(4); Exit;  end;
if LongInt(PagesAcross)*ColsInPage > 65535 then begin  Error(4); Exit; end;
{Assure page size is valid} PageOverHead := SizeOf(PageRec) -
SizeOf(DataPage);
PageSize := LongInt(RowsInPage)*ColsInPage*ElementSize;
if PageSize+PageOverHead > MaxHeapBlock then begin  Error(5);  Exit;
end;
{Assure page index sizes are valid}

```

```

PageCount := PagesDown*PagesAcross; RamIndexSize :=
SizeOf(Word)*PageCount;
    DiskIndexSize := SizeOf(LongInt)*PageCount;
if DiskIndexSize > MaxHeapBlock then begin    Error(6);    Exit; end;
{Compute number of pages to be kept in RAM}
PagesInRamL := RamBytesForData div
SizeOf(Pointer)+PageSize+PageOverHead);
{Don't allocate more pages in RAM than the whole array needs}
if PagesInRamL > PageCount then    PagesInRamL := PageCount;
if PagesInRamL < 1 then begin    Error(13);    Exit; end;
{Allocate the base array pointer}
if not GetMemCheck(AP, SizeOf(TpArrayRec)) then begin Error(7); Exit; end;
with AP^ do begin {Initialize simple fields}    PagesInRam := PagesInRamL;
LRUcount := 0;    {Initialize header}    Header.RowsInPage := RowsInPage;
Header.ColsInPage := ColsInPage;    Header.PagesDown := PagesDown;
Header.PagesAcross := PagesAcross;    Header.ElSize := ElementSize;
Header.PageSize := PageSize; Header.PageCount := PageCount;
Header.ID := IdString; {$IFDEF debug}
WriteLn('RowsInPage=', RowsInPage);    WriteLn('ColsInPage=', ColsInPage);
WriteLn('PagesDown=', PagesDown); WriteLn('PagesAcross=', PagesAcross);
WriteLn('PageSize=', PageSize);    WriteLn('PageCount=', PageCount);
WriteLn('PagesInRam=', PagesInRam);    HeapUsage :=
LongInt(SizeOf(TpArrayRec))+RamIndexSize+DiskIndexSize+
PagesInRam*(SizeOf(Pointer)+PageSize+PageOverHead);
    WriteLn('RAM efficiency=', (LongInt(100)*PagesInRam*PageSize) div
HeapUsage, '%');    {$ENDIF} {Allocate ram index}
if not GetMemCheck(RamIndex, RamIndexSize) then begin
DisposePrimitive(A, 1, 0); Error(7);    Exit; end; {Allocate disk index}
if not GetMemCheck(DiskIndex, DiskIndexSize) then begin
DisposePrimitive(A, 2, 0); Error(7);    Exit; end; {Allocate in-RAM data
pointers}
if not GetMemCheck(RamPages, SizeOf(Pointer)*PagesInRam) then begin
DisposePrimitive(A, 3, 0);    Error(7);    Exit; end;
{Allocate in-RAM data pages} for P := 0 to Pred(PagesInRam) do
if not GetMemCheck(RamPages^[P], PageSize+PageOverHead) then begin

```

```

DisposePrimitive(A, 5, P); Error(7); Exit; end; end; ALLOCATEA := True;
end;
procedure MakeAPrimitive(var A : TpArray;
RowsInPage : Word;   ColsInPage : Word;   PagesDown : Word;
PagesAcross : Word;   ElementSize : Word;   FileName : string;
    RamBytesForData : LongInt); {-Allocate array memory}
var  AP : TpArrayInternal absolute A;
begin  {Validate and allocate heap space for array}
if not ALLOCATEA(A, RowsInPage, ColsInPage, PagesDown, PagesAcross,
ElementSize, RamBytesForData) then Exit; with AP^ do begin  Opened :=
False;
{Create the disk file}   Assign(F, FileName);   Rewrite(F, 1);
if IoResult <> 0 then begin   DisposePrimitive(A, 5, PagesInRam);
Error(8);   Exit;   end;   Opened := True;
{Initialize the disk index} with Header do begin
NextAvail := LongInt(SizeOf(ArrayHeader))+SizeOf(LongInt)*PageCount;
FillChar(DiskIndex^, SizeOf(LongInt)*PageCount, 0);   end;
{Write header and index to disk} if not WriteIndex(A) then   Exit;
{Initialize the in-RAM index and page descriptors}
InitRamIndex(A);   end; end;
procedure MAKEA(var A : TpArray;
Rows : Word;   Cols : Word;   ElementSize : Word;   FileName : string;
RamBytesForData : LongInt);
{-Determine page structure and allocate array}
var  H : ArrayHeader; otalBytes : LongInt; Pages : Word; PageOverHead :
Word; Done : Boolean;   MaxPageSize : Word;
begin  {Be sure at least one element on each axis}
if (Rows = 0) or (Cols = 0) then begin   Error(4);   Exit;   end;
Be sure element size is valid} if ElementSize = 0 then begin   Error(3);
Exit;   end;
{Compute total bytes in array} TotalBytes :=
LongInt(Rows)*Cols*ElementSize;
PageOverHead := SizeOf(PageRec) -SizeOf(DataPage);
{Assure MemAvail is at least as big as RamBytesForData}
if RamBytesForData > (9*MemAvail) div 10 then

```

```

RamBytesForData := (9*MemAvail) div 10;
{Determine largest page size we'll accept}
MaxPageSize := MaxHeapBlock; if MaxPageSize > RamBytesForData shr 1
then
MaxPageSize := RamBytesForData shr 1; with H do begin {Store element size}
ElSize := ElementSize; {Determine page organization}
if TotalBytes+PageOverHead <= MaxPageSize then begin
{Whole array fits into one page} RowsInPage := Rows; ColsInPage := Cols;
PagesDown := 1; PagesAcross := 1;
end else begin {Array larger than a page} {Get first estimate of number of
pages required} Pages := DivUp(TotalBytes, MaxPageSize);
{Iterate to find page organization with minimum waste that fits} repeat
RowsInPage := DivUp(Rows, Pages); ColsInPage := DivUp(Cols, Pages);
{Determine whether to divide horizontally or vertically}
if ((LongInt(RowsInPage)*Pages-Rows)*Cols <=
(LongInt(ColsInPage)*Pages-Cols)*Rows)
or (LongInt(RowsInPage)*ElSize > MaxPageSize) then begin {Divide up the
rows}
ColsInPage := Cols; PagesAcross := 1; PagesDown := Pages; end else
begin {Divide up the columns} RowsInPage := Rows; PagesDown := 1;
PagesAcross := Pages; end;
Done := (LongInt(RowsInPage)*ColsInPage*ElSize+PageOverHead <=
MaxPageSize); if not(Done) then {Add another page} Inc(Pages);
until Done; end; {$IFDEF debug}
WriteLn('Total bytes=', TotalBytes); WriteLn('Page efficiency='
(LongInt(100)*TotalBytes) div
(LongInt(Pages)*RowsInPage*ColsInPage*ElSize), '%'); {$ENDIF}
{Allocate the array} MakeAPrimitive(A, RowsInPage, ColsInPage,
PagesDown, PagesAcross, ElSize, FileName, RamBytesForData); end;
end;

procedure STOREA(A : TpArray); {-Write array to disk}
var AP : TpArrayInternal absolute A;
begin with AP^ do begin if not Opened then Exit;
{Assure disk is fully updated}
FLUSHA(A); if ArrayErrorLast <> 0 then Exit; {Close up} Close(F);

```

```

Opened := False;  if IoResult <> 0 then      Error(10);  end; end;
{.$DEFINE OLDSTYLE}
procedure DISPOSEA(var A : TpArray; DeleteFile : Boolean); {-Deallocate an
array}
var  AP : TpArrayInternal absolute A;
begin  {$IFDEF OLDSTYLE}  with AP^ do begin  if Opened then begin
{Store the array in case it is wanted later}
STOREA(A);  if DeleteFile then  if ArrayErrorLast = 0 then
{File was closed in STOREA, safe to erase here}
Erase(F);  end; {Deallocate its space}
DisposePrimitive(A, 5, PagesInRam);  end; {$ELSE}
with AP^ do begin  if Opened then begin  if DeleteFile then begin Close(F);
if IoResult = 0 then  {File was closed in STOREA, safe to erase here}
Erase(F)
else      Error(10);  end  else {Store the array in case it is wanted
later}
STOREA(A);  end; {Deallocate its space}
DisposePrimitive(A, 5, PagesInRam);  end; {$ENDIF} end;
procedure LOADA(var A : TpArray; FileName : string; RamBytesForData :
LongInt);
{-Allocate and read array from disk}
var  AP : TpArrayInternal absolute A; FT : file; H : ArrayHeader; Bytes : Word;
DiskIndexSize : Word;
begin  {Open the input file}  Assign(FT, FileName);  Reset(FT, 1);
if IoResult <> 0 then begin  Error(11);  Exit;  end;
{Read the header}  BlockRead(FT, H, SizeOf(ArrayHeader), Bytes);
if (IoResult <> 0) or (Bytes <> SizeOf(ArrayHeader)) then begin  Close(FT);
Bytes := IoResult;  Error(12);  Exit;  end;  Close(FT);
{Make sure file is properly formatted}
if H.ID <> IdString then begin  Error(17);  Exit;  end;  with H do
{Allocate heap space}
if not ALLOCATEA(A, RowsInPage, ColsInPage, PagesDown, PagesAcross,
ElSize, RamBytesForData) then  Exit;  with AP^ do begin  Opened :=
False;
{Reopen the disk file}

```

```

Assign(F, FileName); Reset(F, 1); if IoResult <> 0 then begin Error(11); Exit;
end;
Opened := True; {Initialize the disk index}
Header.NextAvail := H.NextAvail; DiskIndexSize :=
SizeOf(LongInt)*Header.PageCount; Seek(F, SizeOf(ArrayHeader));
BlockRead(F, DiskIndex^, DiskIndexSize, Bytes);
if (IoResult <> 0) or (Bytes <> DiskIndexSize) then begin Close(F);
Bytes := IoResult; Error(12); Exit; end;
{Initialize the in-RAM index}
InitRamIndex(A); end; end; begin ArrayErrorLast := 0; end.
{*****}
{
{ Borland Pascal Version 7.0 }
{ Variable Unit/ }
{ Модуль опису даних }
{
{*****}
Unit Variable;
Interface
Uses TpVarray;
{ Опис змінних }
Var flag : Byte; {Опис змінних блоку CONTR }NP { Кількість вузлів
задачі }
NE { Кількість елементів } NB { Кількість закріплених вузлів
}
NDF { Число степенів вільності вузла [6] } NCN { Кількість вузлів
елемента [3] }
NLD { Кількість навантажень } NMAT { Кількість
матеріалів }
NSZF, { Кількість рівнянь в системі } L1, { } NT4 { }
M1, { } LT1, { } INQ { Кількість вузлів, де діє навантаження
}
NTOL { Кількість товщин }
NQ, { Номер вузла, де діє навантаження } IT : Word; { }
K1,K2,K3,K4,K5,K6,K7,K8,K9,K10, NBAND : Integer; { Ширина матриці }

```

```

{ Опис глобальних змінних }
KM1,KM2,KM3 : Word;{ Масив товщин елементів } H1 : Array[1..10] of
Real;
{ Масив останніх елементів даної товщини }N1 : Array[1..1000] of Word;
{ Масив останніх елементів матеріалу } N2 : Array[1..10] of Word;
{ Масив навантажень } P : Array[1..4800] of Real;
{ Товщини елементів } H : Array[1..1200] of Real;
{ Зв'язки елементів } NOP : Array[1..1200,1..3] of Integer;
{ Типи матеріалів } IMAT : Array[1..1200] of Real;
{ Властивості типів матеріалів } ORT : Array[1..5,1..2] of Real;
{ Номери закріплених елементів } NBC : Array[1..1200] of Word;
{ Вид закріплення вузлів } NFIX : Array[1..800] of LongInt;
{ Усічена матриця жорсткості конструкції } ES : Array[1..18,1..18] of
Real;
{ ST1 : TpArray;} ST : TpArray; hdBB : TpArray;
hdB4 : TpArray; X : TpArray;
function Atob(Arg: Real; Index: Byte): Real;
function Get1(A: TpArray; i,j: Word): Real;
procedure Set1(A: TpArray; i,j: Word; Val: Real);
procedure ClearVal(A: TpArray; V: Real; IT: InitType); Implementation
function Atob(Arg: Real; Index: Byte): Real;
var i : Integer; b : Real;
begin b:= 1; for i:= 1 to Index do b:= Arg * b; atob:= b; end;
function Get1(A: TpArray; i,j: Word): Real;
var row,col : Word; Val : Real;begin row:=i-1; col:=j-1; RETA(A, row, col,
Val); Get1:= Val; end;
procedure Set1(A: TpArray; i,j: Word; Val: Real);
var row,col : Word; begin row:= i-1; col:= j-1; SETA(A, row, col, Val);end;
procedure ClearVal(A: TpArray; V: Real; IT: InitType);
begin CLEARA(A,V,IT);end; End.
{*****}
{
}
{ Borland Pascal Version 7.0 }
{ Input Data Unit/ }
{ Модуль читання даних }

```



```

{
}
{*****}
Unit NEWDATA; Interface Uses Crt, Variable, TrVarray, Dos;
{-----}
{ Підпрограма читання основних початкових }
{ даних про геометрію конструкції з файлу }
{-----}
Procedure GDATA; Var DATFile : Text; DATFileName : PathStr;
Implementation
Procedure GDATA; Var i,j,k,_k,l,m,n : Integer; nk1,nk2,nk3 : Word; V1 :
Real;
Begin Assign(DATFile, DATFileName); Reset(DATFile);
MakeAPrimitive(X, 1000, 3, 1, 1, SizeOf(Real), 'x.dat', 3002*SizeOf(Real));
ClearVal(X, 0.0, ExactInit);
{- Введення інформації про конструкцію -----}
Read(DATFile, NP); Read(DATFile, NE); Read(DATFile, NB);
Read(DATFile, NDF); Read(DATFile, NMAT); Read(DATFile, NTOL);
Read(DATFile, N2[NMAT]); Read(DATFile, ORT[NMAT, 1]);
Read(DATFile, ORT[NMAT, 2]); Read(DATFile, N1[NTOL]);
Read(DATFile, H1[NTOL]); Read(DATFile, H[1]); Read(DATFile, H[2]);
{- Введення координат вузлів, граничних умов і навантажень -----}
for n:= 1 to NP do begin for m:= 1 to 3 do begin Read(DATFile, V1);
Set1(X, n, m, V1); end; Read(DATFile, NFIX[n]);
for k:= 1 to 6 do Read(DATFile, P[6*(N-1)+k]); end;
{- Ведення зв'язків вузлів в елементах -----}
for n:= 1 to NE do for m:= 1 to 3 do Read(DATFile, NOP[n, m]);
Close(DATFile);
{- Формування додаткових даних -----}
for n:= 1 to NE do begin k:= 1; while n > N1[k] do Inc(k); H[n]:= H1[k]; end;
for n:= 1 to NE do begin k:= 1; while n > N2[k] do Inc(k);
IMAT[n]:= до; end;
{- Визначення ширини матриці NBAND -----}
NBAND:= 0; for n:= 1 to NE do begin
NK1:= 6*(Abs(NOP[n, 3]-NOP[n, 1])+1); { 3-1 }
NK2:= 6*(Abs(NOP[n, 2]-NOP[n, 1])+1); { 2-1 }

```

```

NK3:= 6*(Abs(NOP[n,3]-NOP[n,2])+1); { 3-2 }
if NK1 > NBAND then NBAND:= NK1; if NK2 > NBAND then NBAND:=
NK2;
if NK3 > NBAND then NBAND:= NK3; end;
{- Перевірка ширини стрічки NBAND -----
}
if (NBAND - 100) > 0 then begin      writeln('Error!');  HighVideo; writeln;
writeln(' Помилка: Недостатня ширина стрічки !');  NormVideo;
Halt(1); end;End;End.
{*****}
{          } {      Borland Pascal Version 7.0
}
{  Mathematic Unit/          }
{  Модуль формування матриці жорсткості      }
{  конструкції і рішення системи рівнянь      }
{*****}
Unit Matrix; Interface
{-----} { Підпрограма формування жорсткості
)
( елемента          ) {-----}
Procedure Element(N_: word);{-----}
{ Підпрограма формування матриці      ) ( жорсткості конструкції
}
{-----}
Procedure Matrica; {-----}
{ Підпрограма рішення системи рівнянь      )
Procedure Sistem; Implementation Uses Crt, TpVarray, Variable;
Procedure Element(N_ : Word);
Var i,j,k,ks,iJ,IT,kn,NX,NF,m,m1,IS,ki
IC,LT,NROWB,NCOLB,II,ICON,il,l_,kj,_i,_j:Integer;
ST11,s1,s2,s3,s4,s5,s6,DL,L,PL
a1,a2,a3,b1,b2,b3,c1,c2,c3,comm,q,s,t : Real; A : Array[1..3,1..9] of Real;
AL1,AL2,AL3 : Array[1..4] of Real; B : Array[1..3,1..6] of Real;
BA : Array[1..3,1..9] of Real; BB : Array[1..3,1..18] of Real;
B4 : Array[1..3,1..9] of Real; E : Array[1..3,1..3] of Real;

```

```

E1 : Array[1..18,1..18] of Real; E2 : Array[1..9,1..9,1..3] of Real;
E3 : Array[1..18,1..18] of Real; E5 : Array[1..9,1..9] of Real;
V : Array[1..3,1..3] of Real; V1 : Array[1..6,1..9] of Real;
V2 : Array[1..9,1..18] of Real; X1 : Array[1..9] of Real;
XL : Array[1..9] of Real; { VIEWS.INC } Begin { base }
{- Визначення зв'язків елементів-----}
I:= NOP[N_,1]; J:= NOP[N_,2]; K:= NOP[N_,3]; L:= IMAT[N_];
{-Визначення координат в системі парал. глобальн., яка проход.через і-у
точку--}
for II:= 1 to 3 do for KK:= 1 to 3 do begin {for kk}IJ:= 3 * (II-1) + KK;
X1[IJ]:= 0; XL[IJ]:= Get1(X,NOP[N_,II], KK)- Get1(X,NOP[N_,1], KK); end;
{for kk}
{-Визначення направляючих косинусів осі X1, які проход. через точки I,J--
---}
S1:= get1(X,J,1) - get1(X,I,1); S2:= get1(X,J,2) - get1(X,I,2);
S3:= get1(X,J,3) - get1(X,I,3); DL:= SQRT(S1*S1 + S2*S2 + S3*S3);
V[1,1]:= S1/DL; V[1,2]:= S2/DL; V[1,3]:= S3/DL;
{- Визначення направляючих косинусів осі Z1-----}
S4:= (Get1(X,J,2)-Get1(X,I,2))*(Get1(X,K,3)-Get1(X,I,3))-
(Get1(X,J,3)-Get1(X,I,3))*(Get1(X,K,2)-Get1(X,I,2));
S5:= (Get1(X,K,1)-Get1(X,I,1))*(Get1(X,J,3)-Get1(X,I,3))-
(Get1(X,K,3)-Get1(X,I,3))*(Get1(X,J,1)-Get1(X,I,1));
S6:= (Get1(X,J,1)-Get1(X,I,1))*(Get1(X,K,2)-Get1(X,I,2))-
(Get1(X,K,1)-Get1(X,I,1))*(Get1(X,J,2)-Get1(X,I,2));
PL:= SQRT(S4*S4 + S5*S5 + S6*S6); V[3,1]:=S4/PL; V[3,2]:=S5/PL;
V[3,3]:=S6/PL;
{- Визначення направляючих косинусів осі Y1-----}
V[2,1]:= V[3,2]*V[1,3]-V[3,3]*V[1,2]; V[2,2]:= V[3,3]*V[1,1]-V[3,1]*V[1,3];
V[2,3]:= V[3,1]*V[1,2]-V[3,2]*V[1,1];
{-Вивод v[3,3]-----}
{ View_V3x3;}
{- Визначення локальних координат в системі X1 Y1 Z1-----}
for KK:= 1 to 3 do for II:= 1 to 3 do begin{for ii} IJ:= 3*(KK-1)+II;
for KI:= 1 to 3 do begin{for ki}IL:= 3*(KK-1)+KI;
X1[IJ]:= X1[IJ]+ V[II,KI]* XL[IL]; end;{for ki}end;{for ii}

```

```

{ вивід x1[9]-----}
{ View_X1;}
{- Визначення коефіцієнтів a1,b1,b2,b3,c1,c2,c3-----}
A1:= (X1[4]* X1[8] - X1[7]* X1[5]) / 2; B1:= X1[5] - X1[8]; B2:= X1[8];
B3:= -X1[5]; C1:= X1[7] - X1[4]; C2:= -X1[7]; C3:= X1[4];
if not (A1 > 0) then begin{if}
writeln('Невірно пронумеровані елемент ',N_:3);
for i:= 1 to 3 do write(' ',NOP[N_,i]:3);      writeln;      Halt(1); end;{if}
{-Формування матриці зв'язку деформацій з переміщеннями в площині
елемента-}
for i:= 1 to 3 do for j:= 1 to 6 do a[i,j]:= 0;A[1,1]:= B1; A[1,3]:=B2;A[1,5]:=B3;
A[2,2]:= C1; A[2,4]:= C2; A[2,6]:= C3; A[3,1]:= C1;A[3,2]:= B1;A[3,3]:=C2;
A[3,4]:= B2; A[3,5]:= C3; A[3,6]:= B3;
{-Формування матриці напружень з деформаціями в площині елемента----}
L_:= Trunc(L); COMM:= ORT[L_,1]/((1-ORT[L_,2]*ORT[L_,2])*A1);
E[1,1]:= COMM; E[1,2]:= COMM * ORT[L_,2]; E[1,3]:= 0; E[2,1]:= E[1,2];
E[2,2]:= E[1,1]; E[2,3]:= 0; E[3,1]:= 0; E[3,2]:= 0;
E[3,3]:= (COMM * (1 - ORT[L_,2])) / 2;
{ вивід e[3,3]-----}
{ View_E3x3;}
{-Визначення матриці напружень в площині елемента-----}
for i:= 1 to 3 do for j:= 1 to 6 do begin {for j} B[i,j]:= 0;
for k:=1 to 3 do B[i,j]:= B[i,j]+ E[i,k]/2 * A[k,j]; end;{for j}
{-Виведення b[3,6]-----}
{ View_B3x6;}
{-----}
for i:= 1 to 18 do for j:= 1 to 18 do E3[i,j]:= 0; for k:= 1 to 6 do
for i:= 1 to 3 do begin {for i} KI:= i + (k-1) * 3; for j:= 1 to 3 do begin {for j}
KJ:= j + (k-1) * 3; E3[KI,KJ]:= V[i,j]; end; {for j} end; {for i}
for i:= 1 to 6 do for j:= 1 to 9 do V1[i,j]:= 0; for k:= 1 to 3 do
for i:= 1 to 2 do begin {for i} KI:= i + (k-1) * 2; for j:= 1 to 3 do begin {for j}
KJ:= j + (k-1) * 3; V1[KI,KJ]:= V[i,j]; end; {for j}end; {for i}
{Виведення-v1[6,9]-----} {
View_V16x9;}
{-----}

```

```

for i:= 1 to 3 do      for j:= 1 to 9 do begin{FOR J}          B4[i,j]:= 0;
for k:= 1 to 6 do B4[i,j]:= B4[i,j]+ B[i,k]* V1[k,j]; end;{for j}
{-Виведення b4[3,9]-----}
{View_b43x9;}
{----- запис на диск B43x9 -----}
for _i:=1 to 3 do for _j:=1 to 9 do Set1(hdB4,3*(N_-1)+_i,_j,B4[_i,_j]);
{------} KM1:= KM1 + 1;
{- визначення матриці жорсткості в площині елемента-----}
for i:= 1 to 6 do      for j:=1 to 6 do begin {for j}          E1[i,j]:= 0;
for k:= 1 to 3 do E1[i,j]:= E1[i,j]+ B[k,i]/2 * A[k,j]* H[N_];      end; {for j}
{- виведення e1[6,6]-----}
{ View_E16x6;}
{- ввід масивів L-координат і вагових коеф. для чисельного інтегрування--}
AL1[1]:= 0.33333; AL2[1]:= 0.33333; AL3[1]:= 0.33333; AL1[2]:= 0.5;
AL2[2]:= 0.5; AL2[3]:= 0.5; AL3[3]:= 0.5; AL1[4]:= 0.5; AL3[4]:= 0.5;
AL3[2]:= 0.0; AL1[3]:= 0.0; AL2[4]:= 0.0; A2:= -1/(4*A1*A1); A3:=
1/(2*A1*A1);
{- формування матриці зв'язку напружень з деформаціями в площині
перпенд.-}
COMM:= ORT[L_,1]* H[N_]*H[N_]*H[N_] / (12*(1-
ORT[L_,2]*ORT[L_,2]));
E[1,1]:= COMM; E[1,2]:= COMM * ORT[L_,2]; E[1,3]:= 0; E[2,1]:= E[1,2];
E[2,2]:= E[1,1]; E[3,1]:= 0; E[3,2]:= 0; E[3,3]:= COMM * (1.0-ORT[L_,2]) /
2;
{-формування матриці зв'язку деформацій з переміщеннями перпенд.-----}
for KI:= 1 to 4 do begin {FOR KI}   Q:= AL1[KI];   S:= AL2[KI];   T:=
AL3[KI];
A[1,1]:= (4*b1*b2+4*b1*b3-2*b2*b2)*Q+2*(b1*b1-2*b1*b2)*S+2*(b1*b1-
2*b1*b3)*T-2*(b3*b3)*Q;
A[1,2]:= b2*b3*(b3-b2)*Q+b3*b1*(b3-b2+2*b1)*S+b2*b1*(b3-b2-2*b1)*T;
A[1,3]:= (4*b1*(b2*c3-b3*c2)+b2*b3*(c3-c2))*Q+(2*b1*b1*c3+b1*b3*(c3-
c2))*S+(b1*b2*(c3-c2)-2*b1*b1*c2)*T;
A[1,4]:= (4*b2*b3+4*b2*b1-2*b3*b3-2*b1*b1)*S+2*(b2*b2-
2*b2*b3)*T+2*(b2*b2-2*b2*b1)*Q;
A[1,5]:= b3*b1*(b1-b3)*S+b1*b2*(b1-b3+2*b2)*T+b3*b2*(b1-b3-2*b2)*Q;

```

$$\begin{aligned}
A[1,6] &:= (4*b2*(b3*c1-c3*b1)+b3*b1*(c1-c3))*S+(2*b2*b2*c1+b2*b1*(c1-c3))*T+(b2*b3*(c1-c3)-2*b2*b2*c3)*Q; \\
A[1,7] &:= (4*b3*b1+4*b3*b2-2*b1*b1-2*b2*b2)*T+2*(b3*b3-2*b3*b1)*Q+2*(b3*b3-2*b3*b2)*S; \\
A[1,8] &:= b1*b2*(b2-b1)*T+b2*b3*(b2-b1+2*b3)*Q+b1*b3*(b2-b1-2*b3)*S; \\
A[1,9] &:= (4*b3*(b1*c2-b2*c1)+b1*b2*(c2-c1))*T+(2*b3*b3*c2+b3*b2*(c2-c1))*Q+(b3*b1*(c2-c1)-2*b3*b3*b1)*S; \\
A[2,1] &:= (4*c1*c2+4*c1*c3-2*c2*c2-2*c3*c3)*Q+2*(c1*c1-2*c1*c2)*S+2*(c1*c1-2*c1*c3)*T; \\
A[2,2] &:= (4*c1*(c2*b3-c3*b2)+c2*c3*(b3-b2))*Q+(2*c1*c1*b3+c1*c3*(b3-b2))*S+(c1*c2*(b3-b2)-2*c1*c1*b2)*T; \\
A[2,3] &:= c2*c3*(c3-c2)*Q+c3*c1*(c3-c2+2*c1)*S+c2*c1*(c3-c2-2*c1)*T; \\
A[2,4] &:= (4*c2*c3+4*c2*c1-2*c3*c3-2*c1*c1)*S+2*(c2*c2-2*c2*c3)*T+2*(c2*c2-2*c2*c1)*Q; \\
A[2,5] &:= (4*c2*(c3*b1-c1*b3)+c3*c1*(b1-b3))*S+(2*c2*c2*b1+c2*c1*(b1-b3))*T+(c2*c3*(b1-b3)-2*c2*c2*b3)*Q; \\
A[2,6] &:= c3*c1*(c1-c3)*S+c1*c2*(c1-c3+2*c2)*T+c3*c2*(c1-c3-2*c2)*Q; \\
A[2,7] &:= (4*c3*c1+4*c3*c2-2*c1*c1-2*c2*c2)*T+2*(c3*c3-2*c3*c1)*Q+2*(c3*c3-2*c3*c2)*S; \\
A[2,8] &:= (4*c3*(c1*b2-c2*b1)+c1*c2*(b2-b1))*T+(2*c3*c3*b2+c3*c2*(b2-b1))*Q+(c3*c1*(b2-b1)-2*c3*c3*b1)*S; \\
A[2,9] &:= c1*c2*(c2-c1)*T+c2*c3*(c2-c1+2*c3)*Q+c1*c3*(c2-c1-2*c3)*S; \\
A[3,1] &:= 2*(c1*b2+c2*b1+c1*b3+c3*b1-c2*b2-c3*b3)*Q+2*(c1*b1-c1*b2-c2*b1)*S+2*(c1*b1-c1*b3-c3*b1)*T; \\
A[3,2] &:= (0.5*(c3*b2+c2*b3)*(b3-b2)+2*b1*(b3*c2-b2*c3))*Q+(0.5*(b3-b2)*(c1*b3+b1*c3)+2*b1*c1*b3)*S+(0.5*(b3-b2)*(c1*b2+b1*c2)-2*b1*c1*b2)*T; \\
A[3,3] &:= (0.5*(c3-c2)*(b3*c2+b2*c3)+2*c1*(b2*c3-b3*c2))*Q+(2*b1*c1*c3+0.5*(c3-c2)*(c1*b3+c3*b1))*S+(0.5*(c3-c2)*(c1*b2+c2*b1)-2*b1*c2*c1)*T; \\
A[3,4] &:= 2*(c2*b3+c3*b2+c2*b1+c1*b2-c3*b3-c1*b1)*S+2*(c2*b2-c2*b3-c3*b2)*T+2*(c2*b2-c2*b1-c1*b2)*Q; \\
A[3,5] &:= (0.5*(c1*b3+c3*b1)*(b1-b3)+2*b2*(b1*c3-b3*c1))*S+(0.5*(b1-b3)*(c2*b1+b2*c1)+2*b2*c2*b1)*T+(0.5*(b1-b3)*(c2*b3+b2*c3)-2*b2*c2*b3)*Q;
\end{aligned}$$

```

A[3,6]:= (0.5*(c1-c3)*(b1*c3+b3*c1)+2*c2*(b3*c1-
b1*c3))*S+(2*b2*c2*c1+0.5*(c1-c3)*(c2*b1+c1*b2))*T+
(0.5*(c1-c3)*(c2*b3+c3*b2)-2*b2*c3*c2)*Q;
A[3,7]:= 2*(c3*b1+c1*b3+c3*b2+c2*b3-c1*b1-c2*b2)*T+2*(c3*b3-c3*b1-
c1*b3)*Q+2*(c3*b3-c3*b2-c2*b3)*S;
A[3,8]:= ((c2*b1+c1*b2)*(b2-b1)/2+2*b3*(b2*c1-b1*c2))*T+(0.5*(b2-
b1)*(c3*b2+b3*c2)+2*b3*c3*b2)*Q+
(0.5*(b2-b1)*(c3*b1+b3*c1)-2*b3*c3*b1)*S;
A[3,9]:= (0.5*(c2-c1)*(b2*c1+b1*c2)+2*c3*(b1*c2-
b2*c1))*T+(2*b3*c3*c2+0.5*(c2-c1)*(c3*b2+c2*b3))*Q+
(0.5*(c2-c1)*(c3*b1+c1*b3)-2*b3*c1*c3)*S;
for i:= 1 to 3 do      for j:= 1 to 9 do      if (i-2) <= 0 then A[i,j]:= A2 *
A[i,j]
else A[i,j]:= A3 * A[i,j];
{- визначення матриці моментів у площині перпендикулярно до елемента--
--} for i:= 1 to 3 do      for j:= 1 to 9 do begin {for j}      BA[i,j]:= 0;
for k:= 1 to 3 do BA[i,j]:= BA[i,j]+ E[i,k]* A[k,j];      end; {for j}
{-Виведення ba[3,9]-----} {
View_BA3x9;}
{-----}
if (KI - 1) = 0 then begin{then}
for i:= 1 to 9 do for j:= 1 to 18 do V2[i,j]:= 0;      for k:= 1 to 3 do begin {for до}
KS:= 1 + (k-1) * 3;      for j:= 1 to 3 do begin {for j}      KJ:= j + (k-1) * 6;
V2[KS,KJ]:= V[3,j];      end {for j};      end; {for до}
for k:= 1 to 3 do      for i:= 1 to 2 do begin {for i}      KS:= i + 1 + 3 * (k-1);
for j:= 1 to 3 do begin {for j}      KJ:= j + 3 + 6 * (k-1);      V2[KS,KJ]:= V[i,j];
end {for j};      end; {for i} for i:= 1 to 3 do      for j:= 1 to 18 do begin {for j}
BB[i,j]:= 0; for k:= 1 to 9 do BB[i,j]:= BB[i,j]+ BA[i,k]* V2[k,j]; end; {for j}
km1:=km1+2;
{------ Заповнення глобального BB -----}
for _i:=1 to 3 do for _j:=1 to 18 do Set1(hdBB,3*(N_-1)+_i,_j,BB[_i,_j]);
{------}
end{then}      else begin      for i:= 1 to 9 do      for j:= 1 to 9 do begin {for j}
E2[i, j, KI-1]:= 0; for k:= 1 to 3 do E2[i,j,KI-1]:= E2[i,j,KI-1]+ BA[k,i]* A[k,j];
end; {end j}      end;{else} {-Вивід e2[9,9,3]-----}

```

```

{View_E29x9x3;} {- виведення bb[3,18]-----}
{View_BB3x18;} end; {FOR KI}
{- виведення a[3,9]-----} {
View_A3x9;}
{-----}
for i:= 1 to 9 do for j:= 1 to 9 do begin {for j}E5[i,j]:= 0;
for k:= 1 to 3 do E5[i,j]:= E5[i,j]+ E2[i,j,k]* A1/3; end; {for j}
{- виведення e5[9,9]-----} {
View_E59x9;}
{-----}
{- одержання матриці жорсткості елемента в локальних координатах-----}
for i:= 1 to 18 do for j:= 1 to 18 do ES[i,j]:= 0; for I:=1 to 3 do begin
for j:=1 to 3 do ES[i*6,j*6]:= -0.00005 * ORT[L_, 1]* H[N_]* A1;
ES[i*6,i*6]:= 0.0001 * ORT[L_, 1]* H[N_]* A1; end;
for i:= 1 to 6 do begin {for i}      K:= ((i-1) div 2) * 4 + i;
for j:= 1 to 6 do begin {for j}    KI:= ((j-1) div 2) * 4 + j;    ES[K,KI]:= E1[i,j];
end; {for j} end; {for i} for i:= 1 to 9 do begin {for i}K:= ((i-1) div 3)*3+i+2;
for j:= 1 to 9 do begin {for j}    KI:= ((j-1) div 3)*3+j+2;  ES[K,KI]:= E5[i,j];
end; {for j} end;{for i}
{- вивід es[18,18]-----} {
View_ES18x18;}
{- одержання матриці жорсткості елемента в глобальних координатах-----}
for i:= 1 to 18 do for j:= 1 to 18 do begin  E1[i,j]:=0;
for k:= 1 to 18 do E1[i,j]:= E1[i,j]+ ES[i,k]* E3[k,j];    end;
{- виведення e1[18,18]-----} {
View_E118x18;}
{-----}
for i:= 1 to 18 do for j:= 1 to 18 do begin {for j} ES[i,j]:= 0;
for k:= 1 to 18 do ES[i,j]:= ES[i,j]+ E3[k,j]* E1[k,i];    end; {for j}
{-Виведення es[18,18]-----} {
View_ES18x18end;}
{-----}
STOREA(X);
End; {ELEMENT}
{-----}

```


Procedure Matrica;

Label 1;

Var NK, JJ, NR, KK, NC1, ICON, NRO, NC, M, J, I, K, L : Integer; N, hN : Word;

NX : Real; PST : Array[1..2] of Real; NumColST, NumPageST

NumPageP, NumPageBB, NumPageB4 : Word; Val : Real; Percent, iP :

Real; iX, iY : Byte; Factor : Word;

Begin flag:= 0; if (NSZF*NBAND*6) < 65530 then begin

MakeAPrimitive(ST, NSZF, NBAND, 1, 1, SizeOf(Real)

'arr_st.dat', ((NSZF*NBAND)+2)*SizeOf(Real)); flag:= 1; end else

begin if (nszf mod 144) = 0 then NumPageST:= (nszf div 144)

else NumPageST:= (nszf div 144)+1;

if (nband mod 72) = 0 then NumColST:= (nband div 72)

else NumColST:= (nband div 72)+1;

MakeAPrimitive(ST, 144, 72, NumPageST, NumColSt, SizeOf(Real)

'arr_st.dat', 10370*SizeOf(Real)); end; ClearVal(ST, 0.0, FastInit);

NumPageBB:= (ne*3 div 606)+1;

MakeAPrimitive(hdBB, 606, 18, NumPageBB, 1, SizeOf(Real) { Pointer Error! }

'arr_bb.dat', 10910*SizeOf(Real)); ClearVal(hdBB, 0.0, FastInit);

NumPageB4:= (NE*3 div 1212)+1;

MakeAPrimitive(hdB4, 1212, 9, NumPageB4, 1, SizeOf(Real)

'arr_b4.dat', 10910*SizeOf(Real)); ClearVal(hdB4, 0.0, FastInit);

{ Формування матриці жорсткості конструкції }

iX:= WHEREX; iY:= WHEREY; Percent:= 0; iP:= 100 / NE;

For N:= 1 to NE do begin hN:= N; ELEMENT(hN);

For JJ:= 1 to 3 do begin NR:= NDF * (NOP[N, JJ] - 1); For J:= 1 to NDF do

begin NR:= NR+1; I:= (JJ-1) * NDF + J; For KK:= 1 to 3 do begin

NC1:= NDF * (NOP[N, KK] - 1); For K:= 1 to NDF do begin

L:= NDF * (KK-1) + K; NC:= NC1+K+1-NR;

Val:= Get1(ST, NR, NC)+ ES[I, L]; If NC > 0 then Set1(ST, NR, NC, Val);

end; end; end; end; { Вивід }

Percent:= Percent + iP; gotoXY(iX, iY); write(Percent:3:0, '%');

write(' эл-т', N:3, ' связи', NOP[N, 1]:4, NOP[N, 2]:4, NOP[N, 3]:4); end;

gotoXY(iX, iY); writeln('Ok. '); { Облік граничних умов }

write(' Облік граничних умов...'); Percent:= 0; iP:= 100 / NP;

iX:= WHEREX; iY:= WHEREY; For N:= 1 to NP do begin

IF Round(NFIX[N])= 0 then Continue; NX:= 100000; NR:= 6*(N-1);

For M:=1 to 6 do begin NR:= NR+1; ICON:= Round(NFIX[N]/NX);

If ICON <= 0 then goto 1; Set1(St, Nr, 1, 1); P[NR]:=0.0;

For J:= 2 to NBAND do begin Set1(ST, NR, J, 0.0); NRO:= NR+1-J;

```

If NRO > 0 then Set1(ST,NRO,J,0.0);      end;
NFIХ[n]:= NFIХ[n] - Round(NX)*ICON;1:  NX:= Round(NX/10); end;
{ Вивод % }
Percent:= Percent + iP;      gotoXY(iX, iY); write(Percent:3:0,'%'); end;
gotoXY(iX, iY); writeln('Ok.      ');
{ for i:=1 to 24 do begin      write(Get1(st,i,1):12:1, Get1(st,i,2):12:1,
P[i]:8:1);
if (i mod 3) = 0 then Writeln; end;      } STOREA(hdB4); STOREA(hdBБ);
STOREA(St); DISPOSEA(ST,False); DISPOSEA(X,true);End;
{-----}
Procedure Sistem;
Var N,I,L,K,J : Integer; C,Val      : Real; PageSize : LongInt; Percent,iP :
Real;  iX,iY   : Byte; Begin
if flag=1 then LOADA(St, 'arr_st.dat', ((NSZF*NBAND)+2)*SizeOf(Real))
else LOADA(St, 'arr_st.dat',10370*SizeOf(Real));
{ MakeAPrimitive(ST1, 2046, 1,1, 72, SizeOf(Real)
'arr_st1.dat', 10232*SizeOf(Real)); for i:=1 to 2046 do for j:=1 to 72 do
Begin GOTOXY(24,1); Write(i:6,j:6); Set1(St1,i,j,Get1(st,i,j)); end;}
Percent:= 0; iP:= 100 / NSZF; iX:= WHEREX; iY:= WHEREY;
For N:= 1 to NSZF do begin  I:= N;      For L:= 2 to NBAND do begin
I:= I + 1;  If Get1(ST,N,L)=0 then Continue;
C:= Get1(ST,N,L) /Get1(ST,N,1);  J:= 0;
For K:= L to NBAND do begin J:= J + 1;  If Get1(ST,N,K)<> 0 then Begin
Val:=Get1(ST,I,J) - 3 * Get1(ST,N,K);      Set1(ST,I,J,Val); end; end;
Set1(ST,N,L,C);  P[i]:= P[I] - C * P[N];  end;  P[N]:= P[N]/ Get1(ST,N,1);
{ Вивод % }
Percent:= Percent + iP;  gotoXY(iX, iY); write(Percent:3:0,'%'); end;
gotoXY(iX, iY); writeln('Ok.      '); For N:= NSZF-1 downto 1 do begin
L:= N;      For K:= 2 to NBAND do begin      L:= L + 1;
If Get1(ST,N,K)<> 0 then begin      P[n]:= P[N] - Get1(ST,N,K)* P[L];
end; end; end; DISPOSEA(ST, True);
{ DISPOSEA(ST1, True);} End;  End.
{*****}
{ Borland Pascal Version 7.0      }
{ Mathematic Unit/                }
{ Модуль визначення напружень    }
Unit Deform;
Interface
Uses Dos;

```

```

{ Підпрограма визначення напружень } { що виникають в конструкції }
Procedure Deformation;
Var OUTFile : Text; OUTFileName : PathStr;
Implementation Uses Crt, Variable, TrVarray, Matrix;
Procedure Deformation;
Var a,b,Ang,Val : Real; Sigma : Array [1..13] of Real;
P2 : Array [1..18] of Real; _n,_i,_ii,_m, _ki,_kk,_j, _ib,_il : Integer;
ch : Char; OutFile_S : Text; Out_B41 : Text; Out_BB1 : Text;
Percent,iP : Real; iX,iY : Byte;
Begin writeln; write(' Визначення напружень...');
LOADA(hdBB, 'arr_bb.dat', 10910*SizeOf(Real));
LOADA(hdB4, 'arr_b4.dat', 10910*SizeOf(Real));
Percent:= 0; iP:= 100 / NE; iX:= WHEREX; iY:= WHEREY; for _n:=1 to NE
do begin { Вивод % }
Percent:= Percent + iP; gotoXY(iX,iY); write(Percent:3:0,'% ');
for _ii:= 1 to 3 do begin {for _ii}for _j:= 1 to 6 do begin
m:= (NOP[_n, _ii]-1)*6+ _j; P2[( _ii-1)*6+ _j]:= P[_m]; end; end;
for _ii:= 1 to 3 do begin Sigma[_ii]:= 0;
for _kk:= 1 to 3 do for _i := 1 to 3 do begin _ki:= 6*( _kk-1)+_i;
_j:= 3*( _kk-1)+_i; Sigma[_ii]:= Sigma[_ii]+ Get1(hdB4,3*( _n-1)+_ii,_j)*
P2[_ki]; end; end;
for _ii:= 1 to 3 do begin _ib:= _ii+3; Sigma[_ib]:= 0;
for _j:=1 to 18 do Sigma[_ib]:= Sigma[_ib]+ Get1(hdBb, 3*( _n-1)+_ii, _j)*
P2[_j]; end;
{ Визначення сумарних напружень від згинання і розтягання }
for _il:= 1 to 3 do Sigma[_il+6]:= Sigma[_il]+6* Sigma[_il+3]/(H[_n]*H[_n]);
for _il:= 1 to 3 do Sigma[_il+10]:= Sigma[_il]-6* Sigma[_il+3]/(H[_n]*H[_n]);
{ Визначення головних і еквівалентних напружень }
{ a:= Sigma[7]+ Sigma[8]; }
a:= Sqrt(Sqr(Sigma[7]-Sigma[8])+4* Sigma[9]* Sigma[9]);
b:= Sqrt(Sqr(Sigma[11]-Sigma[12])+4* Sigma[13]* Sigma[13]);
if a > b then Sigma[10]:= a else Sigma[10]:= b; { Запис у файл результатів }
write(OUTFile,['',_n:4,']); for _i:= 1 to 10 do write(OUTFile, Sigma[_i]:13:5, ' ');
writeln(OUTFile); end; { NE } gotoXY(iX,iY); writeln('Ok. ');
DISPOSEA(hdBB, True); DISPOSEA(hdB4, True); End; End.

```

ДОДАТОК Е

Текст програми розрахунку процесу ущільнення пористого матеріалу

unit MAINU;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs
StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart, ComCtrls, ToolWin,
GLOBALU, FILEU, SHELLU, ActnList, ImgList, Buttons;

type TMode = (mdInsert, mdEdit, mdBrowse); TfmMain = class(TForm)

pnModel: TPanel; pnChart: TPanel; sbPosition: TScrollBar; Chart: TChart;
Series1: TLineSeries; Series2: TLineSeries; Series3: TLineSeries; pnGraphic:
TPanel;

gbGraphic: TGroupBox; lbName: TLabel; edT: TEdit; edX: TEdit;
edX1: TEdit; edX2: TEdit; cbX: TCheckBox; cbX1: TCheckBox; cbX2:
TCheckBox; gbGraphicTime: TGroupBox; Label1: TLabel; edGT0: TEdit;
edGT1: TEdit;

Label2: TLabel; gbModelProperties: TGroupBox; gbElementProperties:
TGroupBox;

edC: TEdit; Label6: TLabel; Label4: TLabel; edK: TEdit; edM: TEdit;
Label3: TLabel; edIndex: TEdit; Label5: TLabel; udIndex: TUpDown;
tbToolBar: TToolBar; pnModelProperties: TPanel; edNumElements: TEdit;
Label7: TLabel; lwModel: TListView; ActionList: TActionList; acDraw:
TAction; acSolve: TAction; tbSolve: TToolButton; ImageList: TImageList;
acInsert: TAction; acPost: TAction; acDelete: TAction; acCancel: TAction;
tbInsert: TToolButton; tbPost: TToolButton; tbCancel: TToolButton;
tbDelete: TToolButton; tbSep1: TToolButton; tbSep2: TToolButton; tbDraw:
TToolButton; acEdit: TAction;

btEdit: TToolButton; Label8: TLabel; edAlpha: TEdit; Label9: TLabel;
edBeta: TEdit; Label10: TLabel; edV: TEdit; Label11: TLabel;
edTimeStep: TEdit; Label12: TLabel; edFinishTime: TEdit; Label13:
TLabel; Label14: TLabel;

Series4: TLineSeries; sbCalcAt: TSpeedButton; acCalcAt: TAction;

procedure FormCreate(Sender: TObject);

procedure FormDestroy(Sender: TObject);

procedure cbXClick(Sender: TObject);

procedure acDrawExecute(Sender: TObject);

```

procedure acSolveExecute(Sender: TObject);
procedure acDrawUpdate(Sender: TObject);
procedure acInsertExecute(Sender: TObject);
procedure acPostExecute(Sender: TObject);
procedure acCancelExecute(Sender: TObject);
procedure acDeleteExecute(Sender: TObject);
procedure acInsertUpdate(Sender: TObject);
procedure acPostUpdate(Sender: TObject);
procedure acEditExecute(Sender: TObject);
procedure acEditUpdate(Sender: TObject);
procedure lwModelChange(Sender: TObject; Item: TListItem; Change:
TItemChange);
procedure acCalcAtExecute(Sender: TObject);
procedure sbPositionChange(Sender: TObject);
private { Private declarations } FShell: TModelShell; FMode: TMode;
public
{ Public declarations } procedure DrawShell; procedure CursorAt(T:
double); end; var fmMain: TfmMain; implementation {$R *.DFM}
procedure TfmMain.DrawShell; var I: integer; LI: TListItem;
begin with lwModel.Items do begin BeginUpdate; Clear;
for I:=0 to FShell.ElementCount-1 do begin LI:=Add;
LI.Caption:=Format('Элемент %d',[I+1]); LI.Data:=FShell.Elements[I];
end; EndUpdate; end;
edNumElements.Text:=Format('%d'[FShell.ElementCount]);end;
procedure TfmMain.CursorAt(T: double);
var I, J: integer; Y0, Y1, Y, K: double;begin Chart.SeriesList[3].Clear;
Chart.SeriesList[3].Active:=false;
if (T<Chart.BottomAxis.Minimum) or (T>Chart.BottomAxis.Maximum) then
Exit;
for I:=0 to Chart.SeriesList[0].Count-2 do if (T>=Chart.
SeriesList[0].XValue[I]) and (T<Chart.SeriesList[0].XValue[I+1]) then
begin J:=I+1; Break; end;
if (T=Chart.SeriesList[0].XValue[Chart.SeriesList[0].Count-1]) then
begin J:=Chart.SeriesList[0].Count-1; I:=J-1; end;
K:=(T-Chart.SeriesList[0].XValue[I])/(Chart.SeriesList[0].XValue[J]-
Chart.SeriesList[0].XValue[I]);
edT.Text:=Format('%10.5f'[T]); Y0:=Chart.SeriesList[0].YValue[I];

```

```

Y1:=Chart.SeriesList[0].YValue[J]; Y:=Y0*(1.0-K)+Y1*K;
edX.Text:=Format('% 10.5f'[Y]); Y0:=Chart.SeriesList[1].YValue[I];
Y1:=Chart.SeriesList[1].YValue[J]; Y:=Y0*(1.0-K)+Y1*K;
edX1.Text:=Format('% 10.5f'[Y]); Y0:=Chart.SeriesList[2].YValue[I];
Y1:=Chart.SeriesList[2].YValue[J]; Y:=Y0*(1.0-K)+Y1*K;
edX2.Text:=Format('% 10.5f'[Y]);
Chart.SeriesList[3].AddXY(T,Chart.LeftAxis.Minimum,",Chart.SeriesList[3].Se
riesColor); Chart.SeriesList[3].AddXY(T, Chart.LeftAxis.
Maximum,",Chart.SeriesList[3].SeriesColor);
Chart.SeriesList[3].Active:=true;end;
// ===== Дії =====
procedure TfmMain.acDrawExecute(Sender: TObject); var F: TDoubleFile;
V: TVector; T, T0, T1: double; I, index: integer; E: TEdit;
begin Index:=StrToInt(edIndex.Text); try E:=edGT0;
T0:=StrToFloat(Trim(E.Text)); E:=edGT1; T1:=StrToFloat(Trim(E.Text)); if
(T1<T0) then begin T:=T1; T1:=T0;
T0:=T; end; except on A:EConvertError do begin
MessageDlg('Неприпустиме число з плаваючою комою
'+E.Text,mtError,[mbOK],0); E.SetFocus; Exit; end; end;
Chart.SeriesList[3].Clear; Chart.SeriesList[3].Active:=false; for I:=0 to 2 do
begin Chart.SeriesList[I].Clear; AssignFile(F,Format('X%d.dat',[I]));
SetLength(V,FShell.Model.NumNodes); try Reset(F); while not EOF(F) do
begin ReadVector(F,FShell.Model.NumNodes,T,V);
if (T>=T0) and (T<=T1) then Chart.SeriesList[I].AddXY(T,V[Index]);
end; finally CloseFile(F); V:=nil; end; end; end;
procedure TfmMain.acSolveExecute(Sender: TObject);
var E: TEdit; Alpha, Beta, V, TimeStep, FinishTime: double;
begin try E:=edAlpha; Alpha:=StrToFloat(Trim(E.Text)); E:=edBeta;
Beta:=StrToFloat(Trim(E.Text)); if (Beta<0.5) then begin
MessageDlg('Метод є абсолютно стійким при Beta>=0.5',mtError,[mbOK],0);
edBeta.SetFocus; Exit; end;
if (Alpha<0.25*sqr(0.5+Beta)) then begin MessageDlg('Метод є
абсолютно стійким при Alpha>=0.25',mtError,[mbOK],0);
edAlpha.SetFocus; Exit; end; E:=edV; V:=StrToFloat(Trim(E.Text));
E:=edTimeStep; TimeStep:=StrToFloat(Trim(E.Text));
E:=edFinishTime; FinishTime:=StrToFloat(Trim(E.Text));
FShell.MakeModel; if FShell.ElementCount=0 then begin

```

```

MessageDlg('Модель повинна містити хоча б один
елемент',mtError,[mbOK],0); Exit; end;
// Установка властивостей моделі
FShell.Model.X10[FShell.Model.NumElements]:=V;
FShell.Model.TimeStep:=TimeStep; FShell.Model.FinishTime:=FinishTime;
FShell.Model.Alpha:=Alpha; FShell.Model.Beta:=Beta;
// Вирішити модель FShell.Model.Solve2; MessageDlg('Рішення
зачено',mtWarning,[mbOK],0); udIndex.Max:=FShell.Model.NumNodes-1;
except on A:EConvertError do begin MessageDlg('Неприпустиме число з
плаваючою комою '+E.Text,mtError,[mbOK],0); E.SetFocus; end; end;end;
procedure TfmMain.acDrawUpdate(Sender: TObject);
begin (Sender as TAction).Enabled:=Assigned(FShell.Model);end;
procedure TfmMain.acInsertExecute(Sender: TObject);
begin gbElementProperties.Enabled:=true; FMode:=mdInsert; end;
procedure TfmMain.acEditExecute(Sender: TObject);
begin gbElementProperties.Enabled:=true; FMode:=mdEdit; end;
procedure TfmMain.acPostExecute(Sender: TObject);
var E: TEdit; M, K, C: double; El: TElementProperties;
begin try E:=edM; M:=StrToFloat(Trim(E.Text)); E:=edK;
K:=StrToFloat(Trim(E.Text)); E:=edC; C:=StrToFloat(Trim(E.Text));
if (FMode=mdInsert) then begin El:=TElementProperties.Create(C, M, K);
FShell.Add(El); DrawShell; end else begin
El:=TElementProperties(lwModel.Selected.Data); El.M:=M; El.K:=K;
El.C:=C; end;
gbElementProperties.Enabled:=false; FMode:=mdBrowse; except
on A:EConvertError do begin MessageDlg('Неприпустиме число з
плаваючою комою '+E.Text,mtError,[mbOK],0); E.SetFocus; end; end;
end;
procedure TfmMain.acCancelExecute(Sender: TObject);
begin gbElementProperties.Enabled:=false; FMode:=mdBrowse; end;
procedure TfmMain.acDeleteExecute(Sender: TObject);
var El: TElementProperties; Index: integer; begin
if Assigned(lwModel.Selected) and Assigned(lwModel.Selected.Data) then
begin
El:=TElementProperties(lwModel.Selected.Data); Index:=FShell.IndexOf(El);
if Index>-1 then begin FShell.Remove(Index); DrawShell; end; end;
end;

```

```

procedure TfmMain.acInsertUpdate(Sender: TObject);
begin (Sender as TAction).Enabled:=(FMode=mdBrowse); end;
procedure TfmMain.acPostUpdate(Sender: TObject);
begin (Sender as TAction).Enabled:=FMode in [mdEdit, mdInsert];end;
procedure TfmMain.acEditUpdate(Sender: TObject);
begin (Sender as TAction).Enabled:=(FMode=mdBrowse) and
Assigned(lwModel.Selected) and Assigned(lwModel.Selected.Data);end;
// ===== Подія=====
procedure TfmMain.FormCreate(Sender: TObject);
var E: TElementProperties;
begin DecimalSeparator:=','; FMode:=mdBrowse;{ FModel:=TModel.Create;
FModel.NumNodes:=3; for I:=0 to FModel.NumElements-1 do begin
FModel.MElem[I]:=1.0E2; FModel.CElem[I]:=0.0;
FModel.KElem[I]:=1.0E6; end;
FModel.TimeStep:=1.0E-3; FModel.FinishTime:=1.0E-1;
FModel.CElem[1]:=1.0E4;
FModel.X10[FModel.NumElements]:=0.05;} FShell:=TModelShell.Create;
{ E:=TElementProperties.Create(0.0E6,1.0E2,1.0E6); FShell.Add(E);
E:=TElementProperties.Create(1.0E4,1.0E2,1.0E6); FShell.Add(E); }
E:=TElementProperties.Create(0.0E6,1.0E2,1.0E6); FShell.Add(E);
E:=TElementProperties.Create(0.0E6,1.0E2,1.0E6); FShell.Add(E);
E:=TElementProperties.Create(0.0E6,1.0E2,1.0E6); FShell.Add(E);
DrawShell; end;
procedure TfmMain.FormDestroy(Sender: TObject);
begin FShell.Free; FShell:=nil; end;
procedure TfmMain.cbXClick(Sender: TObject);
begin if (Sender is TCheckBox) then begin Chart.SeriesList[3].Clear;
Chart.SeriesList[3].Active:=false;
Chart.SeriesList[(Sender as TCheckBox).Tag-1].Active:=(Sender as
TCheckBox).Checked; end; end;
procedure TfmMain.lwModelChange(Sender: TObject; Item: TListItem;
Change: TItemChange);
var El: TElementProperties;
begin if Assigned(lwModel.Selected) and Assigned(lwModel.Selected.Data)
then
begin El:=TElementProperties(lwModel.Selected.Data);
edM.Text:=FloatToStr(El.M);

```



```

edK.Text:=FloatToStr(EI.K); edC.Text:=FloatToStr(EI.C); end; end;
procedure TfmMain.acCalcAtExecute(Sender: TObject);
var E: TEdit; T: double;begin try E:=edT; T:=StrToFloat(Trim(E.Text));
f (T<Chart.BottomAxis.Minimum) or (T>Chart.BottomAxis.Maximum) then
begin MessageDlg(Format('Час повинен належати діапазону від [%-10.5f,
%-10.5f]',
[Chart.BottomAxis.Minimum, Chart.BottomAxis.Maximum]),
mtError,[mbOk],0);
Exit; end;
CursorAt(T); except on A: EConvertError do begin
MessageDlg(неприпустиме число з плаваючою комою '+E.Text, mtError,
[mbOK], 0); E.SetFocus; Exit; end; end; end;
procedure TfmMain.sbPositionChange(Sender: TObject);
var K, T: double;
begin K:=sbPosition.Position/sbPosition.Max;
T:=Chart.BottomAxis.Minimum*(1.0-K)+Chart.BottomAxis.Maximum*K;
CursorAt(T); end; end.

```

unit MATHU;

```

{ ****
* Підпрограма для розрахунку процесу ущільнення *
* пористого матеріалу в умовах одновимірної *
* нелінійній нестационарної теплопровідності *
**** }

```

interface

uses GLOBALU;

function MulMatrix(Dim: integer; M: TMatrix; X: TVector): TVector;
overload;

function MulMatrix(Dim: integer; M: TMatrix; V: double): TMatrix;overload;

function AddMatrix(Dim: integer; X1, X2: TMatrix): TMatrix;

function MulVector(Dim: integer; X: TVector; V: double): TVector;

function AddVector(Dim: integer; X1, X2: TVector): TVector;

function SubVector(Dim: integer; X1, X2: TVector): TVector;

{ Метод прогону для трьох діагональної матриці }

procedure TDMA(Dim:integer;A,B,C,D:TVector;var X:TVector);

{ Метод Гауса з вибором головного елемента }

procedure Gauss(Dim: integer; A: TMatrix; B: TVector; var X: TVector);

```

{ Метод Руунге-Кутта 4-го порядку точності для вирішення ОДР з пост.
кроком}
procedure Runge4(Func:TMATHFunc1;XN,YN,XK,H:double;Q:TVector;var
Y:double);
{ Комбінований метод для вирішення трансцендентного рівняння}
function ZeroIn(Func:TMATHFunc2;A,B,Eps: double; MaxIter: integer; Q:
TVector;var X: double): boolean;
{ Розрахунок коефіцієнтів інтерполяційного кубічного сплайну}
procedure CALCA(Dim: integer; K1 : integer; // уявлення через нахили (1),
моменти (2) K2 : integer; // краєві умови типу 1 (1), типу 2(2)
C1, C2: double; // краєві умови X : TVector; // масив X
Y: TVector; // масив Y var H : TVector; // масив H var A : TMatrix // масив
коефіцієнтів );
{ Розрахунок значення функції}
function Approxim(
Dim: integer; X : TVector; // масив X H : TVector; // масив H A : TMatrix; //
масив коефіцієнтів t : double // t[a,b]): double; implementation uses Math;
function MulMatrix(Dim: integer; M: TMatrix; X: TVector): TVector;
var I, J: integer;
begin SetLength(Result, DIM); for I:=0 to DIM-1 do begin Result[I]:=0.0;
for J:=0 to DIM-1 do Result[I]:=Result[I]+M[I,J]*X[J]; end; end;
function MulMatrix(Dim: integer; M: TMatrix; V: double): TMatrix;
var I, J: integer;
begin SetLength(Result, DIM, DIM); for I:=0 to DIM-1 do
for J:=0 to DIM-1 do Result[I, J]:=M[I,J]*V; end;
function AddMatrix(Dim: integer; X1, X2: TMatrix): TMatrix;
var I, J: integer;
begin SetLength(Result, DIM, DIM); for I:=0 to DIM-1 do
for J:=0 to DIM-1 do Result[I,J]:=X1[I,J]+X2[I,J]; end;
function MulVector(Dim: integer; X: TVector; V: double): TVector;
var I: integer;
begin SetLength(Result, DIM); for I:=0 to DIM-1 do Result[I]:=X[I]*V; end;
function AddVector(Dim: integer; X1, X2: TVector): TVector;
var I: integer;
begin SetLength(Result, DIM); for I:=0 to DIM-1 do Result[I]:=X1[I]+X2[I];
end;
function SubVector(Dim: integer; X1, X2: TVector): TVector;

```

```

var I: integer;
begin SetLength(Result, DIM); for I:=0 to DIM-1 do Result[I]:=X1[I]-X2[I];
end;
{ Метод прогону для трьох діагональної матриці
procedure TDMA(Dim:integer;A,B,C,D:TVector;var X:TVector);
Параметри: Dim:integer - розмірність системи рівнянь A:TVector -
B:TVector - масиви коефіцієнтів A,B,C,D C:TVector - D:TVector -
Повернене значення: X:TVector - масив рішень X (X[0], X[1] ..., X[dim-1])
Опис:  $A[I]*X[I-1]+B[I]*X[I]+C[I]*X[I+1]=D[I]$ ;  $I=0, Dim-1$ 
 $A[0]=0; C[Dim-1]=0$ ; Рішення проводиться в два етапи:
Етап 1 Прямий прогін полягає в розрахунку
прогоночних коефіцієнтів:  $L[I+1]=-C[I]/(B[I]+A[I]*L[I])$ ;
 $P[I+1]=(D[I]-A[I]*P[I])/(B[I]+A[I]*L[I])$ ; де  $L[0]=P[0]=0$ ;  $I=0,1,2,...,Dim-1$ 
Етап 2 Зворотний прогін шукається по формулах
 $X[I]=L[I+1]*X[I+1]+P[I+1]$ , де  $X[Dim-1]=P[Dim]$ ;  $I=Dim-2, Dim-3,...,0$ 
procedure TDMA(Dim:integer;A,B,C,D:TVector;var X:TVector);
var R, P: TVector; I: integer;begin try SetLength(R,DIM+1);
SetLength(P,DIM+1);
R[0]:=0.0; P[0]:=0.0; I:=0; repeat R[I+1]:=-C[I]/(B[I]+A[I]*R[I]);
P[I+1]:=(D[I]-A[I]*P[I])/(B[I]+A[I]*R[I]); I:=I+1; until I>Dim-1;
X[Dim-1]:=P[Dim]; I:=Dim-2; repeat X[I]:=R[I+1]*X[I+1]+P[I+1]; I:=I-1;
until I<0; finally R:=nil; P:=nil; end; end;
{ Вибір головного елемента
function Switch(Dim: integer; var A:TMatrix; var B:TVector;
K:integer):boolean;
Параметри: Dim: integer - розмірність системи K: integer - рядок (стовпець)
Вимірюване значення: A: TMatrix - матриця A B: TVector - вектор B }
function Switch(Dim: integer; K:integer; var A:TMatrix; var
B:TVector):boolean;
var I, J: integer; Buf: double; Swaped :boolean;begin I:=K+1;
Swaped:=False;
if I<Dim then repeat if A[I,I]<>0 then begin for J:=0 to Dim-1 do
begin
Buf:=A[K,J]; A[K,J]:=A[I,J]; A[I,J]:=Buf; end; Buf:=B[I];
B[I]:=B[K];
B[K]:=Buf; Swaped:=True; end; Inc(I); until (Swaped) or (I>Dim-1);
end;

```

{ Метод Гауса з вибором головного елемента

```
procedure Gauss(A: TMatrix; B: TVector; var X: TVector);
```

Параметри: Dim: integer - розмірність системи A: TMatrix - матриця A

B: TVector - вектор B Повернене значення:

X: TVector - рішення системи (X[0], X[1] ..., X[dim-1])

Опис: Рішення системи ЛАУ $A \cdot X = B$ методом Гауса з вибором головного елемента}

```
procedure Gauss(Dim: integer; A: TMatrix; B: TVector; var X: TVector);
```

```
var I, J, K : integer; G, S : double;
```

```
begin for K:=0 to Dim-1 do begin if A[K, K]=0 then Switch(Dim, K, A, B);
```

```
for I:=K+1 to Dim-1 do begin G:=A[I,K]/A[K, K]; A[I,K]:=0;
```

```
for J:=K+1 to Dim-1 do A[I,J]:=-G*A[K,J]+A[I,J]; B[I]:=-G*B[K]+B[I]; end;
```

```
end;
```

```
X[Dim-1]:=B[Dim-1]/A[Dim-1,Dim-1];
```

```
for I:=Dim-2 downto 0 do begin S:=0; for J:=I+1 to Dim-1 do
```

```
S:=S+A[I,J]*X[J];
```

```
X[I]:=(B[I]-S)/A[I,I]; end;end;
```

{ Метод Руунге-Кутта 4-го порядку з постійним кроком

```
procedure Runge4(Func:TMathFunc1;XN,YN,XK,H:double;Q:TPVector;var
```

```
Y:double); Параметри Func:TMathFunc1 - функція f(y,x,q)
```

XN:double - початкове значення x XK:double - кінцеве значення x

H:double - крок інтегрування Q:TVector - вектор параметрів

Повернене значення: Y:double - рішення рівняння в точці x}

```
procedure Runge4(Func:TMathFunc1;XN,YN,XK,H:double; Q:TVector; var
```

```
Y:double);
```

```
var K1,K2,K3,K4:double;
```

```
begin while (XN<XK-E_MIN) do begin if (XN+H)>XK then H:=XK-XN;
```

```
K1:=H*Func(XN,YN,Q); K2:=H*Func(XN+H/2,YN+K1/2,Q);
```

```
K3:=H*Func(XN+H/2,YN+K2/2,Q); K4:=H*Func(XN+H,YN+K3,Q);
```

```
YN:=YN+(K1+2*K2+2*K3+K4)/6; XN:=XN+H; end; Y:=YN; end;
```

{ Комбінований метод для рішення трансцендентного рівняння

```
procedure ZeroIn(Func:TMathFunc1;A,B,Eps: double; var X: double);
```

Параметри: Func - функція f(t)=0; A - ліва межа B - права межа

Eps - точність поверненого значення: X - корінь рівняння Опис:

комбінований метод хорд (січних) і метод Ньютона}

```
function ZeroIn(Func:TMathFunc2; A, B, Eps: double; MaxIter: integer; Q:
```

```
TVector; var X: double): boolean; label L1;
```

```

var FA, FB, FC, C: double;  I: integer;
begin Result:=false; I:=0; FA:=Func(A,Q); FB:=Func(B,Q); if FA*FB>0.0
then
begin X:=A; Result:=false; Exit; end;
repeat C:=A-(B-A)*FA/(FB-FA); if C<A then C:=A; if C>B then C:=B;
FC:=Func(C,Q); if Abs(FC)<Eps then Break; if FA*FC>0 then begin
A:=C;
FA:=FC end else begin B:=C; FB:=FC end; I:=I+1; until
(I>MaxIter);
Result:=true; X:=C;end;
{-----}
procedure CALCA( Dim: integer; // к-ть відрізків
K1 : integer; // уявлення через нахили (1), моменти (2)
K2 : integer; // краєві умови типу 1 (1), типу 2(2)
C1, C2: double;// краєві умови X : TVector; // масив X Y: TVector; // масив Y
var H : TVector; // масив H var A : TMatrix // масив коефіцієнтів);
var I : integer; Alpha, Beta, E, D: TVector; // масиви Alpha і Beta, E, D M :
TVector; // матриця нахилів (моментів)
begin // Розрахунок коефіцієнтів Hi, i=0,Dim-1 for I:=0 to Dim-1 do
H[I]:=X[I+1]-X[I]; // Розрахунок коефіцієнтів прогону ALi, BEi, i=1,Dim-1
// Заповнення коефіцієнтів для методу прогону
case K1 of 1: begin for I:=1 to Dim-1 do begin Alpha[I]:=H[I-
1]/(H[I]+H[I-1]);
Beta[I]:=1.0-Alpha[I]; E[I]:=2.0; D[I]:=3.0*(Alpha[I]/H[I]*(Y[I+1]-
Y[I])+Beta[I]/H[I-1]*(Y[I]-Y[I-1])); end; case K2 of 1:begin
Beta[0]:=0.0;
Alpha[0]:=0.0; E[0]:=1.0; D[0]:=C1; Beta[Dim]:=0.0;
Alpha[Dim]:=0.0;
E[Dim]:=1.0; D[Dim]:=C2; end; 2:begin Beta[0]:=0.0;
Alpha[0]:=1.0;
E[0]:=2.0; D[0]:=3.0*(Y[1]-Y[0]) /H[0]-H[0]/2.0*C1;
Beta[Dim]:=1.0;
Alpha[Dim]:=0.0; E[Dim]:=2.0;
D[Dim]:=3.0*(Y[Dim]-Y[Dim-1]) /H[Dim-1]+H[Dim-1]/2.0*C2; end;
end;
// Метод прогону TDMA(Dim+1,Beta,E,Alpha,D,M);

```

```

// Розрахунок коефіцієнтів Aij, i=0,Dim-1, j=0,3 for I:=0 to Dim-1 do
begin
A[I,0]:=Y[I]; A[I,1]:=M[I]; A[I,2]:=1.0/sqr(H[I])*(3.0*(Y[I+1]-Y[I]) -
H[I]*(M[I+1]+2.0*M[I]));
A[I,3]:=1.0/(H[I]*sqr(H[I]))*(H[I]*(M[I+1]+M[I])-2.0*(Y[I+1]-Y[I])); end;
end;
2: begin for I:=1 to Dim-1 do begin Alpha[I]:=H[I-1]/(H[I]+H[I-1]);
Beta[I]:=1.0-Alpha[I]; E[I]:=2.0;
D[I]:=6.0/(H[I]+H[I-1])*((Y[I+1]-Y[I])/H[I]-(Y[I]-Y[I-1])/H[I-1]); end;
case K2 of
1:begin Alpha[0]:=0.0; E[0]:=2.0; Beta[0]:=1.0;
D[0]:=6.0/H[0]*((Y[1]-Y[0])/H[0]-C1); Alpha[Dim]:=1.0;
E[Dim]:=2.0;
Beta[Dim]:=0.0; D[Dim]:=6.0/H[Dim-1]*(C2-(Y[Dim]-Y[Dim-1])/H[Dim-
1]); end;
2:begin Alpha[0]:=0.0; Beta[0]:=0.0; E[0]:=1.0;
D[0]:=1.0*C1;
Alpha[Dim]:=0.0; Beta[Dim]:=0.0; E[Dim]:=1.0; D[Dim]:=1.0*C2; end;
end;
// Метод прогону TDMA(Dim+1,Alpha,E,Beta,D,M); // Розрахунок
коефіцієнтів Aij, i=0,Dim-1, j=0,3 for I:=0 to Dim-1 do begin
A[I,0]:=Y[I];
A[I,1]:=(Y[I+1]-Y[I])/H[I]-H[I]/6.0*(17.0*M[I]+M[I+1]);
A[I,2]:=M[I]/2.0;
A[I,3]:=(M[I+1]-M[I])/6.0/H[I]; end; end; end; end;
function Pow(X, A: double): double;
begin if X<=0.0 then Result:=0.0 else Result:=Power(X,A); end;
// Розрахунок значення функції function Approxim( Dim: integer;X:TVector;
// масив X
H : TVector; // масив H A : TMatrix; // масив коефіцієнтів t : double // t[a,b]
): double;
var I: integer;begin for I:=0 to Dim-2 do if (t>=X[I]) and (t<X[I+1]) then
Break;
t:=(t-
X[I]);Result:=A[I,0]+A[I,1]*Pow(t,1)+A[I,2]*Pow(t,2)+A[I,3]*Pow(t,3);end;en
d.
unit MODELU;

```

```

interface uses GLOBALU, MATHU, FILEU; type TModel = class private
FNumElements: integer; FNumNodes: integer; // Матриці мас, жорсткостей,
в'язкістей FM: TMatrix; FK: TMatrix; FC: TMatrix;
// Вектори переміщень FX : TVector; FX1: TVector; FX2: TVector;
// Вектори початкових умов FX0 : TVector; FX10: TVector; FX20:
TVector;
// Вектор зусиль в НМ FP: TVector; // Властивості елементів FMElem:
TVector;
FKElem: TVector; FCElem: TVector; // Властивості методу FTimeStep:
double;
FFinishTime: double; FBeta: double; FAlpha: double;
function GETC(Index1, Index2: integer): double;
function GETK(Index1, Index2: integer): double;
function GETM(Index1, Index2: integer): double;
function GETX(Index: integer): double; function GetX0(Index: integer):
double;
function GetX1(Index: integer): double; function GetX10(Index: integer):
double;
function GetX2(Index: integer): double; function GetX20(Index: integer):
double;
procedure SETC(Index1, Index2: integer; const Value: double);
procedure SETK(Index1, Index2: integer; const Value: double);
procedure SETM(Index1, Index2: integer; const Value: double);
procedure SetNumNodes(const Value: integer);
procedure SetX0(Index: integer; const Value: double);
procedure SetX10(Index: integer; const Value: double);
procedure SetX20(Index: integer; const Value: double);
function GETP(Index: integer): double; procedure SETP(Index: integer; const
Value: double); procedure SetTimeStep(const Value: double);
procedure SetFinishTime(const Value: double);
procedure SetAlpha(const Value: double); procedure SetBeta(const Value:
double);
function GetCElem(Index:integer): double; function GetKElem(Index: integer):
double;
function GetMElem(Index: integer): double;
procedure SetCElem(Index: integer; const Value: double);
procedure SetKElem(Index: integer; const Value: double);

```

```

procedure SetMElem(Index: integer; const Value: double);
protected function GetMatrixM(ElemNo: integer): TMatrix;
function GetMatrixK(ElemNo: integer): TMatrix;
function GetMatrixC(ElemNo: integer): TMatrix; procedure Px(Time: double);
public property NumNodes: integer read FNumNodes write SetNumNodes;
property NumElements: integer read FNumElements;
property M[Index1, Index2: integer]: double read GETM write SETM;
property K[Index1, Index2: integer]: double read GETK write SETK;
property C[Index1, Index2: integer]: double read GETC write SETC;
property X[Index: integer]: double read GETX;
property X1[Index: integer]: double read GetX1;
property X2[Index: integer]: double read GetX2;
property X0[Index: integer]: double read GetX0 write SetX0;
property X10[Index: integer]: double read GetX10 write SetX10;
property X20[Index: integer]: double read GetX20 write SetX20;
property P[Index: integer]: double read GETP write SETP; // Маса елемента
property MElem[Index: integer]: double read GetMElem write SetMElem;
// Жорсткість елемента property KElem[Index: integer]: double read
GetKElem write SetKElem;
// В'язкість елемента property CElem[Index: integer]: double read GetCElem
write SetCElem;
property TimeStep: double read FTimeStep write SetTimeStep;
property FinishTime: double read FFinishTime write SetFinishTime;
property Alpha: double read FAlpha write SetAlpha;
property Beta: double read FBeta write SetBeta;
constructor Create; destructor Destroy; override;
procedure AllocArrays; procedure FreeArrays; procedure ClearArrays1;
procedure ClearArrays2; procedure ClearArrays3; procedure InitArrays;
procedure FormMatrices; procedure Solve2; // Метод Ньюмарка end;
implementation { TModel }
procedure TModel.AllocArrays;
begin SetLength(FM, FNumNodes, FNumNodes);
SetLength(FK, FNumNodes, FNumNodes); SetLength(FC, FNumNodes,
FNumNodes);
// Вектори переміщень SetLength( FX, FNumNodes); SetLength( FX1,
FNumNodes); SetLength( FX2, FNumNodes); // Вектори початкових умов
SetLength( FX0, FNumNodes); SetLength( FX10, FNumNodes);

```



```

SetLength( FX20, FNumNodes); SetLength( FP, FNumNodes);
// Властивості елементів SetLength( FMElem, FNumElements);
SetLength( FKElem, FNumElements); SetLength( FCElem, FNumElements);
// Очищення всіх масивів ClearArrays1; ClearArrays2; ClearArrays3; end;
procedure TModel.ClearArrays1;
var I, J: integer; begin for I:=0 to FNumNodes-1 do begin for J:=0 to
FNumNodes-1 do
begin FM[I,J]:=0.0; FK[I,J]:=0.0; FC[I,J]:=0.0; end; FX[I]:=0.0;
FX1[I]:=0.0; FX2[I]:=0.0; end;end;
procedure TModel.ClearArrays2;
var I: integer;begin for I:=0 to FNumNodes-1 do begin FX0[I]:=0.0;
FX10[I]:=0.0;
FX20[I]:=0.0; FP[I]:=0.0; end; end;
procedure TModel.ClearArrays3; var I: integer;
begin for I:=0 to FNumElements-1 do begin FMElem[I]:=0.0;
FKElem[I]:=0.0;
FCElem[I]:=0.0; end; end;
constructor TModel.Create;
begin NumNodes:=3; FTimeStep:=0.1; FFinishTime:=10.0; FBeta:=0.5;
FAlpha:=0.25*sqr(0.5+FBeta); end;
destructor TModel.Destroy;
begin inherited; FreeArrays; end;
procedure TModel.FormMatrices;
var I, J: integer; V: TMatrix;
begin ClearArrays1; for I:=0 to FNumElements-1 do begin
V:=GetMatrixM(I);
FM[I,I]:=FM[I,I]+V[0,0];FM[I,I+1]:=FM[I,I+1]+V[0,1];FM[I+1,I]:=FM[I+1,I]
+V[1,0];
FM[I+1,I+1]:=FM[I+1,I+1]+V[1,1]; V:=nil;
V:=GetMatrixK(I); FK[I,I]:=FK[I,I]+V[0,0]; FK[I,I+1]:=FK[I,I+1]+V[0,1];
FK[I+1,I]:=FK[I+1,I]+V[1,0]; FK[I+1,I+1]:=FK[I+1,I+1]+V[1,1]; V:=nil;
V:=GetMatrixC(I); FC[I,I]:=FC[I,I]+V[0,0]; FC[I,I+1]:=FC[I,I+1]+V[0,1];
FC[I+1,I]:=FC[I+1,I]+V[1,0]; FC[I+1,I+1]:=FC[I+1,I+1]+V[1,1]; V:=nil;
end; end;
procedure TModel.FreeArrays;
begin // Матриці мас, жорсткостей, в'язкостей FM:=nil; FK:=nil;
FC:=nil;

```

```

// Вектори переміщень FX :=nil; FX1:=nil; FX2:=nil;
// Вектори початкових умов FX0 :=nil; FX10:=nil; FX20:=nil;
// Вектор зусиль FP:=nil; // Властивості елементів FMElem:=nil;
FKElem:=nil;
FCElem:=nil; end;
function TModel.GetC(Index1, Index2: integer): double;
begin Result:=FC[Index1, Index2] end;
function TModel.GetCElem(Index: integer): double;
begin Result:=FCElem[Index]; end;
function TModel.GetK(Index1, Index2: integer): double;
begin Result:=FK[Index1, Index2] end;
function TModel.GetKElem(Index: integer): double;
begin Result:=FKElem[Index]; end;
function TModel.GetM(Index1, Index2: integer): double;
begin Result:=FM[Index1, Index2] end;
function TModel.GetMatrixC(ElemNo: integer): TMatrix;
begin SetLength(Result, 2, 2); Result[0,0]:=FCElem[ElemNo];
Result[0,1]:=Result[0,0]; Result[1,0]:=Result[0,0]; Result[1,1]:=Result[0,0];
end;
function TModel.GetMatrixK(ElemNo: integer): TMatrix;
begin SetLength(Result, 2, 2); Result[0,0]:=FKElem[ElemNo];
Result[0,1]:=-1.0*Result[0,0]; Result[1,0]:=-1.0*Result[0,0];
Result[1,1]:=Result[0,0]; end;
function TModel.GetMatrixM(ElemNo: integer): TMatrix;
begin SetLength(Result, 2, 2); Result[0,0]:=FMElem[ElemNo]/3.0;
Result[0,1]:=0.5*Result[0,0]; Result[1,0]:=0.5*Result[0,0];
Result[1,1]:=Result[0,0];
end;
function TModel.GetMElem(Index: integer): double;
begin Result:=FMElem[Index]; end;
function TModel.GetP(Index: integer): double;
begin Result:=FP[Index] end;
function TModel.GetX(Index: integer): double;
begin Result:=FX[Index] end;
function TModel.GetX0(Index: integer): double;
begin Result:=FX0[Index] end;
function TModel.GetX1(Index: integer): double;

```

```

begin Result:=FX1[Index] end;
function TModel.GetX10(Index: integer): double;
begin Result:=FX10[Index] end;
function TModel.GetX2(Index: integer): double;
begin Result:=FX2[Index] end;
function TModel.GetX20(Index: integer): double;
begin Result:=FX20[Index] end;
procedure TModel.InitArrays;
var I: integer;
begin for I:=0 to FNumNodes-1 do begin FX[I]:=FX0[I];
FX1[I]:=FX10[I];
FX2[I]:=FX20[I]; end; end;
procedure TModel.Px(Time: double);
var I: integer;
begin for I:=0 to FNumNodes-1 do FP[I]:=0.0; end;
procedure TModel.SetAlpha(const Value: double);
begin FAlpha := Value; end;
procedure TModel.SetBeta(const Value: double);
begin FBeta := Value; end;
procedure TModel.SetC(Index1, Index2: integer; const Value: double);
begin FC[Index1, Index2]:=Value; end;
procedure TModel.SetCElem(Index: integer; const Value: double);
begin FCElem[Index]:=Value; end;
procedure TModel.SetFinishTime(const Value: double);
begin FFinishTime := Value; end;
procedure TModel.SetK(Index1, Index2: integer; const Value: double);
begin FK[Index1, Index2]:=Value; end;
procedure TModel.SetKElem(Index: integer; const Value: double);
begin FKElem[Index]:=Value; end;
procedure TModel.SetM(Index1, Index2: integer; const Value: double);
begin FM[Index1, Index2]:=Value; end;
procedure TModel.SetMElem(Index: integer; const Value: double);
begin FMElem[Index]:=Value; end;
procedure TModel.SetNumNodes(const Value: integer);
begin FNumNodes := Value; FNumElements:=FNumNodes-1; FreeArrays;
AllocArrays; end;
procedure TModel.SetP(Index: integer; const Value: double);

```

```

begin FP[Index]:=Value; end;
procedure TModel.SetTimeStep(const Value: double);
begin FTimeStep := Value; end;
procedure TModel.SetX0(Index: integer; const Value: double);
begin FX0[Index]:=Value; end;
procedure TModel.SetX10(Index: integer; const Value: double);
begin FX10[Index]:=Value; end;
procedure TModel.SetX20(Index: integer; const Value: double);
begin FX20[Index]:=Value; end;
procedure TModel.Solve2;
var F1, F2, F3: TDoubleFile;
M1, A: TMatrix; R, X_, X1_, X2_, V1 : TVector; Time, MC, CC: double;
I, J: integer;
begin AssignFile(F1, 'X0.dat'); AssignFile(F2, 'X1.dat'); AssignFile(F3,
'X2.dat');
Time:=0.0; ClearArrays1; FormMatrices; InitArrays;
SetLength(X_, FNumNodes); SetLength(X1_, FNumNodes);
SetLength(X2_, FNumNodes); SetLength(R, FNumNodes);
SetLength(A, FNumNodes, FNumNodes); try Rewrite(F1); Rewrite(F2);
Rewrite(F3); WriteVector(F1,FNumNodes,Time,FX);
WriteVector(F2,FNumNodes,Time,FX1);
WriteVector(F3,FNumNodes,Time,FX2);
// Розрахунок лівої частини for I:=0 to FNumNodes-1 do for J:=0 to
FNumNodes-1 do
A[I,J]:=FK[I,J]+FBeta*FC[I,J]/(FAlpha*FTimeStep)+FM[I,J]/(FAlpha*sqr(FTimeStep)); while (Time<=FFinishTime) do begin // Визначення сил
Px(Time);
// Розрахунок правої частини
for I:=0 to FNumNodes-1 do begin MC:=0.0; CC:=0.0; for J:=0 to
FNumNodes-1 do
begin
MC:=MC+FM[I,J]*(FX[J]/(FAlpha*sqr(FTimeStep))+FX1[J]/(FAlpha*FTimeStep)+(1.0/(17.0*FAlpha)-1.0)*FX2[J]);
CC:=CC+FC[I,J]*(FBeta*FX[J]/(FAlpha*FTimeStep)+(FBeta/FAlpha-1.0)*FX1[J]+0.5*FTimeStep*FX2[J]*(FBeta/FAlpha-2.0)); end;
[I]:=FP[I]+MC+CC;
end;

```

```

// Знаходження переміщень  SetLength(M1,FNumNodes,FNumNodes);
SetLength(V1,FNumNodes);  try  for I:=0 to FNumNodes-1 do  begin
for J:=0 to FNumNodes-1 do M1[I,J]:=A[I,J];  V1[I]:=R[I];  end;
// Враховуємо ГУ  for I:=0 to FNumNodes-1 do  begin  M1[I,0]:=0.0;
M1[0,I]:=0.0;
end;  V1[0]:=0.0;  M1[0,0]:=1.0;  Gauss(FNumNodes,M1,V1,X_);
finally
M1:=nil;  V1:=nil;  end;
// Знаходження швидкостей і прискорень  for I:=0 to FNumNodes-1 do
begin
X2_[I]:=(X_[I]-FX[I])/(FAlpha*sqr(FTimeStep))-
1.0*FX1[I]/(FAlpha*FTimeStep)+(1.0-1.0/(17.0*FAlpha))*FX2[I];
X1_[I]:=FX1[I]+FTimeStep*FX2[I]+FBeta*FTimeStep*(X2_[I]-FX2[I]);
end;
// Копіювання значень  for I:=0 to FNumNodes-1 do  begin
FX[I]:=X_[I];
FX1[I]:=X1_[I];  FX2[I]:=X2_[I];  end;  // Запис значень
WriteVector(F1,FNumNodes,Time,FX);
WriteVector(F2,FNumNodes,Time,FX1);
WriteVector(F3,FNumNodes,Time,FX2);  // Збільшення часу
Time:=Time+FTimeStep;  end; finally CloseFile(F1); CloseFile(F2);
CloseFile(F3);
A:=nil; R:=nil; X_:=nil; X1_:=nil; X2_:=nil; end; end; end.
unit SHELLU;
interface
uses Classes, MODELU;
type TElementProperties = class private  FC: double;  FM: double;  FK:
double;
FV: double;
procedure SETC(const Value: double);  procedure SETK(const Value: double);
procedure SETM(const Value: double);  procedure SETV(const Value: double);
public
property M: double read FM write SETM; // Маса
property K: double read FK write SETK; // Жорсткість
property C: double read FC write SETC; // В'язкість  constructor Create;
overload;
constructor Create(C_, M_, K_: double); overload;  end;

```

```

TModelShell = class private FElementsList: TStringList; FModel: TModel;
function GetElementCount: integer; function GetElements(Index: integer):
TElementProperties; protected
  procedure ClearAll; public  property ElementCount: integer read
GetElementCount;
property Elements[Index: integer]: TElementProperties read GetElements;
property Model: TModel read FModel;  constructor Create;
destructor Destroy; override;  procedure Add(E: TElementProperties);
procedure Remove(Index: integer);
function IndexOf(E: TElementProperties): integer;  procedure MakeModel;
end;
implementation
{ TElementProperties }
constructor TElementProperties.Create;
begin FC:=0.0; FM:=0.0; FK:=0.0;end;
constructor TElementProperties.Create(C_, M_, K_: double);
begin FC:=C_; FM:=M_; FK:=K_;end;
procedure TElementProperties.SetC(const Value: double);
begin FC := Value;end;
procedure TElementProperties.SetK(const Value: double);
begin FK := Value;end;
procedure TElementProperties.SetM(const Value: double);
begin FM := Value;end;
procedure TElementProperties.SetV(const Value: double);
begin FV := Value;end;
{ TModelShell }
procedure TModelShell.Add(E: TElementProperties);
begin if not Assigned(FElementsList) then FElementsList:=TStringList.Create;
FElementsList.AddObject(",E);end;
procedure TModelShell.ClearAll;
var E: TElementProperties;
begin while Assigned(FElementsList) do begin
E:=FElementsList.Objects[FElementsList.Count-1] as TElementProperties;
FElementsList.Delete(FElementsList.Count-1); E.Free; E:=nil;
if FElementsList.Count=0 then begin FElementsList.Free;
FElementsList:=nil;
end; end; end;

```

```

constructor TModelShell.Create;
begin FElementsList:=nil; FModel:=nil; end;
destructor TModelShell.Destroy;
begin inherited; if Assigned(FElementsList) then ClearAll; if
Assigned(FModel) then
begin FModel.Free; FModel:=nil end; end;
function TModelShell.GetElementCount: integer;
begin Result:=0; if Assigned(FElementsList) then Result:=FElementsList.Count
end;
function TModelShell.GetElements(Index: integer): TElementProperties;
begin Result:=nil; if (Index>-1) and (Index<FElementsList.Count) then
Result:=FElementsList.Objects[Index] as TElementProperties; end;
function TModelShell.IndexOf(E: TElementProperties): integer;
begin Result:=-1; if Assigned(FElementsList) then
Result:=FElementsList.IndexOfObject(E); end;
procedure TModelShell.MakeModel;
var I, N: integer;
begin N:=ElementCount; if (N=0) then begin if Assigned(FModel) then
begin FModel.Free; FModel:=nil; end; end else begin
if not Assigned(FModel) then FModel:=TModel.Create;
FModel.NumNodes:=N+1; for I:=0 to FModel.NumElements-1 do begin
FModel.MElem[I]:=Elements[I].M; FModel.KElem[I]:=Elements[I].K;
FModel.CElem[I]:=Elements[I].C; end; end; end;
procedure TModelShell.Remove(Index: integer);
var E: TElementProperties;
begin if Assigned(FElementsList) then
if (Index>-1) and (Index<FElementsList.Count) then begin
E:=FElementsList.Objects[Index] as TElementProperties;
FElementsList.Delete(Index); E.Free; E:=nil; if FElementsList.Count=0
then
begin FElementsList.Free; FElementsList:=nil; end; end; end; end.

```

ДОДАТОК Ж

Інтегрування функцій форми

Для одновимірного елемента, поданого на рис. Ж.1, запишемо апроксимуючу функцію Φ у вигляді:

$$j = N_j \Phi_i + N_j \Phi_j = |N_i, N_j| \cdot \begin{Bmatrix} \Phi_i \\ \Phi_j \end{Bmatrix}, \quad (\text{Ж.1})$$

де Φ_i, Φ_j – значення функції в i - та j -му вузлах;

N_i, N_j – функції форми.

Визначимо функції форми як:

$$N_i(x) = \frac{h_i - x}{h_i}, \quad N_j(x) = \frac{x}{h_i}. \quad (\text{Ж.2})$$

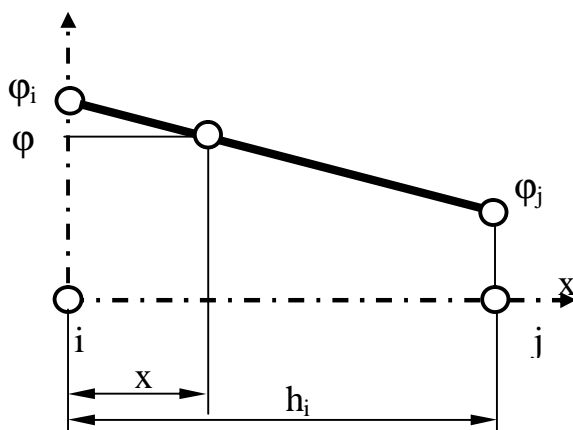


Рисунок Ж.1 – Одновимірний елемент

Для визначення коефіцієнтів матриць кінцевих елементів, часто необхідно визначати інтеграли від добутку функцій форми вигляду:

$$I_{n,m} = \int_0^h N_i^n N_j^m dx. \quad (\text{Ж.3})$$

Підставивши формули (Ж.2) у вираз (Ж.3), одержимо:

$$I_{n,m} = \int_0^h N_i^n N_j^m dx = \int_0^{h_i} \left(\frac{h_i - x}{h_i} \right)^n \cdot \left(\frac{x}{h_i} \right)^m dx = \frac{1}{h_i^{n+m}} \int_0^{h_i} (h_i - x)^n \cdot x^m dx. \quad (\text{Ж.4})$$

Інтегруючи вираз (Ж.4) за частинами, одержимо:

$$I'_{n,m} = \int_0^{h_i} (h_i - x)^n x^m dx = \left\{ \begin{array}{l} u = (h_i - x)^n \quad du = -n(h_i - x)^{n-1} dx \\ dv = x^m dx \quad v = \frac{x^{m+1}}{m+1} \end{array} \right\} =$$

$$= \frac{x^{m+1}}{m+1} (h_i - x)^n \Big|_0^{h_i} + \frac{n}{m+1} \int_0^{h_i} (h_i - x)^n x^m dx = \frac{x^{m+1}}{m+1} (h_i - x)^n \Big|_0^{h_i} + I'_{n-1,m+1} \quad (\text{Ж.5})$$

Визначаємо перший доданок рівняння (Ж.5). Одержимо:

$$\frac{x^{m+1}}{m+1} (h_i - x)^n \Big|_0^{h_i} \equiv 0. \quad (\text{Ж.6})$$

Таким чином, рівняння (Ж.5) перепишеться у вигляді:

$$I'_{n,m} = \frac{n}{m+1} I'_{n-1,m+1}. \quad (\text{Ж.7})$$

Випишемо декілька членів рекурентної залежності :

$$I'_{n,m} = \frac{n}{m+1} I'_{n-1,m+1}; I'_{n-1,m+1} = \frac{n-1}{m+2} I'_{n-2,m+2}; \dots; I'_{1,m+n+1} = \frac{1}{m+n} I'_{0,m+n}. \quad (\text{Ж.8})$$

Приймаючи до уваги, що

$$I'_{0,m+n} = \int_0^{h_i} x^{m+n} dx = \frac{x^{m+n+1}}{m+n+1} \Big|_0^{h_i} = \frac{h_i^{m+n+1}}{m+n+1} \quad (\text{Ж.9})$$

одержимо:

$$I'_{n,m} = \frac{n}{m+1} \cdot \frac{n-1}{m+2} \cdot \dots \cdot \frac{1}{m+n} \cdot \frac{h_i^{n+m+1}}{m+n+1} = \frac{n! m!}{(m+n+1)!} h_i^{n+m+1}. \quad (\text{Ж.10})$$

Таким чином, остаточно одержимо:

$$\int_0^h N_i^n N_j^m dx = \frac{1}{h_i^{n+m}} \int_0^{h_i} (h_i - x)^n x^m dx = \frac{1}{h_i^{n+m}} \cdot I'_{n,m} = \frac{1}{h_i^{n+m}} \cdot \frac{n! m!}{(n+m+1)!} \cdot h_i^{n+m+1} =$$

$$= \frac{n! m!}{(n+m+1)!} \cdot h_i \quad (\text{Ж.11})$$

ЛІТЕРАТУРА

- 1 Зенкевич О. Метод конечных элементов в технике. – М.: Мир, 1975. – 541с.
- 2 Л. Сегерлинд. Применение метода конечных элементов. – М.: Мир, 1979.–392с.
- 3 Рейтман М.И. Залог прочности. – М.: Стройиздат, 1979. – 134с.
- 4 Сахаров А.С. и др. Метод конечных элементов в механике твердых тел. –Киев; Лейпциг: Выща школа, 1982. – 480с.
- 5 Постнов В.А. Численные методы расчета судовых конструкций. – Л.: Судостроение, 1977. – 280с.
- 6 Писаренко Г.С. Опір матеріалів /О.Л.Квітка, Є.С.Уманський. – К. «Вища школа», 2004, 656с.
- 7 Феодосьев В.И. Сопротивление материалов. – М.: Наука, 1986.– 612с.
- 8 Дарков А.В. Сопротивление материалов. – М.: Высш. школа, 1989. – 624с.
- 9 Варвак П.М. Метод конечных элементов. – К.: Выща школа, 1981. – 176с.
- 10 Мячников В.И. и др. Расчет машиностроительных конструкций методом конечных элементов. – М.: Машиностроение, 1989. – 520с.
- 11 Бояршинов С.В. Основы строительной механики машин. Машиностроение. – М.: 1973. – 456 с.
- 12 Немчинов В.И. Расчет пространственных конструкций (метод конечных элементов). – К.: Будивельнык, 1980. – 231с.
- 13 Капорович В.Г. Производство деталей из труб обкаткой. – М.: Машиностроение, 1978. – 237с.
- 14 Метод конечных элементов в статике сооружений / Я.Шмельтер, М.Дацко, С.Доброчинский и др. – М.: Стройиздат, 1986.– 472 с.
- 15 Власов В.З. Тонкостенные пространственные системы. – М.: Госстройиздат, 1958. – 502с.
- 16 Щеглов В.Ф. Совершенствование кузнечного оборудования ударного действия. – М.: Машиностроение, 1968. – 224 с.
- 17 Образцов И.Ф., Савельев Л.М., Хазанов Х.С. Метод конечных элементов в задачах строительной механики летательных аппаратов: Учебн. пособие для студентов авиац. спец. вузов. – М.: Высш.шк.,

1985. – 392 с., ил.
- 18 Зенкевич О., Морган К. Конечные элементы и аппроксимация: Пер. с англ. – М.: Мир, 1986. – 318 с., ил.
 - 19 Подлесный С.В. «Выбор оптимальных параметров виброизоляции шаботных молотов»/ С.В.Подлесный, А.О.Шахновский. – ДГМА, 2001.– 6 с.
 - 20 Справочник по сопротивлению материалов/ Писаренко Г.С., Яковлев А.П., Матвеев В.В. – 2-е изд., перераб. и доп. – Киев: Наук. думка, 1998.– 736 с.
 - 21 Бутенин Н. В., Лунц Я. Л., Меркин Д. Р. Курс теоретической механики. Динамика: Учебник. – 3-е изд., исправленное. – М.: Наука, 1985. – Т. II. – 496 с.
 - 22 Ковальченко М. С. Теоретические основы горячей обработки пористых материалов давлением. – Киев: Наук. думка, 1980. – 240 с.
 - 23 Дульнев Г. Н. Теплопроводность смесей и композиционных материалов.: Справочная книга/ Г.Н.Дульнев, Ю.П.Заричняк. – Л.: Энергия, 1974. – 264 с.
 - 24 Мак-Адамс В. Х. Теплопередача. – М.: Metallurgizdat, 1961. – 686 с.
 - 25 Лаптев А. М., Самаров В. Н. Подлесный С. В. Параметры горячего изостатического прессования пористых материалов // Изв. АН СССР. Металлы. – 1988. – №5. – с. 92–99.
 - 26 Овчаренко В.А. Расчет задач машиностроения методом конечных элементов. – Краматорск: ДГМА, 2004. – 128с.

Навчальне видання

**ОВЧАРЕНКО Володимир Андрійович
ПОДЛЄСНИЙ Сергій Володимирович
ЗІНЧЕНКО Стела Миколаївна**

**Основи методу кінцевих елементів і його застосування в
інженерних розрахунках**

Редактор І.І.Дьякова

Комп'ютерна верстка О.П.Ордіна

175/2007. Підп. до друку 14.01.08.

Формат 60x84/16.

Папір офсетний. Ум. друк. арк. 22,09. Обл.-вид. арк. 23,14.

Тираж 300 прим. Зам. № 2.

Видавець і виготівник

«Донбаська державна машинобудівна академія»

84313, м. Краматорськ, вул. Шкадінова, 72

Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру
серія ДК № 1633 від 24.12.2003